



Zulu Installation Guide

Zulu Releases: Zulu 15.28 (15.0.1+8), 13.34 (13.0.4.0.101+5),
13.35 (13.0.5+3), 11.42 (11.0.8.0.101+5), 11.43 (11.0.9+11),
8.49 (8u271-b09), 8.50 (8u272-b17), 7.41 (7u281-b07), 7.42
(7u282-b10)

Release Date: October 20, 2020

Azul® Zulu® is a Java Development Kit (JDK), and a compliant implementation of the Java Standard Edition (SE) specification that contains all the Java components needed to build and run Java SE applications. Zulu has been verified by passing all tests of the OpenJDK Community Technology Compatibility Kit (TCK) as available for each respective Java SE version.

This document describes how to install the Zulu JDK on Microsoft® Windows®, Linux, Apple® macOS®, or Solaris.

Revision History

Revision	Date	Description
1.0	October 20, 2020	Initial version of the document
1.1	November 5, 2020	Updated section: Using the Timezone Updater Tool.

Preparing the Zulu Installation Platform	9
Identify the Tasks to Install Zulu in Your Environment	9
Prepare for Zulu through Microsoft Azure Cloud	11
Using an Azure Marketplace Virtual Machine	11
Using a Pre-Installed Azure Marketplace Portal Virtual Machine	12
Prepare for Zulu on Amazon AWS	13
Prepare for Zulu on Other Cloud Service	14
Prepare for Zulu on a Physical or Virtual Machine	14
Validate Downloaded Packages	15
Checksum of Download Package on Windows	16
Checksum of Download Package on Linux	16
Checksum of Download Package on macOS	17
Add Azul Package Repositories	18
Add the Yum Repository on RHEL, SLES, or Oracle Linux	18
Add the APT Repository on Ubuntu or Debian	19
Add the APK Repository on Alpine Linux	20
Installing Zulu	22
Variable References	22
Installation on Windows Using the Zulu ZIP File	25
Installation on Windows Using the Zulu MSI File	25
Installation on Linux Using the Zulu TAR.GZ File	28

Installation on Alpine Linux Using an APK Repository	28
Installation on Linux Using a Yum Repository	29
Installation on Linux Using an APT Repository	30
Installation on Linux Using RPM or DEB Package	31
Installation on Linux Using the Zulu Docker Registry	32
Installation on macOS Using a DMG Package through the GUI	35
Installation on macOS Using the Zulu ZIP File	37
Installation on macOS Using the Zulu TAR.GZ File	37
Installation on macOS Using a DMG Package through the CLI	38
Installation on Solaris Using the Zulu ZIP File	40
Post Installation Tasks	41
Locate Zulu Installed Files	41
Verify Installation	42
Verify Installation on Windows	42
Verify Installation on Linux	43
Verify Installation on Linux from a Docker Container	44
Verify Installation on macOS	46
Verify Installation on Solaris	46
Add Paths to Zulu	47
Add Paths to Zulu on Windows	47
Add Paths to Zulu on Linux or Solaris	48

Add Paths to Zulu on macOS	49
Configure Zulu	49
Switching Between Java Alternatives	50
Switch Between your Windows Zulu Installations	50
Switch Between your Linux Zulu Installations	51
Switch Between your macOS Zulu Installations	52
Upgrading Zulu	54
Upgrade Zulu on Windows	54
Upgrade Zulu on Linux Using RPM or DEB Packaging	55
Upgrade Zulu on Linux Using a Yum, APT, or APK Repository	55
Upgrade Zulu on macOS	56
Uninstalling Zulu	57
Uninstall Zulu on Windows	57
Uninstall Zulu Extracted from a Zip File onto Windows	57
Uninstall Zulu on Windows by Using an MSI File	58
Uninstall Zulu on Linux	58
Uninstall Zulu on Linux Using RPM or DEB Packaging	59
Uninstall Zulu on Linux Using a Yum, APT, or APK Repository	60
Uninstall Zulu on macOS	61
Uninstall Zulu on Solaris	62
Using the Zulu Commercial Compatibility Kit	63

Zulu Commercial Compatibility Kit Installed Contents for Zulu JDK from 6 to 8	64
Zulu Commercial Compatibility Kit Requirements for Zulu JDK from 6 to 8	64
Installing the Zulu Commercial Compatibility Kit	65
Prepare to Install the Zulu CCK for Zulu JDK from 6 to 8	65
Install on Windows Using the Zulu CCK MSI File for Zulu JDK from 6 to 8	65
Install on Windows Using the Zulu CCK ZIP File for Zulu JDK from 6 to 8	66
Install on Linux Using the Zulu CCK Shell Script for Zulu JDK from 6 to 8	68
Install on macOS Using the Zulu CCK Shell Script for Zulu JDK from 6 to 8	69
Uninstalling the Zulu Commercial Compatibility Kit from Zulu JDK from 6 to 8	70
Enabling Rhino JavaScript Engine in Zulu 7	71
Appendices	73
Working with Older Versions of Zulu on Linux Platforms	73
Installation on Linux Using a Yum or APT Repository	73
Upgrade Zulu on Linux Using a Yum or APT Repository	75
Downgrade Zulu on Linux	76
Uninstall Zulu on Linux Using a Yum or APT Repository	77
Updating Windows Environment Variables	78
Advanced System Properties	78

setx Command	78
Using the Timezone Updater Tool	79
Zulu MSI Installer and Windows Registry Settings	81
Support	82

Preparing the Zulu Installation Platform

You can install Zulu in a cloud, virtual, or physical environment.

Microsoft Azure cloud service provides a method to directly install Zulu into a virtual machine. Other cloud service providers, such as Amazon AWS, require that you create a virtual machine first, then install Zulu into the virtual machine.

Before you proceed with Zulu installation steps, ensure you implemented the following preparation tasks:

- [Identify the Tasks to Install Zulu in Your Environment](#)
- [Prepare for Zulu through Microsoft Azure Cloud](#)
- [Prepare for Zulu on Amazon AWS](#)
- [Prepare for Zulu on Other Cloud Service](#)
- [Prepare for Zulu on a Physical or Virtual Machine](#)
- [Validate Downloaded Packages](#)
- [Add Azul Package Repositories](#)

Identify the Tasks to Install Zulu in Your Environment

You have several options for installing Zulu. Select your options from the list and complete the steps in the referenced sections.

1. Choose a method to obtain Zulu.

Select a method to obtain Zulu and complete the steps in the appropriate section.

[Prepare for Zulu through Microsoft Azure Cloud](#)

[Prepare for Zulu through Amazon AWS](#)

[Prepare for Zulu through Other Cloud Service](#)

[Prepare for Zulu on a Physical or Virtual Machine](#)

2. Validate downloaded installation packages.

If you are using a ZIP, MSI, RPM, or DMG installer, complete the steps in:

[Validate Downloaded Packages](#)

3. Attach the Azul package repository.

If you are using a Yum, APK, or APT repository, complete the steps in:

[Attach Azul Package Repositories](#)

4. Choose the Operating System Context for your Zulu Installation.

Operating system context includes the operating system used, installation package type, and whether you are using a software repository or not. The installation tasks described here assume common implementation for each of the listed operating systems. For example, typically items installed on

- Windows use a Microsoft Installer (MSI).
- Ubuntu or Debian use either Debian (DEB) packaging or an APT repository.
- RHEL, SLES, or Oracle Linux use either RPM packaging or a Yum repository.
- Alpine Linux use an APK repository.

Complete the steps in the referenced sections for the operating system context that matches your environment. For descriptions of commands not included in the referenced sections, refer to your operating system documentation.

- [Installation on Windows Using the Zulu ZIP File](#)
- [Installation on Windows Using the Zulu MSI File](#)
- [Installation on Linux using the Zulu TAR.GZ File](#)

- [Installation on Alpine Linux using an APK Repository](#)
- [Installation on Linux using a Yum Repository](#)
- [Installation on Linux using an APT Repository](#)
- [Installation on Linux Using RPM, DEB or APT Packaging](#)
- [Installation on Linux using the Zulu Docker Registry](#)
- [Installation on macOS Using a DMG Package through the GUI](#)
- [Installation on macOS using the Zulu ZIP File](#)
- [Installation on macOS Using a DMG Package through the CLI](#)
- [Installation on Solaris Using the Zulu ZIP File](#)

Prepare for Zulu through Microsoft Azure Cloud

There are two ways to use Zulu within the Virtual Machine services of the Microsoft Azure cloud.

- For any general purpose Zulu usage, you can always download Zulu from the Azul website to your Azure virtual machine and install it, using the binary zip or native installer appropriate for the VM's Operating System, which may be Windows or Linux.
- Alternatively, you can simply visit the [Azure Marketplace](#) and launch a fully-configured Virtual Machine image based on Windows Server 2019 or Ubuntu 20.04 with Zulu 15, Zulu 13, Zulu 11, Zulu 8, and Zulu 7 preinstalled.

Refer to [Microsoft Azure blog](#) to learn more about Java Cloud development with Azure.

Using an Azure Marketplace Virtual Machine

The Microsoft Azure Marketplace of virtual machines have the listed operating systems installed and configured.

To prepare to install Zulu using a Microsoft Azure virtual machine from the Azure Marketplace:

1. Login to Microsoft Azure and navigate to the [Azure Marketplace](#) of virtual machines.
2. Select and create your Azure virtual machine.

In the Azure Portal, select any of the virtual hard drives (VHD), that are pre-installed with an operating system that meets the Zulu operating system requirements. This includes: Microsoft Windows Server, Ubuntu, SuSE Linux, and other Linux images.

3. Login to your Azure virtual machine.
4. Proceed to [Prepare for Zulu on a Physical or Virtual Machine](#).

Using a Pre-Installed Azure Marketplace Portal Virtual Machine

Verify the Zulu version. On Windows or Ubuntu, go to the Zulu installation directory and run the `java -version` command.

The Microsoft Azure Marketplace of virtual machines have Windows and Zulu installed and configured.

To prepare to use Zulu from a Microsoft Azure virtual machine from the Azure Marketplace:

1. Login to Microsoft Azure and navigate to the [Azure Marketplace](#) of virtual machines.
2. In the Azure Portal, locate any of the virtual hard drives (VHD), that are pre-installed with Zulu running on Windows or Ubuntu. Click the Zulu OpenJDK icon listing your Java version.



3. Review the Zulu OpenJDK package description, and click **Create Virtual Machine**.
4. Log in to your Azure virtual machine.

The `PATH` and `JAVA_HOME` environment variables are set to include the path to Zulu when the virtual machine boots. This ensures that:

- Zulu can be called by default by Java from command line.
 - Zulu is recognized by JDK-based applications automatically.
5. Verify the Zulu version. On Windows or Ubuntu, go to the Zulu installation directory and run the `java -version` command.

Zulu is preinstalled in Program Files.

Zulu is already installed in a Windows virtual machine and ready for use.

Prepare for Zulu on Amazon AWS

To install Zulu on Amazon AWS:

1. Create and log into your [AWS service](#).
2. Open the Amazon EC2 console at <https://console.aws.amazon.com/ec2/>.
3. Launch your [Linux](#) or [Windows](#) EC2 Instance.
4. Create a virtual machine. Ensure the virtual machine has sufficient resources to support Zulu, and as appropriate, your Java applications.
5. Proceed with the installation steps. See [Installing Zulu](#).

Prepare for Zulu on Other Cloud Service

To install Zulu on any cloud service:

1. Create and log into your cloud service.
2. Create a virtual machine.

Ensure the virtual machine has sufficient resources to support Zulu, and as appropriate, your Java applications.

3. Complete any preparatory steps required by your cloud service provider.
4. Proceed with the installation steps. See [Installing Zulu](#).

Prepare for Zulu on a Physical or Virtual Machine

To install Zulu on a physical or virtual machine:

1. Configure your physical machine or create your virtual machine, ensuring that it has sufficient resources to support Zulu, and as appropriate, your Java applications.
2. If you are using a yum or APT repository, skip the following steps and proceed to the section, [Attach Package Repositories](#).
3. If you are using Docker, skip the following steps and proceed to the section, [Installation on Linux using the Zulu Docker Registry](#).
4. Locate the Zulu installer package from Azul Zulu downloads webpage.

<https://www.azul.com/downloads/zulu/>

Zulu installation package options

- Zulu 9 and later (Windows, Linux, or macOS)
 - Zulu 8 (Windows, Linux, macOS, or Solaris)
 - Zulu 7 (Windows, Linux, or macOS)
5. Select an installer package. There is a `<zulu_package>` for each listed package type.

`<zulu_package>.zip`

`<zulu_package>.msi`

`<zulu_package>.rpm`

`<zulu_package>.dmg`

`<zulu_package>.deb`

6. Proceed to the section, [Validate Downloaded Packages](#).

For macOS DMG installers, skip the Validate Download Package step.

Proceed to either [Installation on macOS Using a DMG Package through the GUI](#) or [Installation on macOS Using a DMG Package through the CLI](#).

Validate Downloaded Packages

When you download the Zulu installation package from the Azul Systems site, a sha256 checksum is provided so you can verify that the file was downloaded completely and is not corrupted. Complete the steps for your operating system. See the steps to validate the downloaded package based on your operating system from one of the links below:

- [Checksum of Download Package on Windows](#)
- [Checksum of Download Package on Linux](#)
- [Checksum of Download Package on macOS](#)

Checksum of Download Package on Windows

1. Use your favorite program to calculate md5 sums.

For example, md5sum is available as a part of installation of GNU tools.

<http://gnuwin32.sourceforge.net/packages/coreutils.htm>

2. After downloading Zulu, at the Windows command line, run the checksum command.

```
> md5sum <zulu_package>.msi
```

Sample response

```
730c347461955f055e1410d6c41f7b6d zulu1.8.0_31-8.5.0.1-  
win64.msi
```

or

```
334c9f28ac7f108f5c35db9cd8499962 zulu8.12.0.1-jdk8.0.71-  
win_x64.msi
```

3. Check the MSI package signature. Right-click **Properties/Digital Signatures**.
MSI packages are signed. Verify the name of the signer is Azul Systems, Inc.
4. Proceed to the installation section corresponding to your operating system and Zulu installer package.

Checksum of Download Package on Linux

1. After downloading Zulu, at the command line, run the checksum command.

```
$ sha256sum <zulu_package>.rpm
```

Sample response:


```
b3d1c044decbbe2b736d342226145667dc9c62f0d22802d0886b699c07
5b4b2f  zulu11.41.23-ca-jdk11.0.8-linux.x86_64.rpm
```

2. View the list of installed public keys.

Install GnuPG if it is not present. See <http://www.gpgtools.org>.

```
$ rpm -q gpg-pubkey --qf '%{name}-%{version}-%{release} --
> %{summary}\n'
```

3. For RHEL, SLES, or Oracle Linux systems, check the package signature.

```
$ rpm -vK <zulu_package>.rpm
```

Sample response

```
<zulu_package>.rpm:
Header V4 RSA/SHA256 Signature, key ID 219bd9c9: OK
Header SHA1 digest: OK
(a0752b6b1b69687a0ad8f9965a881d3775e1b8e5)
V4 RSA/SHA256 Signature, key ID 219bd9c9: OK
MD5 digest: OK (c0ef98f624a35020494f724ca212554a)
```

If you have not imported Azul public key or did not use a software repository to download the Zulu installation file, `NOKEY` is displayed instead of `OK` in the Signature line.

4. Proceed to the installation section to corresponding to your operating system and Zulu installer package.

Checksum of Download Package on macOS

After downloading Zulu, run the checksum command.

- Verify the checksum of the zip package:

```
$ shasum -a 256 <zulu_package>.zip
```

Sample response:

```
c94a968d272b211ba3b24e78dcee701e77d3ba6ab24f8b9be29620b37e
61c776  zulu11.41.23-ca-jdk11.0.8-macosx_x64.zip
```

- Verify the checksum of the `dmg` package:

```
$ shasum -a 256 <zulu_package>.dmg
```

Sample response:

```
c94a968d272b211ba3b24e78dcee701e77d3ba6ab24f8b9be29620b37e
61c776  zulu11.41.23-ca-jdk11.0.8-macosx_x64.dmg
```

Add Azul Package Repositories

Zulu on Linux supports the following command line package management utilities yum, apk, zypper, dnf, and apt. To use these utilities, you need to configure their respective repositories on your system.

- [Add the Yum Repository on RHEL, SLES, or Oracle Linux](#)
- [Add the APT Repository on Ubuntu or Debian](#)
- [Add the APK Repository on Alpine Linux](#)

Add the Yum Repository on RHEL, SLES, or Oracle Linux

Download and install the repository setup package.

1. Log in as `root` or use `sudo`.
2. If you plan to install the repository on the SLES system, import the Azul's

public key from <https://www.azul.com/signing-keys/>:

```
$ sudo rpm --import
https://www.azul.com/files/0xB1998361219BD9C9.txt
```

For other systems, skip this step.

3. Download the installation package from the Azul Systems site <https://cdn.azul.com/zulu/bin/zulu-repo-1.0.0-1.noarch.rpm>.
4. Install the downloaded package with the available package installation tools.

When the Yum repository is installed, proceed to [Installation on Linux Using a Yum Repository](#).

Add the APT Repository on Ubuntu or Debian

You can add the Zulu repository to your apt sources by installing the `zulu-repo` package.

Note that the Azul repository contains DEB packages for the x86 architectures. If the system architecture of your machine is different from x86 (for example, arm32), consider downloading an appropriate tar.gz package for your architecture from [Zulu Downloads](#).

The following steps explain how to install the `zulu-repo` package:

1. Log in as `root`, or use `sudo` when running the commands.
2. Import Azul's public key.

```
$ sudo apt-key adv --keyserver
hkp://keyserver.ubuntu.com:80 --recv-keys
0xB1998361219BD9C9
```

3. Download the installation package from the Azul Systems site https://cdn.azul.com/zulu/bin/zulu-repo_1.0.0-2_all.deb.

4. Install the downloaded package by running the following command in your terminal:

```
$ sudo apt-get install ./zulu-repo_1.0.0-2_all.deb
```

5. Update the information about available packages.

```
$ sudo apt-get update
```

When the APT repository is installed, proceed to [Installation on Linux Using an APT Repository](#).

Add the APK Repository on Alpine Linux

1. Log in as `root` or prefix the commands listed below with `sudo`.
2. Import Azul's public key.
 1. Download the product signing key for Alpine Linux `alpine-signing@azul.com-5d5dc44c.rsa.pub` from <https://www.azul.com/openpgp-keys>.

```
$ wget https://cdn.azul.com/public_keys/alpine-signing@azul.com-5d5dc44c.rsa.pub
```

2. Copy the product signing key into `/etc/apk/keys/` directory.

```
$ cp alpine-signing@azul.com-5d5dc44c.rsa.pub /etc/apk/keys/
```

3. Add the Azul APK repository.

```
# echo "https://repos.azul.com/zulu/alpine" >> /etc/apk/repositories
```

4. Update the information about available packages.

```
# apk update
```

5. Proceed to [Installation on Alpine Linux Using an APK Repository](#).

Installing Zulu

Before you install Zulu, complete the tasks in the section, [Preparing the Zulu Installation Platform](#). Then complete the steps in the installation section corresponding to your operating system and Zulu installer package.

Note: Our software distributions are signed by an OpenPGP key when appropriate. To verify your download, see [Product Signing Keys \(OpenPGP\)](#) for details.

- [Installation on Windows Using the Zulu ZIP File](#)
- [Installation on Windows Using the Zulu MSI File](#)
- [Installation on Linux using the Zulu TAR.GZ File](#)
- [Installation on Alpine Linux using an APK Repository](#)
- [Installation on Linux using a Yum Repository](#)
- [Installation on Linux using an APT Repository](#)
- [Installation on Linux Using RPM or DEB Package](#)
- [Installation on Linux using the Zulu Docker Registry](#)
- [Installation on macOS Using a DMG Package through the GUI](#)
- [Installation on macOS using the Zulu ZIP File](#)
- [Installation on macOS Using the Zulu TAR.GZ File](#)
- [Installation on macOS Using a DMG Package through the CLI](#)
- [Installation on Solaris Using the Zulu ZIP File](#)

Variable References

The following variables are used throughout the guide to simplify the sample commands. Substitute the value that is appropriate for your operating system context.

- **<zulu_dir>**

The Zulu parent directory where the Zulu package is located.

```
(Windows) C:\Program Files\Zulu
```

```
(Linux) /usr/lib/jvm/
```

```
(macOS) /Library/Java/JavaVirtualMachines
```

- **<zulu_package>**

The name of your Zulu installation package. It corresponds to the following pattern:

```
zulu<zulu_version>-jdk<java_version>-  
<platform/architecture>
```

For example, `zulu8.12.0.1-jdk8.0.71-win_x64.zip`

For the releases that precede 8.12, 7.13 and 6.11 an installation package looks as follows:

```
zulu<java_version>-<zulu_version>-  
<platform/architecture>
```

For example, `zulu1.7.0_91-7.12.0.3-x861x64.zip`

- **<java_version>**

The Zulu package label Java version designation. For example,

- 8.0.71

For the releases that precede 8.12, 7.13 and 6.11 the java version looks as follows:

- 1.8.0_65
- 1.7.0_91
- **<zulu_version>**

The Zulu package label Zulu version designation. For example,

- 8.12.0.1
- 8.10.0.1
- 7.12.0.3
- **<platform> or <architecture>**

The Zulu package label platform designation for Windows, RPM, or Debian files.

- **<zulu_jdk>**

Corresponds to your Zulu JDK version referenced with repositories. For example,

- zulu-8
- zulu-8-amd64
- **<cck_package>**

Corresponds to your Zulu Commercial Compatibility Kit installation package. For example,

- zcck8-8.0.0.2-win64.msi
- **<zulu_docker_tag>**

Corresponds to your Zulu Docker repository images. For example,

- latest
- 8u66

- 7
- 6u105

Depending upon your operating system context, `<zulu_dir>/<zulu_jdk>` could be:

- C:\Program Files\Zulu\zulu-8\
- /usr/lib/jvm/zulu-6-amd64
- /Library/Java/JavaVirtualMachines/zulu-7.jdk

Installation on Windows Using the Zulu ZIP File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.
2. Unzip the Zulu zip package, using any unzip tool. For example, right-click `<zulu_package>.zip`, and then click **Extract All**.

All Zulu files for the OpenJDK are expanded into the corresponding folder.

For example,

```
%HOMEPATH%\Downloads\zulu8.48.0.51-ca-jdk8.0.262-win_x64\
```

Installation on Windows Using the Zulu MSI File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.
2. Choose a method for running the installer.
 - a. To install Zulu in default location, run the package installer as Administrator, from the Windows command line:

```
> msixec /i <zulu_package>.msi /qn
```

Zulu will be installed in `C:\Program Files\Zulu\<zulu-jdk>\`.

Where `jdk` in `<zulu-jdk>` stands for the version of Zulu you plan to install, for example, `zulu-11`.

- b. To install Zulu in a non-default location, run the package installer as Administrator, from the Windows command line.

Choose one of the following options:

- Install Zulu in a specific directory:

```
> msixec /i <zulu_package>.msi
INSTALLDIR="<absolute_path_to_dir>" /qn
```

Example to make `java.exe` available as

```
c:\java\jdk8\bin\java.exe:
```

```
> msixec /i zulu8.42-sa-jdk8.0.232-win_
x64.msi INSTALLDIR="c:\java\jdk8" /qn
```

- Install Zulu in the `zulu-jdk` folder at a specific location:

```
> msixec /i <zulu_package>.msi
APPLICATIONROOTDIRECTORY="<absolute_path_to_
dir>" /qn
```

The `\<zulu-jdk>\` folder will be created in the specified directory, and Zulu will be installed in it.

Example to make `java.exe` available at `c:\java\zulu-8\bin\java.exe:`

```
> msixec /i zulu8.42-sa-jdk8.0.232-win_x64.msi
APPLICATIONROOTDIRECTORY="c:\java" /qn
```

- c. To install Zulu using the **Setup Wizard**, double-click the `<zulu_package>.msi` file, make your selections and click **Next** as the screens are displayed. Screens include:

1. **Welcome** page
2. **Destination Folder** page

Accept the default destination folder `C:\Program Files\Zulu\<zulu-jdk>`, or click **Change** to enter an alternative destination.

Change destination Folder window

Select or create a different directory through **Look in** option. However, if you alter the installation path, the `\<zulu_jdk>` subdirectory will not be created and selected directory becomes the root Zulu directory. For example, if you select `C:\temp`, Zulu will be installed into the `C:\temp` directory and the path to `java.exe` will be `C:\temp\bin\java.exe`.

Click **OK** to confirm your choice.

3. **Ready to install** page.

Click **Install** to start the installation.

The Zulu MSI Installer updates the PATH environment variable with Zulu installation directory and creates .jar file association with Java launcher.

Note:

For details on Windows registry keys, see [Zulu MSI Installer and Windows Registry Settings](#).

Installation on Linux Using the Zulu TAR.GZ File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package from [Azul Downloads](#).
2. Extract the downloaded `<zulu_package>.tar.gz` file.

```
$ tar -xzvf <zulu_package>.tar.gz
```

The command extracts all Zulu files into the `<zulu_package>` folder in the current directory.

3. [Verify your Zulu installation](#).

Installation on Alpine Linux Using an APK Repository

The Zulu product package is a set of installation packages.

Zulu Package Name	Description	Dependencies
zuluX	zuluX-jdk alias	None
zuluX-jre-headless	Headless Java Runtime (without GUI support)	None
zuluX-jre	Headful Java Runtime (with GUI support)	zuluX-jre-headless
zuluX-jdk-headless	Headless Java Development Kit (without GUI support)	zuluX-jre-headless
zuluX-jdk	Headful Java Development Kit (with GUI support)	zuluX-jdk-headless zuluX-jre
zuluX-src	Java Development Kit public source snapshot	None
zuluX-doc	Java Development Kit documentation (man pages)	None
zuluX-demo	Java Development Kit demos and examples	zuluX-jdk

Where **X** is the Zulu version: 15, 13, 11, and 8.

1. Log in as `root` or prefix the installation command with `sudo`.
2. Complete the steps described in [Add the APK Repository on Alpine Linux](#).
3. Install a Zulu package.

```
# apk add <zulu_package_name>
```

Example

Zulu 8

- To install *zulu8-jre-headless*, run:

```
# apk add zulu8-jre-headless
```

- To install *zulu8-jdk*, run:

```
# apk add zulu8-jdk
```

See also Azul Repository Tips in [Troubleshooting](#).

Installation on Linux Using a Yum Repository

The layered Zulu product package includes the following installation packages.

Zulu Package Name	Description	Dependencies
zuluX	zuluX-jdk alias	None
zuluX-jre-headless	Headless Java Runtime (without GUI support)	None
zuluX-jre	Headful Java Runtime (with GUI support)	zuluX-jre-headless
zuluX-jdk-headless	Headless Java Development Kit (without GUI support)	zuluX-jre-headless
zuluX-jdk	Headful Java Development Kit (with GUI support)	zuluX-jdk-headless zuluX-jre
zuluX-src	Java Development Kit public source snapshot	None
zuluX-doc	Java Development Kit documentation (man pages)	None

Zulu Package Name	Description	Dependencies
zuluX-demo	Java Development Kit demos and examples	zuluX-jdk

Where **X** is the Zulu version: 15, 13, 11, 8, or 7.

1. Log in as `root` or use `sudo`.
2. Complete the steps described in [Add the Yum Repository on a RHEL, SLES, or Oracle Linux System](#).
3. Install a Zulu package. Select the command for your environment.

- For RHEL6,7 or Oracle Linux

```
$ sudo yum install <zulu_package_name>
```

- For RHEL8

```
$ sudo dnf install <zulu_package_name>
```

- For SLES

```
$ sudo zypper install <zulu_package_name>
```

Example

Zulu 8

To install *zulu8-jre-headless* on RHEL8, run:

```
$ sudo dnf install zulu8-jre-headless
```

Installation on Linux Using an APT Repository

The layered Zulu product package is a set of installation packages.

Zulu Package Name	Description	Dependencies
zuluX	zuluX-jdk alias	None
zuluX-jre-headless	Headless Java Runtime (without GUI support)	None
zuluX-jre	Headful Java Runtime (with GUI support)	zuluX-jre-headless
zuluX-jdk-headless	Headless Java Development Kit (without GUI support)	zuluX-jre-headless
zuluX-jdk	Headful Java Development Kit (with GUI support)	zuluX-jdk-headless zuluX-jre
zuluX-src	Java Development Kit public source snapshot	None
zuluX-doc	Java Development Kit documentation (man pages)	None
zuluX-demo	Java Development Kit demos and examples	zuluX-jdk

Where **X** is the Zulu version: 15, 13, 11, 8, or 7.

1. Log in as `root` or use `sudo`.
2. Complete the steps described in [Add the APT Repository on Ubuntu or Debian](#).
3. Install a Zulu package.

```
$ sudo apt-get install <zulu_package_name>
```

Example

Zulu 11

To install *zulu11-jdk*, run:

```
$ sudo apt-get install zulu11-jdk
```

Installation on Linux Using RPM or DEB Package

This chapter explains how to install the Zulu RPM or DEB package using your OS package manager.

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. If you have not done it, go to the [Downloads](#) page and download the installation package to your machine.
2. Installing with a package manager requires root privileges. Log in to your `root` account, or prefix the commands listed below with `sudo`.
3. Install Zulu by running one of the following commands from the directory where you saved the installation package. Replace the `<zulu_package>` placeholder with an actual name of the package file.

- For RHEL or Oracle Linux:

```
$ sudo yum install <zulu_package>.rpm
```

- For SLES:

```
$ sudo zypper install <zulu_package>.rpm
```

- For Ubuntu or Debian:

```
$ sudo apt install ./<zulu_package>.deb
```

The command may ask you to confirm that you want to install the dependencies.

Installation on Linux Using the Zulu Docker Registry

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Log in to your system as `root` or use `sudo`.
2. Install Docker. See <https://docs.docker.com/installation/>.

- a. Install the Docker runtime onto your Linux system.

Example:

```
$ sudo yum install docker
```

- b. Start the Docker service in the background.

Example:

```
$ sudo service docker start
```

Alternatively, to have the Docker service start at boot time, use the

```
chkconfig docker
```

 command.

3. Locate the Zulu Docker files from the Docker registry at:
<https://registry.hub.docker.com/u/azul/zulu-openjdk/>.
4. Identify a Zulu version from the list of Docker tags. Click the **Tags** tab.
5. Run the Zulu Docker container. From your command line:

- For Ubuntu

```
$ docker run -it --rm azul/zulu-openjdk:<zulu_  
docker_tag> java -version
```

For example, to use version 7 of JDK, run the following:

```
$ docker run -it --rm azul/zulu-openjdk:7 java -  
version
```

- For Debian

```
$ docker run -it --rm azul/zulu-openjdk-debian:<zulu_
docker_tag> java -version
```

For example, to use version 7 of JDK, run the following:

```
$ docker run -it --rm azul/zulu-openjdk-debian:7 java
-version
```

- For CentOS

```
$ docker run -it --rm azul/zulu-openjdk-centos:<zulu_
docker_tag> java -version
```

The following example shows Docker checking for an existing Zulu image, then pulling the image from the repository.

```
$ docker run -it --rm azul/zulu-openjdk-centos:7 java -
version
```

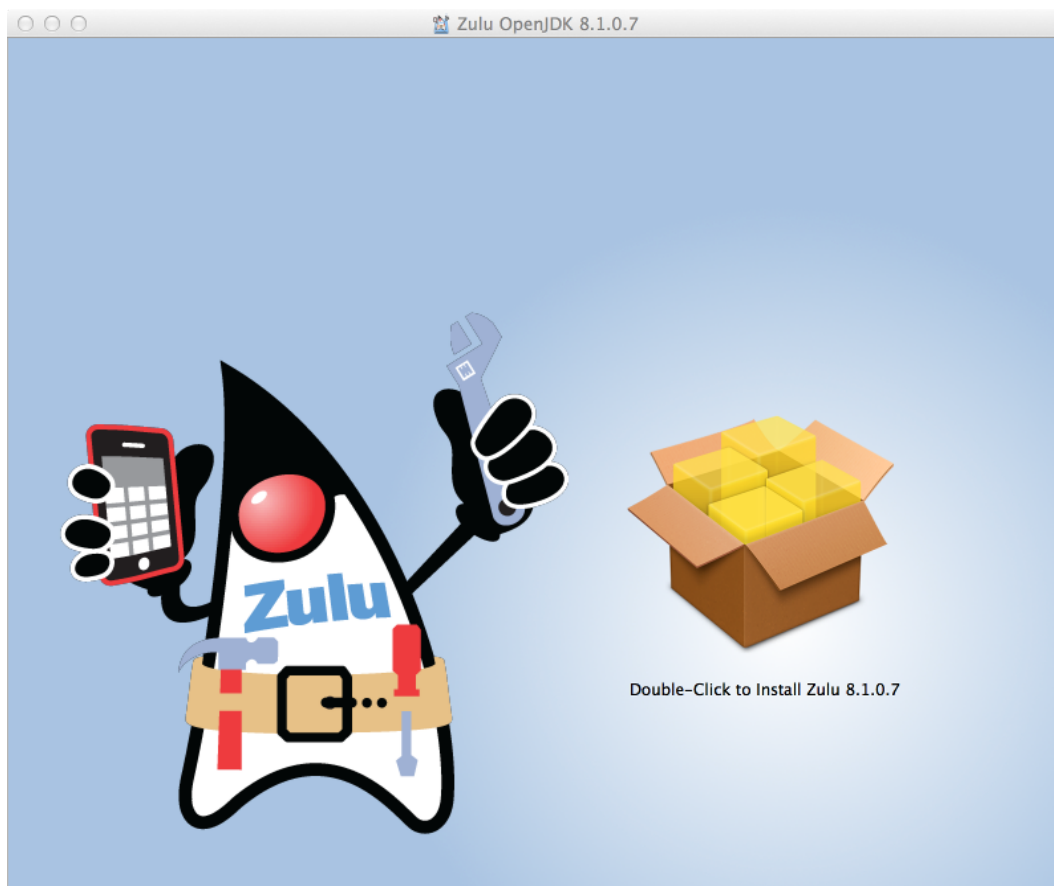
Response shows Docker checks for an existing Zulu image, then pulls the image from the repository.

```
Unable to find image 'azul/zulu-openjdk-centos:7' locally
7: Pulling from azul/zulu-openjdk-centos
ab5ef0e58194: Pull complete
b711fe66e784: Pull complete
Digest:
sha256:612957dc50cddef75301b55033366135fd376a7323de3877435
3a05d30fbea56
Status: Downloaded newer image for azul/zulu-openjdk-
centos:7
openjdk version "1.7.0_262"
OpenJDK Runtime Environment (Zulu 7.38.0.11-CA-linux64)
(build 1.7.0_262-b10)
OpenJDK 64-Bit Server VM (Zulu 7.38.0.11-CA-linux64)
(build 24.262-b10, mixed mode)
```

Installation on macOS Using a DMG Package through the GUI

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.
2. Locate, select, and double-click the `Zulu.dmg` file.
3. From the installer opening panel, double-click the box image.



4. On the Introduction statement panel, click **Continue**.



5. On the installation location panel, click **Install**.

The default location is `/Library/Java/JavaVirtualMachines`. This location is typically preconfigured into the `JAVA_HOME` environment variable.

To change the default installation location, click **Change Install Location**, select an installation drive, then click **Install**.

6. If prompted, enter your password to accept the installation on your Mac.
7. When installation is complete, the Congratulations panel is displayed. Click **Close**.

All Zulu files for the OpenJDK are expanded into appropriate folders relative to the `<zulu_dir>/<zulu_jdk>` subfolder. For example, `/Library/Java/JavaVirtualMachines/zulu-8.jdk`

Installation on macOS Using the Zulu ZIP File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.

Extract the downloaded `<zulu_package>.zip` file. Choose a method.

- From a Finder window, double-click the `<zulu_package>.zip` file.
- From a terminal command line, run:

```
$ unzip <zulu_package>.zip
```

For example, to expand the package `zulu11.41.23-ca-jdk11.0.8-macosx_x64.zip`, run:

```
$ unzip -q zulu11.41.23-ca-jdk11.0.8-macosx_x64.zip
```

All Zulu files for the OpenJDK are expanded into the corresponding folder.

For the above example, this will be the folder:

2. `~/Downloads/zulu11.41.23-ca-jdk11.0.8-macosx_x64/`

Installation on macOS Using the Zulu TAR.GZ File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.
2. Expand the downloaded `<zulu_package>.tar.gz` file. Choose a method:
 - From a Finder window, double-click the `<zulu_package>.tar.gz` file.

- From a terminal command line, run:

```
$ tar -xzvf <zulu_package>.tar.gz
```

For example, to expand the package `zulu13.33.25-ca-jdk13.0.4-macosx_x64.tar.gz`, run:

```
$ tar -xzvf zulu13.33.25-ca-jdk13.0.4-macosx_x64.tar.gz
```

All Zulu files for the OpenJDK are expanded into the corresponding folder.

For the above example, this will be the folder:

```
~/Downloads/zulu13.33.25-ca-jdk13.0.4-macosx_x64/
```

Installation on macOS Using a DMG Package through the CLI

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to your local **Downloads** folder.

Open the `Terminal.app` and use the following commands to mount the `.dmg` package.

- 2.

```
$ cd Downloads
```

```
$ hdiutil mount <zulu_package>.dmg
```

3. In the last line of the output you can find the mounted path, e.g.:

```
[...]
/dev/disk2s1      Apple_HFS      /Volumes/Zulu OpenJDK
11.33+15
```

4. Run the installer located at that path.

```
$ sudo installer -pkg "/Volumes/Zulu OpenJDK
11.33+15/Double-Click to Install Zulu 11.pkg" -
verbose -target /
```

Sample system response:

```
installer: Package name is Zulu 11.33+15
installer: Upgrading at base path /
installer: Preparing for installation.....
installer: Preparing the disk.....
installer: Preparing Zulu 11.33+15.....
installer: Waiting for other installations to complete.....
installer: Configuring the installation.....
installer:
installer: Writing files.....
installer: Writing files.....
installer: Moving items into place.....
installer:
installer: Writing package receipts.....
installer: Validating packages.....
installer: Running installer actions
installer:
installer: Finishing the Installation.....
installer: The software was successfully installed.....
installer: The upgrade was successful.
```

All Zulu files for the OpenJDK are expanded into appropriate folders relative to the `<zulu_dir>/<zulu_jdk>` subfolder. For the above example:

```
/Library/Java/JavaVirtualMachines/zulu-11.jdk
```

Installation on Solaris Using the Zulu ZIP File

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

1. Download the installation package to a reasonable location on your system.
For example:

```
/usr/java/
```

2. Extract the downloaded .zip file.

```
$ unzip <zulu_package>.zip
```

All Zulu files for the OpenJDK are extracted into appropriate folders relative to the `<zulu_dir>/<zulu_jdk>` subfolder. For example:

```
/usr/java/zulu8.27.0.8-jdk8.0.162-solaris/
```

For Zulu 7 64-bit installation you need to install it over the 32-bit version. For example

```
$ unzip zulu7.22.0.4-sa-jdk7.0.171-solaris_i686.zip
```

```
$ unzip -o zulu7.22.0.4-sa-jdk7.0.171-solaris_x64.zip
```


Post Installation Tasks

After you have installed Zulu, complete the following tasks, as needed.

- [Locate Zulu installed files](#)
- [Verify Installation](#)
- [Add Paths to Zulu](#)
- [Configure Zulu](#)

Locate Zulu Installed Files

When installed, all Zulu files are expanded into appropriate subfolders relative to the Zulu installation folder.

The Zulu installation folder includes:

- Sub folders:
 - For Zulu 8, 7: `bin`, `demo`, `include`, `jre`, `lib`, `man` and `sample`.
 - For Zulu 9 and later: `bin`, `conf`, `demo`, `include`, `jmods`, `legal`, `lib`, and `man`.
- Release information: `Welcome.html`, `readme.txt`, and `release`.
- Legal information:
 - For Zulu 8, 7: `ASSEMBLY_EXCEPTION`, `CLASSPATH_EXCEPTION_NOTE`, `DISCLAIMER`, `LICENSE`, and `THIRD_PARTY_README`.
 - For Zulu 9 and later: `DISCLAIMER` and all files from the `legal` folder.

The Zulu installation directories are:

- For Windows:

```
C:\Program Files\Zulu\<zulu_jdk>\
```

- For RHEL, SLES, Oracle Linux systems:

```
/usr/lib/jvm/<zulu_jdk>/
```

- For Ubuntu or Debian Linux systems:

```
/usr/lib/jvm/<zulu_jdk>-<architecture>/
```

- For macOS:

```
/Library/Java/JavaVirtualMachines/<zulu_jdk>.jdk/Contents/Home
```

Verify Installation

Verify Installation of Zulu. Select the instructions that are linked here based on your operating system:

- [Verify Installation on Windows](#)
- [Verify Installation on Linux](#)
- [Verify Installation on Linux from a Docker Container](#)
- [Verify Installation on macOS](#)

Verify Installation on Windows

1. Run a simple Java command.

```
> java -version
```

If needed, provide the fully qualified path.

```
<zulu_dir>\<zulu_jdk>\bin\
```

Sample command:

```
> "C:\Program Files\Zulu\zulu8\bin\java" -version
```

2. Look for the Zulu component reference in the system response. It should reflect the target OpenJDK and Zulu versions as well as the target platform, for example,

```
openjdk version "1.8.0_201"
```

```
OpenJDK Runtime Environment (Zulu 8.34.0.1-CA-win64)  
(build 1.8.0_201-b10)
```

```
OpenJDK 64-Bit Server VM (Zulu 8.34.0.1-CA-win64) (build  
25.201-b10, mixed mode)
```

Verify Installation on Linux

Complete the steps appropriate to your operating system context. These steps apply to Linux ZIP, RPM, or DEB packages and yum or APT repository installations.

1. Run a simple Java command.

```
$ java -version
```

If needed, provide the fully qualified path.

- For RHEL, SLES, or Oracle Linux

```
/usr/lib/jvm/<zulu_jdk>/bin
```

- For Ubuntu or Debian

```
/usr/lib/jvm/<zulu_jdk>/bin
```

Sample commands:

```
$ /usr/lib/jvm/zulu8/bin/java -version
```

```
$ /usr/lib/jvm/zulu8-amd64/bin/java -version
```

2. Look for the Zulu component reference in the system response. It should reflect the target OpenJDK and Zulu versions as well as the target platform, for example,

```
openjdk version "1.8.0_72"
```

```
OpenJDK Runtime Environment (Zulu 8.13.0.5-linux64) (build 1.8.0_72-b15)
```

```
OpenJDK 64-Bit Server VM (Zulu 8.13.0.5-linux64) (build 25.72-b15, mixed mode)
```

Verify Installation on Linux from a Docker Container

1. View the list of Zulu images in your Docker container.

```
$ sudo docker images
```

Response:

REPOSITORY	TAG	IMAGE ID	CREATED	VIRTUAL SIZE
azul/zulu-openjdk	8	047131b9cd77	4 days ago	416.7 MB

2. Run a Java command.

```
$ sudo docker run -it --rm azul/zulu-openjdk:<zulu_
docker_tag> java -version
```

The following sample command invokes the Java runtime in the the `/azul/zulu-openjdk:8` sandbox.

```
$ sudo docker run -it --rm azul/zulu-openjdk:8 java -
version
```

3. Look for the Zulu component reference in the system response.

If the Zulu image is not found, Docker pulls the Zulu image. Since, at this point, the Zulu image is in the Docker container, the Java runtime command is executed. For this sample command, the Java version is listed:

```
openjdk version "1.8.0_20"
```

```
OpenJDK Runtime Environment (Zulu 8.3.0.1-linux64) (build
1.8.0_20-b26)
```

```
OpenJDK 64-Bit Server VM (Zulu 8.3.0.1-linux64) (build
25.20-b23, mixed mode)
```

Verify Installation on macOS

1. From a terminal command line, run a simple Java command:

```
$ java -version
```

If you accepted the default installation path, a fully qualified path should not be needed. If you installed Zulu into a non-default path, include that path in the command line.

2. Look for the Zulu component reference in the system response. It should reflect the target OpenJDK and Zulu versions as well as the target platform, for example,

```
openjdk version "1.8.0_11"
```

```
OpenJDK Runtime Environment (Zulu 8.2.0.1-macosx) (build  
1.8.0_11-b12)
```

```
OpenJDK 64-Bit Server VM (Zulu 8.2.0.1-macosx) (build  
25.11-b03, mixed mode)
```

Verify Installation on Solaris

Complete the steps appropriate to your operating system context. These steps apply to Solaris packages and repository installations.

1. Run a simple Java command.

```
$ java -version
```

If needed, provide the fully qualified path.

Sample command:

```
$ /usr/java/zulu8.27.0.8-sa-jdk8.0.162-solaris/bin/java -
version
```

2. Look for the Zulu component reference in the system response. It should reflect the target OpenJDK and Zulu versions as well as the target platform, for example,

```
openjdk version "1.8.0_162"
```

```
OpenJDK Runtime Environment (Zulu 8.27.0.8-solaris) (build
1.8.0_162-b01)
```

```
OpenJDK 64-Bit Server VM (Zulu 8.27.0.8-solaris) (build
25.162-b01, mixed mode)
```

Add Paths to Zulu

Add Paths to Zulu and the OpenJDK directories. Select the instructions that are linked here based on your operating system:

- [Add Paths to Zulu on Windows](#)
- [Add Paths to Zulu on Linux or Solaris](#)
- [Add Paths to Zulu on macOS](#)

Add Paths to Zulu on Windows

Complete the steps appropriate to your operating system context.

- Create a `JAVA_HOME` environment variable with the path to the installation folder.

To permanently apply this change, use the **Environment Variables** dialog. See [Appendix: Updating Windows Environment Variables](#) for additional information.

To temporarily apply this change, use the set command.

```
> set JAVA_HOME="<zulu_dir>\<zulu_jdk>\"
```

Sample command:

```
> set JAVA_HOME="C:\Program Files\Zulu\zulu8\"
```

- Update your Windows `PATH` variable to prepend the path, `<zulu_dir>\<zulu_jdk>`, to the Zulu OpenJDK `\bin` folder.

```
> set PATH=%JAVA_HOME%\bin;%PATH%
```

- Add the path to your Java application .jar file to the `PATH` variable.
- Add the path to your application Class files to the `CLASSPATH` variable.

For example,

```
> set CLASSPATH=%CLASSPATH%;<path_to_myClassFile.zip>
```

Add Paths to Zulu on Linux or Solaris

Complete the steps appropriate to your operating system context. These steps apply to Solaris, Linux ZIP, RPM, or DEB packages and yum or APT repository installations.

- Update your Solaris or Linux `JAVA_HOME` variable to add path to the Zulu OpenJDK `bin`.

```
$ export JAVA_HOME=/usr/lib/jvm/<zulu_jdk>
```

Sample command:

```
$ export JAVA_HOME=/usr/lib/jvm/zulu8/
```


- Only if you want to permanently fix your path to a specific Zulu version, include your Java application `.jar` file.

```
$ export PATH=/${JAVA_HOME}/bin:$PATH
```

Add Paths to Zulu on macOS

If you installed Zulu using the `.dmg` package and accepted the default installation path, you do not need to add anything to the `JAVA_HOME` environment variable.

`JAVA_HOME` is configured to search the correct path.

If you installed Zulu using the `.zip` package, update your macOS `PATH` variable to add path to the Zulu OpenJDK:

```
$ export JAVA_HOME=/Library/Java/JavaVirtualMachines/<zulu_
package>/Contents/Home/
```

Configure Zulu

After completing the verification steps, you can configure your Zulu instance.

Specify the Java Heap Size

Use the `-Xmx` command-line option to specify the Java heap size. The Zulu instance uses the specified Java heap size to calculate the size of reserved memory.

Specify the Graphics Renderer

Starting with Zulu 8.20, Marlin is set as the default 2D Graphics renderer. However, at any time you can get back to the previous renderer, Pisces. Set up the corresponding system property as follows:

```
-Dsun.java2d.renderer=sun.java2d.pisces.PiscesRenderingEngine
```

Switching Between Java Alternatives

Zulu can have several versions installed at the same time and switch between them. For example, between Zulu 7.x and Zulu 8.x. On Windows systems, you change the `PATH` environment variable. On Linux systems, you can use `alternatives`. On macOS systems, if you have it available, you can use the `java_home` utility.

Additional information on switching between Java alternatives based on your operating systems, is available at:

- [Switch Between your Windows Zulu Installations](#)
- [Switch Between your Linux Zulu Installations](#)
- [Switch Between your macOS Zulu Installations](#)

Switch Between your Windows Zulu Installations

To switch between your Zulu installations on Windows, modify the `PATH` environment variable as follows:

1. From the Desktop, right click the **This PC** icon.
2. Select **Properties** from the context menu.
3. Click the **Advanced system settings** link.
4. Click the **Environment Variables** button. In the **System Variables** area, find the `PATH` environment variable and select it.
5. Click the **Edit** button.

In the **Variable value** field, specify the `PATH` environment variable adding it to the beginning of the string. For example, `C:\Program`

```
Files\Zulu\zulu8.12.0.1-jdk8.0.71-win_x64\bin.
```

6. Click **OK**. Close all remaining windows by clicking **OK**.

Switch Between your Linux Zulu Installations

To switch between your Linux Zulu installations, select a mode.

To set the Java version interactively:

1. Log in as `root` or use `sudo`.
2. View the Java alternatives.

```
$ sudo update-alternatives --config java
```

This command lists the Java installations and prompts to select from the choices.

Sample response

```
There are 4 programs which provide 'java'.
Selection Command
-----
1 /usr/lib/jvm/jre-1.7.0-openjdk.x86_64/bin/java
* 2 /usr/lib/jvm/zulu-6/bin/java
3 /usr/lib/jvm/zulu-7/bin/java
+ 4 /usr/lib/jvm/zulu-8/bin/java
```

3. Select a Java version, at the prompt, type a number.

```
Press enter to keep the default[*], or type selection
number:
```

Sample response

```
Using '/usr/lib/jvm/zulu-7/bin/java' to provide 'java'.
```

4. Verify the switch, check the Java version.

```
$ java -version
```

Sample response

```
openjdk version "1.7.0_65"
```

```
OpenJDK Runtime Environment (Zulu 7.6.0.1-linux64) (build
1.7.0_65-b17)
```

```
OpenJDK 64-Bit Server VM (Zulu 7.6.0.1-linux64) (build
24.65-b04, mixed mode)
```

To set the Java version by path:

- a. Log in as `root` or use `sudo`.
- b. Specify the path to use when running Java.

```
$ sudo update-alternatives --set java
/usr/lib/jvm/zulu8/bin/java
```

Switch Between your macOS Zulu Installations

To switch between Java versions, when Zulu is installed on the macOS either uses your `JAVA_HOME` environment variable or use the full path to the desired Java version when you run the java command.

1. Query the path specified in the `JAVA_HOME` environment variable using the `/usr/libexec/java_home` utility. From the terminal command line, type

```
$ /usr/libexec/java_home -v 1.7
```

```
/Library/Java/JavaVirtualMachines/1.7.0.jdk/Contents/Home
```

2. Set the `JVMVersion` parameter in the `Info.plist` file to your Zulu Java version. For example, add

```
1.7.0_65-zulu-7.6.0.1
```

3. Confirm the `JAVA_HOME` priority.

```
$ JAVA_HOME='/usr/libexec/java_home -v 1.7' java -version
openjdk version "1.7.0_65"
OpenJDK Runtime Environment (Zulu 7.6.0.1-linux64) (build
1.7.0_65-b17)
OpenJDK 64-Bit Server VM (Zulu 7.6.0.1-linux64) (build
24.65-b04, mixed mode)
```

Upgrading Zulu

Zulu can have several versions installed at the same time. You can upgrade from one version to another. For example, from Zulu 8.0 to Zulu 8.1. If you installed using the .zip file, you can have multiple versions of Zulu on the same system, provided when you actually install Zulu you use different directories.

Native installers (like MSI) work by permitting side-by-side across Java major levels, yet do inplace upgrades for point releases within a Java major level. That means within a major level, their interior folder is therefore common, and affixed to the major level version (11, 9, 8, 7, or 6.)

Zip installers can use no logic nor have any knowledge of their version, so side by side install is accomplished by always using a unique interior folder name, no matter what major or minor Java level. This also means no inplace upgrades are possible.

See the following sections:

- [Upgrade Zulu on Windows](#)
- [Upgrade Zulu on Linux using RPM or DEB Packaging](#)
- [Upgrade Zulu on Linux Using a Yum, APT, or APK Repository](#)
- [Upgrade Zulu on macOS](#)

Upgrade Zulu on Windows

Use the `.msi` installer to install more than one Zulu JDK version on your Windows system at a time. For example, you can install Zulu 7 and Zulu 8 on your Windows system, provided you install using separate installation directories, such as the default, `C:\Program Files\Zulu\<zulu_jdk>`. Each Zulu JDK version has its own installation directory: `C:\Program Files\Zulu\zulu-6\`, `C:\Program Files\Zulu\zulu-7\`, `C:\Program Files\Zulu\zulu-`

```
8\, C:\Program Files\Zulu\zulu-9\, or C:\Program  
Files\Zulu\zulu-11\.
```

Use the `.msi` installer to upgrade from one Zulu JDK to a new version of the same JDK. For example, you can upgrade Zulu 7.11 to Zulu 7.12. You cannot have multiple minor versions of the same Zulu JDK installed on your Windows system at the same time. When you upgrade a Zulu JDK version the previous version is removed. For example, if you upgrade from Zulu 7.11 to Zulu 7.12, the Zulu 7.11 version is replaced with Zulu 7.12.

You cannot upgrade mismatched Zulu JDK versions. For example, you cannot upgrade Zulu 6.9 to Zulu 8.10 using the Zulu 8.10 installer.

Upgrade Zulu on Linux Using RPM or DEB Packaging

To upgrade Zulu using RPM or DEB installation packages, install new package file as described in [Installation on Linux Using RPM or DEB Package](#).

Upgrade Zulu on Linux Using a Yum, APT, or APK Repository

You can upgrade Zulu with the available package installation tools.

If you prefer upgrading from the command line, do the following steps:

1. Log in as a user with `sudo` privileges.
2. Ensure there are no running processes that use Zulu.

For example, run `ps -ef | grep bin/java` and make sure that the output does not contain Zulu.

3. Identify the package you want to upgrade. For information on package names, see the Zulu package components list in [Layered Zulu Packages](#).
4. Run the upgrade command on your Zulu package.

- For RHEL6,7 or Oracle Linux

```
$ sudo yum upgrade <zulu_package_name>
```

- For RHEL8

```
$ sudo dnf upgrade <zulu_package_name>
```

- For SLES

```
$ sudo zypper update <zulu_package_name>
```

- For Ubuntu or Debian

```
$ sudo apt upgrade <zulu_package_name>
```

- For Alpine Linux:

```
$ sudo apk add -u <zulu_package_name>
```

Where `<zulu_package_name>` is the name of the package you want to upgrade.

Example

To upgrade `zulu8-jre-headless` on RHEL8, run:

```
$ sudo dnf upgrade zulu8-jre-headless
```

Upgrade Zulu on macOS

To upgrade a Zulu JDK version on a Mac system, run the `.dmg` installer for the newer version. The older version is replaced by a newer version.

Uninstalling Zulu

Uninstalling Zulu removes all the Zulu and Java elements, this includes the distribution elements Java runtime, Java compiler, Java class hierarchy, and the Java JAR files that comprise the core runtime (`java` and `rt.jar`), and the source compiler (`javac`, `tools.jar`), respectively.

Further information on uninstalling Zulu, based on your operating system is available at:

- [Uninstall Zulu on Windows](#)
- [Uninstall Zulu on Linux](#)
- [Uninstall Zulu on macOS](#)
- [Uninstall Zulu on Solaris](#)

Uninstall Zulu on Windows

Uninstalling Zulu removes Zulu files from the installed directories. Uninstalling one Zulu JDK version does not affect any other Zulu JDK versions you have installed. For example, removing Zulu 7 from your system does not affect a Zulu 8 version installation.

Two methods for uninstalling Zulu on Windows are available at:

- [Uninstall Zulu Extracted from a Zip File onto Windows](#)
- [Uninstall Zulu on Windows by Using an MSI File](#)

Uninstall Zulu Extracted from a Zip File onto Windows

To uninstall Zulu installed with a `.zip` package:

1. Login to your system.
2. Ensure there are no running processes using Zulu.
3. Identify the installed Zulu and Java directories and delete them.
4. If you modified your `PATH` or `CLASSPATH` environment variables, remove the Zulu path information.

Uninstall Zulu on Windows by Using an MSI File

To uninstall Zulu installed with an `.msi` package:

1. Login to your system.
2. Ensure there are no running processes using Zulu.
3. Choose a method
 - From Windows command line, run the command as Administrator.

```
> msixexec /x <zulu_package>.msi /qn
```

- From the Windows **Control Panel**, open the **Remove Program**, select **Zulu**, and click **Uninstall**.
- Run the Windows `.msi` installer, choose **Remove**, and follow the prompts to uninstall Zulu.

Uninstall Zulu on Linux

Uninstalling Zulu removes Zulu files from `/usr/lib/jvm` and `/etc/share/man/man1` directories. Three methods for uninstalling Zulu on Linux are available at:

- [Uninstall Zulu on Linux Using RHEL or Oracle Linux System Tools](#)
- [Uninstall Zulu on Linux Using RPM or DEB Packaging](#)

- [Uninstall Zulu on Linux Using a Yum, APT, or APK Repository](#)

Uninstall Zulu on Linux Using RPM or DEB Packaging

1. Login as `root` or use `sudo`.
2. Ensure there are no running processes using Zulu.
3. Run the uninstall command on your Zulu package.

To uninstall a specific version of Zulu, list the corresponding package name.

- For RHEL or Oracle Linux

```
$ sudo yum remove <zulu_package_name>
```

- For SLES

```
$ sudo zypper remove <zulu_package_name>
```

- For Ubuntu or Debian

```
$ sudo apt remove <zulu_package_name>; apt
autoremove
```

Where `<zulu_package_name>` is `zulu-<major_version>` or `zre-<major_version>`

Example

To uninstall Zulu 8 on RHEL system, run:

```
$ sudo yum remove zulu-8
```

Uninstall Zulu on Linux Using a Yum, APT, or APK Repository

1. Login as `root` or use `sudo`.
2. Ensure there are no running processes that use Zulu.

For example, run `ps -ef | grep bin/java` and make sure that the output does not contain Zulu.

3. Identify the package you want to uninstall. For information on package names, see the Zulu package components list in [Layered Zulu Packages](#).
4. Run the uninstall command on your Zulu package.

- For RHEL or Oracle Linux

```
$ sudo yum remove <zulu_package_name>
```

- For SLES

```
$ sudo zypper remove <zulu_package_name>
```

- For Ubuntu or Debian

```
$ sudo apt remove <zulu_package_name>; apt  
autoremove
```

- For Alpine Linux:

```
$ sudo apk del <zulu_package_name>
```

Where

`<zulu_package_name>` is the name of the package you want to uninstall.

Example

To uninstall *zulu8-jdk*, run:

- For RHEL or Oracle Linux

```
$ sudo yum remove zulu8-jdk
```

- For SLES

```
$ sudo zypper remove zulu8-jdk
```

- For Ubuntu or Debian

```
$ sudo apt remove zulu8-jdk
```

- For apk:

```
$ sudo apk del zulu8-jdk
```

Uninstall Zulu on macOS

To uninstall Zulu installed with a `.zip` package:

1. Ensure there are no running processes using Zulu.
2. Identify the installed Zulu and Java directories and delete them.

If you accept the default installation directory with a `.dmg` installer, the Zulu installation directory is, for example,

```
/Library/Java/JavaVirtualMachines/zulu-8.jdk
```

3. If you modified your `JAVA_HOME` variable, remove the Zulu path information.

Uninstall Zulu on Solaris

To uninstall Zulu installed with a `.zip` package:

1. Login to your system.
2. Ensure there are no running processes using Zulu.
3. Identify the installed Zulu and Java directories and delete them.
4. If you modified your `PATH` or `CLASSPATH` environment variables, remove the Zulu path information.

Using the Zulu Commercial Compatibility Kit

The Zulu Commercial Compatibility Kit (CCK) is designed to narrow the compatibility gap between OpenJDK and the closed-source Oracle JDK. Zulu is a commercialized build of OpenJDK that is compliant with the Java SE standard, is 100% open source, is freely downloadable, and as an OpenJDK distribution adheres to GPL v2 with Classpath Exception license. Certain items in the closed-source Oracle JDK (originally as the Sun Hotspot JDK) are not included with the OpenJDK project, and are not in Zulu. When Sun first released OpenJDK, many of the third party components present in the Sun Hotspot JDK, called encumbrances, required open replacements. Not all encumbrances could be made open at that time, and some persist today as potential compatibility pitfalls. The Zulu CCK is an add-on for Zulu that narrows the Oracle JDK compatibility gap, but is itself not open source, and aims to provide alternatives to the otherwise missing encumbered components.

To download the Zulu CCK, you must first acknowledge its Terms of Use. You must also adhere to the terms when applying the Zulu CCK into your Zulu deployment. The common procedure is to download and install the base Zulu using whichever packaging and deployment option you prefer, then apply the corresponding CCK packaging file that matches the base Zulu packaging you used. For Zulu ZIP packaging, use the CCK ZIPs with their internal install scripts; for native Zulu RPM, DEB, MSI or DMG installers, use the corresponding CCK native installer. One other constraint to consider is that the Zulu CCK is matched to a specific Java major version of Zulu, so CCK 8.x mates with Zulu 8.x, 7.x with 7.x, etc. The final constraint is that each Zulu CCK requires a minimum level of Zulu, because older Zulu deployments may not be detected by the CCK installer.

Additional information on using the Zulu Commercial Compatibility kit, is available at:

- [Zulu Commercial Compatibility Kit Installed Contents for Zulu JDK from 6 to 8](#)
- [Zulu Commercial Compatibility Kit Requirements for Zulu JDK from 6 to 8](#)
- [Installing the Zulu Commercial Compatibility Kit](#)
- [Enabling Rhino JavaScript Engine in Zulu 7](#)

Zulu Commercial Compatibility Kit Installed Contents for Zulu JDK from 6 to 8

Currently Zulu CCK consists of and installs the following files:

- Monotype fonts: installed into `<zulu_dir>/jre/lib/fonts`.
- Monotype license in RTF format: installed into `<zulu_dir>/etc/`.
- Zulu CCK has additional license terms. These are listed in the Zulu CCK supplemental third party license, `ThirdPartyLicenses.txt`. Review this file for Zulu CCK terms of use. This file is installed into `<zulu_dir>/etc/`.

Zulu Commercial Compatibility Kit Requirements for Zulu JDK from 6 to 8

The Zulu CCK is available for each Zulu JDK version from 6 to 8. Zulu CCK can only be installed on systems with a matching Zulu installation. The Zulu CCK version must match the Zulu version.

Install the following Zulu version prior to installing Zulu CCK:

- Zulu CCK 8 – requires Zulu 8.2 or later
- Zulu CCK 7 – requires Zulu 7.6 or later
- Zulu CCK 6 – requires Zulu 6.5 or later

Installing the Zulu Commercial Compatibility Kit

For Zulu JDK from 6 to 8, Zulu CCK is installed in addition to and after Zulu is installed. The installation process checks for a valid and corresponding Zulu version. For example, Zulu CCK 6 cannot be installed on a Zulu 7 installation. When versions checks pass, Zulu CCK applies the selected components to your installed Zulu instance.

For Zulu JDK from 6 to 8, the Zulu CCK ships in ZIP, shell script (`.sh`), and native installer packages: `.msi`, `.rpm`, and `.deb`.

Note:

The Bigelow & Holmes Lucida family fonts (Lucida Sans, Lucida Bright, and Lucida Typewriter) fonts are licensed by Azul from Monotype for internal use only. If your product plans require that these fonts be made available to your end customers, please contact your Azul representative immediately.

Prepare to Install the Zulu CCK for Zulu JDK from 6 to 8

Complete the steps for the Zulu CCK, as you would for Zulu. Referencing the following Zulu preparation steps, substitute the Zulu CCK package information for the Zulu package information.

1. Prepare for Zulu CCK on a physical or virtual machine. See [Prepare for Zulu on a Physical or Virtual Machine](#).
2. Validate the downloaded Zulu CCK package. See [Validate Downloaded Packages](#).

Install on Windows Using the Zulu CCK MSI File for Zulu JDK from 6 to 8

1. From a browser, access the Azul download page:
<http://www.azul.com/products/zulu/cck-downloads/>
2. Download the installation package to a reasonable location on your system.

For example,

```
C:\Downloads
```

3. Choose a method for running the installer.

Run the package installer as Administrator, from the Windows command line.

```
> msexec /i <cck_package>.msi /qn
```

- Use the Setup Wizard. Double-click the `<cck_package>.msi` file, make your selections and click Next as the screens are displayed.

Screens include:

- Zulu Commercial Compatibility Kit Welcome page
- Finishing Completed Zulu Commercial Compatibility Kit Setup Wizard

As part of Zulu CCK installation a Zulu version check is performed, the Zulu installation path is identified from the Registry, and Zulu CCK is integrated with Zulu. For example, if you accepted the installation defaults, Zulu CCK is integrated with Zulu at:

```
C:\Program Files\Zulu\
```

- Locate and review the Zulu CCK terms of use in the supplemental third party license file at:

```
<zulu_dir>\etc\ThirdPartyLicenses.txt
```

Install on Windows Using the Zulu CCK ZIP File for Zulu JDK from 6 to 8

The Windows Zulu CCK .zip file is a general ZIP archive. It includes a BAT file that prompts for the full path to your Zulu installation.

1. From a browser, access the Azul download page:

<http://www.azul.com/products/zulu/cck-downloads/>

2. Download the installation package to a reasonable folder on your system. For example,
3. Unzip the Zulu CCK zip package using any unzip tool.

```
C:\Downloads
```

4. Right-click `<cck_package>.zip`. For example, `zcck8.0.0.4-win_x64.zip`. Click **Extract All**.

Note:	Do not copy the files manually
--------------	--------------------------------

- b. Select a **Destination**. This can be a temporary location. For example,

```
C:\Users\<user_name>\Desktop\ZuluCCK\
```

- c. Click **Extract**.

5. Run the Zulu CCK bat file.

- a. Right-click `<cck_jdk>.bat`. For example, `zcck-8-win64.bat`. Click **Run as administrator**. Confirm allowing changes at the prompt, click **Yes**.

- b. When prompted, provide the full path to your Zulu installation.

As part of Zulu CCK installation a Zulu version check is performed. Then all Zulu CCK files are integrated into appropriate Zulu folders relative to the `<zulu_dir>\` subfolder.

6. Locate and review the Zulu CCK terms of use in the supplemental third party license file at:

```
<zulu_dir>\etc\ThirdPartyLicenses.txt
```

Install on Linux Using the Zulu CCK Shell Script for Zulu JDK from 6 to 8

1. From a browser, access the Azul download page:

<http://www.azul.com/products/zulu/cck-downloads/>

2. Download the installation package to a reasonable location on your system. For example,

```
$HOME/Zulu
```

3. Expand the downloaded .sh file.

```
$ sudo <cck_package>.sh
```

Example

```
$ sudo sh zcck8.0.0.4-linux_x64.sh
```

4. At the prompt, specify the full path to your Zulu installation. For example,

```
/usr/lib/jvm/zulu8/
```

As part of Zulu CCK installation a Zulu version check is performed. Then all Zulu CCK files are integrated into appropriate Zulu folders relative to the `<zulu_dir>/` subfolder.

5. Locate and review the Zulu CCK terms of use in the supplemental third party license file at:

```
<zulu_dir>/etc/ThirdPartyLicenses.txt
```

Install on macOS Using the Zulu CCK Shell Script for Zulu JDK from 6 to 8

Follow these listed steps if you installed Zulu on your macOS system using the ZIP file. Installing the Zulu CCK is currently not supported on a macOS system where Zulu was installed using DMG installer.

1. From a browser, access the Azul download page:

<http://www.azul.com/products/zulu/cck-downloads/>

2. Download the installation package to a reasonable location on your system. For example,

```
$HOME/Zulu
```

3. Expand the downloaded `<cck_package>.sh` file. Choose a method.

- From a Finder window, double-click the `<cck_package>.sh` file.
- From a terminal command line.

1. Expand the downloaded `.sh` file.

```
$ sudo sh <cck_package>.sh
```

Example

```
$ sudo sh zcck8.0.0.4-macosx_x64.sh
```

2. At the prompt, specify the full path to your Zulu installation. For example,

```
/usr/lib/jvm/zulu-8/
```

As part of Zulu CCK installation a Zulu version check is performed. Then all Zulu CCK files are integrated into appropriate Zulu folders relative to the `<zulu_dir>/` subfolder.

4. Locate and review the Zulu CCK terms of use in the supplemental third party license file at:

```
<zulu_dir>/etc/ThirdPartyLicenses.txt
```

Example

```
/Library/Java/JavaVirtualMachines/zulu-8.jdk/Contents/Home/etc/ThirdPartyLicenses.txt
```

Uninstalling the Zulu Commercial Compatibility Kit from Zulu JDK from 6 to 8

To uninstall the Zulu CCK, uninstall Zulu. The Zulu CCK package is uninstalled during the Zulu uninstallation. See [Uninstalling Zulu](#).

To uninstall Zulu CCK when ZIP or SH file was used for installation:

1. Delete the Zulu CCK directories.

To uninstall Zulu CCK on Windows when an MSI was used for installation:

1. Open the Control Panel and select the **Uninstall** option.
2. Scroll through the list, select the Zulu installation.
3. Right-click to see the options, and select **Remove**.

To uninstall Zulu CCK on Linux when an RPM or DEB installer was used for installation:

1. From the command line

- For RHEL, SLES, or Oracle Linux

```
$ sudo rpm -e <cck_package>
```

Example

```
$ sudo rpm -e zcck8.0.0.4-linux.x86_64.rpm
```

- For Ubuntu or Debian

```
$ sudo dpkg -r <cck_package>
```

Example

```
$ sudo dpkg -r zcck8.0.0.4-linux_amd64.deb
```

Enabling Rhino JavaScript Engine in Zulu 7

Rhino JavaScript engine is included into the Zulu CCK though it is absent from OpenJDK 7.

To enable Rhino JavaScript engine in Zulu 7, complete the following steps:

1. Install Zulu7. See [Installing Zulu](#) for details.
2. Install ZCCK 7. See [Installing the Zulu Commercial Compatibility Kit](#) for details.

This adds the following files:

- `jre/lib/azul-rhino-1.7.7.2.jar`
- `jre/lib/ext/rhino-1.7.7.2.jar`
- `jre/lib/rhino.jar`

Use the following code to verify the installation:

```
ScriptEngineManager manager = new ScriptEngineManager();  
ScriptEngine engine = manager.getEngineByName("JavaScript");
```

If `engine` is not `null` but

`com.sun.script.javascript.RhinoScriptEngine`, your installation is successful.

Appendices

This section provides additional information that is related to Zulu installation for help with the following:

- [Working with Older Versions of Zulu on Linux Platforms](#)
- [Updating Window Environment Variables](#)
- [Using the Timezone Updater Tool](#)
- [Zulu MSI Installer and Windows Registry Settings](#)

Working with Older Versions of Zulu on Linux Platforms

This section contains the archive of instructions for older versions of Zulu released before July 2020.

- [Installation on Linux Using a Yum or APT Repository](#)
- [Upgrade Zulu on Linux Using a Yum or APT Repository](#)
- [Downgrade Zulu on Linux](#)
- [Uninstall Zulu on Linux Using a Yum or APT Repository](#)

Installation on Linux Using a Yum or APT Repository

Before you install Zulu, complete the steps described in the section [Preparing the Zulu Installation Platform](#).

Install Zulu.

1. Log in as `root` or use `sudo`.
2. Complete the steps:

- For yum

[Add the Yum Repository on a RHEL, SLES, or Oracle Linux System](#)

- For APT

[Add the APT Repository on Ubuntu or Debian](#)

3. Install Zulu by using a repository. Select the command for your environment.

- For yum

```
$ sudo apt-get install <zulu_jdk>
```

Example:

```
$ sudo apt-get install zulu-8
```

- For APT

- For RHEL or Oracle Linux

```
$ yum install <zulu_jdk>
```

Example:

```
$ sudo yum install zulu-8
```

- For SLES

```
$ sudo zypper install <zulu_jdk>
```

Example:

```
$ sudo zypper install zulu-8
```

4. Links in `/usr/bin` and `/usr/share/man` are updated to point to the newly installed JVM at `<zulu_dir>/<zulu_jdk>`.

Example:

```
/usr/lib/jvm/zulu-8/bin
```

```
/usr/lib/jvm/zulu-8-amd64/bin
```

See also Azul Repository Tips in [Troubleshooting](#).

Upgrade Zulu on Linux Using a Yum or APT Repository

1. Login as `root` or use `sudo`.
2. Ensure there are no running processes using Zulu.
3. Identify the installed Zulu packages.

- For yum

```
$ yum list "zulu*"
```

- For APT

```
$ apt-cache showpkg <zulu_jdk>
```

4. Run the upgrade command on your Zulu package.

- For yum

```
$ sudo yum upgrade <zulu_jdk>
```

- For APT

```
$ sudo apt-get install <zulu_jdk>
```

Downgrade Zulu on Linux

To downgrade Zulu on a Linux system you must have installed Zulu through a yum or APT repository. A downgrade option if you used a ZIP, RPM, or DEB package is not available.

1. Log in as `root` or use `sudo`.
2. Ensure there are no running processes using Zulu.
3. Identify the installed Zulu packages.

- For yum

```
$ yum list "zulu*"
```

- For APT

```
$ apt-cache showpkg <zulu_jdk>
```

4. Run the downgrade command on your Zulu package.

- For yum

```
$ sudo yum downgrade <zulu_jdk>
```

For example

```
$ sudo yum downgrade zulu-8
```

- For APT

```
$ sudo apt-get install <zulu_jdk>-<zulu_version>
```

Uninstall Zulu on Linux Using a Yum or APT Repository

1. Login as `root` or use `sudo`.
2. Ensure there are no running processes using Zulu.
3. Identify the installed Zulu packages installed with a repository.

- For yum

```
$ yum list "zulu*"
```

- For APT

```
$ apt-cache showpkg <zulu_jdk>
```

4. Run the uninstall command on your Zulu package.

- For RHEL or Oracle Linux

```
$ sudo yum remove <zulu_jdk>
```

- For SLES

```
$ sudo zypper remove <zulu_jdk>
```

- For Ubuntu or Debian

```
$ sudo apt-get remove <zulu_jdk>
```

Updating Windows Environment Variables

Two methods for updating Windows environment variables, include using either:

- [Advanced System Properties](#)
- [setx Command](#)

Advanced System Properties

To change the environment variable settings, using the Advanced System Properties:

1. From the Windows command line, run the command:

```
> sysdm.cpl
```

2. In the dialog window that opens, select **Advanced Tab, Environment Variables**.

setx Command

To make persistent changes to your local user environment:

From the Windows command line, run the command:

```
> SETX PATH %PATH%
```

This sets the value of `PATH` in Windows Registry entries for the current user to the `CURRENT` value of the `PATH` environment variable.

For more information, see <http://ss64.com/nt/setx.html>.

Using the Timezone Updater Tool

Timezone definitions are updated automatically within every new Zulu release. When you need to deploy updated IANA (Internet Assigned Numbers Authority)-based timezone definitions onto an installed Zulu instance, we recommend to update timezone data with the ZIUpdater tool provided by Azul. Note that ZIUpdater modifies the Zulu instance that is used to execute the tool. To update timezone data successfully, ensure that you have privileges to modify [Zulu installation directories](#).

Perform the following steps:

For Java 8 and later

1. Download the ZIUpdater archive from the following location:

<https://www.azul.com/products/open-source-tools/ziupdater-time-zone-tool/>

For more details about the tool refer to the `Readme.html` file available in the archive.

2. Download the latest Time Zone Database distribution from the following location:

<https://data.iana.org/time-zones/tzdata-latest.tar.gz>

3. Stop any running instances of Zulu.
4. Run the ZIUpdater tool:

```
java -jar ziupdater-1.1.1.1.jar -l file://<path to  
tzdata archive>tzdata-latest.tar.gz
```

For example:

```
java -jar ziupdater-1.1.1.1.jar -l file:tzdata-  
latest.tar.gz
```

5. Start Zulu instances.

For Java 7

1. Download the ZIUpdater archive from the following location:

<https://www.azul.com/products/open-source-tools/ziupdater-time-zone-tool/>

For more details about the tool, refer to the `Readme.html` file available in the archive.

2. Download the latest Time Zone Database distribution from the following location:

<https://www.iana.org/time-zones/repository/tzdb-latest.tar.lz>

3. Convert the distribution to the `rearguard` format using the `make tarballs` command. See <https://mm.icann.org/pipermail/tz-announce/2018-May/000050.html> for more information.

Note:

Starting with the 2018e version, the Time Zone distribution uses a new `vanguard` format that the ZIUpdater tool does not support for Zulu 7.

4. Stop any running instances of Zulu.
5. Run the ZIUpdater tool:

```
java -jar ziupdater.jar -l file://<path to tzdata archive>/tzdata<version>-rearguard.tar.gz
```

For example:

```
java -jar ../../ziupdater-1.1.1.1.jar -l file:tzdata2020d-rearguard.tar.gz
```

6. Start Zulu instances.

ZIUpdater performs the same function as Oracle [TZUpdater](#). As a viable alternative, you may continue to use the TZUpdater tool provided by Oracle to update timezone data.

Zulu MSI Installer and Windows Registry Settings

When you install Zulu on Windows, the MSI installer uses the registry to store path and version information. The installer creates two sets of registry keys: Zulu keys and Generic keys. The second set includes registry keys that are typically created by Java installers, and it is used to ensure Zulu is compatible with other tools such as Launch4j and WinRun4J.

The table below shows the registry keys created during installation of various types of Zulu packages.

Bin-ary type	Zulu keys	Generic keys
JRE 64-bit	HKLM\SOFTWARE\Azul Systems\Zulu\zulu-<version>-jre	HKLM\SOFTWARE\JavaSoft\Java Runtime Environment
JRE 32-bit	HKLM\SOFTWARE\WOW6432Node\Azul Systems\Zulu 32-bit\zulu-<version>-jre	HKLM\SOFTWARE\WOW6432Node\JavaSoft\Java Runtime Environment
JDK 64-bit	HKLM\SOFTWARE\Azul Systems\Zulu\zulu-<version>	HKLM\SOFTWARE\JavaSoft\Java Development Kit
JDK 32-bit	HKLM\SOFTWARE\WOW6432Node\Azul Systems\Zulu 32-bit\zulu-<version>	HKLM\SOFTWARE\WOW6432Node\JavaSoft\Java Development Kit

Where `<version>` in `zulu-<version>-jre` stands for the version of Zulu, for example, `zulu-8-jre`.

The MSI Installer creates registry keys during Zulu installation and removes these keys during uninstallation. Please note that the MSI Installer creates the Generic keys only if Oracle Java is not installed on the computer. If Java is installed, this means that the keys are already created, and the installer does not overwrite them.

Support

You can contact the Zulu support team, download software, and review product documentation from the [Azul Customer Support](#) site.

You can find Zulu-related topics on the [Stack Overflow](#) website under the tags "[azure-zulu](#)" and "[azul-zulu](#)".