Installation of the JDK on macOS

This topic includes the following sections:

- System Requirements for Installing the JDK on macOS
- Determining the Default JDK Version on macOS
- Installing the JDK on macOS
- Uninstalling the JDK on macOS
- Installation FAQ on macOS Platform

System Requirements for Installing the JDK on macOS

The following are the system requirements for installing the JDK on macOS:

- Any Intel-based computer running macOS.
- Administrator privileges.

  You cannot install Java for a single user. Installing the JDK on macOS is performed on a systemwide basis for all users. Administrator privileges are required to install the JDK on macOS.

Determining the Default JDK Version on macOS

When starting a Java application through the command line, the system uses the default JDK.

There can be multiple JDKs installed on the macOS system.

You can determine which version of the JDK is the default by entering `java -version` in a Terminal window. If the installed version is 11 Interim 0, Update 0, and Patch 0, then you see a string that includes the text `11`. For example:

```
Copy$  java -version

java version "11"

Java(TM) SE Runtime Environment (build 11)

Java HotSpot(TM) 64-Bit Server VM (build 23.2-b04, mixed mode)
```

To run a different version of Java, either specify the full path, or use the `java_home` tool. For example:

```
$ /usr/libexec/java_home -v 11 --exec javac -version
```

Installing the JDK on macOS

To install the JDK on macOS:

1. Download the JDK `.dmg` file, `jdk-11.`*`interim.update.patch`*`-macosx-x64.dmg`.

   Before the file can be downloaded, you must accept the license agreement.

2. From either the browser **Downloads** window or from the file browser, double-click the `.dmg` file to start it.

   A **Finder** window appears that contains an icon of an open box and the name of the `.pkg` file.

3. Double-click the `JDK 11.pkg` icon to start the installation application.

   The installation application displays the **Introduction** window.

4. Click **Continue**.

   The **Installation Type** window appears.

5. Click **Install**.

   A window appears that displays the message: `Installer is trying to install new software. Enter your password to allow this.`

6. Enter the Administrator user name and password and click **Install Software**.

   The software is installed and a confirmation window is displayed.

After the software is installed, you can delete the `.dmg` file if you want to save disk space.

Uninstalling the JDK on macOS

To uninstall the JDK on macOS:

You must have Administrator privileges.
Note:

Do not attempt to uninstall Java by removing the Java tools from `/usr/bin`. This directory is part of the system software and any changes will be reset by Apple the next time that you perform an update of the OS.

1. Go to `/Library/Java/JavaVirtualMachines`.

2. Remove the directory whose name matches the following format by executing the `rm` command as a root user or by using the `sudo` tool:

   `/Library/Java/JavaVirtualMachines/jdk-`*`interim.update.patch`*`.jdk`

   For example, to uninstall 11 Interim 0 Update 0 Patch 0:

   ```
   $ rm -rf jdk-11.jdk
   ```

Installation FAQ on macOS Platform

This topic provides answers for the following frequently asked questions about installing JDK on macOS computers.

1. How do I find out which version of Java is the system default?

When you run a Java application from the command line, it uses the default JDK. If you do not develop Java applications, then you do not need to worry about this. See Determining the Default JDK Version on macOS.

2. How do I uninstall Java?

See Uninstalling the JDK on macOS.

3. After installing Java for macOS 2012-006, can I continue to use Apple's Java 6 alongside the macOS JDK for Java 11?

If you want to continue to develop with Java 6 using command-line, then you can modify the startup script for your favorite command environment. For bash, use this:

```
$ export JAVA_HOME=`/usr/libexec/java_home -v 11`
```

Some applications use `/usr/bin/java` to call Java. After installing Java for macOS 2012-006, `/usr/bin/java` will find the newest JDK installed, and will use that for all of the Java-related command-line tools in `/usr/bin`. You may need to modify those applications to find Java 6, or contact the developer for a newer version of the application.

4. Can I restore Apple Java after uninstalling Oracle Java?

Go back to Apple Java using the following instructions:

1. Uninstall Oracle Java by deleting the plug-in file. From a command-line, enter:

   ```
   $ sudo rm -rf "/Library/Internet Plug-Ins/JavaAppletPlugin.plugin"
   ```

2. Create a symlink using the following command, entered on a single line:

   ```
   $ sudo ln -s
   /System/Library/Java/Support/CoreDeploy.bundle/Contents/JavaAppletPlu
   gin.plugin "/Library/Internet Plug-Ins/JavaAppletPlugin.plugin"
   ```

5. What happened to the Java Preferences app in Application Utilities?

The Java Preferences app was part of the Apple Java installation and is not used by Oracle Java. Therefore, macOS releases from Apple that do not include Apple Java will not include Java Preferences.

Installation of the JDK on Microsoft Windows Platforms

This topic includes the following sections:

- System Requirements for Installing the JDK on 64-Bit Windows Platform
- JDK Installation Instruction Notation for Windows
- JDK Installation Instructions for Windows
- Beginning to Use the JDK
- Uninstalling the JDK on Windows
- JDK Installation Troubleshooting

System Requirements for Installing the JDK on 64-Bit Windows Platform

For supported processors and browsers, see Oracle JDK Certified Systems Configurations.

JDK Installation Instruction Notation for Windows

For any text in this document that contains the following notation, you must substitute the appropriate update version number:

interim.update.patch

For example, if you are downloading the JDK installer for 64-bit systems for update 11 Interim 0, Update 0, and Patch 0, then the file name `jdk-11.`*interim.update.patch*`_windows-x64_bin.exe` becomes `jdk-11_windows-x64_bin.exe`.

JDK Installation Instructions for Windows

You run a self-installing executable file to unpack and install the JDK on Windows computers.

Install JDK on Windows computers by performing the actions described in the following topics:

- Downloading the JDK Installer
- Running the JDK Installer
- Installing the JDK Silently
- Setting the PATH Environment Variable

Downloading the JDK Installer

Access Java SE Downloads page and click **Accept License Agreement**. Under the **Download** menu, click the **Download** link that corresponds to the `.exe` for your version of Windows.

Download the file `jdk-11.`*interim.update.patch*`_windows-x64_bin.exe`.

Note:

Verify the successful completion of file download by comparing the file size on the download page and your local drive. Alternatively, you can ensure that the downloaded file's checksum matches the one provided on the Java SE Downloads page.

Running the JDK Installer

You must have administrator privilege to install the JDK on Microsoft Windows.

To run the JDK installer:

1. Start the JDK 11 installer by double-clicking the installer's icon or file name in the download location.
2. Follow the instructions provided by the Installation wizard.
3. After the installation is complete, delete the downloaded file to recover the disk space.

Installing the JDK Silently

Instead of double-clicking or opening the JDK installer, you can perform a silent, non interactive, JDK installation by using command-line arguments.

The following table lists example installation scenarios and the commands required to perform them. The notation *jdk* stands for the downloaded installer file base name, such as `jdk-11_windows-x64_bin.exe`.

Installation Scenario	Command
Install JDK in silent mode.	`jdk.exe /s`
Install development tools and source code in silent mode.	`jdk.exe /s ADDLOCAL="ToolsFeature,SourceFeature"`

Setting the PATH Environment Variable

It is useful to set the `PATH` variable permanently for JDK 11 so that it is persistent after rebooting.

If you do not set the `PATH` variable, then you must specify the full path to the executable file every time that you run it. For example:

```
CopyC:\> "C:\Program Files\Java\jdk-11\bin\javac" MyClass.java
```

To set the `PATH` variable permanently, add the full path of the `jdk-11\bin` directory to the `PATH` variable. Typically, the full path is:

```
CopyC:\Program Files\Java\jdk-11\bin
```

To set the `PATH` variable on Microsoft Windows:

1. Select **Control Panel** and then **System**.
2. Click **Advanced** and then **Environment Variables**.
3. Add the location of the `bin` folder of the JDK installation to the `PATH` variable in **System Variables**.

   Note:

The `PATH` environment variable is a series of directories separated by semicolons (;) and is not case-sensitive. Microsoft Windows looks for programs in the `PATH` directories in order, from left to right.

You should only have one `bin` directory for a JDK in the path at a time. Those following the first instance are ignored.

If you are not sure where to add the JDK path, append it.

The new path takes effect in each new command window that you open after setting the `PATH` variable.

The following is a typical value for the `PATH` variable:

```
CopyC:\WINDOWS\system32;C:\WINDOWS;"C:\Program
Files\Java\jdk-11\bin"
```

Beginning to Use the JDK

Use the **Java Development Kit** in the Windows **Start** menu to access information related to Reference Documentation.

During JDK install, Java menu items are added to the Windows **Start** menu to provide easy access to Reference Documentation, which is online documentation web page.

During JDK installation and uninstallation processes, the appropriate start menu items are updated so that they are associated with the latest JDK version on the system

Note:

The Windows 7 and Windows 10 have a **Start** menu; however, the menu is not available in Windows 8 and Windows 8.1. The JDK and Java information in Windows 8 and Windows 8.1 is available in the following Start directory: `%ALLUSERSPROFILE%\Microsoft\Windows\Start Menu\Programs`.

Uninstalling the JDK on Windows

To uninstall JDK 11, use the **Add/Remove Programs** utility in the Microsoft Windows Control Panel.

Uninstalling the JDK in Silent Mode

You can use the command line for uninstalling the JDK.

Use the following command to uninstall the JDK in silent mode:

```
CopyMsiExec.exe/X{<UninstallString>}
```

For example, to uninstall JDK 11, run the command:

```
CopyMsiExec.exe/X{E973EEA1-3C61-5347-8DF7-30494D4EC697}
```

To find the UninstallString, see Finding the JDK Registry Key and UninstallString Value .

Note:

- This command can be run from anywhere.
- The `msiexec.exe` executable is located in the windows system directory.
- A reboot is required only if some files are in use during uninstallation; it is not necessary everytime. However, to manually suppress reboot while uninstalling, append `REBOOT=R` option to the command.
- Append `/l "C:\<path>setup.log"` option to the command if you want to create a log file describing the uninstallation status.

Windows Installer dialog appears prompting you for confirmation. Click **Yes** to uninstall JDK.

Finding the JDK Registry Key and UninstallString Value

1. Go to **Start** and type **Regedit**.
2. In the Registry Editor, go to `HKEY_LOCAL_MACHINE/Software/Microsoft/Windows/CurrentVersion/Uninstall`.

   Under the `Uninstall` folder, you will find many registry entries within curly brackets.
3. Click **Edit** and then **Find**.

   Note:

Highlight `Uninstall` folder before performing search for a particular registry.

4. Enter version string as value to find corresponding registry key. For example, enter **jdk-11**.

   The registry key is highlighted on the right-hand side of the pane and values of various uninstall strings are displayed on the left-hand pane.

5. Note the value of the **UninstallString**.

JDK Installation Troubleshooting

The following sections provide tips for resolving issues, if any, while installing JDK.

System Error During Decompression

If you see the error message: `system error during decompression`, then there might not be enough space on the disk that contains the `TEMP` directory.

Program Cannot Be Run in DOS Mode

If you see the error message: `This program cannot be run in DOS mode`, then do the following:

1. Open the MS-DOS shell or command prompt window.
2. Right-click the title bar.
3. Select **Properties**.
4. Select the **Program** tab.
5. Click **Advanced**.
6. Ensure that the item **Prevent MS-DOS-based programs from detecting Windows** is not selected.
7. Select **OK**.
8. Select **OK** again.
9. Exit the MS-DOS shell.
10. Restart your computer.

Characters That Are Not Part of the System Code Page

A 1722 error may occur if the installation directory is not part of the system locale's code page. To prevent this, ensure that the user and system locales are identical, and that the installation path contains only characters that are part of the system locale's code page. User and system locales can be set in the **Regional Options** or **Regional Settings** control panel.

The associated bug number is 4895647.

Cleanup the Registry After a Failed JDK Uninstall

Sometimes, attempts to uninstall JDK through the Windows **Add/Remove** program leave behind some Java entries in the registry that are not fully removed. These left behind registry entries can cause problems in installing a new version of Java. The following are the methods to cleanup registry entries:

- Program Install and Uninstall troubleshooter (Recommended Method)
- Manual Registry Edit

Program Install and Uninstall troubleshooter (Recommended Method)

Run the Program Install and Uninstall troubleshooter to repair the corrupted registry keys that prevent programs from being completely uninstalled, or blocks new installations and updates.

Manually edit the registry (Use this only if the Fix It utility does not work)

Incorrectly editing your registry may severely damage your system. You should back up any valued data from your computer before making changes to the registry.

Use the `File->Export` functionality of the registry editor to save the registry key before deleting. In case you deleted the wrong registry key, you can restore the registry from your saved backup file, by using the `File->Import` functionality.

To delete the registry key:

1. Determine the correct Registry Key. See Finding the JDK Registry Key and UninstallString Value.
2. Highlight the key, **Right click** and select **Delete**.
3. Click **Yes** when prompted.

The JDK gets uninstalled.