

Java Platform, Standard Edition

Installation Guide



Release 12
F13814-02
December 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE®

Copyright © 1993, 2020, Oracle and/or its affiliates.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

Contents

	Preface	
	Audience	v
	Documentation Accessibility	v
	Related Documents	v
	Conventions	v
1	Overview of JDK Installation	
2	Version-String Format	
3	Installation of the JDK on Linux Platforms	
	General Notes About Installing the JDK on Linux Platforms	3-1
	System Requirements for Installing JDK on Linux Platforms	3-2
	JDK Installation Instruction Notation for Linux Platforms	3-2
	Installing the JDK on Linux from Archive Binaries and RPM Packages	3-2
	Installing the 64-Bit JDK 12 on Linux Platforms	3-3
	Installing the 64-Bit JDK on RPM-Based Linux Platforms	3-3
4	Installation of the JDK on macOS	
	System Requirements for Installing the JDK on macOS	4-1
	Determining the Default JDK Version on macOS	4-1
	Installing the JDK on macOS	4-1
	Uninstalling the JDK on macOS	4-2
	Installation FAQ on macOS Platform	4-2
5	Installation of the JDK on Microsoft Windows Platforms	
	System Requirements for Installing the JDK on 64-Bit Windows Platform	5-1
	JDK Installation Instruction Notation for Windows	5-1

JDK Installation Instructions for Windows	5-1
Downloading the JDK Installer	5-2
Running the JDK Installer	5-2
Installing the JDK Silently	5-2
Setting the PATH Environment Variable	5-3
Beginning to Use the JDK	5-3
Uninstalling the JDK on Windows	5-4
JDK Installation Troubleshooting	5-4

6 Installed Directory Structure of JDK

Preface

Java Platform, Standard Edition Installation Guide describes how to install the Java Development Kit (JDK) on Linux, Microsoft Windows, and macOS computers.

Audience

This document is intended for users who are installing any variant of the Java SE platform.

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc>.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info> or visit <http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs> if you are hearing impaired.

Related Documents

See [JDK 12 Documentation](#).

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

Overview of JDK Installation

The type of installation depends on your requirement and the platform that you choose to install.

Installation Requirements and Related Tasks

To develop or run Java applications, download and install JDK.

Version-String Format

To understand the version-string scheme that is used to distinguish various JDK releases, see [Version-String Format](#).

Installation

The JDK can be installed on the following platforms:

- [Microsoft Windows](#)
- [Linux](#)
- [macOS](#)

You can download JDK from [Java SE Development Kit Downloads](#) page.

For supported processors and browsers, see [Oracle JDK Certified System Configurations](#).

The JDK documentation is a separate download. See [Java SE Documentation](#).

Microsoft Windows

[Installation of the JDK on Microsoft Windows Platforms](#): Instructions include steps to install the JDK on 64-bit Microsoft Windows operating systems.

Linux

[Installation of the JDK on Linux Platforms](#): Instructions include installation from Archive binaries and Red Hat Package Manager (RPM).

macOS

The installation of the JDK 12 requires macOS 10.11 or later.

[Installation of the JDK on macOS](#): Instructions include steps to install the JDK on the macOS platform.

Directory Structure

To learn about the files and directories that are created after JDK installation, see [Installed Directory Structure of JDK](#) .

2

Version-String Format

Java SE platform has adopted time-based release model with the JDK being released every six months.

As of JDK 10 and later, the format of the version string, which reflects the Java SE platform's time-based release model, is `$FEATURE.$INTERIM.$UPDATE.$PATCH`.

- **\$FEATURE** is the version number that is incremented for every feature release. The feature release contains new features and changes to the existing features as specified by the Java SE platform specification. The version number is incremented every six months. For example, the version number for the March 2018 release is 10, the version number for the September 2018 release is 11, and so on.
- **\$INTERIM** is the version number that is incremented for every interim release, which contains bug fixes and enhancements. An interim release does not contain incompatible changes, feature removals, nor any changes to the standard APIs. The version number for the interim release is always zero (0) as the six-month release model does not include interim releases. However, this version number is reserved for future interim releases, if any.
- **\$UPDATE** is the version number that is incremented for an update release, which includes fixes for security issues, regressions, and bugs in new features. The version number is incremented one month after the **\$FEATURE** release and every three months thereafter. For example, the full version string for the October update release is 12.0.1, the full version string for the January update release is 12.0.2, and so on.
- **\$PATCH** is the version number that is incremented for an emergency patch release to fix a critical issue.

The version string doesn't have trailing zero elements. For example, if the value of **\$FEATURE** is 12, the value of **\$INTERIM** is 0, the value of **\$UPDATE** is 1, and the value of **\$PATCH** is 0, then the full version string is 12.0.1.

3

Installation of the JDK on Linux Platforms

This topic describes the installation procedures of the JDK on the Linux platform, along with the system requirements.

This topic includes the following sections:

- [General Notes About Installing the JDK on Linux Platforms](#)
- [System Requirements for Installing JDK on Linux Platforms](#)
- [JDK Installation Instruction Notation for Linux Platforms](#)
- [Installing the JDK on Linux from Archive Binaries and RPM Packages](#)

General Notes About Installing the JDK on Linux Platforms

This topic describes general information about installing JDK 12 on Linux platforms.

Root Access

Installing the JDK automatically creates a directory called `jdk-12.interim.update.patch`.

Note:

If you install Java SE Runtime Environment in a specific location such as `/usr/jdk` (for JDK) then you must log in with root credentials to gain the necessary permissions. If you do not have root access, then either install the Java SE Runtime Environment in your home directory or a subdirectory for which you have write permissions.

Overwriting Files

If you install the software in a directory that contains a subdirectory named `jdk-12.interim.update.patch`, then the new software overwrites files of the same name in that directory. Ensure that you rename the old directory if it contains files that you would like to keep.

System Preferences

By default, the installation script configures the system such that the backing store for system preferences is created inside the JDK installation directory as applicable. If the JDK is installed on a network-mounted drive, then the system preferences can be exported for sharing with Java runtime environments on other machines.

System Requirements for Installing JDK on Linux Platforms

See [Oracle JDK Certified Systems Configurations](#) for information about supported platforms, operating systems, and browsers.

JDK Installation Instruction Notation for Linux Platforms

Instructions for installing either JDK contains a *version* notation that represents the Feature, Interim, and Update version information.

For example, if you are installing JDK 12 Interim 0, Update 0, and Patch 0, then the following string represents the name of the bundle:

```
jdk-12.interim.update.patch-linux-x64_bin.tar.gz
```

This string becomes:

```
jdk-12-linux-x64_bin.tar.gz
```

Installing the JDK on Linux from Archive Binaries and RPM Packages

You can install the Java Development Kit (JDK) on a Linux platform from archive binary files or from Red Hat Package Manager (RPM) packages.

Installation can be performed by using one of the following processes:

- **From archive binaries** (.tar.gz): This allows you to install a private version of the JDK for the current user into any location, without affecting other JDK installations. However, it may involve manual steps to get some of the features to work. See the following table for the options available for downloading and installing the JDK from archive binaries.
- **From RPM packages** (.rpm): This allows you to perform a systemwide installation of the JDK for all users, and requires root access. RPM-based Oracle Linux platforms are based on Red Hat and SuSE. See the following table for the options available for downloading and installing the JDK from RPM packages.



Note:

JDK can be used to test applications and a separate JRE is not required.

The following table lists the options and instructions for downloading and installing the JDK 12 release on a Linux platform:

Download File	Instructions	Architecture	Who Can Install
<code>jdk-12.interim.update.patch-linux-x64.tar.gz</code>	Installing the 64-Bit JDK 12 on Linux Platforms	64-bit	Anyone
<code>jdk-12.interim.update.patch-linux-x64.rpm</code>	Installing the 64-Bit JDK on RPM-Based Linux Platforms	64-bit RPM-based Linux	Root

Installing the 64-Bit JDK 12 on Linux Platforms

You can install the JDK for 64-bit Linux from an archive binary file (`.tar.gz`).

To install the 64-bit JDK on a Linux platform:

1. Download the file, `jdk-12.interim.update.patch_linux-x64_bin.tar.gz`.

Before you download a file, you must accept the license agreement. Anyone (not only by root users) can install the archive binary in any location having write access.

The `.tar.gz` archive file (also called a tarball) is a file that can be uncompressed and extracted in a single step.

2. Change the directory to the location where you want to install the JDK, then move the `.tar.gz` archive binary to the current directory.
3. Unpack the tarball and install the JDK:

```
$ tar zxvf jdk-12.interim.update.patch_linux-x64_bin.tar.gz
```

The Java Development Kit files are installed in a directory called `jdk-12.interim.update.patch`.

4. Delete the `.tar.gz` file if you want to save disk space.

Installing the 64-Bit JDK on RPM-Based Linux Platforms

You can install the JDK on 64-bit RPM-based Linux platforms, such as Red Hat and SuSE by using an RPM binary file (`.rpm`) in the system location.

Use the `sudo` command to run the RPM-based installation as a root user.

To install the 64-bit JDK on an RPM-based Linux platform:

1. Download the file, `jdk-12.interim.update.patch_linux-x64_bin.rpm`.

Before you download a file, you must accept the license agreement.

2. Ensure that you have root user access by running the command `su` and entering the superuser password.
3. Install the package using the following command:

```
$ rpm -ivh jdk-12.interim.update.patch_linux-x64_bin.rpm
```

 **Note:**

JDK 12 can coexist with earlier versions of JDK. For each version, a new directory is created, the default directory being `/usr/java/jdk-12.interim.update.patch`.

Upgrade the package using the following command:

```
$ rpm -Uvh jdk-12.interim.update.patch_linux-x64_bin.rpm
```

4. Delete the `.rpm` file if you want to save disk space.
5. Exit the root shell.

It is not required to reboot.

The JDK installation is integrated with the *alternatives* framework. After installation, the *alternatives* framework is updated to reflect the binaries from the recently installed JDK. Java commands such as `java`, `javac`, `javadoc`, and `javap` can be called from the command line.

Using the `java -version` command, users can confirm the default (recently installed) JDK version.

In addition, users can check which specific RPM package provides the `.java` files:

```
$ rpm -q --whatprovides java
```

4

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 12 Interim 0, Update 0, and Patch 0, then you see a string that includes the text 12. For example:

```
$ java -version
java version "12"
Java(TM) SE Runtime Environment (build 12)
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 12 --exec javac -version
```

Installing the JDK on macOS

To install the JDK on macOS:

1. Download the JDK .dmg file, `jdk-12.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 12.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 12 Interim 0 Update 0 Patch 0:

```
$ rm -rf jdk-12.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 12?

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 12"
```

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/JavaAppletPlugin.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.

5

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 12 Interim 0, Update 0, and Patch 0, then the file name `jdk-12.interim.update.patch_windows-x64_bin.exe` becomes `jdk-12_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-12.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 12 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-12_windows-x64_bin.exe`.

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

Setting the PATH Environment Variable

It is useful to set the `PATH` variable permanently for JDK 12 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:

```
C:\> "C:\Program Files\Java\jdk-12\bin\javac" MyClass.java
```

To set the `PATH` variable permanently, add the full path of the `jdk-12\bin` directory to the `PATH` variable. Typically, the full path is:

```
C:\Program Files\Java\jdk-12\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:

```
C:\WINDOWS\system32;C:\WINDOWS;"C:\Program Files\Java\jdk-12\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 12, use the **Add/Remove Programs** utility in the Microsoft Windows Control Panel.

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.

Source Files in Notepad

In Microsoft Windows, when you create a new file in Microsoft Notepad and then save it for the first time, Notepad usually adds the `.txt` extension to the file name. Therefore, a file that you name `Test.java` is actually saved as `Test.java.txt`. Note that you cannot see the `.txt` extension unless you turn on the viewing of file extensions (in Microsoft Windows Explorer, deselect **Hide file extensions for known file types** under **Folder Options**). To prevent the `.txt` extension, enclose the file

name in quotation marks, such as "Test.java" when entering information in the Save As dialog box.

Characters That Are Not Part of the System Code Page

It is possible to name directories using characters that are not part of the system locale's code page. If such a directory is part of the installation path, then generic error 1722 occurs, and installation is not completed. Error 1722 is a Windows installer error code. It indicates that the installation process has failed. The exact reason for this error is not known at this time.

To prevent this problem, 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.

6

Installed Directory Structure of JDK

The modular runtime image of JDK 12 contains the following folders:

 **Note:**

In macOS, the JDK installation path is `/Library/Java/JavaVirtualMachines/jdk-interim.update.patch.jdk/Contents/Home`.

`/jdk-interim.update.patch.jdk`

The root directory of the JDK software installation. This directory also contains Copyright, README, and Release files.

`/jdk-interim.update.patch.jdk/Contents/Home/`

The root directory of the JDK software installation on macOS.

`/jdk-interim.update.patch.jdk/bin`

The executable and command-line launchers that are defined by the modules linked to the image.

`/jdk-interim.update.patch.jdk/conf`

The `.properties`, `.policy`, and other configuration files intended to be edited by developers, deployers, and end users.

`/jdk-interim.update.patch.jdk/lib`

The private implementation details of the runtime system. These files are not intended for external use and must not be modified.

The `lib` directory on macOS or the `lib/$ARCH` directory on Linux contains the runtime system's dynamically linked native libraries.

`/jdk-interim.update.patch.jdk/jmods`

The compiled module definitions.

`/jdk-interim.update.patch.jdk/legal`

The copyright and license files for each module.

`/jdk-interim.update.patch.jdk/lib/src.zip`

The archive that contains the source code for the Java platform.

`/jdk-interim.update.patch.jdk/include`

The C-language header files that support native-code programming with the Java Native Interface and the Java Virtual Machine (JVM) Debugger Interface.

 **Note:**

You can install two versions of JDK simultaneously. Update version of JDK does not replace the previous version, however, a new directory is created. For example, you can install JDK 12 and JDK 12.0.1 simultaneously. In the JDK installation location, `jdk-12` and `jdk-12.0.1` directories get created.