Java Platform, Standard Edition

JDK Installation Guide

Release 11

E95199-12

July 2025

Java Platform, Standard Edition JDK Installation Guide, Release 11

E95199-12

Contents

Preface

1 Overview of JDK Installation

2 Version-String Format

3 Installation of JDK on Oracle Solaris

4 Installation of the JDK on Linux Platforms

5 Installation of the JDK on macOS

6 Installation of the JDK on Microsoft Windows Platforms

7 Installation of the JDK with a Configuration File on Microsoft Windows Platform

8 Installed Directory Structure of JDK

Preface

Java Platform, Standard Edition Installation Guide describes how to install the Java Development Kit (JDK) on Oracle Solaris, 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 11 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.
italic	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:

- Oracle Solaris
- 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.

Oracle Solaris

Installation of JDK on Oracle Solaris: Instructions include installation from Image Packaging System (IPS) and Archive binaries.

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 11 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 11.0.1, the full version string for the January update release is 11.0.2, 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 `$FEATURE` is 11, the value of `$INTERIM` is 0, the value of `$UPDATE` is 1, and the value of `$PATCH` is 0, then the full version string is 11.0.1.

3 Installation of JDK on Oracle Solaris

This topic includes the following sections:

- System Requirements for Installing the JDK on Oracle Solaris
- JDK Installation Instruction Notation for Oracle Solaris
- Installation of JDK on Oracle Solaris from Archive Binaries
- Installing JDK on Oracle Solaris 11 from IPS Packages
- Specifying the Default Java Platform

System Requirements for Installing the JDK on Oracle Solaris

JDK 11 is supported on Oracle Solaris SPARC (64-bit), version 11 or later.

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

JDK Installation Instruction Notation for Oracle Solaris

Instructions for installing the JDK contain the *interim.update.patch* notation that represents the update version number.

For JDK installation instructions that contain the notation *interim.update.patch*, substitute the appropriate update version number. For example, if you are installing JDK 11, Interim 0, Update 0, and Patch 0, then the following string represents the name of the bundle:

```
JDK-11.interim.update.patch_solaris-sparcv9_bin.tar.gz
```

This string becomes:

```
JDK-11_solaris-sparcv9_bin.tar.gz
```

Installation of JDK on Oracle Solaris from Archive Binaries

You can install JDK 11 on an Oracle Solaris platform from archive binaries (`.tar.gz`) and Image Packaging System (IPS), which is the Solaris Update Repository.

The following list provides details about the download bundle:

- Download File: `jdk-11.`*`interim.update.patch`*`_solaris-sparcv9_bin.tar.gz`
- Architecture: 64-bit SPARC
- Who Can Install: Anyone

The installation instructions for using these files are organized by the `.tar.gz` file type. This technique allows you to install a private version of the JDK for the current user in any location without affecting other JDK installations. However, it involves manual steps to get some features to work.

> **Note:**
>
> See Installing JDK on Oracle Solaris 11 from IPS Packages for the recommended approach.

Installing JDK on Oracle Solaris from Archive Binaries

You can install the JDK archive binary in any location that you can write to. It does not replace the system version of the Java platform provided by the Oracle Solaris OS.

To install JDK from archive binaries (`.tar.gz`) on Oracle Solaris:

1. Download the bundle.

   Ensure that you accept the license agreement before downloading the file. The archive binaries can be installed by anyone in any location that you can write to.

   You can simultaneously uncompress and extract the `.tar.gz` archive file (also called a tarball) in one step.

2. Change the directory to the location where you want the JDK to be installed.
3. Move the `.tar.gz` archive binaries to the current directory.
4. Unpack the tarball and install the JDK on SPARC processor:

   ```
   $ gzip -dc jdk-11.interim.update.patch_solaris-sparcv9_bin.tar.gz | tar xf
   -
   ```

   The JDK is installed in a directory called `jdk-11.`*`interim.update.patch`* in the current directory. For example, the directory for the JDK 11 Interim 0, Update 0, and Patch 0 is named:

   `jdk-11`

Installing JDK on Oracle Solaris 11 from IPS Packages

You can install JDK 11 on Oracle Solaris by using an IPS package.

To install JDK 11 on Oracle Solaris 11 by using the `jdk-11.`*`interim.update.patch`* package:

1. Ensure that the `jdk-11.`*`interim.update.patch`* package is available from your IPS publisher.

   For example:
   `$ pkg list -a jdk-11.`*`interim.update.patch`*

Name (Publisher)	Version	IFO
`developer/java/jdk-11.`*`interim.update.patch`*	`11.0.1.0.0`	i---

   - Name: the name of the package. If the publisher from which the package is installed (or available) is not the first in the publisher search order, then the publisher name is listed in parentheses after the package name.
   - Version: the release and branch versions of the package.
   - IFO:

the set of flags that display the status of the package:

  - `i` in the **I** column implies that the package is installed.
  - `f` in the **F** column implies that the package is frozen.
  - `o` in the **O** column implies that the package is obsolete.
  - `r` in the **O** column implies that the package has been renamed.

  This package is available from the Oracle Solaris publisher at `pkg.oracle.com` and also from other publisher origins. If you see a message that no such package is found, then use the `pkg publisher` command to check your publisher origin and contact your system administrator or Oracle Support representative.

2. Ensure that you have permission to install IPS packages:

   - Use the `profiles` command to list the rights profiles that are assigned to you. If you have the Software Installation rights profile, then you can use the `pfexec` command to install and update packages.

     ```
     $ pfexec pkg install jdk-11.interim.update.patch
     ```

     Other rights profiles also provide installation privileges, such as the System Administrator rights profile.

   - Depending on the security policy at your site, you might be able to use the `sudo` command with your user password to execute a privileged command:

     ```
     $ sudo pkg install jdk-11.interim.update.patch
     ```

   - Use the roles command to list the roles that are assigned to you. If you have the root role, you can use the `su` command with the root password to assume the root role:

     ```
     $ pkg install jdk-11.interim.update.patch
     ```

     > **Note:**
     >
     > The JDK 11 installation from the SVR4 package (through the `pkgadd` command) is not supported. Install JDK 11 using the IPS package.

Specifying the Default Java Platform

If there are multiple versions of Java installed on the same image, then you can specify the default or preferred Java version using `Mediator`. A `Mediator` contains a set of links to different implementations of an application, where every link has the same mediator name and link path, but different target link paths.

View the Current Default Version

Use `pkg mediator` command to display all mediators in the image and the current default Java version.

```
$ /usr/bin/pkg mediator | grep java
```

Set the Default Java Version

Use the `pkg set-mediator` command to set Java 11 as the default or preferred version.

```
$ usr/bin/pkg set-mediator -V 11.interim.update.patch java
```

4

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 from Archive Files
- General Notes About Installing the JDK on Linux Platforms from RPM packages
- System Requirements for Installing JDK on Linux Platforms
- Installing the JDK on Linux from Archive Files, Debian Packages, RPM Packages, and Oracle Linux Repositories

General Notes About Installing the JDK on Linux Platforms from Archive Files

This topic describes general information about installing JDK 11 on Linux platforms. Installing the JDK automatically creates a directory called `jdk-feature.interim.update.patch` where `feature (11)`,`interim`, `update`, and `patch` represent corresponding values of version information.

Root Access

If you install JDK in a specific location such as `/opt`, you must log in with root credentials to gain the necessary permissions. If you do not have the root access, either install the JDK in your home directory or a sub directory for which you have the write permissions.

Overwriting Files

If you install the software in a directory that contains a sub directory named `jdk-11.interim.update.patch`, 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

Installing the JDK from archive files doesn't configure the backing store for the system node of Java Preferences API (`java.util.prefs` Java package). If this functionality of the JDK is required, you need to manually configure it by making sure either `/etc/.java/.systemPrefs` or `jdk-11.interim.update.patch/.systemPrefs` directory exists.

General Notes About Installing the JDK on Linux Platforms from RPM packages

This topic describes general information about installing JDK 11 from RPM packages on Linux platforms.

You can install only one version of JDK of the same feature release. If you try to install the newer version of the same feature release while the older version exists, the installer uninstalls the older version and installs the new version. For example, you can't install `jdk-11` and `jdk-11.0.1` simultaneously. If you attempt to install `jdk-11.0.1` after `jdk-11` is installed, the installer uninstalls `jdk-11` and installs `jdk-11.0.1`.

Every update release will be installed in a separate directory, such as `/usr/lib/jvm/jdk-<VERSION>-oracle-<ARCH>` directory, where <VERSION> is a full version string (example: 23, 23.0.1), `<ARCH>` equals `x64` or `aarch64`. Additionally `/usr/java/jdk-11` symbolic link that is pointing to the installation directory is created for backward compatibility.

> **Note:**
>
> If you install an older version of a JDK when the newer version of the same feature family already exists, an error is displayed, prompting you to uninstall a newer JDK version if an older version has to be installed.

Root Access

You must log in with root credentials to install or update JDK.

System Preferences

By default, the installation script configures the system such that the backing store for the system node of Java Preferences API (java.util.prefs Java package) is configured in `/etc/.java/.systemPrefs` directory.

System Integration

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, you can confirm the default (recently installed) JDK version.

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

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

System Requirements for Installing JDK on Linux Platforms

See Oracle JDK Certified Systems Configurations for information about supported platforms, operating systems, and browsers.

Installing the JDK on Linux from Archive Files, Debian Packages, RPM Packages, and Oracle Linux Repositories

You can install the JDK on a Linux platform from archive files, debian packages, downloadable Red Hat Package Manager (RPM) packages files, or RPM packages from Oracle Linux (OL) repositories.

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

- **From archive files** (`.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 files. The bundles are available for Linux x64 and Linux aarch64 (64-bit ARM) systems. Install the required bundle.
- **From Debian packages** (`.deb`): This allows you to perform a system-wide JDK installation on Debian-based Linux platforms, and requires root access.
- **From downloadable RPM packages** (`.rpm`): This allows you to perform a system-wide JDK installation on RPM-based Linux platforms for all users, and requires root access. See the following table for the options available for downloading and installing the JDK from Linux x64 and Linux aarch64 (64-bit ARM) RPM packages.
- **From Oracle Linux repositories**: This allows you to perform a system-wide JDK installation on OL platforms starting from OL7 for all users, and requires root access. JDK RPM packages in Oracle Linux RPM repositories are available for x64 and aarch64 (64-bit ARM) OL platforms. Explicit download of JDK RPM packages is not required, the package manager will do it automatically for you.

JDK RPM packages for Generic Linux platforms and Oracle Linux platforms differ. The following table lists the differences:

Table 4-1 Difference between Generic Linux and OL platforms

Feature	Downloadable JDK RPM Package for Generic Linux Platforms	JDK RPM package for Oracle Linux platforms
Supported Linux Platforms	Any RPM-based Linux platform. For example, Red Hat, SuSE	OL platforms starting from OL7

Table 4-1 (Cont.) Difference between Generic Linux and OL platforms

Feature	Downloadable JDK RPM Package for Generic Linux Platforms	JDK RPM package for Oracle Linux platforms
Package name	JDK image is packed in a single package named `jdk-11`	JDK image is split into two packages: `jdk-11-headless` and `jdk-11-headful`. • `jdk-11-headless` is a headless Java Runtime for running non-GUI applications. • `jdk-11-headful` is headful Java Runtime and Development Tools for developing and running all types of applications. `jdk-11-headful` package requires `jdk-11-headless` package. When these packages are installed together, they provide the same functionality as the `jdk-11` package for Generic Linux Platforms.
Installation directory name	`/usr/lib/jvm/jdk-11-oracle-<ARCH>`	`/usr/lib/jvm/jdk-11-oracle-<ARCH>`
Available for download from Java SE Downloads	Yes	No. The packages are hosted in Oracle Linux RPM repositories.
Integration with the alternatives framework	`jdk-11` package registers `java` and `javac` groups with the alternatives framework.	`jdk-11-headless` package registers `java` group with the alternatives framework. `jdk-11-headful` package registers `javac` group with the alternatives framework.
Will automatically install the required 3rd party packages?	No. You need to manually install the required packages.	Yes. All required packages will be installed automatically.

Downloading the JDK Installer

Access Java SE Downloads page and click **Accept License Agreement**. Under the **Download** menu, click the **Download** link corresponding your requirement.

The following table lists the options and instructions for downloading and installing the JDK 11 release on a Linux platform:

Download File	Instructions	Architecture (CPU Type)	Who Can Install
For Linux x64 systems: `jdk-11.interim.update.patch_linux-x64.tar.gz`	Installing the 64-Bit JDK on Linux Platforms	Intel - 64-bit	Anyone
For Linux aarch64 (64-bit ARM) systems: `jdk-11.interim.update.patch_linux-aarch64.tar.gz`	Installing the 64-Bit JDK on Linux Platforms	ARM - 64-bit	Anyone

Download File	Instructions	Architecture (CPU Type)	Who Can Install
For Linux x64 systems: `jdk-11.interim.update.patch_linux-x64.deb`	Installing the 64-Bit JDK Package on Debian-based Linux Platforms	Intel - 64-bit RPM-based Linux	Root
For Linux x64 systems: `jdk-11.interim.update.patch_linux-x64.rpm`	Installing the 64-Bit JDK on RPM-Based Linux Platforms	Intel - 64-bit RPM-based Linux	Root
For Linux aarch64 (64-bit ARM) systems: `jdk-11.interim.update.patch_linux-aarch64.rpm`	Installing the 64-Bit JDK on RPM-Based Linux Platforms	ARM - 64-bit RPM-based Linux	Root

Installing the 64-Bit JDK on Linux Platforms

You can install the JDK for 64-bit Linux from an archive file (`.tar.gz`). The `.tar.gz` archive file (also called a tarball) is a file that can be uncompressed and extracted in a single step.

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

1. Download the required file:

   - For Linux x64 systems: `jdk-11.interim.update.patch_linux-x64_bin.tar.gz`. For example `jdk-11_linux-x64_bin.tar.gz`
   - For Linux aarch64 (64-bit ARM) systems: `jdk-11.interim.update.patch_linux-aarch64_bin.tar.gz`. For example, `jdk-11_linux-aarch64_bin.tar.gz`

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

2. Change the directory to the location where you want to install the JDK, then move the `.tar.gz` archive file to the current directory.

3. Unpack the tarball and install the downloaded JDK:

   ```
   $ tar zxvf jdk-11_linux-x64_bin.tar.gz
   ```

   OR

   ```
   $ tar zxvf jdk-11_linux-aarch64_bin.tar.gz
   ```

   The Java Development Kit files are installed in a directory named `jdk-11.interim.update.patch`.

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

Installing the 64-Bit JDK Package on Debian-based Linux Platforms

You can install the JDK Debian package for 64-bit Linux from the (`.deb`) file.

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

1. Download the required file:

   `jdk-11.interim.update.patch_linux-x64_bin.deb`. For example `jdk-11_linux-x64_bin.deb`

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

2. Install the package using the following command:

```
$ sudo dpkg -i jdk-11_linux-x64_bin.deb
```

The Java Development Kit files are installed into `/usr/lib/jvm/jdk-<VERSION>-oracle-<ARCH>` directory. For example, JDK 11 releases for x64 will be installed in `/usr/lib/jvm/jdk-11-oracle-x64` directory.

3. Delete the `.deb` 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 a downloadable RPM package file (`.rpm`) in the system location.

> **Note:**
>
> You can install the JDK using a downloadable RPM package file on an OL system too, but the preferred way to install the JDK on OL systems is from Oracle Linux RPM repositories.

Ensure that you have the root user access. You can do this by running the command `su` and entering the superuser password.

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

1. Download the required file:

   - For x64 systems: `jdk-11.interim.update.patch_linux-x64_bin.rpm`. For example, `jdk-11_linux-x64_bin.rpm`
   - For aarch64 (64-bit ARM) systems: `jdk-11.interim.update.patch_linux-aarch64_bin.rpm`. For example, `jdk-11_linux-aarch64_bin.rpm`

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

2. Install the required package using the following command:

   ```
   $ sudo rpm -ivh jdk-11_linux-x64_bin.rpm
   ```

   OR

   ```
   $ sudo rpm -ivh jdk-11_linux-aarch64_bin.rpm
   ```

   Upgrade the required package using the following command:

   ```
   $ sudo rpm -Uvh jdk-11_linux-x64_bin.rpm
   ```

   OR

   ```
   $ sudo rpm -Uvh jdk-11_linux-aarch64_bin.rpm
   ```

   > **Note:**
   >
   > JDK 11 can coexist with other feature releases of JDK. For each version release, a separate directory is created, the default directory being `/usr/lib/jvm/jdk-<VERSION>-oracle-<ARCH>`. For example, JDK 11 releases for x64 will be installed in `/usr/lib/jvm/jdk-11-oracle-x64` directory.

3. Delete the `.rpm` file if you want to save disk space.

   It is not required to reboot.

Installation of Public Key on RPM-based Linux Platforms

From JDK version 8, the JDK RPMs are signed with OL keys. RPMs installed on Linux distributions other than OL may display a warning message indicating that security validation of the package failed. This indicates that the public key used to sign this RPM needs to be installed in the system.

A sample warning message and the public key installation steps are as follows:

Sample Warning Message

```
jdk-11.0.23_linux-aarch64_bin.rpm: Header V3 RSA/SHA256 Signature, key
        ID ec551f03: NOKEY
```

Installation Steps

1. Download the key file from https://yum.oracle.com/RPM-GPG-KEY-oracle-ol9 using the following command:

```
wget
    https://yum.oracle.com/RPM-GPG-KEY-oracle-ol9
```

2. Install the key using the following command:

```
sudo rpm --import RPM-GPG-KEY-oracle-ol9
```

> **Note:**
>
> Public key installation needs to be done only once. The key installation is persistent across reboots.

Installing the 64-Bit JDK on Oracle Linux Platforms

There are two OL-specific JDK RPM packages, `jdk-11-headless` and `jdk-11-headful`. When these packages are installed together, they provide full JDK functionality.

Install `jdk-11-headless` package if you only need headless Java Runtime for running non-GUI applications.

Install `jdk-11-headful` package if you need full JDK functionality for running any Java application and development.

> **Note:**
>
> The `jdk-11-headful` package depends on `jdk-11-headless`, hence installing `jdk-11-headful` package will result in the automatic installation of `jdk-11-headless`, if it is not installed yet.

To install the 64-bit JDK headless on an Oracle Linux platform use either `dnf` or `yum` command:

```
sudo dnf install jdk-11-headless
```

OR

```
sudo yum install jdk-11-headless
```

To install the 64-bit JDK headful on an Oracle Linux platform use the following command:

```
sudo dnf install jdk-11-headful
```

OR

```
sudo yum install jdk-11-headful
```

5

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:

```
$  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 required JDK `.dmg` or file from `.tar.gz` page:

- macOS x64 systems: `jdk-11.`*`interim.update.patch`*`_macos-x64_bin.dmg` OR `jdk-11.`*`interim.update.patch`*`_macos-x64_bin.tar.gz`
- macOS aarch64 (64-bit ARM) systems: `jdk-11.`*`interim.update.patch`*`_macos-aarch64_bin.dmg` OR `jdk-11.`*`interim.update.patch`*`_macos-aarch64_bin.tar.gz`

Click **Accept 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` or `.tar.gz` 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-11.`*`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/
   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.

6

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

JDK installers now support only one version of any Java feature release. You can't install multiple versions of the same feature release.

For example, you can't install `jdk-11` and `jdk-11.0.1` simultaneously. If you attempt to install `jdk-11.0.1` after `jdk-11` is installed, the installer uninstalls `jdk-11` and installs `jdk-11.0.1`.

> **Note:**
>
> If you install an older version of a JDK when the newer version of the same feature family already exists, an error is displayed, prompting you to uninstall a newer JDK version if an older version has to be installed.

JDK is installed in `/Program Files/Java/jdk-`*`<FEATURE>`* where *`<FEATURE>`* is the feature release number. For example, JDK 11.0.1 is installed in `/Program Files/Java/jdk-11`.

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

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 installer.
3. After the installation is complete, delete the downloaded file to recover the disk space.

> **Note:**
>
> During installation, the following files are copied to the location mentioned:
>
> ```
> "C:\Program Files\Common Files\Oracle\Java\javapath\java.exe"
> "C:\Program Files\Common Files\Oracle\Java\javapath\javaw.exe"
> "C:\Program Files\Common Files\Oracle\Java\javapath\javac.exe"
> "C:\Program Files\Common Files\Oracle\Java\javapath\jshell.exe"
> ```

Alternatively, you can run these executables directly from the JDK's `bin` directory.

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.

Install JDK in silent mode using the command:

```
jdk.exe /s
```

Note:

The notation *jdk* stands for the downloaded installer file base name, such as `jdk-11_windows-x64_bin.exe`.

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:

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

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

```
MsiExec.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.

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.

7 Installation of the JDK with a Configuration File on Microsoft Windows Platform

You can use a configuration file to standardize JDK installations and specify options that are not available on the command line.

This topic describes the configuration file that supplements JDK command-line installation options. An installer configuration file is an alternative to, and extension of, options specified on the installer command line.

This topic includes the following sections:

- Using a Configuration File
- Configuration File Options
- Sample Configuration File

Using a Configuration File

Use the `INSTALLCFG` command-line option to specify a configuration file while installing the JDK on Windows:

```
jdk INSTALLCFG=configuration_file_path
```

- *`jdk`* is the installer base file name, for example, `jdk-11_windows-x64_bin.exe`.
- *`configuration_file_path`* is the path to the configuration file.

Configuration File Options

The Table lists the installer configuration file options that apply for Windows operating system.

> **Note:**
>
> Substitute the value of **Enable** to `1` and the value **Disable** to `0`.

Table 7-1 Configuration File Options

Option	Operating Systems	Values	Description
`INSTALL_SILENT=`	Windows	`Enable`, `Disable`	Silent (noninteractive) installation. Default: `Disable`

Table 7-1 (Cont.) Configuration File Options

Option	Operating Systems	Values	Description
`INSTALLDIR=`	Windows	*`path`*	Folder/directory into which the files are installed. Default: Operating system default path For Windows, this will work only for first time installation.
`REBOOT=`	Windows	`Enable`, `Disable`	If disabled, then the installer will never prompt you to restart your computer after installing the JDK. However, in most cases, the installer does not need to restart your computer after installing the JRE. Default: `Enable`

Sample Configuration File

The following is a sample configuration file for Windows. It specifies the following:

- Perform a silent installation.
- Install the JDK in the directory `C:\java\jdk`.

Example 7-1 Sample Configuration File for Windows

```
INSTALL_SILENT=Enable
INSTALLDIR=C:\java\jdk
```

8

Installed Directory Structure of JDK

The modular runtime image of JDK 11 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 and Solaris 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 11 and JDK 11.0.1 simultaneously. In the JDK installation location, `jdk-11` and `jdk-11.0.1` directories get created.