

JDK 17 – New Features in Java 17

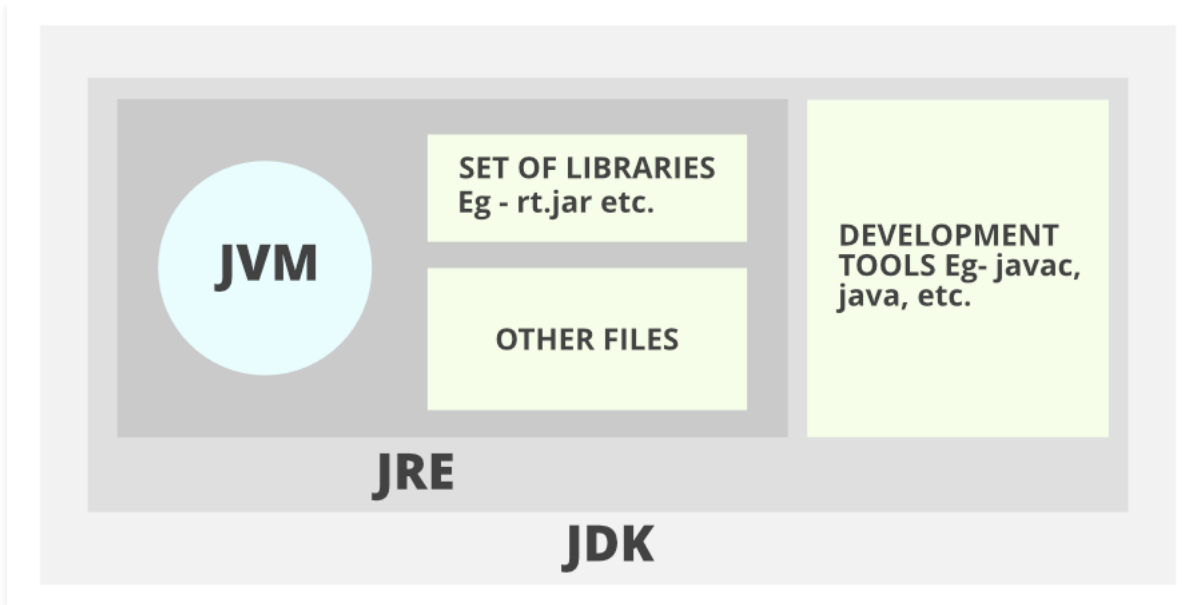
Last Updated : 05 Oct, 2021

Java 17 LTS is the latest long-term support release for the Java SE platform. JDK 17 binaries are free to use in production and free to redistribute, at no cost, under the [Oracle No-Fee Terms and Conditions License](#), where LTS stands for long-term support. It was released **on September 15, 2021**.



Have you ever wondered What do we mean by this?

Without a monotonous mindset and frequent releases by the time one gets familiar and understands operation a newer. Basically, an Open LTS version of JDK17 is out now which consists of development tools, a set of libraries, a java virtual machine, and other files. With every release, we do see new functionalities while keeping a check some of them getting altered or wiped off. Let us discuss the evolution of open LTS releases else directly switch to the installation part or update to it.



Evolution of Open LTS Java Releases

Remember the old days where we get to wait for new java releases such as Java 6 in 2006 to 2012, adhering to Java 7 in 2013 then Java 8 in 2014 where the most popular concepts lambda streams and streams were introduced. It was fine till now as with the new releases it can be interpreted that java is becoming a hot piece for the IT industry. Adhering forward now we see new java releases in very short intervals such as by 2017 we were having Java 9, by 2018 we have Java 10 and in the same year we do see Java 11. But this time java has deeper roots in the industry as most of the companies were opting for Java as their primary language for developing so do demanding skillset from employees.

As we all know that this programming language is owned, designed, and updated by **Oracle systems**, so do this company expands enormously and after the year 2018, we can plot the pattern that every year it is for sure short new releases are supposed to come so do we see Java 12, Java 13 in the single year 2019. By 2020 it becomes general agenda that we were seeing new Java LTS releases cycle is after every 6 months so do Java 14 and Java 15 came in 2020. Now in the present year, we have seen Java 16 as the latest LTS but as usual with the rapid expansion, we were supposed to see a new LTS **Java 17 which was launched on September 15, 2021** back now recently.

Most developers like it, some not from their perceptive but it becomes tedious for the enterprising world as it becomes tension for companies where after every successive 6 months new release roll out with some new feature and at the same time eradicating a few of previous features so do it was made every LTS version is accessible up to years for proper workflow. There are some companies in today's world where some companies are opting for stability and the rest opting for scalability and ease with new features been introduced.

Features Introduced in JDK17

Yes everybody is crazy for major updates in this version in order to transduce workflow but developers should be disappointed as there are not such big release update version checks as we can see from the JDK enhancement proposal commonly referred to as JEPS as provided below:

306: Restore Always-Strict Floating-Point Semantics

356: Enhanced pseudo-Random Number Generators

382: New macOS Rendering pipelines

391: macOS/AArch64 Port

398: Deprecate the Applet API for Removal

403: Strongly Encapsulated JDK Internals

406: Pattern matching for Switch(Preview)

407: Removal RMI Activation

409: Sealed Classes

410: Removal Experimental AOT and JIT Compiler

411: Deprecate the Security manager for Removal

412: Foreign Functions & memory API(Incubator)

414: Vector API(Second Incubator)

415: Context-Specific Deserialization Filters

From this, we perceive there are no significant markup changes there up in this release still getting a deeper dive to understand them for which we are dividing them into 4 categories stuffs as follows:

1. Nice Developer kinds of stuff
2. Specific developer kinds of stuff
3. Keeping up with Apple kinds of stuff
4. Cleaning up kinds of stuff

Category 1: Nice Developer

406: Pattern matching for Switch(Preview)

409: Sealed Classes

1.1 Pattern matching for the switch:

It expands the expressiveness and applicability of switch expressions and statements by allowing patterns and statements by allowing patterns to appear in case labels. It also allows the historical null-hostility of the switch to be relaxed when desired.

There are two new patterns introduced as follows:

- Guarded pattern: Uses pattern && boolean expression for further refinement
- Parenthesized pattern

1.2 Sealed Classes:

Enhances the java programming language with sealed classes and interfaces. Sealed classes and interfaces restrict which other classes or interfaces may extend or implement them.

Syntax:

```
public abstract sealed class Animal
permits Dog, Cat, Rabbit {...}
```

Category 2: Specific Developer

306: Restore Always-Strict Floating-Point Semantics

2.1 Restore Always-Strict Floating-Point Semantics:

It makes floating-point operations consistently strict.

2.2 Enhanced pseudo-Random Number Generators:

It provides a new interface type and implementations for pseudorandom number generators to make it easier to use various PRNG algorithms and to better support stream-based operations.

2.3 Strongly Encapsulated JDK Internals:

It strongly encapsulates all non-critical internal elements of the JDK

2.4 Foreign Functions & memory API(Incubator):

It introduces an API by which java programs can interpret code and data outside of the java runtime.

2.5 Vector API(Second Incubator):

It introduces an API to express vector computations that reliably compile at runtime to optimal vector instructions.

2.6 Context-Specific Deserialization Filters:

It allows applications to configure context-specific and dynamic-selected deserialization filters.

Category 3: Keeping up with Apple kinds of stuff

3.1 New macOS Rendering pipelines:

It changed the java 2D macOS rendering pipeline for macOS to use Apple Metal API instead of deprecated Apple OpenGL API.

3.2 macOS/AArch64 Port

Category 4: Cleaning up kinds of stuff

4.1 Deprecate the Applet API for Removal:

Applet API will be removed as it was deprecated since JDK9 most browsers do not support it anymore.

4.2 Removal RMI Activation:

Although RMI is still used, the RMI activation mechanism is obsolete with the web technology of the last decade.

4.3 Removal Experimental AOT and JIT Compiler:

Remove the experimental java-based ahead-of-time(AOT) and just-in-time(JIT) compiler.

4.4 Deprecate the Security manager for Removal:

Deprecate the Security Manager for removal in a future release. The Security Manager dates from java 1.0. It has not been the primary means of securing client-side java code for so many years.

Installation of JDK17

So now let us discuss how to download the latest release. In order to download the latest LTS version that means the JDK

Step 1: Surf the browser and visit the Oracle website.

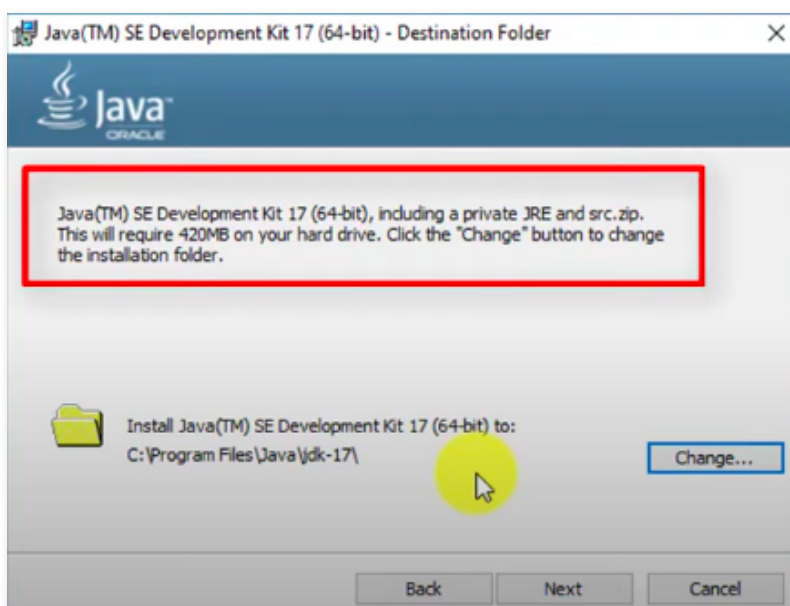


Product/File description	File size	Download
Arm 64 Compressed Archive	166.72 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.tar.gz (sha256 [2])
Arm 64 DMG Installer	166.71 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-aarch64_bin.dmg (sha256 [2])
x64 Compressed Archive	169.24 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.tar.gz (sha256 [2])
x64 DMG Installer	168.64 MB	https://download.oracle.com/java/17/latest/jdk-17_macos-x64_bin.dmg (sha256 [2])

Step 2: Download the version as per your operating systems where in order to ease for different OS refer below as follows:

- .deb for Linux
- .dmg for macOS
- .exe for Windows

Proceed to next and choose the directory where it is to be installed on the local machine and note one can see JRE in-built support during installation itself via the below snapshot so you do not need to add it separately as justified in the header image.



Step 3: Set the path for the system environment variable and in order to check whether this new JDK is installed or not, verify via terminal by entering the below commands.

```
javac --version
```

```
java --version
```

Only Java Can Get The Job Done For You.

So Strengthen Your Foundations and

Start Learning

