Java 9 Features

← prev next →

Oracle has released Java 9 with rich set of new features. It includes various upgrades to the Java programming, JVM, Tools and libraries. In this tutorial, we will discuss all the main features that are given below.

- Platform Module System (Project Jigsaw)
- Interface Private Methods
- Try-With Resources
- Anonymous Classes
- @SafeVarargs Annotation
- Collection Factory Methods
- Process API Improvement
- New Version-String Scheme
- JShell: The Java Shell (REPL)
- Process API Improvement
- Control Panel
- Stream API Improvement
- Installer Enhancement for Microsoft windows and many more

Java Platform Module System (Project Jigsaw)

It is a new kind of Java programing component that can be used to collect Java code (classes and packages). The main goal of this project is to easily scale down application to small devices. In Java 9, JDK itself has divided into set of modules to make it more lightweight. It also allows us to develop modular applications.

For more information click here

Interface Private Methods

In Java 9, we can create private methods inside an interface. Interface allows us to declare private methods that help to **share** common code between **non-abstract** methods.

Before Java 9, creating private methods inside an interface cause a compile time error. For more information click here

Try-With Resources

Java introduced **try-with-resource** feature in Java 7 that helps to close resource automatically after being used.

In other words, we can say that we don't need to close resources (file, connection, network etc) explicitly, try-with-resource close that automatically by using AutoClosable interface.

In Java 7, try-with-resources has a limitation that requires resource to declare locally within its block.

For more information click here

Anonymous Classes Improvement

Java 9 introduced a new feature that allows us to use diamond operator with anonymous classes. Using the diamond with anonymous classes was not allowed in Java 7.

In Java 9, as long as the inferred type is denotable, we can use the diamond operator when we create an anonymous inner class.

For more information click here

Java @SafeVarargs Annotation

It is an annotation which applies on a method or constructor that takes **varargs parameters**. It is used to ensure that the method does not perform unsafe operations on its varargs parameters.

It was included in Java 7 and can only be applied on

- Final methods
- Static methods
- Constructors

For more information click here

Java Collection Factory Methods

Factory methods are special type of static methods that are used to create **unmodifiable instances** of collections. It means we can use these methods to create list, set and map of small number of elements.

It is unmodifiable, so adding new element will throw **java.lang.UnsupportedOperationException**

For more information click here

Java Process API Improvement

Java has improved its process API in Java 9 version that helps to manage and control operating system processes.

In earlier versions, it was complex to manage and control operating system processes by using Java programming. Now, new classes and interfaces are added to perform this task.

For more information click here

Java New Version-String Scheme

Java version-string is a format that contains version specific information. This version-string consists of major, minor, security and patch update releases.

In Java 9, a new version-string scheme is introduced. For more information click here.

JShell: The Java Shell (REPL)

It is an interactive Java Shell tool, it allows us to execute Java code from the shell and shows output immediately. JShell is a REPL (Read Evaluate Print Loop) tool and run from the command line. It is benificial, if we want to test our business logic and get result immediately. For more information click here

Java 9 Control Panel

Java control panel is used to control Java applications that are embedded in browser. This control panel maintains the settings that manage Java application embedded in browser.

In Java 9, control panel was rewritten as a JavaFX application and the storage location has changed. For more information click here.

Java 9 Stream API Improvement

In Java 9, Stream API has improved and new methods are added to the Stream interface. TakeWhile, dropWhile and ofNullable, and one overloaded iterate method are added to perform operations on stream elements. For more information click here.

Installer Enhancement for Microsoft windows

Java 9 includes improved version of Microsoft Windows installer and added the following feature.

This installer allows us to enable and disable web deployement. We can enable web deployment by selecting Custom Setup, install and click on the checkbox from the welcome page of the installer.

Installer Enhancements for macOS

Java 9 included the following features to the macOS installer.

Feature	Description
CPU Version Availability	It provides notification on next CPU availability after uninstalling the current CPU version.
User Experience	It enhanced user experience while updating the JRE.

Add More Diagnostic Commands

Java 9 has added some new commands to improve the diagnose issues. These diagnostic command are used to diagnose Hotspot and JDK.

The Java jcmd utility can be used to send diagnostic command requests to a running Java Virtual Machine (JVM).

Remove Launch-Time JRE Version Selection

Java 9 has removed JRE (Java Runtime Environment) version selection at launch time. Now days, modern application has own active installer that further contains methods to manage the JRE. that?s why JRE version selection has been removed.

Remove the JVM TI hprof Agent

Java 9 has removed the hprof from the existed JDK. It was not intended to be a production tool. Some better featured of the hprof agent have been superseded for better alternatives.

Remove the Jhat Tool

Java has removed the jhat tool in its new release JDK 9. it was an experimental and unsupported tool added in JDK 6, now has outdated.

Validate JVM Command-Line Flag Arguments

Java validates arguments to all numeric JVM command-line flags to avoid failure. If arguments are invalid or out-of-range, it displays an appropriate error message.

Constraint check has been implemented for range and optional that require numeric value.

jlink: The Java Linker

jlink is a tool that can be used to assemble set of modules into a runtime image. It also allows us to assemble module's dependencies into the custom runtime image.

Link time is a phase between the compile and runtime, jlink works there for linking and assemble modules to runtime image.

Datagram Transport Layer Security (DTLS)

DTLS is a protocol which is used to construct TLS over datagram. JSSE (Java Secure Socket Extension) API support DTLS protocol and both versions 1.0 and 1.2.

The TLS protocol requires a TCP, So it can't be used to secure unreliable datagram traffic.

TLS Application-Layer Protocol Negotiation Extension

This extension allows the client and server in TLS connection to negotiate for application protocol. Client server communicate and inform to each other about supported application protocols. The application protocol negotiation accomplished within the TLS handshake.

OCSP Stapling for TLS

OCSP (Online Certificate Status Protocol) helps to the server in a TLS connection to check for a revoked X.509 certificate revocation. During TLS handshaking server contact an OCSP responder for the certificate. Server then staple the revocation information tot the certificate at client.

DRBG-Based SecureRandom Implementation

Java 9 includes the functionality of DRBG (Deterministic Random Bit Generator) mechanisms as specified in NIST SP 800-90Ar1 in the SecureRandom API.

The DRBG mechanisms use modern algorithms as strong as SHA-512 and AES-256. Each of these mechanisms can be configured with different security strengths and features to match user requirements.

Disable SHA-1 Certificates

The security configuration of the JDK has improved. It provides more flexible mechanism to disable X.509 certificate having SHA-1-based signatures.

The jdk.certpath.disabledAlgorithms security property is enhanced with several new constraints that allow greater control over the types of certificates that can be disabled.

Create PKCS12 Keystores by Default

The default keystore type has modified from JKS to PKCS12. The PKCS is an extensible, standard, and widely supported format for storing cryptographic keys. It improves confidentiality by storing private keys, trusted public key certificates etc. This feature also opens opportunities for interoperability with other systems such as Mozilla, Microsoft's Internet Explorer, and OpenSSL that support PKCS12.

SHA-3 Hash Algorithms

New Java version supports SHA-3 cryptographic hash functions.

The java.security.MessageDigest API supports various algorithms like: SHA3-224, SHA3-256, SHA3-384, and SHA3-512.

The following providers support SHA-3 algorithm enhancements:

- SUN provider: SHA3-224, SHA3-256, SHA3-384, and SHA3-512
- OracleUcrypto provider: SHA-3 digests supported by Solaris 12.0

Deprecate the Java Plug-in

Java Plug-in and applet technologies has deprecate in JDK 9. In future releases, these technologies will be removed.

Java Plug-in is require to run applet and JavaFX applications in web browser. It is recommended to rewrite applications as Java Web applications.

Enhanced Java Control Panel

Java control panel has improved, information is easier to locate, a search field is available and modal dialog boxes are no longer used. Note that the location of some options has changed from previous versions of the Java Control Panel.

Modular Java Application Packaging

Java new version integrates features from Project Jigsaw into the Java Packager, including module awareness and custom runtime creation.

We can use jlink tool to create smaller packages.

Simplified Doclet API

The old Doclet API is replaced with a new simplified API that leverages other standard, existing APIs. In Java 9, the standard doclet has been rewritten to use the new Doclet API.

Compiler Control

Now, we can control JVM compilation through compiler directive options. The level of control is runtime-manageable and method-specific. Compiler Control supersedes, and is backward compatible, with CompileCommand.

Segmented Code Cache

Code cache is divided into distinct segments. Each segment is a compiled code and improve performance and enable extensibility.

Unified JVM Logging

Java new version introduces a common logging system for all components of the JVM.

Remove GC Combinations Deprecated in JDK 8

Deprecated garbage collector combination has removed from JDK 9. Following are the garbage collector combination that are not supported in new version.

- DefNew + CMS
- ParNew + SerialOld
- Incremental CMS

Unified GC Logging

Garbage collection logging is reimplemented by using the unified JVM logging framework.

Deprecate the Concurrent Mark Sweep (CMS) Garbage Collector

The Concurrent Mark Sweep (CMS) garbage collector is deprecated. It issue a warning message when requested on the command line. The Garbage-First (G1) garbage collector is intended to be a replacement for most uses of CMS.

Process API Updates

Process API has improved with methods and interfaces. This API handles the operating system processes.

The ProcessHandle class is used to get process's native process ID, arguments, command, start time, accumulated CPU time, user, parent process, and descendants.

Compact Strings

In new version, Java uses more space-efficient internal representation for strings. In previous versions, the String was stored in char array and takes two bytes for each character. Now, the new internal presentation of the string is a byte class.

Platform Logging API and Service

The java.util.logging API helps to log messages, together with a service interface for consumers. An application or library can provide implementation of this service to log messages to logging framework. It uses default implementation, if no implementation is provided.

More Concurrency Updates

More concurrency updates are added to JDK 9. These updates are improved CompletableFuture API and interoperable publish-subscribe framework.

XML Catalogs

Standard XML catalog API is added which supports the organization for the Advancement of OASIS (Structured Information Standards) XML Catalogs version 1.1. This API consists of catalog-resolver which can be used as an intrinsic with the JAXP processors.

Convenience Factory Methods for Collections

This feature makes easier to create instances of collection and maps with small number of elements. Some new methods are added to List, Set and Map to create immutable instances. For example:

Set<String> alphabet = Set.of(1, 2, 3);

Enhanced Deprecation

The @Deprecated annotation is improved to provide better information about the status and intended disposition of an API. Following new elements are added.

- **@Deprecated(forRemoval=true):** It indicates that the API will be removed from the future release of Java.
- **@Deprecated(since="version"):** It contains the Java SE version string that indicates deprecared API element for Java 9.

Spin-Wait Hints

It enables Java code to inform that a spin loop is executing. A spin loop repeatedly checks to see if a condition is true, such as when a lock can be acquired, after which some computation can be safely performed followed by the release of the lock. This API is purely a hint, and carries no semantic behavior requirements.

Filter Incoming Serialization Data

It helps to filter date of incoming stream of object-serialization data to improve both robustness and security. Object-serialization clients can validate their input more easily, and exported Remote Method Invocation (RMI) objects can validate invocation arguments more easily as well.

Serialization clients implement a filter interface that is set on an ObjectInputStream. For RMI, the object is exported through a RemoteServerRef that sets the filter on the MarshalInputStream to validate the invocation arguments as they are unmarshalled.

Stack-Walking API

Java included a stack-walking API that allows lazy access of information in stack. It also allows easy filtering of information.

This stack-walking API allows access to Class objects, if the stack walker is configured.

Parser API for Nashorn

Java added Parser API which allows use to Enable applications, in server-side framework, particular IDEs etc..

It can be used to parse ECMAScript code from a string, URL, or file with methods of Parser class. Methods of this class return an object of CompilationUnitTree class, which represents ECMAScript code as an abstract syntax tree.

Nashorn parser API is located into jdk.nashorn.api.tree package.

Implement Selected ECMAScript 6 Features in Nashorn

Java added some new features to Nashorn in the 6^{th} edition of ECMA-62. Following are the Implemented features.

- Template strings
- let, const, and block scope
- Iterators and for..of loops
- Map, Set, WeakMap, and WeakSet
- Symbols
- Binary and octal literals

Prepare JavaFX UI Controls and CSS APIs for Modularization

Java included public APIs for CSS functionality and JavaFX UI controls. These functionalities previously available only through internal packages, but now can be accessible because of modular approach.

A new package javafx.scene.control.skin is included that consists of a set of classes to provide a default implementation of each UI (User Interface) control.

BeanInfo Annotations

The @beaninfo Javadoc tag is replaced with the annotation types JavaBean, BeanProperty, and SwingContainer.

We can use these attributed directly in the Bean class. It also allows auto removal for automatically created classes and set the corresponding feature attributes during BeanInfo generation at runtime.

TIFF Image I/O

TIFF (Tag Image File Format) is added for reading and writing as standard. It is located into the package javax.imageio. One more new package javax.imageio.plugins.tiff is added to provide classes that simplify the optional manipulation of TIFF metadata.

HiDPI Graphics on Windows and Linux

Automatically scales and sizes AWT and Swing components for High Dots Per Inch (HiDPI) displays on Windows and Linux.

The JDK already supports HiDPI "retina displays" on OS X.

Prior to this release, on Windows and Linux, Java applications were sized and rendered based on pixels, even on HiDPI displays that can have pixel densities two to three times as high as traditional displays. This led to GUI components and windows that were too small to read or use.

Platform-Specific Desktop Features

Some new methods are added to the java.awt.Desktop class. These methods provides the following features.

- Show custom About and Preferences windows.
- Handle requests to open or print a list of files.
- Handle requests to open a URL.
- Open the native help viewer application.
- Set the default menu bar.
- Enable or disable the application to be suddenly terminated.

Enable GTK 3 on Linux

Java new version allows Java graphical applications(JavaFX, Swing, or Abstract Window Toolkit) to use GTK version on Linux.

JDK uses GTK +2 by default, if it is not available, GTK+3 is used.

We can use specific version by setting property of jdk.gtk.version.

Unicode 8.0

Java supports Unicode 8.0 in its new Java 9 version, previously Unicode 6.2 was used.

Java 9 supports, Unicode 6.3, 7.0 and 8.0 standards that combined introduced 10,555 characters, 29 scripts, and 42 blocks.

CLDR Locale Data Enabled by Default

CLDR (Common Locale Data Repository's) represents the locale data provided by the Unicode CLDR project. It was first added in JDK 8 and now default in JDK 9.

We can enable behavior compatible with Java 8 by setting the property java.locale.providers to a value with COMPAT ahead of CLDR.

UTF-8 Properties Files

The UTF-8 is a convenient way to represent non-Latin characters. The new version of java loads properties files in UTF-8 encoding. In earlier versions, ISO-8859-1 encoding was used when loading property resource bundles.

- Java removes the endorsed-standards override mechanism and the extension mechanism.
- The **rt.jar and tools.jar** both has been removed from JRE.
- JDK's internal APIs are inaccessible by default.