# **What's New in JDK 8**

[Java Programming Language](http://docs.oracle.com/javase/8/docs/technotes/guides/language/enhancements.html#javase8)

* Lambda Expressions, a new language feature, has been introduced in this release. They enable you to treat functionality as a method argument, or code as data. Lambda expressions let you express instances of single-method interfaces (referred to as functional interfaces) more compactly.
* Method references provide easy-to-read lambda expressions for methods that already have a name.
* Default methods enable new functionality to be added to the interfaces of libraries and ensure binary compatibility with code written for older versions of those interfaces.
* Repeating Annotations provide the ability to apply the same annotation type more than once to the same declaration or type use.
* Type Annotations provide the ability to apply an annotation anywhere a type is used, not just on a declaration. Used with a pluggable type system, this feature enables improved type checking of your code.
* Improved type inference.
* Method parameter reflection.

[Collections](http://docs.oracle.com/javase/8/docs/technotes/guides/collections/changes8.html)

* Classes in the new java.util.stream package provide a Stream API to support functional-style operations on streams of elements. The Stream API is integrated into the Collections API, which enables bulk operations on collections, such as sequential or parallel map-reduce transformations.
* Performance Improvement for HashMaps with Key Collisions

[Compact Profiles](http://docs.oracle.com/javase/8/docs/technotes/guides/compactprofiles/) contain predefined subsets of the Java SE platform and enable applications that do not require the entire Platform to be deployed and run on small devices.

[Security](http://docs.oracle.com/javase/8/docs/technotes/guides/security/enhancements-8.html)

* Client-side TLS 1.2 enabled by default
* New variant of AccessController.doPrivileged that enables code to assert a subset of its privileges, without preventing the full traversal of the stack to check for other permissions
* Stronger algorithms for password-based encryption
* SSL/TLS Server Name Indication (SNI) Extension support in JSSE Server
* Support for AEAD algorithms: The SunJCE provider is enhanced to support AES/GCM/NoPadding cipher implementation as well as GCM algorithm parameters. And the SunJSSE provider is enhanced to support AEAD mode based cipher suites. See Oracle Providers Documentation, JEP 115.
* KeyStore enhancements, including the new Domain KeyStore type java.security.DomainLoadStoreParameter, and the new command option -importpassword for the keytool utility
* SHA-224 Message Digests
* Enhanced Support for NSA Suite B Cryptography
* Better Support for High Entropy Random Number Generation
* New java.security.cert.PKIXRevocationChecker class for configuring revocation checking of X.509 certificates
* 64-bit PKCS11 for Windows
* New rcache Types in Kerberos 5 Replay Caching
* Support for Kerberos 5 Protocol Transition and Constrained Delegation
* Kerberos 5 weak encryption types disabled by default
* Unbound SASL for the GSS-API/Kerberos 5 mechanism.
* SASL service for multiple host names
* JNI bridge to native JGSS on Mac OS X
* Support for stronger strength ephemeral DH keys in the SunJSSE provider
* Support for server-side cipher suites preference customization in JSSE

J[avaFX](http://docs.oracle.com/javase/8/javase-clienttechnologies.htm)

* The new Modena theme has been implemented in this release. For more information, see the blog at [fxexperience.com](http://fxexperience.com/2013/03/modena-theme-update/).
* The new SwingNode class enables developers to embed Swing content into JavaFX applications. See the [SwingNode](http://docs.oracle.com/javase/8/javafx/api/javafx/embed/swing/SwingNode.html) javadoc and [Embedding Swing Content in JavaFX Applications](http://docs.oracle.com/javase/8/javafx/interoperability-tutorial/embed-swing.htm).
* The new UI Controls include the [DatePicker](http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/DatePicker.html) and the [TreeTableView](http://docs.oracle.com/javase/8/javafx/api/javafx/scene/control/TreeTableView.html) controls.
* The javafx.print package provides the public classes for the JavaFX Printing API. See the [javadoc](http://docs.oracle.com/javase/8/javafx/api/javafx/print/package-summary.html) for more information.
* The 3D Graphics features now include 3D shapes, camera, lights, subscene, material, picking, and antialiasing. The new Shape3D (Box, Cylinder, MeshView, and Spheresubclasses), SubScene, Material, PickResult, LightBase (AmbientLight andPointLight subclasses) , and SceneAntialiasing API classes have been added to the JavaFX 3D Graphics library. The Camera API class has also been updated in this release. See the corresponding class javadoc for javafx.scene.shape.Shape3D,javafx.scene.SubScene, javafx.scene.paint.Material,javafx.scene.input.PickResult, javafx.scene.SceneAntialiasing, and the[Getting Started with JavaFX 3D Graphics](http://docs.oracle.com/javase/8/javafx/graphics-tutorial/javafx-3d-graphics.htm) document.
* The WebView class provides new features and improvements. Review [Supported Features of HTML5](http://docs.oracle.com/javase/8/javafx/embedded-browser-tutorial/index.html) for more information about additional HTML5 features including Web Sockets, Web Workers, and Web Fonts.
* Enhanced text support including bi-directional text and complex text scripts such as Thai and Hindi in controls, and multi-line, multi-style text in text nodes.
* Support for Hi-DPI displays has been added in this release.
* The CSS Styleable\* classes became public API. See the [javafx.css](http://docs.oracle.com/javase/8/javafx/api/javafx/css/package-frame.html) javadoc for more information.
* The new [ScheduledService](http://docs.oracle.com/javase/8/javafx/api/javafx/concurrent/ScheduledService.html) class allows to automatically restart the service.
* JavaFX is now available for ARM platforms. JDK for ARM includes the base, graphics and controls components of JavaFX.

[Tools](http://docs.oracle.com/javase/8/docs/technotes/tools/enhancements-8.html)

* The jjs command is provided to invoke the Nashorn engine.
* The java command launches JavaFX applications.
* The java man page has been reworked.
* The jdeps command-line tool is provided for analyzing class files.
* Java Management Extensions (JMX) provide remote access to diagnostic commands.
* The jarsigner tool has an option for requesting a signed time stamp from a Time Stamping Authority (TSA).

[Javac tool](http://docs.oracle.com/javase/8/docs/technotes/guides/javac/index.html)

* The -parameters option of the javac command can be used to store formal parameter names and enable the Reflection API to retrieve formal parameter names.
* The type rules for equality operators in the Java Language Specification (JLS) Section 15.21 are now correctly enforced by the javac command.
* The javac tool now has support for checking the content of javadoc comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when javadoc is run. The feature is enabled by the new -Xdoclint option. For more details, see the output from running "javac -X". This feature is also available in the javadoc tool, and is enabled there by default.
* The javac tool now provides the ability to generate native headers, as needed. This removes the need to run the javah tool as a separate step in the build pipeline. The feature is enabled in javac by using the new -h option, which is used to specify a directory in which the header files should be written. Header files will be generated for any class which has either native methods, or constant fields annotated with a new annotation of type java.lang.annotation.Native.

[Javadoc tool](http://docs.oracle.com/javase/8/docs/technotes/guides/javadoc/whatsnew-8.html)

* The javadoc tool supports the new DocTree API that enables you to traverse Javadoc comments as abstract syntax trees.
* The javadoc tool supports the new Javadoc Access API that enables you to invoke the Javadoc tool directly from a Java application, without executing a new process. See the [javadoc what's new](http://docs.oracle.com/javase/8/docs/technotes/guides/javadoc/whatsnew-8.html) page for more information.
* The javadoc tool now has support for checking the content of javadoc comments for issues that could lead to various problems, such as invalid HTML or accessibility issues, in the files that are generated when javadoc is run. The feature is enabled by default, and can also be controlled by the new -Xdoclint option. For more details, see the output from running "javadoc -X". This feature is also available in the javac tool, although it is not enabled by default there.

[Internationalization](http://docs.oracle.com/javase/8/docs/technotes/guides/intl/enhancements.8.html)

* Unicode Enhancements, including support for Unicode 6.2.0
* Adoption of Unicode CLDR Data and the java.locale.providers System Property
* New Calendar and Locale APIs
* Ability to Install a Custom Resource Bundle as an Extension

[Deployment](http://docs.oracle.com/javase/8/docs/technotes/guides/jweb/enhancements-8.html)

* For sandbox applets and Java Web Start applications, URLPermission is now used to allow connections back to the server from which they were started. SocketPermission is no longer granted.
* The Permissions attribute is required in the JAR file manifest of the main JAR file at all security levels.

[Date-Time Package](http://docs.oracle.com/javase/8/docs/technotes/guides/datetime/index.html) - a new set of packages that provide a comprehensive date-time model.

[Scripting](http://docs.oracle.com/javase/8/docs/technotes/guides/scripting/enhancements.html#jdk8)

* [Nashorn](http://docs.oracle.com/javase/8/docs/technotes/guides/scripting/nashorn/) Javascript Engine

[Pack200](http://docs.oracle.com/javase/8/docs/technotes/guides/pack200/enhancements.html)

* Pack200 Support for Constant Pool Entries and New Bytecodes Introduced by JSR 292
* JDK8 support for class files changes specified by JSR-292, JSR-308 and JSR-335

[IO and NIO](http://docs.oracle.com/javase/8/docs/technotes/guides/io/enhancements.html#jdk8)

* New SelectorProvider implementation for Solaris based on the Solaris event port mechanism. To use, run with the system property java.nio.channels.spi.Selector set to the value sun.nio.ch.EventPortSelectorProvider.
* Decrease in the size of the <JDK\_HOME>/jre/lib/charsets.jar file
* Performance improvement for the java.lang.String(byte[], \*) constructor and the java.lang.String.getBytes() method.

[java.lang and java.util Packages](http://docs.oracle.com/javase/8/docs/technotes/guides/lang/enhancements.html#jdk8)

* Parallel Array Sorting
* Standard Encoding and Decoding Base64
* Unsigned Arithmetic Support

[JDBC](http://docs.oracle.com/javase/8/docs/technotes/guides/jdbc/)

* The JDBC-ODBC Bridge has been removed.
* JDBC 4.2 introduces new features.

Java DB

* JDK 8 includes Java DB 10.10.

[Networking](http://docs.oracle.com/javase/8/docs/technotes/guides/net/enhancements-8.0.html)

* The class java.net.URLPermission has been added.
* In the class java.net.HttpURLConnection, if a security manager is installed, calls that request to open a connection require permission.

[Concurrency](http://docs.oracle.com/javase/8/docs/technotes/guides/concurrency/changes8.html)

* Classes and interfaces have been added to the java.util.concurrent package.
* Methods have been added to the java.util.concurrent.ConcurrentHashMap class to support aggregate operations based on the newly added streams facility and lambda expressions.
* Classes have been added to the java.util.concurrent.atomic package to support scalable updatable variables.
* Methods have been added to the java.util.concurrent.ForkJoinPool class to support a common pool.
* The java.util.concurrent.locks.StampedLock class has been added to provide a capability-based lock with three modes for controlling read/write access.

[Java XML](http://docs.oracle.com/javase/8/docs/technotes/guides/xml/enhancements.html) - [JAXP](http://docs.oracle.com/javase/8/docs/technotes/guides/xml/jaxp/enhancements-8.html)

[HotSpot](http://docs.oracle.com/javase/8/docs/technotes/guides/vm/)

* Hardware intrinsics were added to use Advanced Encryption Standard (AES). TheUseAES and UseAESIntrinsics flags are available to enable the hardware-based AES intrinsics for Intel hardware. The hardware must be 2010 or newer Westmere hardware. For example, to enable hardware AES, use the following flags:
* -XX:+UseAES -XX:+UseAESIntrinsics
* To disable hardware AES use the following flags:
* -XX:-UseAES -XX:-UseAESIntrinsics
* Removal of PermGen.
* Default Methods in the Java Programming Language are supported by the byte code instructions for method invocation.

[Java Mission Control 5.3 Release Notes](http://www.oracle.com/technetwork/java/javase/jmc53-release-notes-2157171.html)

* JDK 8 includes Java Mission Control 5.3.

Difference between

|  |  |
| --- | --- |
| **JDK 6** | **JDK 7** |
| Features added:  - Support for older win9x versions dropped.  - Scripting lang support: Generic API for integration with scripting languages, & built-in mozilla javascript rhino integration  - Dramatic performance improvements for the core platform, and swing.  - Improved web service support through JAX-WS JDBC 4.0 support  - Java compiler API: an API allowing a java program to select and invoke a java compiler programmatically.  - Upgrade of JAXB to version 2.0: including integration of a stax parser.  - Support for pluggable annotations  - Many GUI improvements, such as integration of swingworker in the API, table sorting and filtering, and true swing double-buffering (eliminating the gray-area effect). | Features Added:  - Upgrade class-loader architecture: A method that frees the underlying resources, such as open files, held by a URLClassLoader  - Concurrency and collections updates: A lightweight fork/join framework, flexible and reusable synchronization barriers, transfer queues, concurrent linked double-ended queues, and thread-local pseudo-random number generators.  - Internationalization Upgrade: Upgrade on Unicode 6.0, Locale enhancement and Separate user locale and user-interface locale.  - More new I/O APIs for the Java platform (NIO.2), NIO.2 filesystem provider for zip/jar archives, SCTP, SDP, TLS 1.2 support.  - Security & Cryptography implemented Elliptic-curve cryptography (ECC).  - Upgrade to JDBC 4.1 and Rowset 1.1.  - XRender pipeline for Java 2D, Create new platform APIs for 6u10 graphics features, Nimbus look-and-feel for Swing, Swing JLayer component, Gervill sound synthesizer.  - Upgrade the components of the XML stack to the most recent stable versions: JAXP 1.4, JAXB 2.2a, and JAX-WS 2.2.  - Enhanced MBeans." Support for dynamically-typed languages (InvokeDynamic): Extensions to the JVM, the Java language, and the Java SE API to support the implementation of dynamically-typed languages at performance levels near to that of the Java language itself  - Strict class-file checking: Class files of version 51 (SE 7) or later must be verified with the typechecking verifier; the VM must not fail over to the old inferencing verifier.  - Small language enhancements (Project Coin): A set of small language changes intended to simplify common, day-to-day programming tasks: Strings in switch statements, try-with-resources statements, improved type inference for generic instance creation ("diamond"), simplified varargs method invocation, better integral literals, and improved exception handling (multi-catch). |

**Difference between JDK 7 and JDK 8**

|  |  |
| --- | --- |
| **JDK7** | **JDK8** |
| Features Added:  -Support for dynamically-typed languages (InvokeDynamic): Extensions to the JVM, the Java language, and the Java SE API to support the implementation of dynamically-typed languages at performance levels near to that of the Java language itself  - Strict class-file checking: Class files of version 51 (SE 7) or later must be verified with the typechecking verifier; the VM must not fail over to the old inferencing verifier.  - Small language enhancements (Project Coin): A set of small language changes intended to simplify common, day-to-day programming tasks: Strings in switch statements, try-with-resources statements, improved type inference for generic instance creation (\"diamond\"), simplified varargs method invocation, better integral literals, and improved exception handling (multi-catch)  - Upgrade class-loader architecture: A method that frees the underlying resources, such as open files, held by a URLClassLoader  - Concurrency and collections updates: A lightweight fork/join framework, flexible and reusable synchronization barriers, transfer queues, concurrent linked double-ended queues, and thread-local pseudo-random number generators.  - Internationalization Upgrade: Upgrade on Unicode 6.0, Locale enhancement and Separate user locale and user-interface locale.  - More new I/O APIs for the Java platform (NIO.2), NIO.2 filesystem provider for zip/jar archives, SCTP, SDP, TLS 1.2 support.  - Security & Cryptography implemented Elliptic-curve cryptography (ECC).  - Upgrade to JDBC 4.1 and Rowset 1.1.  - XRender pipeline for Java 2D, Create new platform APIs for 6u10 graphics features, Nimbus look-and-feel for Swing, Swing JLayer component, Gervill sound synthesizer.  - Upgrade the components of the XML stack to the most recent stable versions: JAXP 1.4, JAXB 2.2a, and JAX-WS 2.2.  - Enhanced Managed Beans. | Code name is Spider. Features Added:  - JSR 335, JEP 126: Language-level support for lambda expressions.  - JSR 223, JEP 174: Project Nashorn, a JavaScript runtime which allows developers to embed JavaScript code within applications.  - JSR 308, JEP 104: Annotation on Java Types.  - Unsigned Integer Arithmetic.  - JSR 337, JEP 120: Repeating annotations.  - JSR 310, JEP 150: Date and Time API.  - JEP 178: Statically-linked JNI libraries.  - JEP 153: Launch JavaFX applications (direct launching of JavaFX application JARs).  - JEP 122: Remove the permanent generation.  - Java 8 is not supported on Windows XP. But as of JDK 8 update 5, it still can run under Windows XP after forced installation by directly unzipping from the installation executable. |