

# JDK 11: Pitfalls For The Unwary

© Copyright Azul Systems 2015

**Simon Ritter**

Deputy CTO, Azul Systems

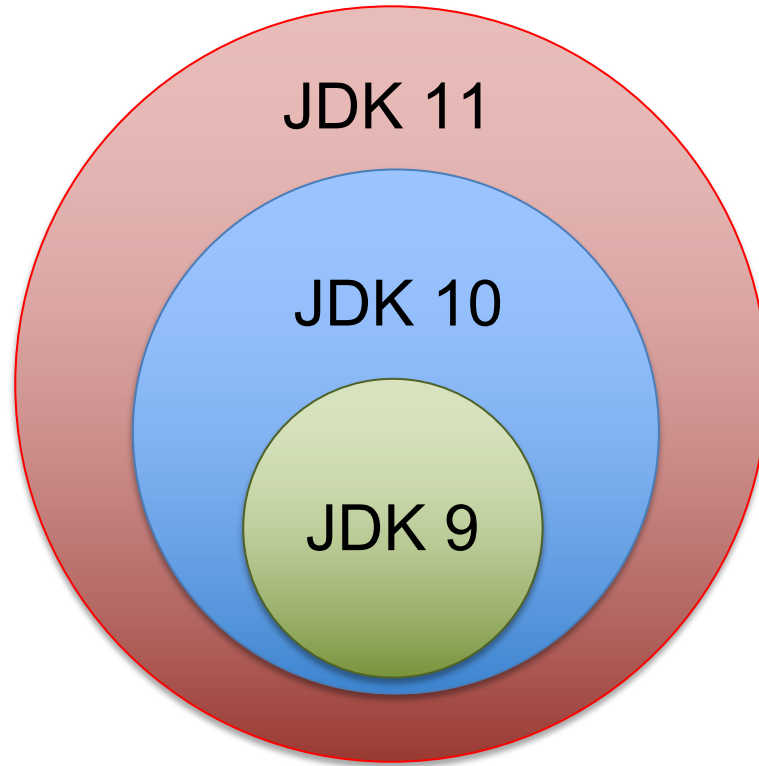
[azul.com](http://azul.com)

© Copyright Azul Systems 2017



@speakjava

# JDK 11 Functionality



# JDK 9: Big And Small Changes

# JDK 9

- Process API Updates
- HTTP 2 Client
- Improve Contended Locking
- Unified JVM Logging
- Compiler Control
- Variable Handles
- Segmented Code Cache
- Smart Java Compilation, Phase 1
- The Modular JDK
- Modular Source Code
- Elide Deprecation Warnings on Import Statements
- Resolve Lint and Doclint Warnings
- Milling Project Coin
- Remove GC Combinations Deprecated in JDK 8
- Tiered Attribution for javac
- Process Import Statements Correctly
- Annotations Pipeline 2.0
- Datagram Transport Layer Security (DTLS)
- Modular Run-Time Image
- Simplified Doclet API
- jschell: The Java Shell (Read-Eval-Print Loop)
- New Version-String Scheme
- HTML5 Javadoc
- Javadoc Search
- UTF-8 Property Files
- Unicode 7.0
- Add More Diagnostic Commands
- Create PKCS12 Keystores by Default
- Remove Launch-Time JRE Version Selection

- Improve Secure Application Performance
- Generate Run-Time Compiler Tests Automatically
- Test Class-File Attributes Generated by javac
- Parser API for Nashorn
- Linux/AArch64 Port
- Multi-Release JAR Files
- Remove the JVM TI hprof Agent
- Remove the jhat Tool
- Remove the JVM Compiler Access API
- Remove the Negotiated Compression
- Validate Command-Line Arguments
- Leverage Constructive Grammar Annotations (CGA)
- Compile for Platform-Specific Instructions
- Make GC Default G1
- OCSF Support for TLS
- Store Image Strings in Searchable Format
- Multi-Resolution Image Support
- Use Scalable Data Layout
- Use JavaFX UI Controls for SS APIs for Authorization
- Escape Strings
- Merge Selected Xerces 2.12 Fixes into JAXP
- BeanInfo Annotations
- Update JavaFX/Media to Newer Version of GStreamer
- HarfBuzz Font-Layout Engine
- Stack-Walking API
- Encapsulate Most Internal APIs
- Module System
- TIFF Image I/O
- HiDPI Graphics on Windows and Linux

- Platform Logging API and Service
- Marlin Graphics Renderer
- More Concurrency Updates
- Unicode 8.0
- XML Catalogs
- Convenience Factory Methods for Collections
- Reserved Stack Areas for Critical Sections
- Unified Class-File Format
- Platform-Specific Features
- DRBG and SecureRandom Implementations
- Enhance Method Handles
- Modular Java Application Packaging
- Dynamic Linking of Libraries and Defined Object Models
- Enhance Object Model
- Add Support for Weak Objects in G1
- Improve Test Failure Tracing and Shooting
- Indify String Concatenation
- HotSpot C++ Unit-Test Framework
- Link The Java Linker
- Enable Java Linker
- New HotSpot System
- Spin-Wait Hints
- SHA-3 Hash Algorithms
- Disable SHA-1 Certificates
- Deprecate the Applet API
- Filter Incoming Serialization Data
- Implement Selected ECMAScript 6 Features in Nashorn
- Linux/s390x Port

# JDK Compatibility

- Compatibility has always been very important
  - Minor issues in JDK 1.4 and JDK 5
- Deprecation introduced in JDK 1.1
  - JDK 8 has 492 deprecated API elements
  - None had ever been removed
  - At least one release warning of removal
- JDK 9 starts an overdue cleanup of the Java platform
  - APIs and features removed
  - Process will continue in future releases



# JDK 9 Onwards And Compatibility

*If your code only uses  
standard Java SE APIs,  
then it will **most likely**  
work without change.*

Mark Reinhold  
Chief Architect of the OpenJDK

# Migration Guidance From Oracle

"Clean applications that just depend on java.se  
*should just work*" - *Oracle Product Management*

# Module System



# Java Platform Module System (JPMS)

- The core Java libraries are now a set of modules (JEP 220)
  - 75 OpenJDK modules: 27 Java SE, 48 JDK
  - Oracle JDK: 14 additional JDK, 8 JavaFX, 2 Oracle specific
- Most internal APIs now encapsulated (JEP 260)
  - `sun.misc.Unsafe`
  - Some can be used with command line options

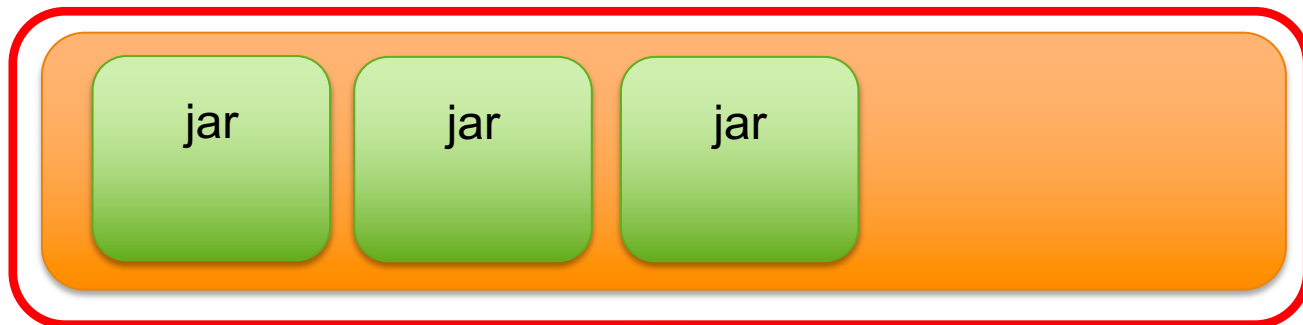
# Migrating Applications to JPMS

- Initially, leave everything on the classpath
- Anything on the classpath is in the unnamed module
  - All packages are exported
  - The unnamed module depends on all modules
- Migrate to modules as required
  - Try automatic modules
  - Move existing jar files from classpath to modulepath

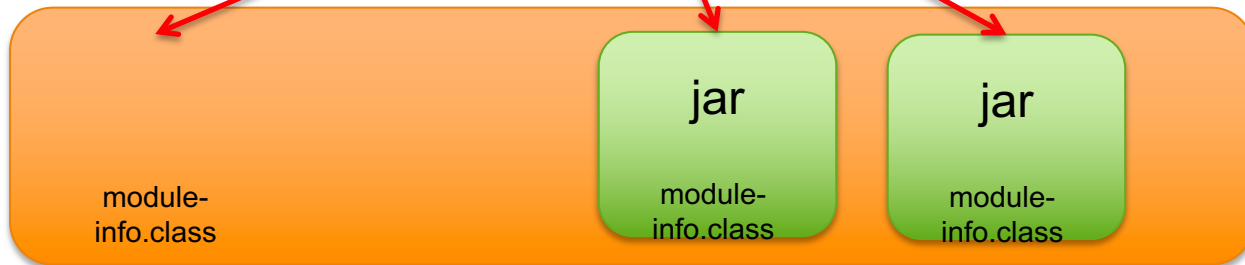
# Classpath v Modulepath

Unnamed module

classpath



modulepath



Automatic module

# Reversing Encapsulation

- "The Big Kill Switch" to turn off encapsulation
  - `--illegal-access`
    - `permit`: Warning for first use of an encapsulated API
    - `warn`: Warning for every use of an encapsulated API
    - `debug`: Warning and stack trace for every use
    - `deny`: No access to encapsulated APIs



# Reversing Encapsulation

- Allowing direct access to encapsulated APIs

- `--add-exports`

- `--add-exports java.management/com.sun.jmx.remote.internal=mytest`

- `--add-exports java.management/sun.management=ALL-UNNAMED`

- Allowing reflective access to encapsulated APIs

- `--add-opens`

- `--add-opens java.base/java.util=ALL-UNNAMED`

# Reversing Encapsulation

- Using the JAR file manifest

Add-Exports: `java.base/sun.security.provider`

# Finding Dependencies: jdeps

```
> jdeps --module-path /opt/javafx-sdk-11/lib
--add-modules=javafx.controls --list-deps FlightTracker.jar
JDK removed internal API/com.sun.media.jfxmediaimpl.platform.ios
java.base
java.datatransfer
java.desktop/java.awt.dnd.peer
java.desktop/sun.awt
java.desktop/sun.awt.dnd
java.desktop/sun.swing
java.logging
java.scripting
java.sql
java.xml
jdk.jsobject
jdk.unsupported
jdk.unsupported.desktop
jdk.xml.dom
```

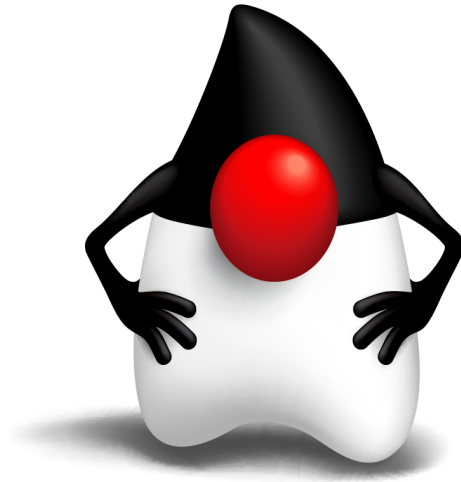
# "Missing" Modules

- Remember, "Clean applications that only use java.se..."
- The `java.se.ee` module not included by default (JDK 9/10)
  - Compilation and runtime
- Affected modules
  - `java.corba`
  - `java.transaction`
  - `java.activation`
  - `java.xml.bind`
  - `java.xml.ws`
  - `java.xml.ws.annotation`

# Using "Missing" Modules

- Use the command line option
  - `--add-modules java.corba`
- All modules (except CORBA) have standalone versions
  - Maven central
  - Relevant JSR RI
- Deploy standalone version on the upgrade module path
  - `--upgrade-module-path <path>`
- Deploy standalone version on the classpath

# Small Incompatibilities



# Milling Project Coin (JEP 213)

- A single underscore is now a keyword in Java

error: as of release 9, '\_' is a keyword, and may not be used as an identifier

- Fear not, two or more underscores can still be used



# Deleted Deprecated Methods

- **Classes**
  - `java.util.jar.Pack200`
  - `java.util.jar.Unpack200`
  - `java.util.logging.LogManager`
- **Methods**
  - `addPropertyChangeListener()`
  - `removePropertyChangeListener()`
- **Removal required for clean modularisation**

# Deleted Deprecated Class

- `com.sun.security.auth.callback.DialogCallbackHandler`
- Part of the Java Authentication and Authorisation Service
  - JAAS
  - Deprecated in JDK 7

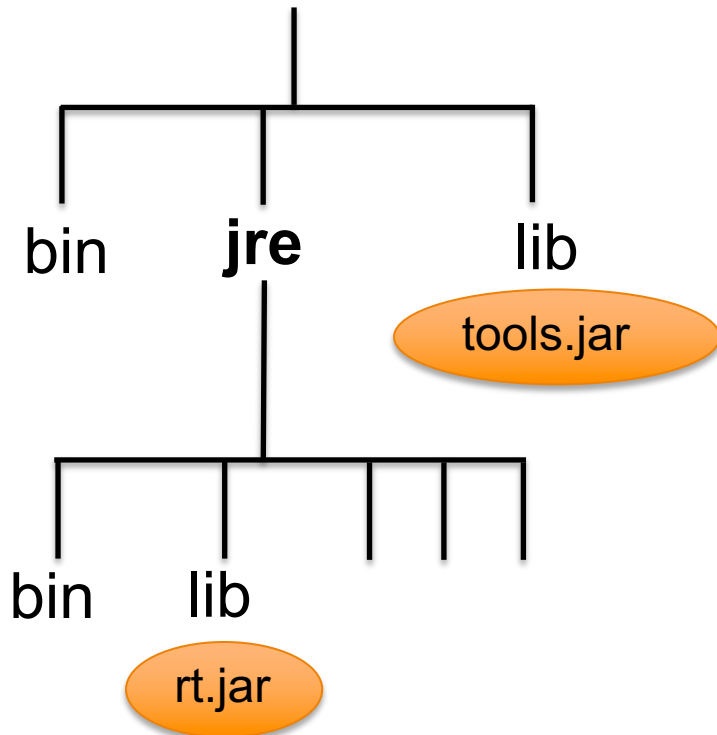
# Finding Deprecated API Use

- `jdeprscan`
  - New tool in JDK 9
  - Statically analyses class files and jar files against Java SE APIs
  - Looks for and reports usage of deprecated Java SE APIs

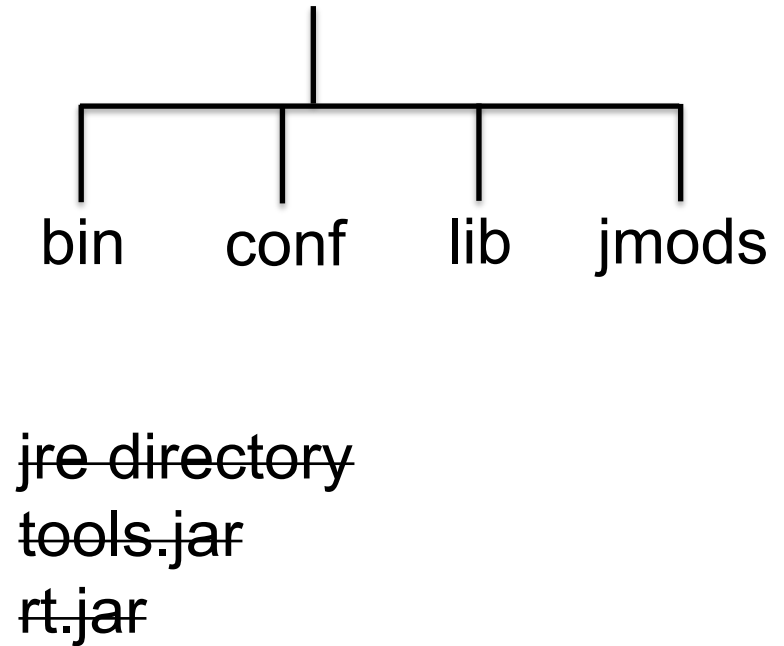
```
$ jdeprscan --class-path classes example.Deprecations
class example/Deprecations uses type java/rmi/RMISeccurityManager deprecated
class example/Deprecations uses method javax/swing/JList getSelectedValues ()[Ljava/
lang/Object; deprecated
class example/Deprecations uses method in type java/rmi/RMISeccurityManager deprecated
class example/Deprecations uses method java/lang/Boolean <init> (Z)V deprecated
```

# JDK/JRE File Structure (JEP 220)

## Pre-JDK 9



## JDK 9



# New Version String Format (JEP 223)

- Old

- Download: Java SE 8u131

```
$ java -version  
java version "1.8.0_131"
```

- Which has more patches, JDK 7u55 or JDK 7u60?

- New

- `#{FEATURE}.#{INTERIM}.#{UPDATE}.#{PATCH}`

- Easy to understand by humans and apps

- Semantic versioning



# Non-Programmatic Issues

- Java Network Launch Protocol (JNLP) [JSR 52]
  - Now uses strict parsing of configuration files
  - Some files that did parse may now fail
- Extension mechanism/Endorsed Standards Override mechanisms removed
  - Directories removed
    - `$JAVA_HOME/lib/ext`
    - `$JAVA_HOME/lib/endorsed`

`<JAVA_HOME>/lib/ext` exists, extensions mechanism no longer supported; Use `-classpath` instead.

Error: Could not create the Java Virtual Machine.

Error: A fatal exception has occurred. Program will exit.

# Removed GC Options (JEP 214)

- Deprecated in JDK 8 (JEP 173)

```
DefNew + CMS      : -XX:-UseParNewGC -XX:+UseConcMarkSweepGC
ParNew + SerialOld : -XX:+UseParNewGC
ParNew + iCMS     : -Xincgc
ParNew + iCMS     : -XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC
DefNew + iCMS     : -XX:+CMSIncrementalMode -XX:+UseConcMarkSweepGC
                  : -XX:-UseParNewGC
CMS foreground    : -XX:+UseCMSCompactAtFullCollection
CMS foreground    : -XX:+CMSFullGCsBeforeCompaction
CMS foreground    : -XX:+UseCMSCollectionPassing
```



# JVM Logging

- Unified JVM logging (JEP 158)
  - Common logging system for all components of JVM
- Unified GC logging (JEP 271)
  - Re-implement GC logging using unified JVM logging
  - Many command line options changed



# Removed JVM Flags: Ignored

- AdaptiveSizePausePolicy
- CodeCacheMinimumFreeSpace
- DefaultThreadPriority
- JNIDetachReleasesMonitors
- LazyBootClassLoader
- NmethodSweepCheckInterval
- NmethodSweepFraction
- PrintOopAddress
- ReflectionWrapResolutionErrors
- StarvationMonitorInterval
- ThreadSafetyMargin
- UseAltSigs
- UseBoundThreads
- UseCompilerSafepoints
- UseFastAccessorMethods
- UseFastEmptyMethods
- BackEdgeThreshold
- PreInflateSpin

Java HotSpot(TM) 64-Bit Server VM warning: Ignoring option <Option>; support was removed in 9.0

# Replaced JVM Flags

JDK 8 Option	JDK 9 Replacement
PrintGC	-Xlog:gc
PrintGCDetails	-Xlog:gc*
CreateMinidumpOnCrash	CreateCoredumpOnCrash (suggestion)
DefaultMaxRAMFraction	MaxRAMFraction (suggestion)
TraceBiasedLocking	-Xlog:biasedlocking=info
TraceClassLoadingPreorder	-Xlog:class+preorder=debug
TraceClassResolution	-Xlog:class+resolve=debug
TraceMonitorInflation	-Xlog:monitorinflation=debug
TraceRedefineClasses	-Xlog:redefine+class*=info
TraceSafepointCleanupTime	-Xlog:safepoint+cleanup=info
TraceClassLoading	-Xlog:class+load=info
TraceClassUnloading	-Xlog:class+unload=info
TraceLoaderConstraints	-Xlog:class+loader+constraints=info
ConvertSleepToYield	NONE

# Deprecated JVM Flags

- Two forms of warning message

warning[gc] -XX:+PrintGC is deprecated. Will use -Xlog:gc instead.

Java HotSpot(TM) 64-Bit Server VM warning: Option CreateMinidumpOnCrash was deprecated in version 9.0 and will likely be removed in a future release. Use option CreateCoredumpOnCrash instead.

# JVM Flags: Non-Starters

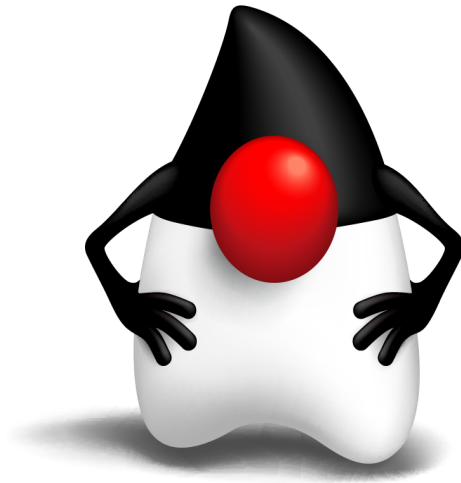
- 50 command line flags from JDK 8
  - Many related to incremental CMS
  - Many for old-style logging
    - -XX:+PrintHeapAtGC
    - -XX:+TraceDynamicGCThreads, etc.
- Use will cause the JVM to abort at start
  - It won't run your application

Unrecognized VM option '<Option>'

Error: Could not create the Java Virtual Machine.

Error: A fatal exception has occurred. Program will exit.

# JDK 10



# Local Variable Type Inference

- var is now a reserved type

```
var var = new ArrayList<String>();
```



```
public class var {  
    public var(String x) {  
        ...  
    }  
}
```





# Deprecated Things Removed

- `jdk.security.auth` module
- `com.sun.security.auth` package
  - `PolicyFile`
  - `SolarisNumericGroupPrincipal`
  - `SolarisNumericUserPrincipal`
  - `X500Principal`
- `com.sun.security.auth.module` package
  - `SolarisLoginModule`
  - `SolarisSystem`

# Deprecated Things Removed

- `java.lang.SecurityManager`
- `inCheck` field
- Methods
  - `classDepth`
  - `classLoaderDepth`
  - `currentClassLoader`
  - `currentLoadedClass`
  - `getInCheck`
  - `inClass`
  - `inClassLoader`

# Deprecated Things Removed

- `java.lang.Runtime`
- Obsolete internationalisation methods removed
  - `getLocalizedInputStream`
  - `getLocalizedOutputStream`

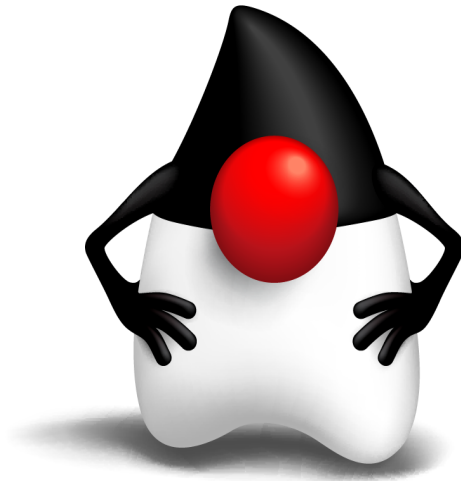
# Miscellaneous Things

- javah removed
  - Use javac -h
- policytool removed

# Command Line Flags

- -d32 and -d64 no longer valid
  - JVM will fail to start
- Also, 5 -X options
  - -Xoss
  - -Xsqnopause
  - -Xoptimize
  - -Xboundthreads
  - -Xusealtsigs

# JDK 11



# Oracle JDK & OpenJDK

- Oracle now produce two binary JDK distributions
  - Oracle JDK (now under commercial license)
  - Oracle OpenJDK JDK (GPLv2 with CPE)
- All functional differences eliminated
  - Significant changes that impact users switching to OpenJDK builds

# Major Absent Features

- JavaFX
  - Available via OpenJFX project and Gluon for binaries
- Browser plugin
  - Is anyone still using this?
- Java Web Start
  - This will hit some people
  - No easy solution
    - Oracle not open sourcing Web Start
    - Some alternatives but no drop in replacement



# Tools Affected

- No more:
  - appletviewer
  - jcontrol
  - CORBA tools
    - idlj
    - orbd
    - servertool
    - tnamesrv
  - Monitoring tool
    - jmc (now standalone package)
- Java web service tools
  - schemagen
  - wsgen
  - wsimport
  - xjc
- Java deployment tools
  - javapackager
  - javaws

# APIs Removed

- `java.se.ee` aggregator module removed completely
- Others
  - `javax.security.auth.Policy`
  - `java.lang.Runtime.runFinalizersOnExit`
  - `java.lang.SecurityManager.checkAwtEventQueueAccess`
  - `java.lang.SecurityManager.checkMemberAccess`
  - `java.lang.SecurityManager.checkSystemClipboardAccess`
  - `java.lang.SecurityManager.checkTopLevelWindow`
  - `java.lang.System.runFinalizersOnExit`
  - `java.lang.Thread.destroy`
  - `java.lang.Thread.stop(java.lang.Throwable)`

# Miscellaneous Things

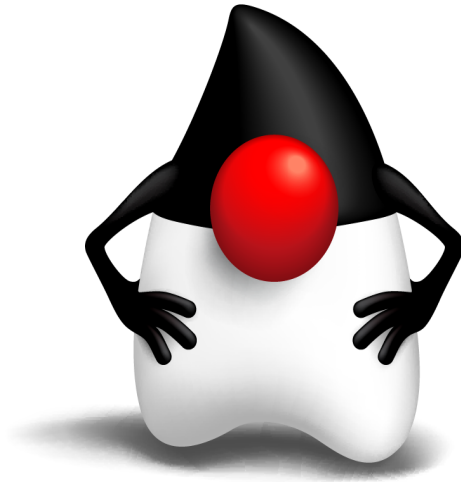
- **SNMP monitoring support**
  - `jdk.snmp` module removed
  - `JVM-MANAGEMENT-MIB.mib` removed
- **Desktop**
  - T2K font rasteriser removed
  - Lucida fonts removed
    - JDK now relies entirely on fonts from the operating system
- **Security certificates**
  - Several root certificates removed from truststore

# Command Line -XX Flags

- Big changes
- JDK 9
  - Removed 187, added 123
- JDK 10
  - Removed 36, added 26
- JDK 11
  - Removed 27, added 53

[chriswhocodes.com/hotspot\\_option\\_differences.html](http://chriswhocodes.com/hotspot_option_differences.html)

# Conclusions



# Migrating To JDK 9, 10 or 11

- Simple applications will run [almost] unchanged
  - Leave everything on the classpath
  - May need to change JVM flags
- Encapsulation
  - Additional JVM flags
  - Identify and rectify issues
- Smaller changes may cause issues
  - Removed APIs
  - JVM flag changes

# Zulu Java

- Azul's binary distribution of OpenJDK
  - Passes all TCK tests
- JDK 6, 7, 8, 9,10 and 11 available
- Wider platform support:
  - Intel 32-bit Windows and Linux
  - ARM 32 and 64-bit
  - PowerPC

[www.azul.com/downloads/zulu](http://www.azul.com/downloads/zulu)

# Thank You!

© Copyright Azul Systems 2015

**Simon Ritter**

Deputy CTO, Azul Systems

[azul.com](http://azul.com)

© Copyright Azul Systems 2017



[@speakjava](https://twitter.com/speakjava)