
ORACLE®


Java™
ORACLE®

This Is Not A Drill

Prepare For JDK 9

Rory O'Donnell
Senior Quality Manager
OpenJDK Quality Lead

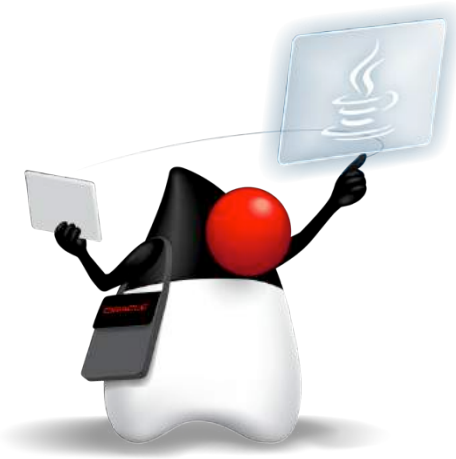
Java Platform Group @ Oracle
March 19th, 2016

Safe Harbor Statement

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Program Agenda

1. Prepare for JDK 9
2. Q & A



“plus ça change, plus c'est la même chose”

– Jean-Baptiste Alphonse Karr (1808-1890)

Image Credit: "Alphonse Karr". Licensed under Public Domain via Commons - https://commons.wikimedia.org/wiki/File:Alphonse_Karr.jpg#/media/File:Alphonse_Karr.jpg

Prepare for JDK 9

Categories of JDK-internal APIs

http://openjdk.java.net/jeps/260

- Non-critical
  - No evidence of use outside of JDK
  - or used only for convenience
- Critical
  - Functionality that would be difficult, if not impossible, to implement outside of the JDK

JEP 260 Proposal

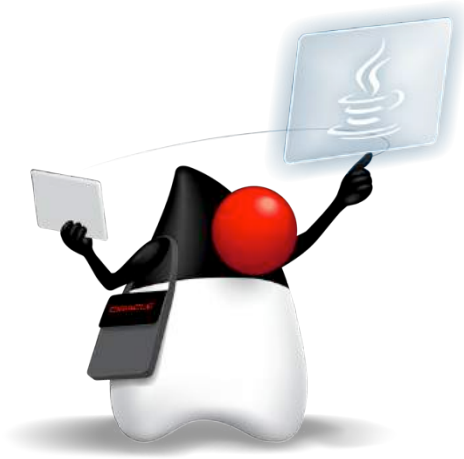
http://openjdk.java.net/jeps/260

- Encapsulate all non-critical internal APIs by default
- Encapsulate all critical internal APIs for which supported replacements exist in JDK 8
- Do not encapsulate critical internal APIs
  - Deprecate them in JDK 9
  - Plan to remove in JDK 10
  - Provide a workaround via command-line flag

JEP 260 Proposal

http://openjdk.java.net/jeps/260

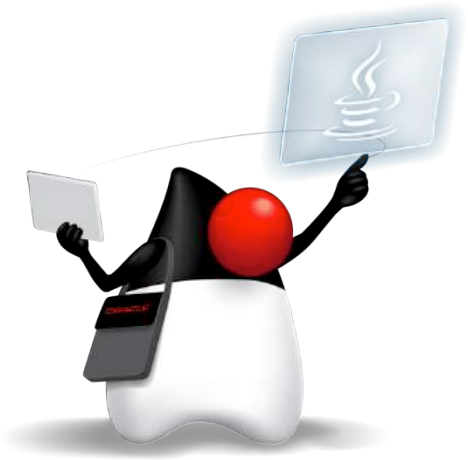
- Propose as critical internal APIs
  - sun.misc.Unsafe
  - sun.misc.{Signal,SignalHandler}
  - sun.misc.Cleaner
  - sun.reflect.Reflection::getCallerClass
  - sun.reflect.ReflectionFactory



Finding uses of JDK-internal APIs

http://openjdk.java.net/jeps/260

- jdeps tool in JDK 8, improved in JDK 9
- Maven JDeps Plugin



Removed 6 deprecated methods

http://openjdk.java.net/jeps/162

- Flagged for removal in JSR 337, and JEP 162
- Removed
  - java.util.logging.LogManager::addPropertyChangeListener
  - java.util.logging.LogManager::removePropertyChangeListener
  - java.util.jar.Pack200.Packer::addPropertyChangeListener
  - java.util.jar.Pack200.Packer::removePropertyChangeListener
  - java.util.jar.Pack200.Unpacker::addPropertyChangeListener
  - java.util.jar.Pack200.Unpacker::removePropertyChangeListener

Change the binary structure of the JRE and JDK

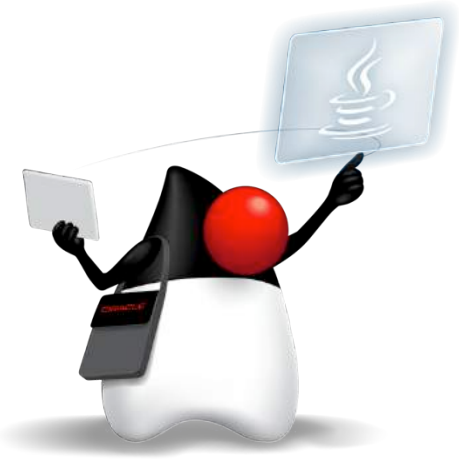
http://openjdk.java.net/jeps/220

- Motivation
- Not an API but still a disruptive change
- Details in JEP 220
- In JDK 9 since late 2014 to give lots of time for the tools to catch up

Removed

http://openjdk.java.net/jeps/220

- Endorsed standards override mechanism
- Extension mechanism



Other changes

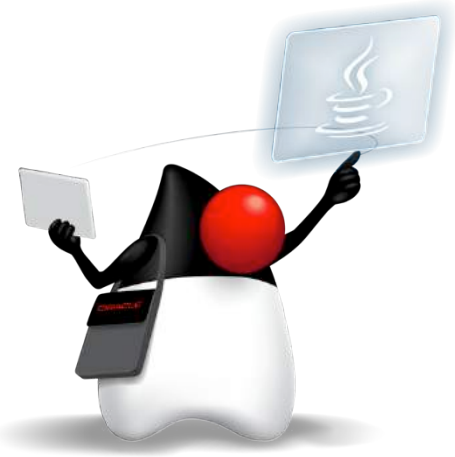
http://openjdk.java.net/jeps/261

- Application and extension class loaders are no longer instances of java.net.URLClassLoader
- Removed: -Xbootclasspath and -Xbootclasspath/p are removed
- Removed: system property sun.boot.class.path
- JEP 261 has the full list of the issues that we know about

New version-string scheme

http://openjdk.java.net/jeps/223

- Old versioning format is difficult to understand
- New format addresses these problems
- Impacts java -version and related properties



New version-string format : Hypothetical Examples

Release type	Old format Long form	Old format Short form		New format Long form	New format Short form
Early Access	1.9.0-ea-b19	9-ea	->	9-ea+19	9-ea
Major	1.9.0-b100	9	->	9+100	9
CPU	1.9.0_5-b20	9u5	->	9.0.1+20	9.0.1
Minor	1.9.0_20-b62	9u20	->	9.1.2+62	9.1.2

What can you do to prepare?

https://wiki.openjdk.java.net/display/Adoption/JDK+9+Outreach

- Check code for usages of JDK-internal APIs with jdeps
- If you develop tools then check code for a dependency on rt.jar or tools.jar or the runtime-image layout
- Check code that might be sensitive to the version change
- Check code for uses of underscore as an identifier
- Check code for uses of unrecognized VM options such as -XX:MaxPermSize
- Test the JDK 9 EA builds and Project Jigsaw EA builds

Thank you!

Quality Outreach effort wiki:

https://wiki.openjdk.java.net/display/quality/Quality+Outreach

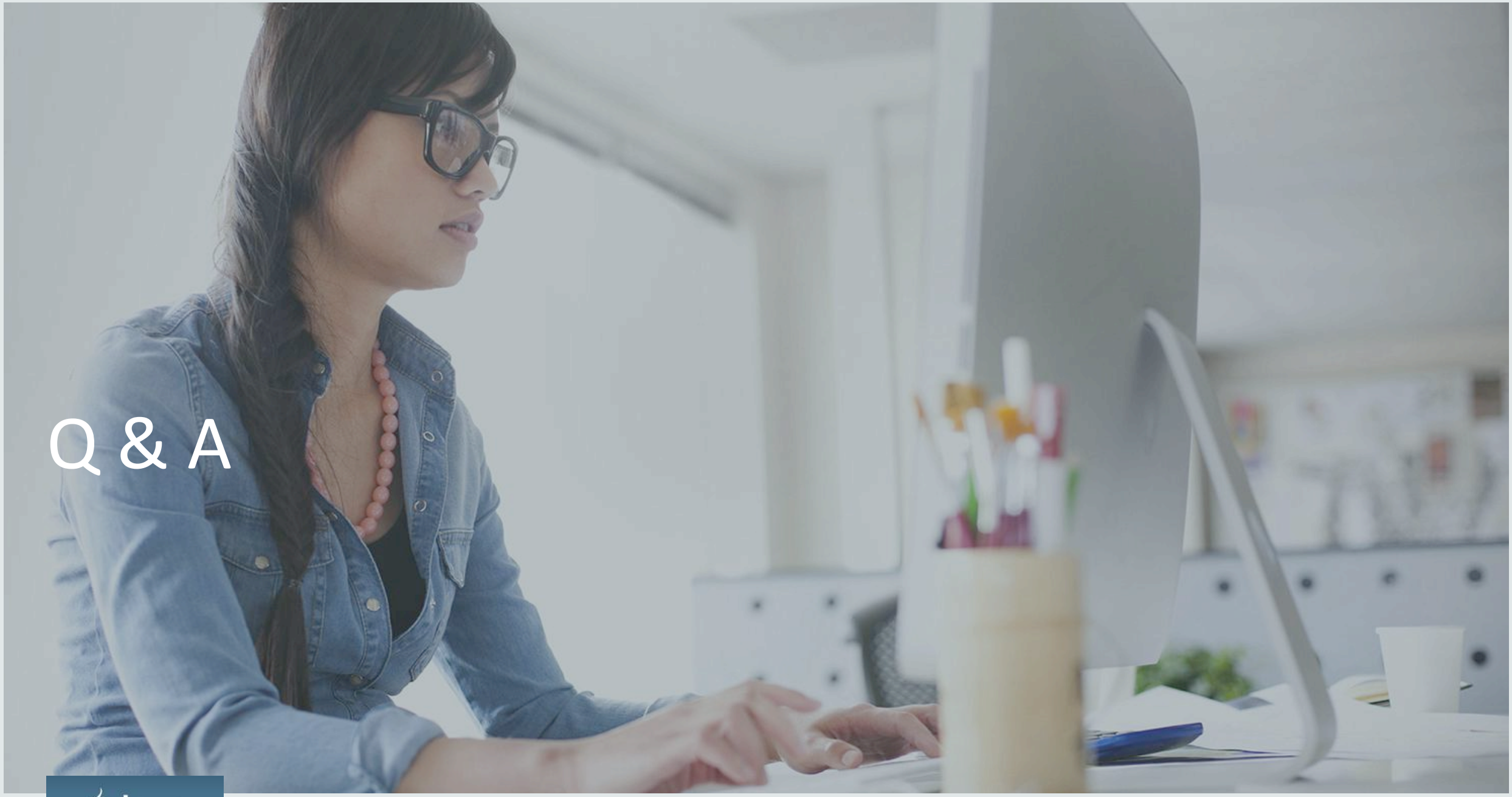
The following testers who submitted significant bug reports deserve special mention:

Apache Lucene, JaCoCo, Apache Groovy , Hazelcast & Apache Ant.

Valuable reports were also received from :

Gradle, Jitwatch, AssertJ, Apache JMeter & ElasticSearch.

(from **Quality Outreach report Sept '2015**)



Safe Harbor Statement

The preceding is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Hardware and Software
Engineered to Work Together


Java™
ORACLE®


ORACLE®