
ORACLE®


Java™
ORACLE®

What's new in Java SE Embedded?

David Holmes
Consulting Member of Technical Staff
Java SE Embedded Group, Oracle
September 29, 2014

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. Java SE Embedded Overview
2. Java SE Embedded 8
3. Java SE Embedded 8u6
4. Java SE Embedded 9?
5. Q & A

Program Agenda

1. Java SE Embedded Overview
2. Java SE Embedded 8
3. Java SE Embedded 8u6
4. Java SE Embedded 9?
5. Q & A

Java Embedded (ME & SE)

Multi-function
Embedded Devices

Personal Devices

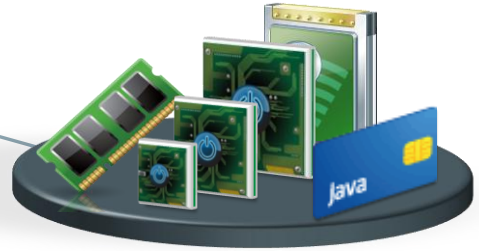
Communications

Industrial controls &
Network Appliances

Smart Appliances &
Consumer Electronics

Connected Vehicles

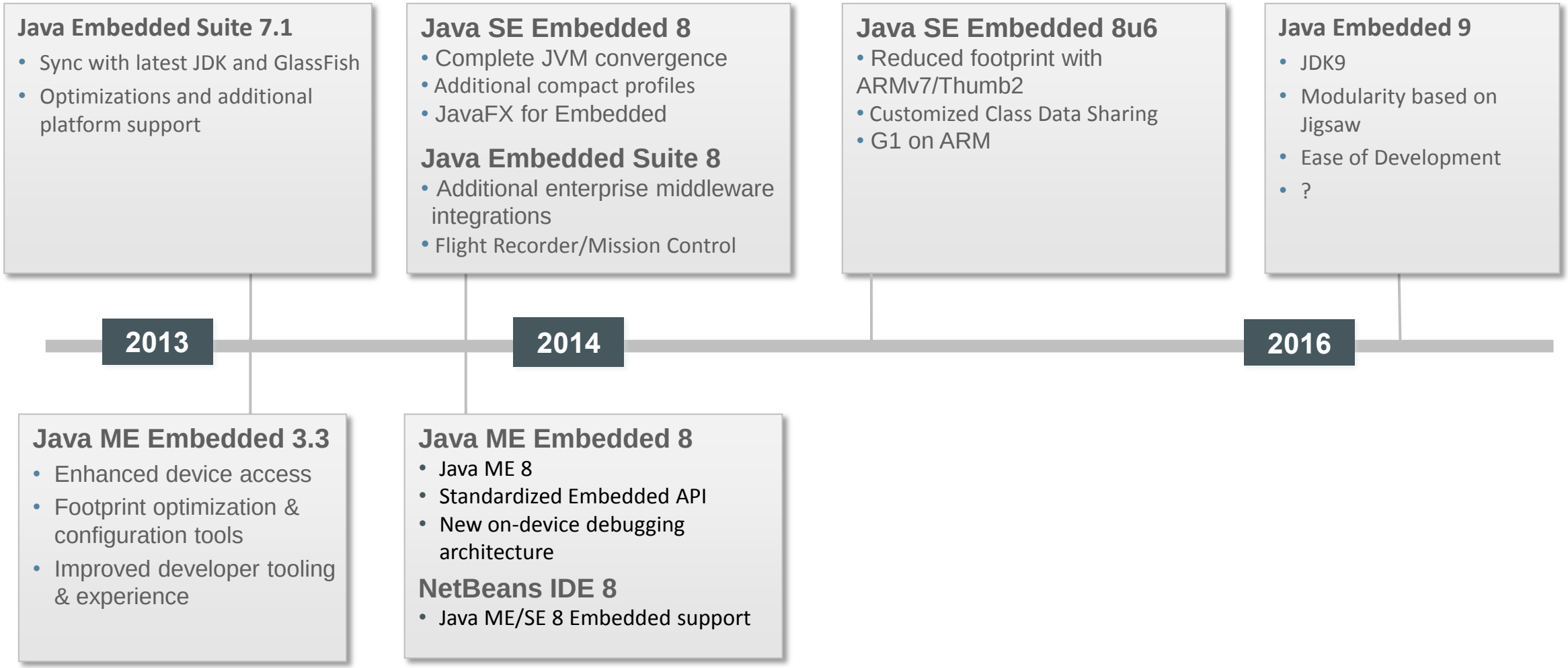
Smart Energy & mHealth



Sensors & Micro controllers

Oracle Java Embedded Offering Today

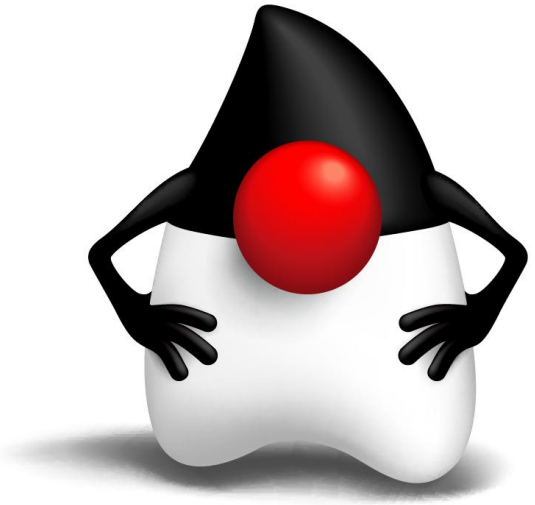
Roadmap



Program Agenda

1. Java SE Embedded Overview
2. Java SE Embedded 8
3. Java SE Embedded 8u6
4. Java SE Embedded 9?
5. Q & A

Java SE 8 – Evolving the Java Platform

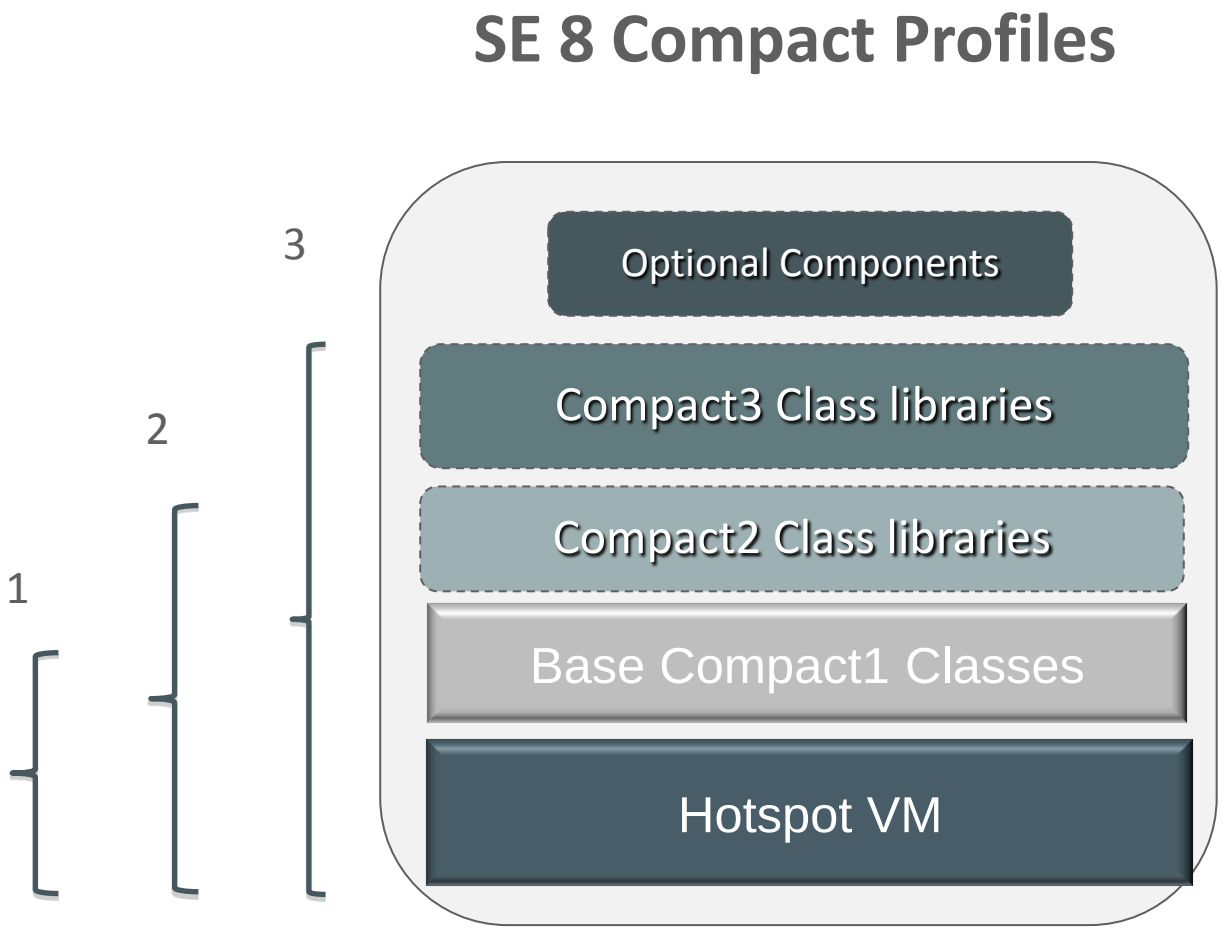


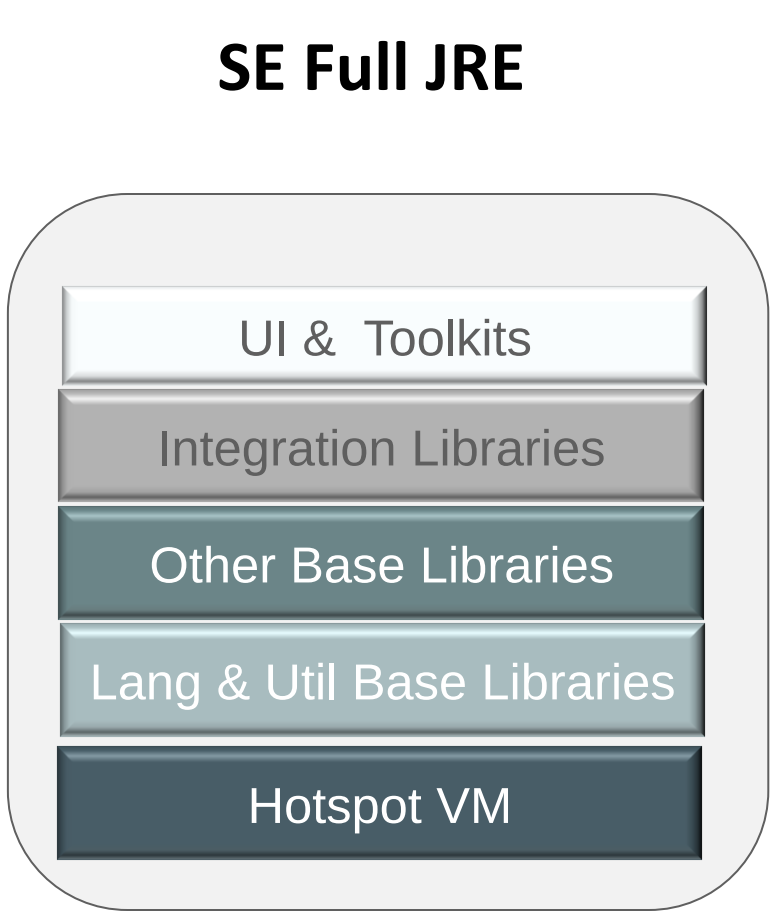
- Language evolution:
  - Lambdas & Default methods
- Library evolution
  - Streams framework for parallel computing
  - New Date/Time API's
- Hotspot evolution
  - PermGen removal; JSR-292 enhancements, Native Memory Tracking
  - Minimal VM
- Configuration evolution
  - Compact Profiles

Compact Profiles and SE Embedded

- Three new Java SE Runtimes (Compact Profiles) in JDK 8
  - Well-defined Java SE compatible subsets
  - Described in Java Enhancement Proposal 161 (http://openjdk.java.net/jeps/161)
- Benefits
  - Significantly smaller base Java runtime
  - Platform for smaller FX only graphical UI stack
  - New tool for Java SE Embedded which allows selection of runtime options
  - Developers select an appropriate runtime for their target hardware
- Basis for Java ME / CDC Converged product
  - Similar to Foundation profile & optional packages
    - Enables migration of CDC headless applications

Java SE 8 Compact Profiles





Java SE 8 API Subsets for Compact Profiles

SE Embedded 8 Features

- Java Flight Recorder (JFR) support
  - But no Mission Control and requires full JRE
- Hotspot minimal VM
  - Reduced footprint client VM with all optional services stripped out
    - 3.5MB or smaller, static footprint
  - Serial GC only; no JVMTI, NMT, CDS, Flat Profiler, JFR, Management APIs
- General performance improvements on ARM/PPC
  - Code cache management
- Optimized vector operations on ARM:
  - Leverage ARM Advanced SIMD (NEON™) in server JIT compiler

Program Agenda

1. Java SE Embedded Overview
2. Java SE Embedded 8
3. **Java SE Embedded 8u6**
4. Java SE Embedded 9?
5. Q & A

SE Embedded 8u6

- First release post JDK 8
  - Independent of SE release (8u20)
  - Incorporates 8u5 CPU release
- Key features for SE Embedded
  - Reduced static footprint
  - Increased performance on server platforms
  - ARM specific enhancements
  - Additional Java FX support on i.MX6 platform

Reduced Static Footprint

- ARMv7 builds using Thumb-2 ISA mode
  - Reduces binary size
  - Supports JNI compiled in both ARM/Thumb-2
- Link-Time Optimization (LTO) used with minimal VM
  - Improves visibility between independently compiled object files
  - Reduces static footprint as well as runtime dynamic footprint
  - Further info: http://gcc.gnu.org/wiki/LinkTimeOptimization

Tiered Compilation for Server VM

- Tiered compilation combines C1 (client) and C2 (server) JIT compilers into one binary
- Initial C1 compilation gives benefit of C1's faster startup
- C1 inserts profiling probes to determine when to invoke C2
- C2 compiles to higher optimization level to improve long running performance
- Feature is disabled by default:
  - Use –XX:+TieredCompilation to enable it

Tiered Compilation Benchmark Results

- The benchmarks were run on a server class system to satisfy the minimum of 4 cores requirement
- DaCapo benchmark on first iteration showed on average an 11% improvement
- Linpack showed a 33% improvement in time measurement and 100% improvement in Mflops/sec measurement
- **Caveat**: as with all performance improvements, results will be application and platform dependent

Provide G1 for JDK on ARM

- G1 (Garbage First):
- One of the garbage collectors in Java SE, still evolving.
  - Refer to SE documents for details, tools and tips
- Developed and tuned for big heaps
  - Explicitly deactivated for all SE Embedded builds
- Ported to ARM for 8u6
  - JDK on ARM only
  - Technology preview release and thus is **not officially supported**

Customized Class Data Sharing (CDS)

- CDS dumps an archive of boot classes that can be preloaded by the JVM:
  - Reduces dynamic footprint when shared by multiple JVM instances
  - Reduces startup time even when used by a single JVM instance
  - List of boot classes to dump is ‘hard-wired’ in SE
- SE Embedded 8u6 allows customization of the archive for each application
  - -XX:DumpLoadedClassList=<classlist_file>
    - Dumps a list of all classes loaded from the bootclasspath when application exits
  - -XX:SharedClassListFile=<classlist_file>
    - Allows user to specify their own classlist when creating archive via –Xshare:dump
  - -XX:SharedArchiveFile=<archive_file>
    - Allows user to specify the location of the shared archive file to dump or use

Program Agenda with

1. Java SE Embedded Overview
2. Java SE Embedded 8
3. Java SE Embedded 8u6
4. Java SE Embedded 9?
5. Q & A

Java SE Embedded 9?

- What will it look like?
  - Jigsaw adds modularity to the platform
  - jlink allows you to configure a JRE with what you need
  - No need for compact profiles or embedded images
- ??? … to be continued …

Program Agenda

1. Java SE Embedded Overview
2. Java SE Embedded 8
3. Java SE Embedded 8u6
4. Java SE Embedded 9
5. Q & A

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®