Java Micro Edition (ME) 8: Bringing Java to the Internet of Things

Robert Clark
Senior Software Development Director
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.
Join Us!

- **Tues@6pm**, Reception & Meet Experts from Oracle to discuss Java 8, Eclipse, Mobile, Cloud or Beer…

- **Wed@1:30pm**, Developing On-Device Java Mobile Apps for iOS ... and Android too

- **Wed@2:15pm**, Java Application Development Lifecycle in the Cloud

- **Wed@6pm**, Java EE Meetup!
Internet of Things
Opportunities & Challenges
IoT Challenges Everything
Overview of Java ME 8
Java ME 8: Key Themes and Features
Next-Generation Software Platform for the Internet of Things

**Themes**
- Unifying Java Embedded ecosystem & unleashing innovation
- Dedicated and optimized embedded software platform
- Enable increased range of use cases and markets

**Key Features**
- Modern, flexible, standards-based software platform
- Value-add new & enhanced features for embedded and wireless
- Improved configurability and optimized footprint
  - Target devices as low as at 128 KB RAM, 1 MB Flash/ROM

**Target Markets**
- Small to mid-embedded covering wide range of use cases/markets
- Intelligent edge devices, communication nodes, healthcare devices, smart sensors, smart meters, general IoT/M2M solutions
# Java ME 8 Platform Overview

<table>
<thead>
<tr>
<th>Additional APIs (Examples)</th>
<th>Use Case Software (e.g. smart pen)</th>
<th>Use Case Software (e.g. wireless module)</th>
<th>Use Case Software (e.g. control unit)</th>
<th>Use Case Software (e.g. smart meter)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Messaging</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Wireless Communication</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Protocols and Data Conversion</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security and Management</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vertical Specific APIs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Services</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sensors</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Additional Optional JSRs</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SATSA (JSR 177)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>On-Device I/O Access</td>
<td>Device I/O API</td>
<td>Generic Connection Framework</td>
<td>Application Platform</td>
<td>Security and Trust Services</td>
</tr>
<tr>
<td>Java ME Embedded Profile (MEEP) 8 (JSR 361)</td>
<td>Java ME Connected Limited Device Configuration (CLDC) 8 (JSR 360)</td>
<td>SATSA (JSR 177)</td>
<td>SATSA (JSR 177)</td>
<td>SATSA (JSR 177)</td>
</tr>
</tbody>
</table>
Top Features of Java ME 8 for the Internet of Things
Aligned with Java SE 8

- Java ME 8 is a major step towards alignment with Java SE 8
- Enables portability of code across the Java Platform, from small to large
- Unifying the development model
  - Brings Java skills, methods, tools, and innovation to embedded
  - Allows 9 million Java developers to target embedded space
- Java ME 8 remains
  - Backward compatible to previous Java ME versions
  - Focused on resource-constrained devices
New Language Features

- New Language Features aligned with SE 8
  - Assertions
  - Generics
  - Enhanced ‘for’ loop
  - Autoboxing
  - Annotations
  - Enums
  - Varargs
  - Static imports
  - Binary literals
  - Strings in switches
  - Multi-catch and rethrow
  - Diamond operator
  - Try-with-resources

```java
Hashtable<String, String> map = new Hashtable<>();
```
New Core Library Features

- New Core Library Features aligned with SE 8
  - Collections (Lists, Maps, Sets, Queues, Iterable/Interators)
  - Platform extensibility via ServiceLoader
  - EventObject and EventListeners
  - NIO Files and NIO Channels, subset of NIO Buffers
  - Logging
  - StringBuilder and String Formatter
  - Comparable Interface
  - Closable and AutoClosable (try-with-resources)
### Designed for Embedded

- Java ME 8 is purpose-built for connected embedded solutions
  - Fully headless operation
  - Robust and secure application execution
  - Application monitoring and recovery
  - Remote software provisioning and management
  - Versatile advanced connectivity, including popular protocols and latest security standards such as TLS 1.2
  - Power management framework
Highly Portable and Scalable

Platform Footprint

10MB-100MB

Java ME

1MB-10MB

Java Card

50KB-1MB

ARM 7 - Cortex M - ARM9/11 - Cortex A - MIPS32 - PPC - Intel Atom

Java SE

Device CPU/GPU/I-O
Low System Requirements

- Java ME 8 is very compact and highly configurable
  - Features and footprint can be customized for the use case
  - "Platform Right-Sizing" allows addressing a wide range of use cases and target markets with a single Java platform
- Java ME 8 implementations can run on
  - Resource-constrained devices with as little as 128 KB RAM and a minimal embedded kernel or RTOS
  - To more powerful general-purpose embedded systems with full operating systems (Linux, etc)
Consistent Across Devices

- Java ME 8 brings consistency to embedded platforms
  - Decouples applications from underlying fragmented embedded hardware and software
  - Consistent Java APIs and functionality across devices
- Enables creating embedded solutions efficiently
  - Create portable Java applications that scale across a variety of devices with minimal effort
  - Replace or update target devices without rewriting
  - Instead of fighting complexity, focus on your business value and reduce cost and time to market
Advanced Application Platform

- Java ME 8: Multi-application model
  - Robust concurrent execution of multiple applications or services
  - Virtual machine provides strict isolation between applications and the system
  - Software management functionality for lifecycle control, including termination with full clean-up

- Remotely manageable, reliable 24x7 operation
  - Supports deeply embedded use cases
  - Optimized and reliable resource management
Enhanced Software Provisioning

- Java ME 8 has built-in Software Provisioning
  - Full provisioning functionality
    - Install, update, remove of software components
    - Standardized Over-the-Air (OTA) provisioning
    - Or deployment-specific provisioning methods and policies
      - All software components are verified, authenticated and assigned to specific security domains
  - Provides out-of-the-box flexible and secure software provisioning
Modularized Software Services

- Multi-Application Model enables software modularization
  - Application logic can be partitioned into functional modules running as individual services
  - Services collaborate to provide complete solution
  - Services can be developed, deployed, and managed independently

- Enables faster, more flexible software development and deployment
  - Increases agility in developing new functionality or updates
  - Allows generalization and reuse of services
  - Reduces footprint, deployment overhead, and time-to-market
Multiple Client Domains ("Partitioning")

- Java ME 8 supports multiple Client Domains on device ("Multi-Tenancy")
  - Example clients: ODM, service provider, system integrator, software developer
  - Each client has its own security domain (policy, privileges, parameters)
  - All software executes in the domain of its client with strict enforcement of security privileges and limitations

- Enables secure sharing of a device by different clients as part of an embedded value chain
Access to Peripheral Devices

- **Device I/O API**
  - Platform-neutral access to peripheral device hardware directly from Java, no native coding involved
  - Allows easy support of use-case specific peripherals, such as sensors, actuators, converters, etc
  - Extensible for specialized devices
  - Supports a range of common I/O
    - GPIO, I2C, SPI, ADC, DAC, UART, AT Commands, Pulse counter, PWM, memory-mapped I/O, and more

- Also planned for Java SE
Compatible with Standard APIs

- Java ME 8 is designed to be compatible with established standards and APIs such as
  - JSR 75 (File API)
  - JSR 120 (Wireless Messaging API)
  - JSR 172 (Web Services API)
  - JSR 177 (Security and Trust Services API)
  - JSR 179 (Location API)
  - JSR 280 (XML API)
  - and more …
Enhanced Embedded Tooling

- **Java ME 8 SDK**
  - Tools and emulation for rapid development of embedded Java ME applications
  - Includes device emulator, application management interface, memory monitor, network monitor, and more
  - Live code deployment and debugging on devices

- **NetBeans Plug-ins**
  - Integration with Java ME SDK
  - Full-featured, integrated development environment for embedded
Eclipse Support for ME 8 Embedded

• Generic support for development of ME 8 Embedded projects:
  o CLDC 8 configuration support
  o MEEP profile support (including new security model and LIBlets)
  o ME 8 Embedded SDKs support

• Bug Fixes

Contribute

Eclipse
Mobile Tools for Java (MTJ)

Java ME SDK
• Emulator
• H/W Devices Connectivity
• Tools

Java ME SDK Tools

Devices Management, CPU Profiler, Memory Monitor, Network Monitor, etc.
Summary & Resources
# Java ME 8: A Modern Embedded Software Platform for the Internet of Things

<table>
<thead>
<tr>
<th>Aspect</th>
<th>Benefit</th>
</tr>
</thead>
</table>
| Modern embedded software platform           | • Robust, secure, cross-platform software execution environment  
• Modular software system and remote operation extends product value and reduced cost/risk  
• Leverage feature-rich platform and focus on your value-add |
| Efficient software development and deployment model | • Address the challenges of traditional embedded development  
• Accelerate time-to-market  
• Enable software portability and economies of scale  
• Leverage large ecosystem of expertise and partners |
| Increased market reach                       | • Platform “right-sizing” allows to address wide range of use cases and target markets with a single software model, from low-footprint devices to more powerful systems |
| Open, based on standards, interoperable      | • Avoid vendor lock-in  
• Participate in and benefit from technology innovation  
• Integratable with many industry standards |
Java ME 8: Learn More

• **Download** Java ME 8 Early Access
  - [oracle.com/technetwork/java/embedded/overview/javame/index.html](http://oracle.com/technetwork/java/embedded/overview/javame/index.html)

• **Documentation** for Java ME 8
  - [http://docs.oracle.com/javame/embedded/embedded.html](http://docs.oracle.com/javame/embedded/embedded.html)
    (see top of page for ME 8 content)

• **Java 8 Central:**
  - [www.oracle.com/java8](http://www.oracle.com/java8)
We’re Hiring!

Java Deployment
Java Graphics
Java Security
Java Quality Assurance
Java Performance
Java Core Libraries
Java Serviceability
Java Deployment
Java Tools
And More...

oracle.com/javajobs
Join Us!

- **Tues@6pm**, Reception & Meet Experts from Oracle to discuss Java 8, Eclipse, Mobile, Cloud or Beer…

- **Wed@1:30pm**, Developing On-Device Java Mobile Apps for iOS ... and Android too

- **Wed@2:15pm**, Java Application Development Lifecycle in the Cloud

- **Wed@6pm**, Java EE Meetup!
Evaluate This Session

1. Sign-in: www.eclipsecon.org

2. Select session from schedule

3. Evaluate: +1  0  -1