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.
Oracle Java for Freescale Embedded Devices
What’s keeping you awake at night?

- Security
- Predictable technology updates
- Everything is networked
- Cost sensitivity

- Time-to-market
- Building a developer ecosystem
- Maturity and reliability
- Standards support
Building an Embedded System?

Factors you’re likely considering:

• Robustness, reliability, and flexibility of the platform
• Ensuring security, both system and application
• How to support a broad range of I/O, including graphics and networking
• How to support diverse processors and operating systems
• Availability of developers, support, and training
• Confidence in your chosen technology and suppliers
Java – A Credible, Proven Platform

9 Million Developers Worldwide
Evans Data Corp.

#1 Choice of Developers
TIOBE Programming Community Index

1.1 Billion Desktops run Java
Nielsen Online, Gartner 2010

3+ Billion Java-enabled devices

150+ Million Java TV Devices
Why Java is Ideal Platform for Embedded

Core Value Propositions

• Era of Standards-Based Platforms: HW and SW / Open Source
• A Ubiquitous Platform – runs on almost anything
• Device to Data Center Solutions:
  – Java-Enabled Intelligent Systems for M2M and the Internet of Things
• Extend Product Lifecycles and Value:
  – Remote In-Market Update and Refresh (app logic mgmt)
• Increase Productivity, Reduce Costs and Time to Market:
  – Java Ecosystem (partners)
• Any Device, Any Size, Any Market:
  – Java is Scalable, Robust, Portable, Full-Featured
Why Freescale Promotes Java

• Secure
  – Hack-proof
  – Works in sandbox on OS - A Java app cannot touch the underlying OS

• Stability
  – TCK compliance of >40K tests before posted as GA
  – Works in sandbox on OS - A Java app cannot touch the underlying OS
  – Garbage collection avoids memory consumption and prevents app crashes

• Performance
  – Optimized on Freescale products
  – Java vs Android (2.5x better performance on Java)
  – Java VM is faster than previous (2000 transactions/second on i.MX6)

• Device-to-Data Center
  – Apps and data are moving to cloud and data centers
  – Current datacenter applications are written in Java
Why Java on Freescale

• Synergy in application spaces
  – We target the same markets
  – We understand those markets

• Optimizations specific to Freescale architectures
  – QorIQ
  – i.MX

• Great Java performance at lower power

• Freescale - Java is completely ready for market
  – Time to Market reduced
  – Complexity is reduced
Java Can Be a Game Changer

Software Flexibility Drives Business Advantages

**Without Java**

Limited flexibility, high cost

- H/w-s/w interdependencies
- Upgrade complexity
- Impact on security/integrity

**With Java**

Extend lifetime, flexibility, and value of your solutions

- Create cross-platform, modular software applications
- In-field s/w upgrades while maintaining system integrity
- Reduce device certification and testing overhead

---

**Traditional Platform**

- Native Application
- OS + Core Services
- Hardware Platform

**Service-Enabled System**

- Loadable Applications and Services
- Java Platform
- OS + Core Services
- Hardware Platform

**Component Cloud**

(Network/Enterprise)
Java Embedded

Example devices powered by Oracle

Small
- RFID Readers
- Parking Meters
- Intelligent Power Module
- Smart Meters

Medium
- Routers & Switches
- Storage Appliances
- Network Management Systems
- Factory Automation Systems
- Security Systems

Large
- Multi Function Printers
- ATMs
- POS Systems
- In-Flight Entertainment Systems
- Electronic Voting Systems
- Medical Imaging Systems
## Java Compared with Native Solutions

### For Small Embedded Systems

<table>
<thead>
<tr>
<th>Feature</th>
<th>Oracle Java</th>
<th>Native</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-platform/multi-architecture</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Sandbox security model</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Robustness</td>
<td>Excellent</td>
<td>Varied</td>
</tr>
<tr>
<td>Multi-processing/multi-threading</td>
<td>Built-in</td>
<td>Add-on, platform-specific</td>
</tr>
<tr>
<td>Customizable</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Performance</td>
<td>Optimized</td>
<td>Varied</td>
</tr>
<tr>
<td>Updatability</td>
<td>Excellent</td>
<td>Varied</td>
</tr>
<tr>
<td>Efficient, scalable development model from small embedded to large systems</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Developer Community</td>
<td>Large</td>
<td>Fragmented</td>
</tr>
</tbody>
</table>
# Java Compared With Other Platforms

<table>
<thead>
<tr>
<th>Feature</th>
<th>Java</th>
<th>Native</th>
<th>HTML</th>
<th>Flash / AIR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Supports headless apps</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Sandbox security model</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Multiple processes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Multiple language support</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Developer Community</td>
<td>Large</td>
<td>Fragmented</td>
<td>Large</td>
<td>Large</td>
</tr>
<tr>
<td>Scalability</td>
<td>Excellent</td>
<td>Excellent</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>Performance</td>
<td>Very good</td>
<td>Excellent</td>
<td>Poor</td>
<td>Poor</td>
</tr>
<tr>
<td>Updatability</td>
<td>Excellent</td>
<td>Fair</td>
<td>Good</td>
<td>Fair</td>
</tr>
</tbody>
</table>
Java Success Story
(Java vs Windows vs C++ vs Qt vs Android)

- Performance of VM
- More and developers coming into the markets
  - Universities teaching Java over C++
  - In IA Ladder logic dying
- Development costs = $0
- Royalty costs << Windows
  - Develop not cheap either
- Deployment much easier than Android
- Google’s Android focus is completely consumer
- Qt is almost proprietary
Java in the Internet of Things
31 billion devices, 4 billion people connected by 2020

- Personal Devices
- Sensors / Microcontrollers
- Med-Large Embedded / Multi-function Devices
- Cloud for Embedded Devices
- VoIP Comm
- Industrial controls / Network Appliances
- Enterprise Data & Applications
- Management / Monitoring / Operations
- Connected Vehicles
- Smart Appliances & electronics
- Meters

Oracle ON DEMAND
M2M: Wireless Modules

• Small, wireless devices add intelligence to:
  - Industrial automation
  - Healthcare applications
  - Security
  - Monitoring

• Smart & programmable
  - CPU, RAM/Flash, I/O, SIM
  - Highly integrated, 2G/3G connectivity (voice + data), low power

• Java adds intelligence and connectivity to vertical solutions
Multi-Function Printers
- Highly reliable office automation equipment
- Integration with enterprise and cloud-based applications

Why Embedded Java?
- Comprehensive set of APIs and tooling
- Connectivity and scalability, OSGi, and proven security
- App-friendly architecture with vibrant developer ecosystem
Oracle Java Embedded

Value Proposition

• **Extend Product Lifecycle**: In-market update and refresh
• **Competitive Advantage**: Focus on business value, reuse commodities from ecosystem
• **Innovation**: Fueled by largest pool of developer talent
• **Increase Market Reach**: Partner in world’s largest developer ecosystem

---

**Grow ROI**

• **Portability**: Increase flexibility to manage BOM and roadmap
• **Shorter Time-to-Market**: Reduce QA cycles and remove reinvention
• **Reduce Support**: Remote update
• **Reduce Risk**: Most widely deployed, secure, reliable development platform
• **Standards-based**: Developed in the open and backed by Oracle
Ask

What are You and Your Company doing to…

- Differentiate from competitors?
- Modernize services and technology?
- Increase ROI?
- Create new value-added services?
- Change business your model for competitive advantage?
- Gain productivity – reduce complexity?
- Be more agile (business and technology)?
- Gain more intelligence your service end-points?
- Have you considered Java? **If Not, Why not?**
Decouples software development from hardware development cycle

Development and testing can be done (mostly) using standard desktop systems

Highly productive language, APIs, runtime, and tools mean quick time to market

Create high-performance, portable, secure, robust, cross-platform applications easily

Java isolates your apps from language and platform variations (e.g. C/C++, kernel, libc differences)

Most popular embedded processors supported allowing design flexibility

Support for key embedded features (low footprint, power mgmt., low latency, etc)

Leverage huge Java developer ecosystem (expertise, existing code)

Easily create end-to-end solutions integrated with Java back-end services

Solutions from constrained devices to server-class systems
Oracle Device to Data Center
The Platform for a Totally Connected World
Which do you consider to be the top priorities in delivering M2M projects?

- Coordinating partners in value chain
- Solution Lifecycle Management
- Integration with IT
- Achieving cost/benefit constraints
- Ensuring end-to-end compatibility
- Ensuring end-to-end security
- Dealing with many different devices
- Delivering to Service Level
- Delivering within timeframe
Thinking about Service from front to back...

Building an IoT / Device to Data Center Application

M2M Application

Device

Event Processing

RDBMS

BigData

Analytics

Email

Portal

Billing

Information

CRM / Service

Social

Integration to Operational Systems

Security

Service Levels
What Are the Top Priorities When Delivering M2M Projects?

#1 - Ensuring end-to-end Security
#2 - Integration with IT systems
#3 - Coordinating partners in the value chain
#4 - Addressing cost/benefit constraints & delivering within timeframe
#5 - Ensuring end-to-end compatibility and lifecycle solution management

Oracle’s Device to Data Center Platform
Complete, Best in Class & Engineered Together

DEVICE
- Java Card
- Java ME Embedded
- Oracle Berkeley DB

GATEWAY
- Java Embedded Suite

NETWORK CLOUD
- Oracle Berkeley DB

MIDDLEWARE
- Oracle Middleware
- Oracle Database

DATABASE
- Oracle Database Mobile Server
- Oracle Big Data Appliance
- Oracle Exadata
- Oracle Exalogic
- Oracle Exalytics

OS & VIRTUAL MACHINE
- Oracle Virtual Machine

ENGINEERED SYSTEMS, SERVERS & STORAGE
- Oracle Engineered Systems

M2M APPLICATION
- Oracle Fusion Middleware
- Oracle Weblogic Server
- Oracle Oracle Enterprise Manager
- Oracle Endeca
- Java ME Embedded
- Java Card
- Java Embedded Suite
Java is an Ideal Platform for Embedded Applications

• Secure, flexible execution environment
  – Industry leading security features such as the 'sandbox' model

• Inherent cross-platform application compatibility

• Scales from deeply-embedded to server-class hardware

• Extensive developer, community, and industry support
  – Java is the #1 programming language among developers worldwide
  – Broad array of tools and support options ensures leading developer productivity

• Based on open standards driven through the Java Community Process
Summary

- Java is a trusted, robust platform ideally-suited to embedded systems where security and reliability are critical
- Java’s scalable architecture and efficient language lower costs by shortening development time and easing maintenance requirements
- Java’s large ecosystem of developers, partners, support, and training supports on-time delivery of projects and products