

# ORACLE JAVA WIRELESS CLIENT

INDUSTRY-LEADING JAVA ME SOLUTION FOR MASS-MARKET PHONES

**KEY FEATURES**

- Leverages an ecosystem with more than 3 billion devices and 45,000 applications
- Best-in-class Java Virtual Machine and graphics performance, including advanced multi-tasking capabilities
- Fully compliant with the Mobile Services Architecture (MSA) and comprehensive Java Specification Request (JSR) support
- Support for the latest display features including touch and screen rotation
- Tight Integration with Native Platforms and support for extensions from leading Mobile Operators

**KEY BENEFITS**

- Complete solution - high-performance, comprehensive implementation of Java ME; includes support for the latest Java standards
- Faster time-to-market – ready-to-deploy “preferred solutions” for mass-market mobile operating systems and leading Carrier networks
- Compatible with Java ME SDK– OJWC is used by the Java ME SDK emulator and includes a proven portability-layer to enable rapid, consistent ports across dozens of chipset and operating system combinations
- Oracle-led technologies such as LWUIT make it even easier to deliver richer, interactive experiences

*Oracle Java Wireless Client is a market-leading Java ME implementation that delivers a comprehensive and feature-rich application environment for mass-market mobile devices. To help mobile OEMs and Operators get to market more quickly with a consistent, top-quality application platform, Oracle builds and delivers customized and optimized implementations of Oracle Java Wireless Client supported by a full complement of developer tools.*

**Java ME - the most broadly deployed mobile platform**

Oracle Java Wireless Client is built around an optimized implementation of Java Platform, Micro Edition (Java ME). Java ME runs on over 3 billion mobile handsets, more devices than any other mobile application platform, and is the best way to bring content to mass-market phones. Operators and OEMs around the world have standardized on Java ME as the primary application technology for feature phones and it is supported by a massive Java ecosystem.

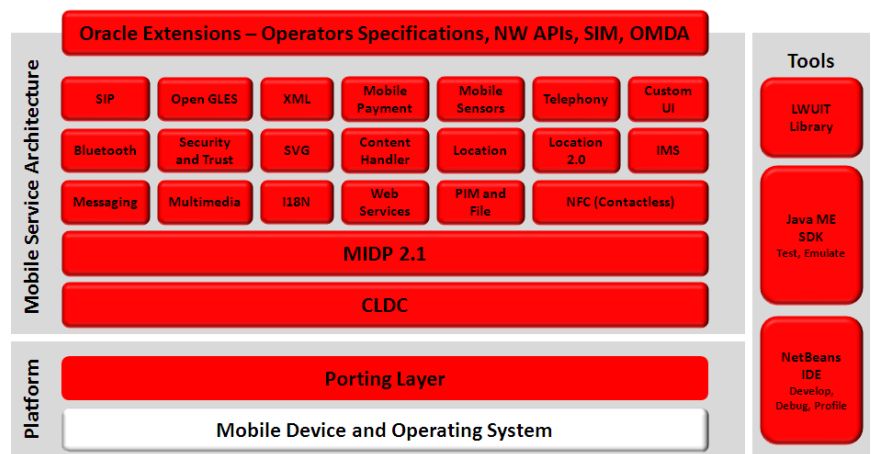


Figure 1. Oracle Java Wireless Client Architecture Diagram

**What's new in Oracle Java Wireless Client 3.1?**

The latest release of OJWC includes the following new functionality:

- **Oracle Mobile Developer APIs (OMDA)** – additional APIs to enable a set of new use cases for richer functionality Java ME applications on feature phones
- **Content Management APIs** – improved application management APIs to aid content discovery and deployment.
- **Network APIs (NW APIs)** – functionality to leverage data from Operator’s network for running smarter Java ME applications on resource constrained devices
- **Support for MTK Platform** – in the form of a reference implementation for a reference device

### Rich user interfaces with the Lightweight User Interface Toolkit (LWUIT)

The Light Weight UI Toolkit (LWUIT) is a UI library that enables developers to create compelling and consistent user interfaces for Java ME applications across a range of devices. LWUIT supports visual components and other UI features such as theming, transitions, animations, integration with Vector Graphics, integration of web content and more for today's mass market devices.

In addition, OJWC has been optimized for LWUIT applications to deliver improved performance. To this end, the combination of LWUIT and OJWC results in applications that look better and run faster than on other Java ME runtimes.

### Supports requirements from leading Operators to improve time-to-market

In addition to Java ME standards, Oracle's Java Wireless Client software also includes support for leading operator extensions. Most of the operator extensions include a combination of the following elements:

- Clarifications to standard Java ME JSRs
- Operator-specific APIs
- Application management system requirements
- User-Interface guidelines
- Testing requirements

Customizing and testing devices to meet operator specific requirements can require a substantial amount of time and engineering effort. Device manufacturers that rely on pre-built extensions in Oracle Java Wireless Client will benefit from reduced overall investment and improved time to market for commercializing devices for these operators.

### Tight integration with native platforms improves user experience

Oracle Java Wireless Client has been deeply optimized and integrated with native platforms to improve the overall execution of Java ME applications. Specifically, OJWC has been ported and deployed across a range of native operating systems including Brew MP, Windows Mobile, MediaTek platform, TDS-CDMA platform and a number of other proprietary real-time operating systems.

### Efficient porting layer improves portability and consistency

To enable Oracle Java Wireless Client to run across a wide range of operating systems efficiently and effectively, the product includes a porting layer for the Java virtual machine. The included porting layer is agnostic to the underlying chipset family, resulting in reduced OEM porting times and effort. In addition, the Oracle solution offers a variety of customization and configuration tools, enabling OEMs and device manufacturers to fine-tune the solution to their target devices without modifying the binary Java virtual machine image. These tools include string locale configuration, Java UI look and feel customization, and VM parameter setting (heap size, screen size etc.).

### Source and Binary variants of the implementation

Oracle Java Wireless Client is available as the source code of optimized implementation of the industry standard JSRs, and also as an extended binary stack which includes not only the standard JSRs but also innovative Oracle proprietary functionality to enable new use cases for Java ME applications such as seamless access of web content from a Java ME application, device location retrieval through system properties, etc. The source variant of the product is available for Win32 platforms, and the binary variant can be made available as a custom port

for a whole range of device OSs.

**Pre-integrated solution for Brew MP**

Oracle Java Wireless Client has been fully optimized for the Brew Mobile Platform to provide a seamless, powerful solution. This pre-integrated stack provides support for a variety of Qualcomm Mobile Station Modem (MSM) / Single-Chip (QSC) chipsets and enables tight integration of Java applications with the Brew MP Application Manager and Windows Manager software components. This solution enables access to running Java applications via the Brew MP home canvas.

Oracle Java Wireless Client software for Brew MP is designed for portability and rapid time to market. The JVM porting layer is pre-integrated with the Brew MP software stack in a way that is agnostic to the underlying chipset family, resulting in OEM porting times that rarely exceed a couple of weeks.

**Comprehensive tool-chain improves developer productivity**

Oracle’s Java Platform, Micro Edition (Java ME) Software Development Kit (SDK) provides a complete development environment for Oracle Java Wireless Client. With the Java ME SDK, mobile developers can write, edit, compile, package, sign, and obfuscate their applications. The applications can then be tested and debugged on Windows or Mac OS X desktops using included device emulators and built-in profiling support. The Java ME SDK also includes the ability to monitor networking and memory usage to enable developers to produce better quality, higher performance applications.



Figure 2. Java ME SDK 3.0 delivers high fidelity emulation

**Key Java specifications supported**

The following table lists the Java Specification Requests (JSRs) supported by Oracle Java Wireless Client

Supported Java Specification Requests (JSRs)	
JSR 075 – File and PIM	Enables access to device file systems and Personal Information Management (PIM) data
JSR 082 – Bluetooth	Allows Java ME enabled devices to integrate into a Bluetooth environment
JSR 118 – MIDP	Standard Java ME platform configuration for resource constrained devices
JSR 120 – Messaging	The Wireless Messaging API (WMA) is an optional package that provides platform-independent access to communication resources like SMS
JSR 135 – Mobile Media	Specifies a small-footprint multimedia API for Java ME, allowing simple, easy access and control of basic audio and multimedia resources
JSR 139 – CLDC (Connected Limited Device Configuration)	Defines a standard platform for small, resource-constrained, connected devices and is compliant with the Java Language and VM specifications
JSR 172 – Web services	Provides standard access from Java ME applications to web services

JSR 177 – Security and Trust	APIs for security and trust services
JSR 179 – Location	Enables location-based applications for resource constrained devices
JSR 180 – SIP	Enables Session Initiation Protocol (SIP) applications to be executed in resource constrained devices powered by Java ME
JSR 184 – 3D Graphics	Scalable, small-footprint, interactive 3D API for use on mobile devices
JSR 205 – Messaging	Defines Wireless Messaging API 2.0, which extends the Wireless Messaging API (JSR 120). It gives Java ME developers the ability to send and receive Multimedia Messaging Service (MMS) messages.
JSR 211 – Content Handler	Allows applications to invoke registered Java and non-Java applications
JSR 226 – Vector Graphics	SVG (Scalable 2D Vector Graphics) API for Java ME. JSR 226 is the standard for interactive and animated 2D graphics on the J2ME platform.
JSR 229 – Payment	Enables initiation of mobile payment transactions in Java ME applications, and supports different payment instruments
JSR 234 – Multimedia Supplements	Optional package for advanced multimedia functionality which is targeted to run as an supplement in connection with MMAPI (JSR-135)
JSR 238 – Internationalization	Provides regionally relevant data formatting, sorting of text strings and application resource processing for MIDlets running in MIDP / CLDC
JSR 239 – Java Binding for the OpenGL ES API	Defines Java Binding for the OpenGL ES API, which enables access to a low level 3D graphics library through a standard Java interface.
JSR 248 – Mobile Services Architecture	Platform definition for mass-market devices that includes a comprehensive API structure to facilitate development of the widest range of highly portable applications across the broadest spectrum of devices.
JSR 253 – Telephony	Exposes common telephony features available in most wireless handsets.
JSR 256 – Mobile Sensor APIs	Enables fetching data from any physical sensor such as magnetometers, accelerometers, etc.
JSR 257 - Contactless Communication API (NFC)	Enables contactless communication based on RFID, NFC or bar codes. Includes ability to access information and bi-directional communication
JSR 258 – Mobile User Interface Customization API	Enables users to customize the look and feel of the user interface components in a mobile device or platform
JSR 281 – IP Multimedia Subsystem (IMS)	Enables access to IMS functionality such as support for IMS registration mechanism, support for co-location of multiple IMS services, use of IMS service sessions, etc
JSR 293 – Location API 2.0	Extends JSR 179 by adding proximity services (geo-boundaries that can trigger application events) and improving support for landmarks

## Contact Us

For more information about Oracle Java Wireless Client, visit [oracle.com](http://oracle.com) or call +1.800.ORACLE1 to speak to an Oracle representative.



Copyright © 2011, Oracle and/or its affiliates. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice. This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. UNIX is a registered trademark licensed through X/Open Company, Ltd. 1010

**Hardware and Software, Engineered to Work Together**