WHITE PAPER

MOTO**MAGX**™

Platform Overview

An Open and Flexible Platform for Motorola Mobile Devices

June 2008

REVISION HISTORY

Version	Date	Description
1.0	12/07/07	First version.
1.1	02/05/08	Updated number of Java mobile devices to 2.1 billion.
1.2	03/25/08	Removed reference to plug-ins.
1.3	06/30/08	Replaced MOTOMAGX logo.

CONTENTS

MOTOROLA AND LINUX

Consumers are demanding more out of their mobile devices – they want devices that are personal, contextually aware, and perform as well as look cool – across a variety of areas including productivity, multimedia and entertainment, sports and active lifestyle, social networking and more. To deliver these compelling mobile experiences, sophisticated software and an advanced, open mobile platform are critical.

Building on the global success of Motorola's earlier Linux-based platforms, Motorola's MOTOMAGX platform is built around mobile Linux®, which enables superior mobile experiences for consumers and the most flexible platform for partners, developers, and operators. A clear leader in mobile Linux innovation and committed champion of open standards, Motorola is driving the vision for mobile Linux and open source platforms, industry-wide.

Motorola chose Linux because it is open, flexible, and widely adopted. The benefits of working with open software solutions include decreased time to market, reduced operational costs, and accelerated innovation – all of which help Motorola and its customers stay ahead of the changing marketplace by producing the right solutions for today and tomorrow. Linux attracts many developers, and major technology companies support and invest in Linux, including Sony, IBM, Mitsubishi, Oracle, and Sun. By supporting Linux, Motorola can bring new capabilities to market sooner and leverage best-of-breed solutions to support customers and subscribers.

- **Linux is flexible and scaleable.** Linux as a platform provides superior flexibility because it has been ported to more processors and chipsets than any other mobile platform, and the majority of Internet servers are using Linux today. Furthermore, tens of thousands of programmers have reviewed the source code to improve performance, eliminate bugs, and strengthen security. No other operating system has ever undergone this level of review.

- **Linux has no vendor lock-in.** The availability of source code means that users and support providers are empowered to get to the root of technical problems quickly and effectively by working with the community to address their technical issues. This contrasts sharply with proprietary operating systems, where even top-tier support providers must rely on the OS vendor for technical information and bug fixes.

- **Linux has a rich ecosystem of developers** and is uniquely designed to draw on the creativity and expertise of a growing community of dedicated developers around the world to deliver rich and compelling applications.

- **Linux is secure.** Viruses, Trojans and other malware rarely, if ever, manage to infect Linux systems, in part because of its multi-user design and its modular nature. A multi-user design isolates a user's applications, files, and directories from the rest of the operating system, while a modular system isolates flaws that exist in one place from the rest of the system. Motorola has enhanced the security of its platform by incorporating robust security features at all levels of the architecture.

With over 10 million Linux-based handsets shipped worldwide (as of Q3 2007), Motorola is one of the top manufacturers to offer a Linux platform for mobile devices shipped globally and is the first to bring Linux to North America with the iconic MOTORAZR^2 V8. Motorola also plays a leading role in driving open standards around mobile Linux and Java™, as the founder, member, and committee leader of such consortiums as the recently-formed Open Handset Alliance, the LiMo Foundation, CELF (Consumer Electronics Linux Forum), the Linux Foundation, OMA (Open Mobile Alliance), OSGi (Open Services Gateway initiative), JCP (Java Community Process), and the Eclipse Foundation.

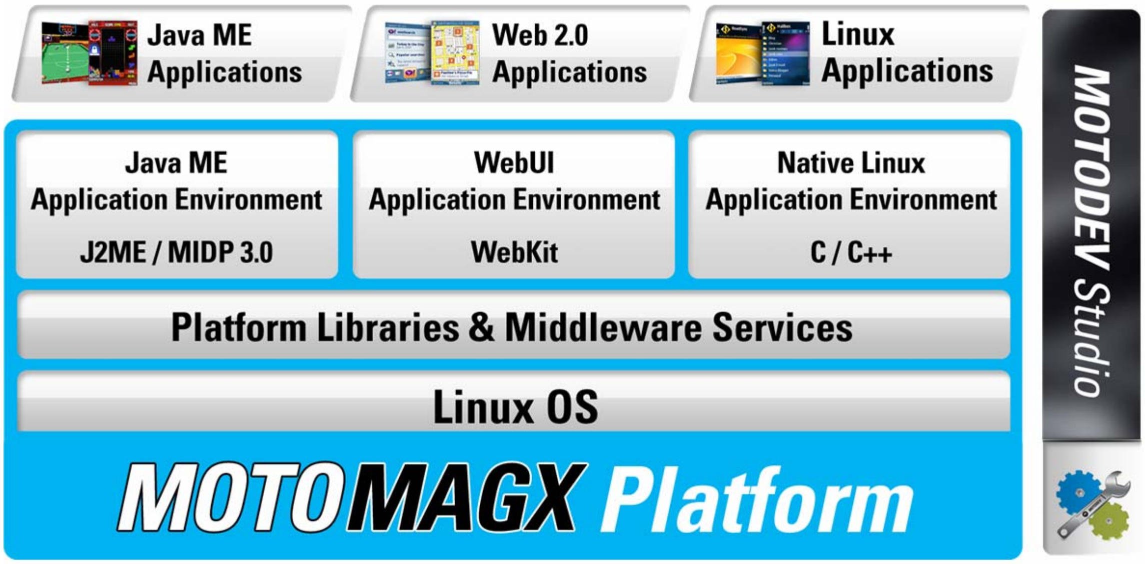
MOTOMAGX PLATFORM ARCHITECTURE

MOTOMAGX is the next generation of Motorola's mobile Linux platform, and plays a key role in demonstrating Motorola's commitment to mobile Linux. Building on the global success of Motorola's earlier Linux-based platforms, MOTOMAGX delivers new levels of openness, flexibility, and support for third-party applications on Motorola mobile devices.

The MOTOMAGX platform has a modular architecture that includes the following components:

- **Linux OS.** The Linux OS incorporates packages from the open source community and additional components to address mobile device requirements.
- **Platform libraries and middleware services.** The platform libraries and middleware services provide the underlying services to the platform, and manage the application life cycle, application interactions, and security of the platform.
- **Application environments** (Java ME, WebUI, native Linux). The application environments include the APIs and services needed to attract a third-party application development community.
- **Applications.** The applications provide the user experiences through interactions with the rest of the components.

The figure below shows the high-level MOTOMAGX architecture.


Java ME Applications
Web 2.0 Applications
Linux Applications
Java ME Application Environment
J2ME / MIDP 3.0
WebUI Application Environment
WebKit
Native Linux Application Environment
C / C++
Platform Libraries & Middleware Services
Linux OS
MOTOMAGX Platform
MOTODEV Studio

Linux OS

The MOTOMAGX Linux OS layer contains the Linux kernel, which is based on MontaVista's Linux Kernel distribution, version 2.6.10. MontaVista's kernel is commercial-grade for faster time to market, but it offers the customizability and control of an open-source Linux environment, with support for the glibc collection of libraries from the GNU project. The kernel also addresses mobile device requirements for power management, real-time performance, fast start-up, small footprint, and security, and includes chipset support for silicon vendors such as Freescale and Texas Instruments, and device drivers for Bluetooth, the camera, USB, the modem, and so on.

Platform Libraries and Middleware Services

The platform libraries and middleware services consist of:

- Middleware components that abstract the hardware and operating system from the platform, such as TAPI (Telephony Application Programming Interface) and database components
- Java and WebUI access to platform services and standard libraries
- UI services
- Application framework

The platform libraries provide daemons, services, and other non-UI aspects of the system that advertise services both among themselves and to the objects in the application framework, UI services, and application layers. Most of the platform services entities have a flexible architecture with a plug-in structure to comply with various standards for media formats, communication protocols, digital rights management specifications, and so on. This architecture allows the integration of best-in-class services or the use of more than one service to perform a specific function, such as Instant Messaging, with engines from Yahoo, AIM, Skype, and others. Many of the platform components are sourced directly from open source projects, such as WebKit, BlueZ, OpenOBEX, and SQLlite, leveraging the innovation power of open source communities.

The Java ME (Java Platform, Micro Edition) application environment provides access to some of the platform services through the JSRs (Java Specification Requests) supported by the platform, while the WebUI application environment enables access to platform services through the WebUI APIs.

The UI services are responsible for defining the look and feel of the user interface. They ensure that an application that has focus receives user input and is able to render to the display. Java ME applications, also known as MIDlets, use the LCDUI API to provide UI services. The LCDUI API contains the user interface classes for Java ME. WebUI and native applications use the services of WebKit, the framework for the Open Source Browser, to provide UI services.

The application framework manages the application life cycle, which consists of installing, starting and stopping, and uninstalling applications. It also supports interactions between applications and components, including the invocation and discovery of services by applications, as well as event-based application interaction. A secure package installer and a runtime environment for downloaded applications ensure the security of the platform.

Application Environments

The MOTOMAGX platform is designed to support a broad range of content created by third-party developers. Today, MOTOMAGX supports applications developed in Java ME, with plans to introduce WebUI and native Linux application environments in upcoming releases. These three application environments, combined with the relevant tools available through Motorola's MOTODEV Studio integrated development environment, will enable MOTOMAGX developers to innovate and accelerate time to market for their applications.

JAVA ME APPLICATION ENVIRONMENT

As one of the largest licensees of Java ME, Motorola has played an active role in the development of the Java ME (formerly known as J2ME™) technology, collaborating with Sun Microsystems to develop an edition of the Java 2 platform for consumer electronics and embedded devices. Motorola continues to drive the creation of Java ME standards, leading the JSR for Mobile Information Device Profile 3.0 (JSR 271, MIDP 3.0) and participating in over 76 JSRs (leading or co-leading nine of these, including Mobile Broadcast Services for Handheld Terminals, JSR 272, for viewing and utilizing digital television services). Motorola is an active participant in establishing Java API specifications and serves as a member of the Executive Committee for the JCP, which is helping the industry to improve the standards-setting process.

The Java ME technology is important for several reasons:

- It speeds the development of light to mid-weight applications that don't tax the hardware.
- It provides a common standard for developers writing applications for mobile devices.
- It enables users to run a wide array of applications that are portable to other devices.
- It is a secure, mature, and supported technology that can provide a rich consumer experience.

On MOTOMAGX, Java ME developers have access to some native services such as SMS, Bluetooth, networking, and location through the Java ME APIs and will continue to have more access as the JSR roadmap evolves.

Currently, the following JSRs are supported:

- JSR 75 – File System API for navigating Java MIDlets file system; PIM API for accessing PIM data through Java MIDlets
- JSR 82 – Bluetooth API
- JSR 118 – Mobile Information Device Profile (MIDP) 2.0 – UI and security model
- JSR 120 – Wireless Messaging APIs for sending and receiving SMS through Java MIDlets
- JSR 135 – Mobile Media APIs for progressive download and playing multimedia files in Java MIDlets
- JSR 139 – Connected Limited Device Configuration (CLDC) 1.1
- JSR 172 – Web Services API for accessing web services
- JSR 184 – Mobile 3D Graphics API for 3D games and UI
- JSR 185 – JTWI – Java API package that includes additions to JSR 118, 120, and 135
- JSR 205 – MMS APIs for sending and receiving MMS through Java MIDlets

MOTOMAGX also provides Java APIs for controlling features such as the secondary display, DRM, and H3G. Furthermore, the MOTOMAGX Java implementation supports a wide range of audio and video formats to enable the creation of compelling applications.

In 2008, support for the following JSRs will be added:

- JSR 177 – Security and trust APIs (partial compliance)
  - SATSA-CRYPTO
  - SATSA-APDU
- JSR 179 – Location-based services API
- JSR 211 – Content handling API
- JSR 226 – Scalable 2D Vector Graphics API

Java is deployed on more than 2.1 billion phones (Sun Microsystems, 02/08), and 8 out of 10 new mobile devices sold in the marketplace today are Java-enabled. Because Java applications abstract the details of the platform, developers can more easily port their applications to other devices, such as set-top boxes and desktop PCs. Today, Java provides a robust platform for developers producing new content that they want to port to multiple devices, and is ideal for light to medium-weight applications such as entertainment and gaming, offering the most mature set of 3D APIs.

Developers can currently create Java applications for MOTOMAGX by downloading MOTODEV Studio for Java ME from the MOTODEV developer network. MOTODEV Studio for Java ME provides a full set of tools to compile, debug, and test mobile Java applications for Motorola handsets. MOTODEV Studio for Java ME is described in greater detail below.

WEBUI APPLICATION ENVIRONMENT

Developers today are using the latest web technologies to create innovative services and applications with exciting interactivity. WebUI, based on the open source WebKit engine, enables rich interactive applications to be quickly developed using the XHTML, JavaScript, CSS, and AJAX technologies that are used to create web pages and Web 2.0 services today. WebUI leverages the expertise of the web development community, and opens up exciting new possibilities for developers to extend their business to the mobile arena.

The WebUI application environment supports two categories of applications and services – WebUI widgets and full-featured WebUI applications. WebUI widgets are lightweight, standalone web applications that are launched from a dedicated visual container (the Widget Library or the Widget Livecon View); provide a miniature live view; and have limited or no interactivity. Examples are a weather widget that displays continuous weather information after the user enters a home zip code, a stock widget that continuously displays stock quotes for a specific stock, or a sports widget that displays scores for a user’s favorite teams.

Full-featured WebUI applications are an alternative to Java ME applications that can be launched from the main menu or a home screen shortcut to deliver richer, full-screen Web 2.0 services such as Google Maps for the mobile handset, mashups, or interactions with a user’s Flickr™, YouTube, or eBay® accounts. WebUI applications are frequently interactive, using the AJAX technology for the more efficient exchange of data with a server. Applications can range from simple to so complex that they require an advanced skill level to write and support, and operators charge a subscription fee for access.

For greater levels of interactivity and personalization, the WebUI application environment provides APIs that enable developers to access the native resources of the MOTOMAGX platform. For example, Google Maps for the mobile handset could be contextualized by location using the location-based services on the device.

WebUI widgets can only access the subset of WebUI APIs that is untrusted, while full-featured WebUI applications require a signature to access a set of WebUI APIs based on their trust level. The WebUI APIs provide access to the following native phone services.

- File system – Read and write access to files in the file system and access to the native file picker UI
- Location – Information about the device's location
- Contacts – Contacts database and native contact list picker UI
- Calendar – Calendar entries
- Internationalization – Language translation and language change notifications
- Application management – Interfacing with other application services
- Event subsystem – Publishing and subscribing to system-defined events
- System – System settings and status bar information
- Soft key management – Getting or setting the soft keys and receiving notification when they are pressed
- Media finder – Searching for files based on metadata tags
- Messaging – Reading, writing, listing, and sending SMS messages
- Download – Downloading and storing files

In an upcoming release, MOTODEV Studio will support WebUI application development by providing a complete set of tools to create, debug, and test WebUI applications for Motorola handsets. MOTODEV Studio for WebUI is described in greater detail below.

NATIVE LINUX APPLICATION ENVIRONMENT

The native Linux application environment includes the APIs that allow native applications to interface with basic mobile handset functions, such as dialing, setup, data synchronization, and device management. Native applications written in C or C++ interact directly with the lower-level services of the MOTOMAGX platform, offering better performance for processor-intensive or media-intensive applications, and greater integration with the platform. Native applications can also provide compelling experiences such as multimedia and games. To create these advanced experiences using the native APIs, developers can leverage the large Linux ecosystem and the resources of the open source community.

In an upcoming release, MOTODEV Studio for Linux will support native application development by providing application developers with a full set of tools to compile, debug, and test native applications for Motorola handsets. MOTODEV Studio for Linux is described in greater detail below.

Applications

The applications layer consists of the applications that interface with the middleware and application framework to provide the user experience. The applications layer includes the standard phone applications, such as calling, phonebook, email, calendar, and the browser, and can also include third-party native applications, Java MIDlets, and WebUI applications.

MOTOMAGX SECURITY

Linux is touted for its openness, but critics say that this makes it less secure. By incorporating robust security components at all levels of its architecture, the MOTOMAGX platform is one of the most secure in the industry. These security features include the following:

- Unauthorized code execution prevention prevents the injection of unauthorized code that can corrupt the operating system.
- Application signing grants a level of trust to an application, which dictates what features are available and whether the subscriber is required to confirm access to a specific feature.
- A security policy model consists of a set of configurable rules to enforce the business logic, including access control rules for determining whether an application can perform a restricted operation; protection domains to which an application is assigned based on the verified identity of the application's originator; and policy updates for preventing new viruses and attacks from compromising the integrity of the platform.
- Cryptographic services include secure communication using SSL and TLS; authentication APIs for requiring a PIN to gain access to a device, application, or SIM card; a certificate store for validating a signer's public key that is used in verifying application signatures; and cryptography APIs for ensuring the confidentiality of information through encryption.
- Protection against unauthorized flashing prevents an unauthorized user from writing over certain areas of memory on the device.
- Secure solutions for protecting the IMEI (International Mobile Equipment Identity), subsidy lock, and boot loader are implemented.

For more information, see the MOTOMAGX Security white paper.

DEVELOPER TOOLS

MOTODEV Studio is an innovative and integrated platform for third-party application development. Based on an open framework enabled by Eclipse, MOTODEV Studio is designed to accelerate the application development life cycle – from requirements definition to testing and deployment. Extensible and modular, the MOTODEV Studio supports plug-ins for new product tools and delivers updates and new features through a web-based service, ensuring that developers always have the latest functionality.

Aligning with Motorola's seamless mobility strategy, MOTODEV Studio will support application development across all of Motorola's products – including mobile devices, home and networks mobility, and enterprise mobility solutions. By providing standardized developer tools, Motorola, developers, and operators can speed the time to market of value-added applications. MOTODEV

Studio for Java ME is currently available to developers of Java applications. MOTODEV Studio for WebUI and native Linux will be available in upcoming releases.

MOTODEV Studio for Java ME

MOTODEV Studio for Java ME provides an integrated development environment supporting mobile application development in Java. MOTODEV Studio for Java ME includes the following features:

- Seamless integration with the popular Eclipse framework
- Easy, central location of Java ME libraries, sample MIDlets and tutorials, and integrated documentation
- Powerful emulation capabilities with an updated Java ME emulator that simulates the MIDlet environment behind an interactive skin of the Motorola handset
- Support for multiple simultaneous handset emulations to conveniently develop peer-to-peer applications

Java developers who want to use another UEI-compliant IDE can download the standalone Motorola Java ME SDK.

MOTODEV Studio for WebUI

WebUI applications typically transfer the processing necessary for the user interface to the client but keep the bulk of the data on the application server. For mobile developers, developing and testing an end-to-end solution represents a challenge because the development environment must be suited for both mobile and server-side applications, and ideally allow testing of the entire mobile application on the desktop. Eclipse has strengths in both server and mobile environments, making it an ideal choice for the development environment provided by MOTODEV Studio.

MOTODEV Studio for WebUI will offer the following features:

- Documentation of exposed WebUI device APIs
- Integration with the popular Eclipse framework and the portions of the Web Tools Platform (WTP) Project for editing and debugging JavaScript, XHTML, and CSS code
- Web-based emulator that fully replicates the handset capabilities

MOTODEV Studio for Linux

MOTODEV Studio for Linux provides an integrated development environment for supporting mobile application development in C or C++.   MOTODEV Studio for Linux will include the following features:

- Complete documentation of Linux and Motorola APIs and services to ease the creation of applications
- Seamless integration with the popular Eclipse framework
- Open source-based toolchains, including utilities such as a compiler, a make utility, a debugger, and a linker (gmake, gcc, gdb, and binutils)
- Libraries and header files for Motorola’s published APIs as well as generic Linux functionality such as stdio, pthreads, and so on

- Emulator that provides the general operations and capabilities of the mobile device while replicating the application UI experience on the development machine by importing the skin for a specific device

ACCELERATING INNOVATION

Motorola is committed to supporting the entire ecosystem through the MOTODEV developer network. Moreover, Motorola supports open source innovation and standardization to speed application development and drive innovation and the adoption of new services. A few examples of Motorola's open initiatives include opensource.motorola.com, the Eclipse Foundation, the LiMo Foundation, and the Open Handset Alliance. Each of these efforts demonstrates Motorola's leadership in mobile Linux innovation and commitment to open standards.

MOTODEV Ecosystem

MOTODEV is a unified, global developer network and business ecosystem, designed to make it easier and more rewarding to innovate with Motorola, its products, and technologies. MOTODEV combines Motorola-wide resources, tools, and technical support to promote developer success by opening new channels for growth and revenue creation while extending Motorola's seamless mobility vision.

MOTODEV provides everything developers need to create innovative solutions, including developer tools, a searchable support knowledgebase, a global team of technical support experts to assist developers, an online documentation library, and a handset loaner program. Launched in September 2007, the Motorola Solutions Catalog enables developers to market their applications by providing information on the application and where it can be purchased.

MOTODEV Fast Track Center gives developers access to business and product development services as well as marketing and go-to-market programs. Aimed at streamlining the application development life cycle, the Fast Track Center enables the quick delivery of innovative, high-quality applications to Motorola customers. Members can leverage a full range of Motorola services plus attractive offers from a variety of industry-leading service providers and channel partners.

Opensource.motorola.com

Opensource.motorola.com is a web resource aimed at sharing source code, open source projects, new ideas, and information with open source developers around the world. Opensource.motorola.com features Linux source code, including the kernel, drivers, and compilers for Motorola's Linux-based devices such as the MOTOMING and the MOTOMAGX-based MOTOROKR™ Z6 and MOTORAZR^2 V8. It also features Java technology including Motorola's Java test framework and test cases. And soon, code, documents, and specifications for Motorola-led JSRs such as MIDP 3.0 will be made available.

Eclipse Foundation

Motorola's membership in the Eclipse Foundation is another important step in reaching the company's goal of increased Linux adoption in the mobile space through the delivery of a truly open software platform. Motorola is a Strategic Developer Member and will contribute code and engineering support through its leadership of the Tools for Mobile Linux project, which supports the development of stronger tools for mobile Linux, including the frameworks and tools for the development and emulation of C++ applications targeting mobile devices.

LiMo Foundation

Motorola, NEC, NTT DoCoMo, Panasonic, Samsung, and Vodafone formed the LiMo Foundation to reduce the fragmentation of mobile Linux and to support the creation of the world's first globally competitive, Linux-based middleware platform for mobile devices. The LiMo Foundation, through a fair and balanced contribution and participation process, focuses on the joint development of a Linux-based middleware platform and set of API specifications. This platform doesn't touch the layers of the stack that directly shape the user experience and are used by individual members for competitive differentiation. As a founding member of the LiMo Foundation, Motorola will make every effort to maintain compliance with the LiMo platform specifications in MOTOMAGX.

Open Handset Alliance

Motorola has joined Google and others in the formation of the Open Handset Alliance to advance the development of Android, an open and comprehensive Linux-based platform for mobile devices. Motorola plans to leverage the Android platform to enable seamless, connected services and rich consumer experiences in future Motorola products.

SUMMARY

For a platform to be successful, it must tap into the open source community, be flexible, and provide a thriving ecosystem. The modular and extensible architecture of MOTOMAGX maximizes flexibility and speeds time to market, while support for open standards and open platform APIs, along with superior tools and support, accelerates innovation. Furthermore, Motorola demonstrates its commitment to Linux and open source through participation in efforts to standardize the Linux platform and tools, and thereby encourage the adoption of Linux as the platform of choice for mobile devices.

MOTOROLA

Motorola, Inc.
www.motorola.com

Motorola reserves the right to make changes to the information contained in this white paper. All information is accurate as of the date of release.
06/08