J2ME Programming Models

Technical Information Paper – v1.0

19 October 2004

Authors:

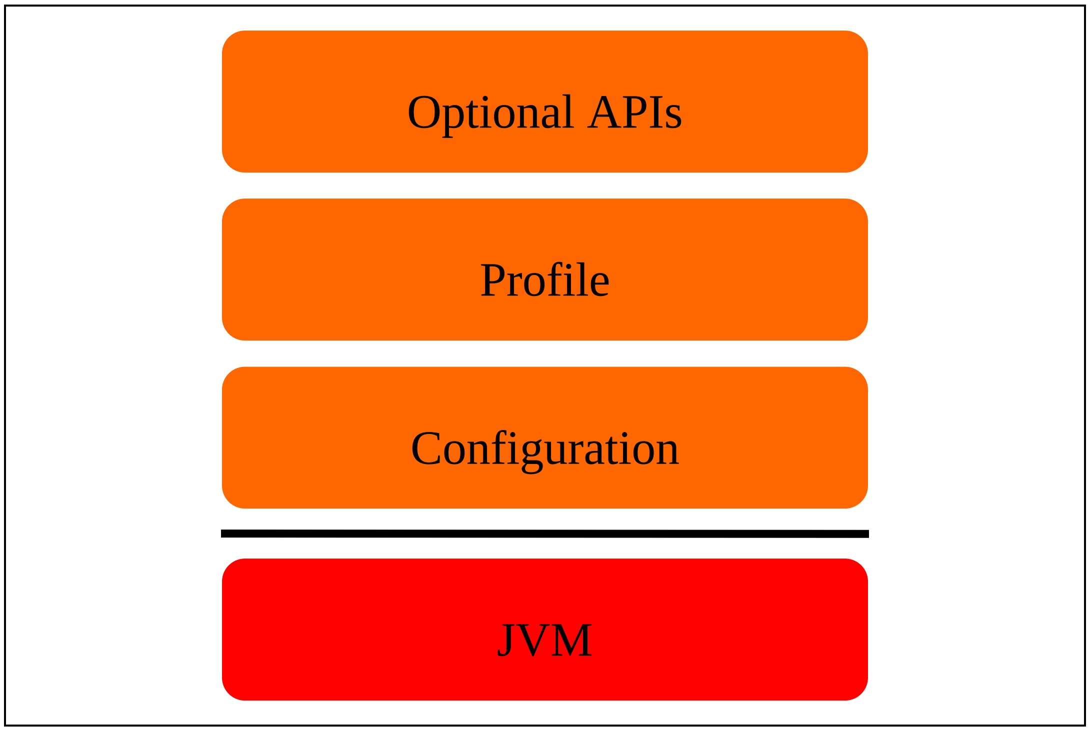
Name	Contact eMail
Kevin Horowitz	wctmepvc@us.ibm.com

What is Java 2, Micro Edition

Java 2, Micro Edition (J2ME) is the programming model introduced to standardize programming for embedded devices, such as phones, personal digital assistants (PDAs), and vehicle telematics. It was formalized during the division of Java 2 into 3 editions, Standard Edition (J2SE) for the desktop and lower requirement business applications, Enterprise Edition (J2EE) for the enterprise environment, and Micro Edition.

J2ME is not a single specification, but a family of specifications for different types of resource-constrained devices in a variety of environments. J2ME is composed of a set of configurations, profiles, and optional APIs that can be combined to give the power and flexibility needed for a particular class of devices. The J2ME specification describes the combination of configurations, profiles, and optional APIs as building blocks that are designed to meet the various needs of the device and user. These building blocks provide the APIs that are used to build an application.

J2ME Layers



IBM's implementation of the J2ME specification, the J9 VM, is based on the Java Virtual Machine Specification Version 1.3.

Configuration

A configuration provides the set of class libraries which provide base functionality for a particular range of devices with similar capabilities. That is, they define the minimum set of features that make up the java runtime. The two currently defined J2ME configurations are the Connected Limited Device Configuration (CLDC) and the Connected Device Configuration (CDC). Configurations provide only a minimal set of APIs.

Profile

A profile adds power to the basic capabilities defined in the configuration. A profile narrows the category of devices that can be supported by a configuration by adding more powerful APIs. These APIs can define functionality such as the user interface, the event models, and device-specific properties. In addition, profiles usually define any available user interfaces and the model used to load and activate an application. The addition of a profile can provide for development of a complete Java application for a particular device. It is assumed that applications built to a particular Profile will be portable across all devices that implement the profile.

WebSphere Studio Device Developer/WebSphere Everyplace Micro Environment (WEME), without any extensions, supports CDC with the Foundation Profile version 1.0 or Personal Profile version 1.0. Alternately, CLDC is supported with the Mobile Information Device Profile (MIDP) version 1.0 and 2.0.

Optional APIs

Optional APIs are additional libraries that can be used on top of a Profile to address specific market needs. Optional APIs can be used to needs such as Web Services or database connectivity.

Java Community Process

All of the Configurations, Profiles, and even J2ME have been defined or are being refined as part of the Java Community Process (JCP). The JCP was introduced by Sun Microsystems, Inc., to develop and revise Java technology specifications. The process creates an expert group to define and approve standards. The following chart shows the Java Specification Requests (JSRs) that are part of the base level WebSphere Studio Device Developer. The JSRs are available at http://www.jcp.org.

JSR Number	Name	Finalized
30	J2ME™ Connected, Limited Device Configuration	May 2000
36	J2ME™ Connected Device Configuration	August 2002
37	Mobile Information Device Profile for the J2ME™ Platform	November 2002
46	J2ME™ Foundation Profile	August 2002
62	Personal Profile Specification	September 2002
118	Mobile Information Device Profile 2.0	September 2000
139	Connected Limited Device Configuration 1.1	March 2003

Connected Limited Device Configuration

The Connected Limited Device Configuration (CLDC) specification defines a standard for a portable resource constrained device. Devices requiring CLDC usually have both power and memory constraints. The broad category of devices assumed to be able to run the CLDC configuration includes cell phones, PDAs, home appliances, and some TV set-top boxes.

CLDC has completed its second approved standard. Version 1.0 is defined in JSR 30 (May 2000). Version 1.1 is defined by JSR 139 (March 2003), and is a backward compatible incremental release minor enhancements and bug fixes. CLDC contains a minimal portion of the J2SE specified APIs, as well as some CLDC specific APIs called the Generic Connection Framework (GCF). The GCF defines an I/O system for devices with limited I/O capabilities. However, it does not define the device implementation.

Mobile Information Device Profile

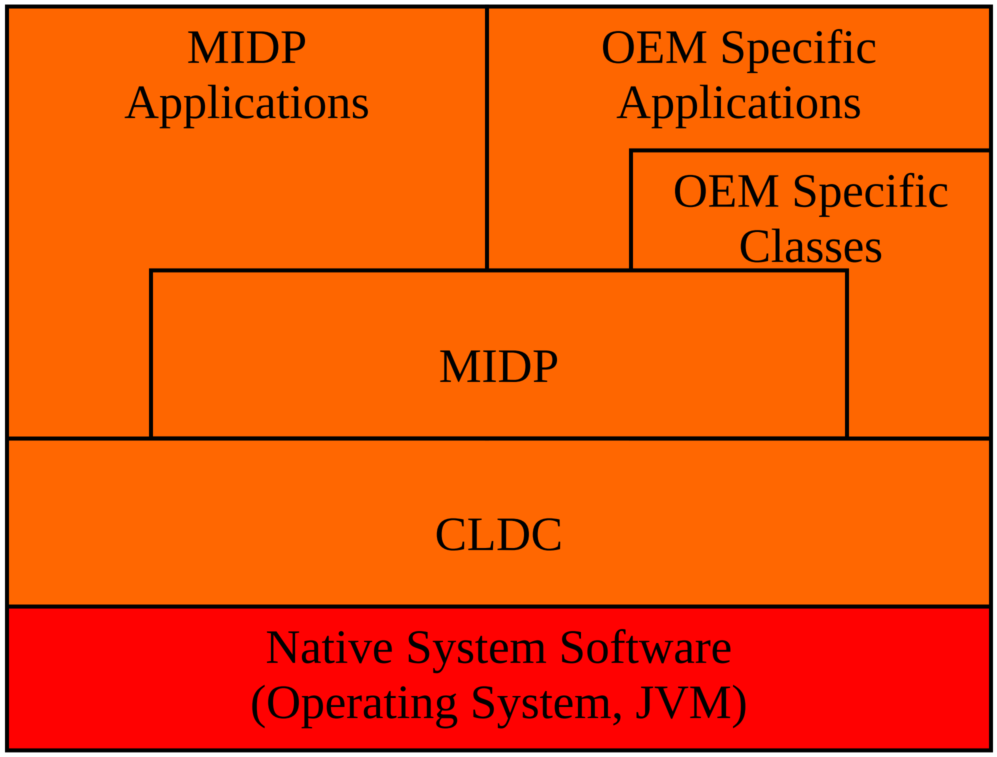
The Mobile Information Device Profile (MIDP) specification defines a set of APIs to enable standardized application development for mobile information devices (MIDs). Applications written to the MIDP specification are know as Midlets.   Examples of devices that MIDP applications run on are cellular phones, two-way pagers, and wireless PDAs.

MIDP is designed to be executed on top of CLDC and has completed its second release cycle.   MIDP 2.0, JSR 118, (September 2000) is designed to be backward compatible with MIDP 1.0, JSR 37 (November 2002).

The MIDP specification defines requirements and APIs for application lifecycle, security model (including application signing), end-to-end transactional security, networking, persistent storage, sound, timers, and a user interface (UI).   The basic requirements to support a MIDP application are a minimum-sized bit addressable screen, an input device such as a keyboard or touch screen, networking capability, and a minimal sound ability.

MIDlets are packaged in a construct called a MIDlet Suite. MIDlet Suites are a standard JAR (Java Archive) with control application information stored in an application descriptor and control information in a manifest file.

MIDP High-Level Architecture View


MIDP Applications
OEM Specific Applications
OEM Specific Classes
MIDP
CLDC
Native System Software (Operating System, JVM)

Connected Device Configuration

The Connected Device Configuration (CDC) specification defines a standard for embedded devices that do not have as many resources as a desktop environment, but have less resource constraints then a device requiring CLDC. The CDC specification defines a richer set of J2SE libraries, while maintaining the CLDC, Generic Connection Framework, APIs. CDC also adds some file I/O operations.

CDC is a superset of CLDC.   CDC, JSR 36, and was finalized in August 2002 through a maintenance update.

Foundation Profile

The J2ME Foundation Profile (FP) is a set of APIs that are designed to run on a CDC configuration. Foundation adds some additional J2SE classes and methods back into the CDC libraries.   Some of the API additions that the Foundation Profile adds to CDC are socket classes, and internationalization and localization support. As its name implies, the Foundation Profile is intended to serve as a foundation for other profiles.

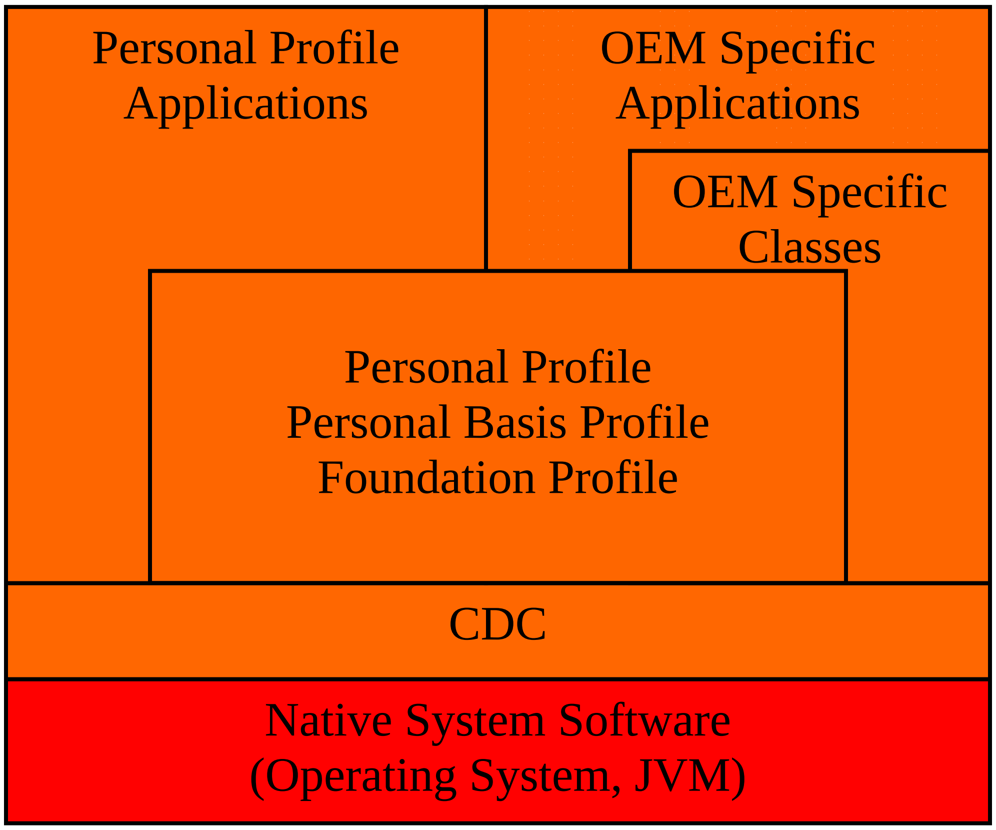
The Foundation Profile specification, JSR 46, was finalized in August 2002.

Personal Profile

The J2ME Personal Profile (PP) is a set of APIs that are designed to run on a CDC configuration, and is a superset of both Foundation Profile and Personal Basis Profile (JSR 129). The PP specification, JSR 62, was finalized in September of 2002.

Personal Profile includes API support for most of Java Abstract Windowing Toolkit (AWT) as well as support for an applet application programming model. Devices that can run Personal Profile include high-end PDAs as well as embedded web browsers.

Personal Profile High-Level Architecture View



Unified Emulator Interface Devices

The Unified Emulator Interface (UEI) defines a set of standards for an emulator to execute J2ME code. UEI allows application developers to support multiple emulators by standardizing the emulator's directory, command and command line arguments. The emulator can run on the local file system or run in a remote, Over The Air (OTA) manner. When running in an OTA-mode, there is no defined manner to install the application on the device. This installation is vendor specific.

The standard is published by Sun Microsystems, Inc., and there are currently two versions of the standard. The version 1.0 Draft was released in June of 2002. The version 1.0.1 was released in December of 2003.

There are a few differences between the versions of the specification. The major difference between the two versions is that some tracing options supported in version 1.0 have been eliminated in version 1.0.1, and some command-line requirements have been modified. It is possible to query the emulator to get a version, and therefore know how to programmatically provide only the options available.

The UEI specification (and the WCTME implementation) is geared around running a MIDP application, however, the specification is clear to state, that there is no requirement on the emulator that specifies what type of configuration or profile are possible.

Java Specification Request

In addition to the base level of support provided in WebSphere Studio Device Developer, optional libraries are available from WCTME to support additional features, and support the mobile environment from a J2EE backend. Many of these optional libraries are implementations of additional JSRs as defined in the Java Community Process. These will be discussed in detail in follow-on papers, but some of the JSRs supported include:

JSR Number	Name	Finalized
66	RMI Optional Package Specification Version 1.0 (RMIOP)	June 2002
75	PDA Optional Packages for the J2ME Platform – File Connection (FC)	June 2004
75	PDA Optional Packages for the J2ME Platform – Personal Information Management (PIM)	June 2004
135	Mobile Media API (MMAPI)	June 2003
169	JDBC Optional Package for CDC/Foundation Profile	April 2004
172	J2ME Web Services Specification	March 2004

OSGi™ Service Platform

The OSGi Alliance (http://www.osgi.org/) is an independent, non-profit organization working to define open, component oriented, standards for the delivery of services to networked devices.

The OSGi Service Platform, Release 3.0 (approved March of 2003) specifies the service delivery framework. The service platform describes a software deployment and configuration management architecture. IBM's implementation is packaged as Service Management Framework (SMF) and is provided as part of the IBM Micro Environment Toolkit for WebSphere Studio and can be exercised in WebSphere Studio Device Developer.

The goals of the service platform are to provide a secure component-based application programming model. Applications are provisioned in a secure sandbox, but can be deployed to be able to dynamically discover services running within the same service platform.