

Java ME Architecture

Version 0.9, Draft



INFORMATION GUIDE

COPYRIGHT

Samsung Electronics Co. Ltd.

This material is copyrighted by Samsung Electronics. Any unauthorized reproductions, use or disclosure of this material, or any part thereof, is strictly prohibited and is a violation under the Copyright Law. Samsung Electronics reserves the right to make changes in specifications at any time and without notice. The information furnished by Samsung Electronics in this material is believed to be accurate and reliable, but is not warranted true in all cases.

Trademarks and Service Marks

The Samsung Logo is the trademark of Samsung Electronics. Java is the trademark of Sun Microsystems.

All other company and product names may be trademarks of the respective companies with which they are associated.



About This Document

This document describes about the Java ME Architecture, explaining the building blocks of Java ME.

Scope:

This article is for novice users who want to know about the basics of Java ME. It explains about most popular CLDC configuration and MIDP Profiles.

Document History:

Date	Version	Comment
02/02/09	0.9	Draft

Reference:

1. Java ME:

<http://java.sun.com/javame/technology/index.jsp>

2. Mobility Article:

<http://developers.sun.com/mobility/midp/articles/optional/>

Abbreviations:

J2EE	Java 2 Platform Enterprise Edition
MIDP	Mobile Information Device Profile
MIDP_NG	MIDP Next Generation
CLDC	Connected Limited Device Configuration
JVM	Java Virtual Machine
KVM	KiloByte Virtual Machine
API	Application Programming Interface
OTA	Over The Air

Table of Contents

Introduction.....	5
Overview	5
Configuration.....	7
Kilobytes Virtual Machine (KVM).....	7
Connected Limited Device Configuration (CLDC).....	7
Profile	7
Mobile Information Device Profile (MIDP).....	7
Optional Packages	8
MIDlet	9

Table of Figures

Figure 1: Java 2 Platform Editions.....	6
Figure 2: Java ME Architecture.....	6



Introduction

Sun Microsystems has classified Java platform into three editions:

- Java 2 Platform, Enterprise Edition (J2EE)
- Java 2 Platform, Standard Edition (J2SE)
- Java 2 Platform, Micro Edition (Java ME)

Java Micro Edition (Java ME) is an application platform to use Java technologies on resource-constrained embedded devices like Mobile Phones, PDAs and Palmtops etc. It is a minimalist version of Java, targeted at devices, which have limited processing power, storage capabilities and intermittent or low-bandwidth network connections.

Overview

The Java ME architecture is generally classified into 3 layers:

- Java Virtual Machine
- Configuration
- Profile



Java Virtual Machine is an interpreter and provides the runtime environment for Java applications. Java source code is compiled into a format called "bytecode" which is executed by Java Virtual Machine.

Configuration is a minimal set of core class libraries that provides the basic functionality for a particular range of devices.

Profile is an extension of the configuration that addresses the specific demands of a device family.

Figure 1 shows Java ME components and other Java Technologies.

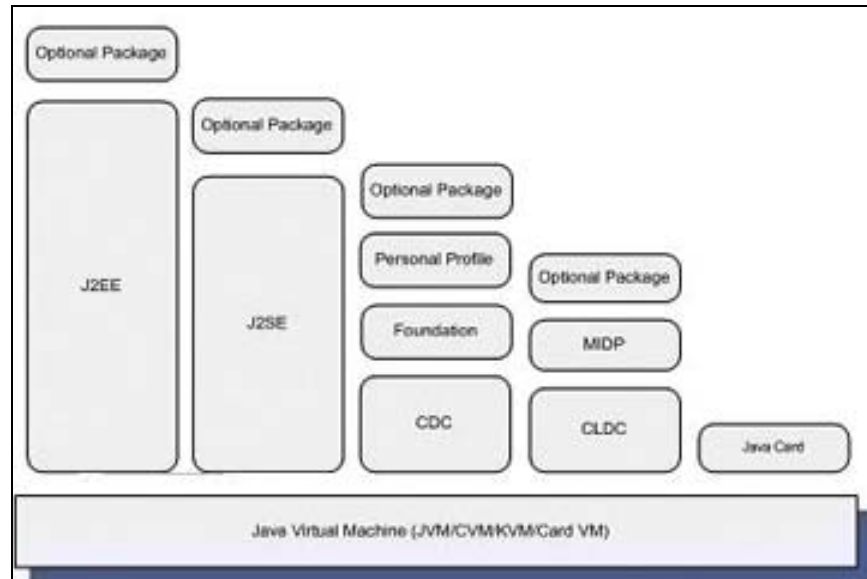


Figure 1: Java 2 Platform Editions

Figure 2 explains the various components that compromise Java ME Architecture.

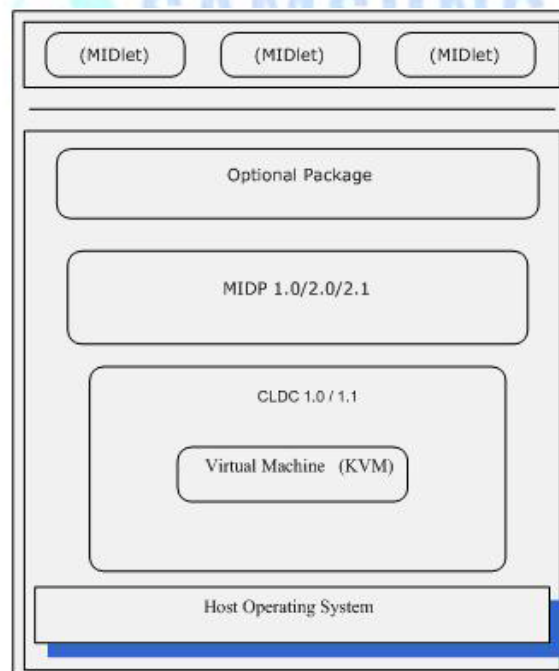


Figure 2: Java ME Architecture

Configuration

Configuration provides the most basic set of libraries and limited Java Virtual Machine capabilities for Java ME devices.

Kilobytes Virtual Machine (KVM)

Kilobytes Virtual Machine (KVM) is a compact, portable Java Virtual Machine designed for resource-constrained devices. These devices typically contain 16 or 32 bit processors and a minimum memory footprint of approximately 128 kilobytes. KVM maintains all the aspects of the Java programming language, and runs in a resource-constrained device with only a few hundred kilobytes of total memory budget.

Connected Limited Device Configuration (CLDC)

CLDC is designed for mobile devices. This configuration typically has the following features:

- 160 – 512K of memory available for Java.
- Typically has limited power or battery operated.
- Network connectivity, often wireless, intermittent, low-bandwidth.
- Core Java class libraries available on a particular category of devices.
- 32 kilobytes memory for runtime memory allocation.

Profile

Profile adds domain-specific classes and helps to create interactive applications. The most popular profile is Mobile Information Device Profile (MIDP). Currently MIDP 1.0, MIDP 2.0 & 2.1 are the versions of MIDP. There are number of profiles available such as Foundation Profile, Personal Basis, Personal and RMI profiles each having its specific features and target devices.

Mobile Information Device Profile (MIDP)

MIDP is set of Java APIs and it is more domain specific than the class libraries of CLDC. This addresses the user to persistence storage, networking and user interface where the prime focus is on the guaranteed interoperability within certain devices family. One of the core features of the Mobile Information Device Profile (MIDP) technology is its support for developing user interfaces.

Mobile Information Device has the following specifications:

The primary focus of the MIDP_NG specification scope will address:

- Backward compatibility with MIDP 1.0
- Continued focus on small, high-volume wireless phones
- Maintain tight footprint objectives to limit growth in the core APIs
- Information learned from MIDP 1.0 deployments to fine tune MIDP 1.0 APIs
- Focus on core functions needed by all devices and applications
- Focus on enabling mCommerce, service-based applications

Optional Packages

Optional Packages are nothing but the bunch of Java Classes, which extend the core platform. These Java classes can be packaged with configurations and profiles to create software pack. Optional package has the functionality, which is independent of any particular device family. It is a set of APIs that is most of the time layered on top of (and thus extends) a profile. It allows the definition of APIs, which can add flexibility on top of other profiles.

Configuration and profiles extend the runtime environment to support device capabilities that are not universal enough to be defined as part of a profile or that need to be shared by different profiles.

Java ME supports number of optional packages. Some of them are listed below:

- Wireless Messaging API (WMA), JSR 120, JSR 205
- Mobile Media API (MMAPI), JSR 135, JSR 234
- Location API for Java ME, JSR 179
- SIP API for Java ME, JSR 180
- Security and Trust Services API (SATSA), JSR 177
- Mobile 3D Graphics, JSR 184
- Web Services APIs (WSA), JSR 172
- Bluetooth API, JSR 82

MIDlet

MIDlet is a MID Profile application. Application should extend the MIDlet class to get the Application Management Software (AMS) to control the MIDlet and to be able to retrieve properties from the application descriptor, notify and request the state change. Application Management Software manages the activities of multiple MIDlets within a runtime environment using these states.

MIDlet has three methods which user must override. They are:

- *startApp()*
- *pauseApp()*
- *destroyApp (boolean)*

With the help of this class method, Application Management Software can create and control the start, pause and can also destroy a MIDlet.

