Java 2
Micro Edition
(J2ME)
Heads up

• Java platforms
• J2ME history
• J2ME positioning
• J2ME concepts
• J2ME components
• Standardization through CLDC & MIDP
• Exemples
Java Platforms

Java Technology Enabled Devices
Java Technology Enabled Desktop
Workgroup Server
High-End Server

Micro Edition
Standard Edition
Enterprise Edition
History

- 1998: Sun Microsystems Laboratories rolls out the "Spotless" research project.

- Stated goals:
  - Assessment on Java usability for embedded platforms.
  - Development resources for a VM that has the following particularities:
    - Small
    - Portable
    - Ease of use and development
History (continued)

- June, 1999: Sun Microsystems Laboratories presents Java 2 Micro Edition (J2ME)
- October 1999: Standardization protocols for J2ME components start
Java 2 Micro Edition

Goals

• End-user hardware appliances and services have a stable base for new applications and services to build onto

• Hardware makers can let others develop standardized applications for their devices

• Users can dynamically download and install full-featured applications on a wide range of devices
Devices

Figure 2-1  Java 2 editions and their target markets
Devices (continued)

- Mobile phones
- PDAs
- TV receivers/decoders
- Gaming consoles
- Bank terminals
- Vehicle-embedded appliances
- Wearable computer
Device constraints

- Limited user interaction (no keyboard/mouse, ...)
- Limited displays (size, color, ...)
- Limited memory
- Communication
- Limited power (battery)
- Size constraints (small, light, ...)
- Shock-resistant
Configuration

- A configuration defines a minimal platform for a given category of devices, based on processing power and memory.

- A configuration defines the smallest set of functionalities to be in the Java platform. It generally applies to a very wide scope of devices.
Existing Configurations

- Connected Device Configuration (CDC)
- Connected Limited Device Configuration (CLDC)
Connected Device Configuration (CDC)

- Identifies end-user "top of the range" devices
- Hardware specifications:
  - Wide range of user interaction devices
  - Total memory > 512 KB
  - 32 bits CPU
  - Stable connection to a broadband network access
- Exemples : TVs et Internet videophones, navigation systems, entertainment appliances.
Connected Limited Device Configuration (CLDC)

- Identifies end-user "low class" devices
- Hardware specifications:
  - Memory: 128 KB to 512 KB of available memory for Java processes
  - 16 or 32 bits CPU
  - Battery powered devices, low consumption
  - Small bandwidth on a potentially intermittent connection
- Exemples: mobile phones, pagers, PDAs, barcode readers, ...
J2SE & Configurations

J2SE

CDC

CLDC
Profile

- A profile is a layer on top of a configuration
- A profile gives a common platform for a family of devices
- A profile specifies a set of functionalities
- A profile may have additional libraries
Existing profiles

• CDC profiles
  - Foundation Profile JSR-46
  - Personal Profile JSR-62
  - Personal Basis Profile JSR-129
  - RMI Profile JSR-66
  - Game Profile JSR-134

• CLDC profiles
  - Mobile Information Device Profile (MIDP) JSR-37 et JSR-118
Mobile Information Device Profile (MIDP)

• Specifications
  • Memory: 128 KB of non-volatile memory for MIDP components, 8 KB of non-volatile memory for persistent data, and 32 KB of memory of volatile memory
  • Display: 96 x 54, 1 bit
  • User interaction: Keyboard (one or two hands), touchscreen
  • Network: Both ways, wireless, intermittent, small bandwidth
Existing VMs

- CDC Configuration
  - CVM (Sun)
  - JVM (Sun)
  - J9 (IBM)
- CLDC Configuration
  - KVM (Sun)
  - Monty (Sun)
  - MicrochaiVM (HP)
  - JBED (Esmertec)
Kuaui Virtual Machine (KVM)

• 16 & 32 bits CPU

• Specifications :
  - ~ 60 KB of static memory usage
  - Highly portable, fully documented
  - Module based for good customization
  - As fast and complete as possible

• Supported OS
  - Windows CE
  - Solaris Operating Environment
  - Palm OS
Architecture

- J2EE (Java 2 Enterprise Edition)
- J2SE (Java 2 Standard Edition)
- J2ME (Java 2 Micro Edition)
- CDC (Connected Device Configuration)
- CLDC (Connected Limited Device Configuration)
- Java Card API

Packages Optionnels

JVM
KVM
Card VM

Système d’exploitation
CLDC & MIDP Standardization

- **Java Application**:
  - A Java application is a class collection in which one of the classes has a main method: `public static void main(String[] args)`

- **Application handling**
  - Download and install applications
  - Inspect existing applications on the device
  - Select and execute applications
  - Uninstall existing applications
CLDC & MIDP
Standardization

• Java
  - No floating point unit
  - No finalize
  - Limited exceptions

• KVM
  - No FPU
  - No JNI
  - No dynamic class loader
  - No reflexivity
  - No finalize
  - No thread groups
Architecture
CLDC & MIDP

Applications MIDP
MIDP
CLDC (KVM)
Système d’exploitation
Applications OEM
API OEM
OEM
Application Life-Cycle
Compilation

"javac" takes java files, and compiles them into class files
Pre-verification

Takes the program and runs it against a "simulated" environment to ensure maximum compatibility
Execution (Phase 1)

- The runtime verifier looks up the sent bytecode and checks for compatibility
- If this verification fails, the program won't run
Execution (Phase 2)

- If the verification completes successfully, the classes are loaded
- Interpretation (execution) can now start
Application cycle

A Java MIDP application is called a « MIDlet »

• States of a « MIDlet »
  – « Active »
  – « Pause »
  – « Destroy »
package examples;
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class HelloMIDlet extends MIDlet implements CommandListener {
    private Command exitCommand;
    private Ticker hi = new ("J2ME is cool");
    public HelloMIDlet() {
        exitCommand = new Command("Exit", Command.EXIT, 1);
    }

    public void startApp() {
        TextBox tb = new TextBox("Hello MIDlet ", "Wireless Internet", 256, 0);
        tb.addCommand(exitCommand);
        tb.setTicker(hi);
        tb.setCommandListener(this);
        Display.getDisplay(this).setCurrent(tb);
    }

    public void pauseApp() {}

    public void destroyApp(boolean unconditional) {

    }

    public void commandAction(Command c, Displayable d) {
        if (c == exitCommand) {
            destroyApp(false);
            notifyDestroyed();
        }
    }
}
Result

- J2ME is cool
- Hello MIDlet
- Wireless Internet

[Image of a mobile phone interface with various options and buttons]
Application Exemples
References

- J2ME : http://java.sun.com/j2me/
- CDC : http://java.sun.com/products/cdc/