

Java ME

Plateformes mobiles et outils de développement

Plateforme	Outils de développement	SDK
Windows Mobile Windows Phone 7	• Visual studio .Net	• Pocket PC SDK • Smartphone SDK • Windows Mobile 6.x SDK • Windows Phone 7 SDK
Symbian OS S60 C++	Performance (code natif c++)	• Très peu portable, même sur des versions différentes Symbian.
JAVA ME	• Standard ouvert (Nokia, Sony-Ericsson, Samsung, Motorola, ...). • Le plus grand parc de téléphones	• Wireless Tool Kit
Androïd	• ADT Plugin pour Eclipse (Java, code natif c++)	• Androïd SDK
IPhone	Xcode (Objective C)	IPhone SDK

Historique Java ME (1/2)

- Architecture
 - Contraintes matérielles
 - Faible capacité mémoire des systèmes cibles
 - Nécessité d'une faible consommation électrique
 - Connectivité intermittente et à faible débit
 - Entrées / Sorties limitées
- Projet initié par sun en 1998.
 - Groupe d'experts
 - AOL, Bull, Ericsson, Fujitsu, Matsushita, Mitsubishi, Motorola, Nokia, NTT DoCoMo, Oracle, Palm, Research In Motion, Samsung, Sharp, Siemens, Sony, Sun, Symbian.
 - Spécifications des machines cibles
 - Machine virtuelle de taille très réduite : KVM
 - Bibliothèques réduites
 - Deux configurations de base
 - CDC (Connected Device Configuration): Appareils électroniques. KVM < 2 Mo
 - CLDC (Connected Limited Device Configuration) KVM < 512 Ko

Historique 2/2

- 1999 standardisation du profil MIDP
 - Point de départ : CLDC
 - Fonctionnalités prise en considération
 - Interface utilisateur,
 - Réseau,
 - Stockage

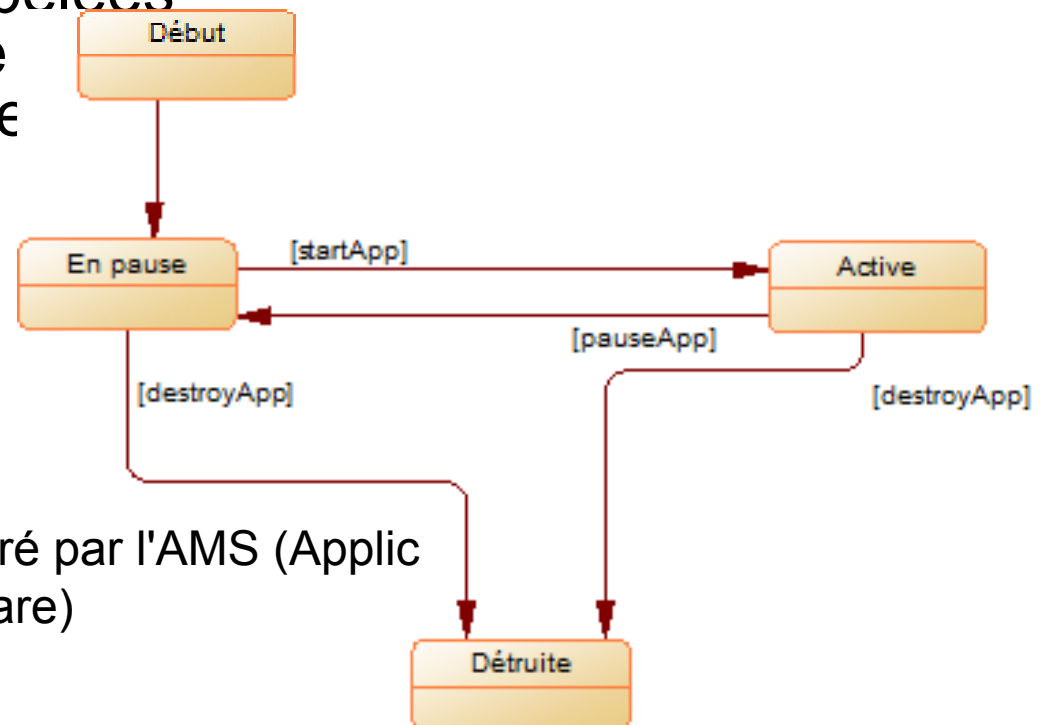
Architecture Java ME

- configuration CLDC (Connected Limited Device Configuration) : contient une version minimale de la JVM pour des terminaux possédant de 128 à 512 Ko de mémoire (comme les téléphones mobiles)
- profil MIDP (Mobile Information Device Profile): fournit l'API nécessaire aux applications Java ; notamment il contient le paquetage javax.microedition.lcdui associé aux écrans LCD. MIDP est utilisé avec CLDC. CDC utilise d'autres profils (Foundation Profil et Personal Profil).
- paquetages optionnels : ensemble d'API qui concernent des fonctionnalités qui ne sont nécessairement supportées par tous les terminaux (exemple: bluetooth, sms et mms...)

MIDP

Les MIDlets

- Package javax.microedition
- Classe de base de toute application MIDP: Toutes les applications MIDP héritent de la classe abstraite javax.microedition.midlet.MIDlet
- Trois méthodes sont appelées pour prendre en charge cycle de vie d'une MIDlet
 - startApp()
 - pauseApp()
 - destroyApp()



Le cycle de vie d'une Midlet est géré par l'AMS (Application Management Software)

Etapes de création d'une MIDlet

- Etape 1 : Design
L'IHM doit être conçu pour tenir compte d'un terminal mobile.
- Etape 2 : Ecriture du code.

```
package ateliers.cours;
import javax.microedition.lcdui.*;
import
javax.microedition.midlet.MIDlet;
public class Ate1 extends MIDlet
{
    Alert t;
    public Ate1()
    {
        t = new Alert("Exemple1");
    }
    public void startApp()
    {
```

```
        Display.getDisplay(this).setCurrent(t);
    }
    public void pauseApp()
    {
    }
    public void destroyApp(boolean
unconditional)
    {
    }
}
```

Etapes

- **Etape 3 : Compilation**

Le compilateur javac crée la classe Ate1.class dans le répertoire C:\WTK2.5.2_01\ateliers\cours>javac -classpath..\lib\clcdapi11.jar;..\lib\midp20.jar intro\Ate1.java

- **Etape 4 : Pré vérification**

Cette étape permet de vérifier que tout est correct vis à vis de l'exécution de l'application sur un terminal mobile.

```
preverify.exe -classpath ..\lib\clcdapi11.jar;..\lib\midp20.jar cours.Ate1
```

Cette commande crée un répertoire output sous le répertoire courant (ici WTK22\ateliers) et y place la version prévérifiée de la classe Ate1.class.

- **Etape 5 : Paquetage**

Plusieurs phases sont nécessaires au paquetage souhaité pour une utilisation sur le terminal mobile.

il faut créé un fichier Manifest.mf qui continet quelques métadonnées :

```
MIDlet-Name: Ate1  
MIDlet-Version: 1.0.0  
MIDlet-Vendor:
```

Ce fichier, **qui se termine par une ligne blanche**, est sauvegardé dans le répertoire output créé précédemment.

Ensuite on crée le fichier JAR qui est la paquetage contenant Ate1.class et Manifest.mf à l'aide de la commande jar

```
C:\WTK22\exemples\output>jar cvfm Ate1.jar Manifest.mf .\cours
```

Etapes

Enfin, il faut maintenant créer un fichier jad qui est un descripteur Ate1.jad permettant au terminal mobile d'installer l'application :

```
MIDlet-1: Ate1  
MIDlet-Name: Ate1  
MIDlet-Version: 1.0.0  
MIDlet-Vendor:  
MIDlet-Jar-URL: ATe1.jar  
MIDlet-Jar-Size: 1058  
MicroEdition-Profile: MIDP-2.0  
MicroEdition-Configuration: CLDC-1.1
```

La taille du fichier jar doit être exacte (elle est de 1058 octets sur la machine utilisée).

- ***Etape 6 : Test avec un émulateur***

On peut utiliser un émulateur de terminal mobile pour vérifier que l'application fonctionne. Le kit WTK propose un émulateur que l'on peut appeler par la commande :

```
C:\WTK22\exemples\output>..\..\bin\emulator.exe -Xdescriptor  
Ate1.jad
```