Créer un “ Log File “ sous NetBeans IDE 6.5 et la plate-forme J2ME

Zied CHEBIL

INTRODUCTION

J2ME (Java 2 Micro Edition) ou Java ME est le framework Java spécialisé dans les applications mobiles.

Des plates-formes Java compatibles avec J2ME sont embarquées dans de nombreux téléphones portables et PDA.

INTRODUCTION

Une plate-forme J2ME est composée :

* d'une KVM (Kilobyte Virtual Machine), une machine virtuelle capable d'exécuter une application Java
* d'une « configuration », une API donnant accès aux fonctions de base du système
* d'un « profil », une API donnant accès aux fonctions spécifiques de la plate-forme.

INTRODUCTION

Les configurations les plus courantes sont :

* CLDC (Connected Limited Device Configuration), que l'on retrouve par exemple dans les téléphones mobiles

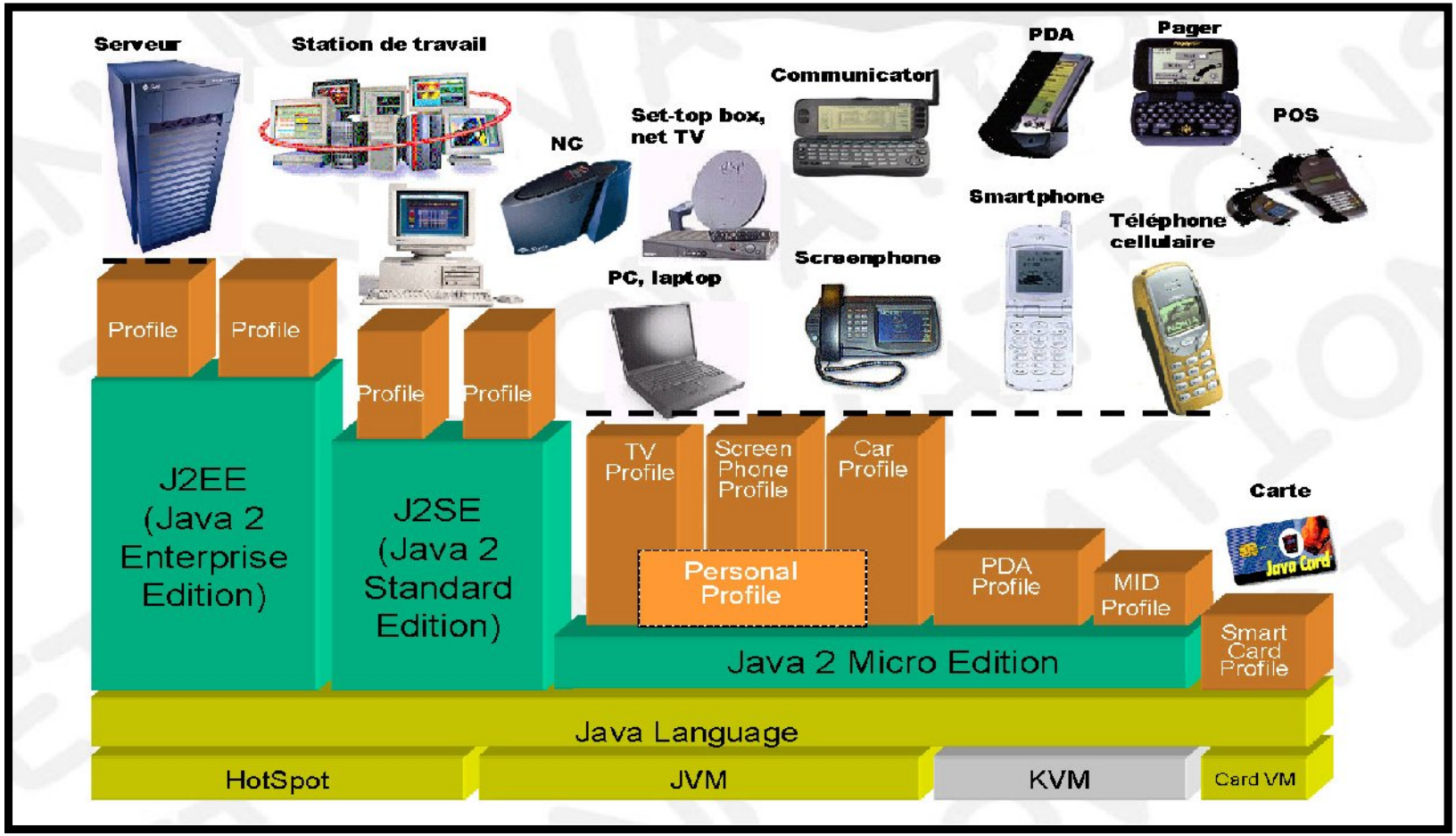
* CDC (Connected Device Configuration), qui est plutôt utilisé dans des décodeurs de télévision numérique

Les profils les plus courants sont :

* MIDP (Mobile Information Device Profile), dont sont équipés les téléphones WAP J2ME

* DoJa, développé par NTT DoCoMo pour les téléphones i-mode J2ME

INTRODUCTION

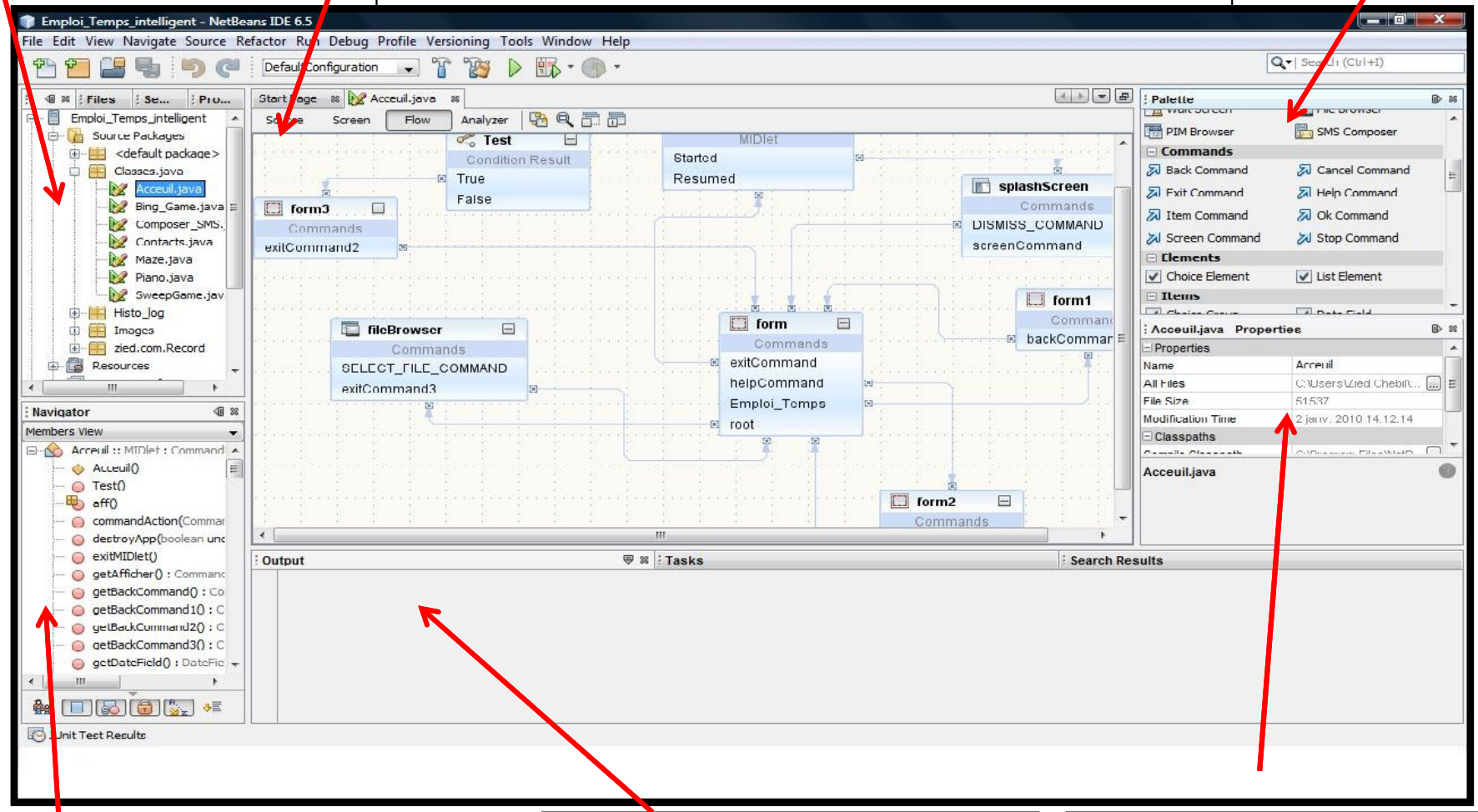


Présentations de NetBeans IDE 6.5

Projects
-Choix du projet actif
-Visualisation des éléments java du projet

Style d`affichage
-Source :code java
-Screen :concepton d`un seul écran de l`IHM en utilisant la palette de composants
-Flow : conception de l`enchainement de l`IHM en utilisant la palette de composants

Palette
-composant graphiques



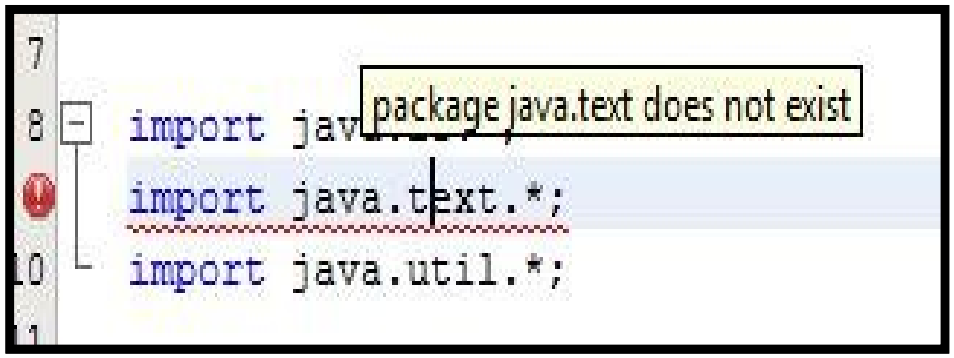
Navigator
-Visualisation de l`organisation des composants graphiques

Output
-Affichage des messages de construction du projet et des sorties console

Properties
-Affichage des messages de construction du projet et des sorties console

Créer un « Log file » sous NetBeans IDE 6.5 avec J2ME

Dans l`IDE NetBeans et sous la plate-forme J2ME, certains paquetages sont insupportables ou absents. Parmi ces paquetages celle de « java.text ».



Malgré que ce paquetage existe dans les différentes plate-forme, de java mais il est non fonctionnel sur la plate-forme J2ME.

Problème

Comment donc créer un Log file sous la plateforme J2ME ?

Solution

Java offre un API pour résoudre ce problème, ce dernier s`appelle « **File Connection API** »

JAVA offre un « File Connection API » dont le but d`accèder aux fichiers système de l`appareil téléphonique.

```
FileConnection fc =
(FileConnection)Connector.open(String URL, int mode) ;
```

Format **URL** :
file:///<host>/<root>/<directory>/<directory>/.../<name>

L`URL débute avec "file:///" pour indiquer que le fichier est accessible dans le localhost

Les modes : Connector.READ, Connector.WRITE, ou Connector.READ_WRITE

Exemple : Ecrire un « Log file »

```
public void ecrireLigne(String s) throws IOException
 {
  OutputStream out;
  PrintStream print;
   FileConnection fc =
(FileConnection)Connector.open("file:///root1/java.txt");
  //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
  out = fc.openOutputStream(fc.fileSize());
  //On ouvre un flux PrintStream depuis le flux de sortie
  //pour faciliter l'écriture ds le fichier
  print = new PrintStream(out);
  //On écrit la chaine avec un retour chariot
  print.println(s);
  //Fermeture des flux
  print.close();
  out.close();

 }
```

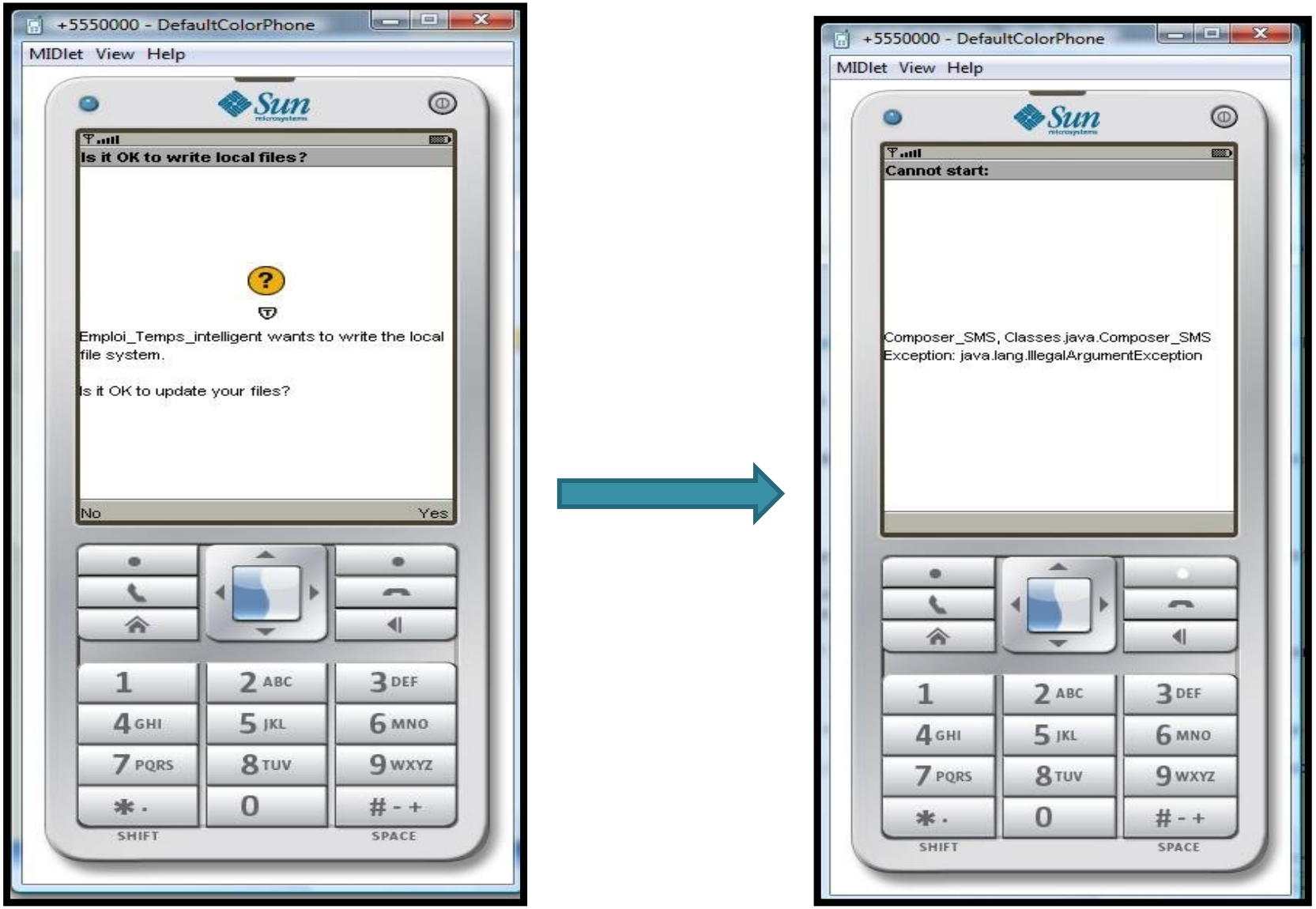
Exemple : Lire un « Log file »

```
public  String readHelpText() throws IOException
 {
   InputStream is ;
    FileConnection fc = (FileConnection)Connector.open("file:///root1/java.txt");
 //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
 is = fc.openInputStream();
String text ="" ;
   try
   {
     StringBuffer sb = new StringBuffer();
     int chr, i = 0;
     // lire jusqu`a la fin de  stream
     while ((chr = is.read()) != -1)
      text =text+((char) chr)  ;
return text;
   }
   catch (Exception e)
   {System.out.println(« Impossible d`acceder au stream"); }
return null;
 }
```

Exemple : Effacer le contenu de « Log file »

```
public void effacer_log_fichier() throws IOException
 {
  InputStream is ;
     FileConnection fc =
(FileConnection)Connector.open("file:///root1/java.txt");
  //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
  is = fc.openInputStream();
  is.reset();
 }
```

Erreur lors d`une ouverture d`un fichier



Problème

L`émulateur ne peut pas accéder au fichier

Solution

- Il faut créer ce fichier dans le bon chemin pour exécuter l`application, car l`émulateur essaye d'ouvrir un fichier qui est réellement n`existe pas.
- On change l`émulateur si le problème persiste , car certain émulateur ne supporte pas cette stratégie.

(l` émulateur Sprint peut être une solution).

Création de fichier Log :

Il faut créer un dossier qui s`appelle “root1”(certains émulateur possède ce dossier par défaut) ensuite créer votre fichier .
'C:\WTK22\appdb\DefaultColorPhone\filesystem\root1\java.txt'

Nous pouvons créer le fichier avec un code :

```
try {
FileConnection fc = (FileConnection)Connector.open("file:///root1/java.txt");
// Create the file if it doesn't exist.
if(!fc .exists())
{fc.create(); }
```

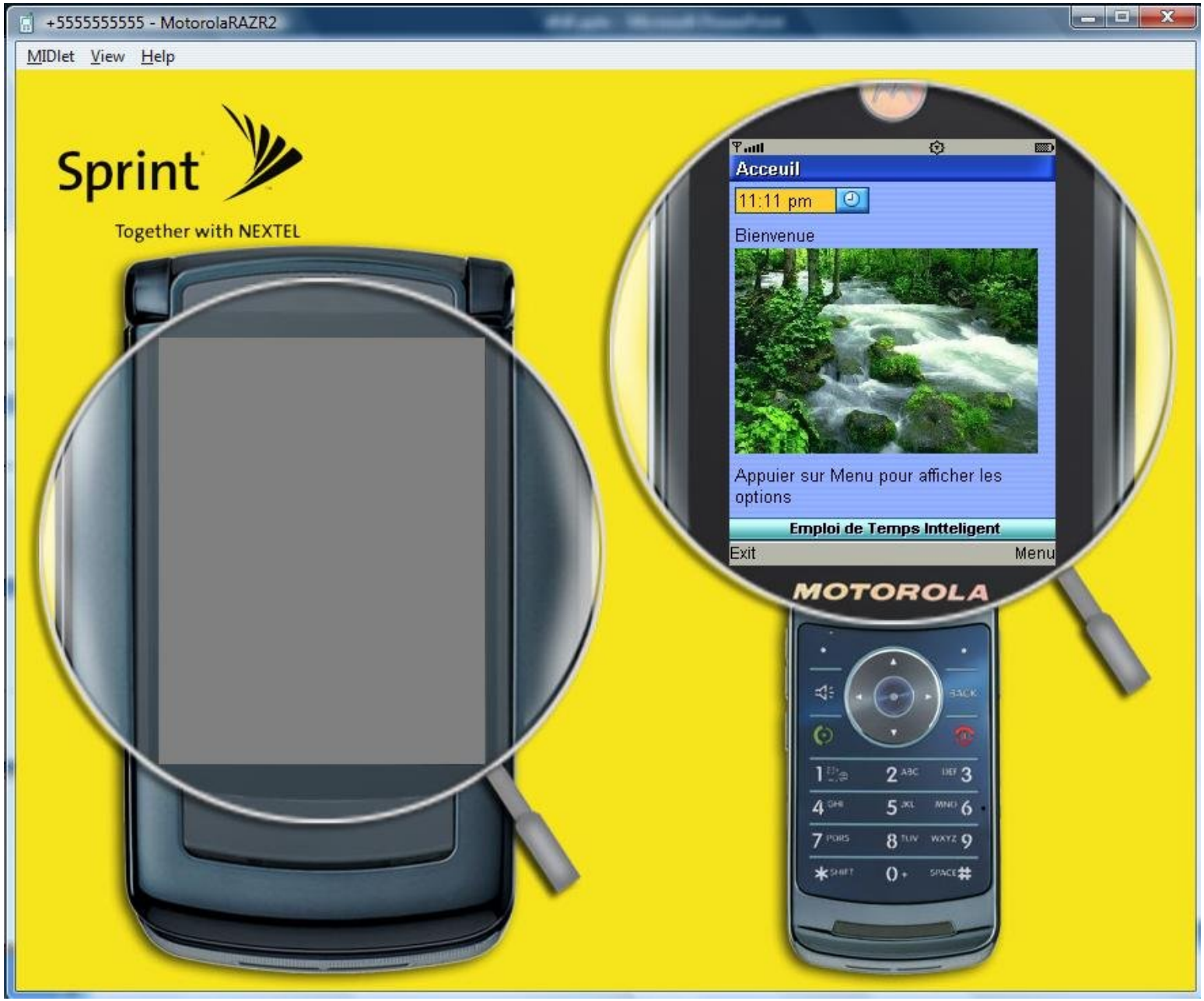
Classe LOG:

```
import java.io.DataOutputStream;
import java.io.IOException;
import java.io.InputStream;
import java.io.OutputStream;
import java.io.PrintStream;
import javax.microedition.io.Connector;
/**
 *
 * @author Zied Chebil
 */
public class LOG {

    public void ecrire(String s) throws IOException
 {
  OutputStream out;
  PrintStream print;
      FileConnection fc = (FileConnection)Connector.open("file:///root1/java.txt");
  //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
  out = fc.openOutputStream(fc.fileSize());
  //On ouvre un flux PrintStream depuis le flux de sortie
  //pour faciliter l'écriture ds le fichier
  print = new PrintStream(out);
  //On écrit la chaine
  print.println(s);
```

```
//Fermeture des flux
 print.close();
 out.close();
}
public  String readText() throws IOException
 {
   InputStream is ;
    FileConnection fc = (FileConnection)Connector.open("file:///root1/java.txt");
  //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
  is = fc.openInputStream();

         // getClass().getResourceAsStream("file:///root1/java.txt");
   String text ="" ;
   try
   {
     StringBuffer sb = new StringBuffer();
     int chr, i = 0;
      // Read until the end of the stream
     while ((chr = is.read()) != -1)
     {
        // sb.append((char) chr);
      text =text+((char) chr)  ;

     }
```

```
System.out.println(text);
      return text;
    }
    catch (Exception e)
    {
      System.out.println("Unable to create stream");
    }
    return null;
  }
public void effacer_log_fichier() throws IOException
 {
  InputStream is ;
     FileConnection fc =
(FileConnection)Connector.open("file:///root1/java.txt");
  //On ouvre un flux de sortie sur le fichier, en se positionnant à la fin
  is = fc.openInputStream();
  is.reset();
 }

}
```

Interface D`accueil du projet : Emploi de Temps intelligent qui utilise le Log file.



Merci pour votre Attention