Mobile Media Player

Prepared By:

Mahmoud abd El-Sabour Ahmed

Sec : 2

Project Description

- It is a mobile application that plays audio and video files.
- User select file from list to be played.

Why Mobile application

- Mobility has become an essential element for the success of any business
- Providing easy access to information form anywhere and at anytime

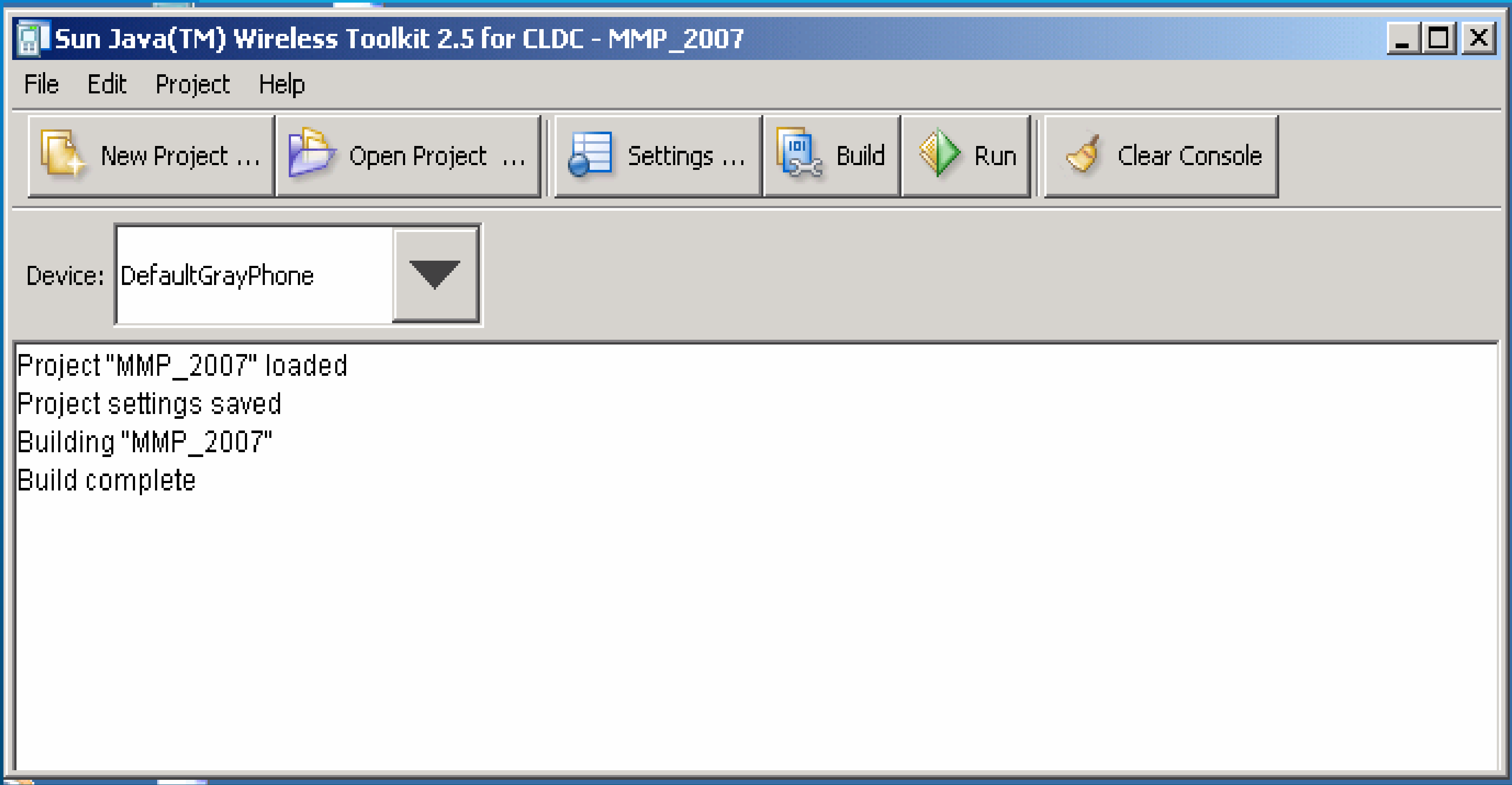
Tools

- **J2ME**
  - Java 2 platform Micro Edition
- Sun Java (TM) Wireless Toolkit 2.5 for CLDC.

CLDC

- Connected Limited Device Configuration.
- It is the configuration that encompass mobile phones and other devices of similar size.

Sun Java (TM) Wireless Toolkit



```
Project "MMP_2007" loaded
Project settings saved
Building "MMP_2007"
Build complete
```

MIDP

- Mobile Information Device Profile
- Mobile Information Device Characteristics

  -128 of non-volatile memory

  -32KB of volatile memory

  -A screen of at least 96x54 pixels

MIDLets

- **It is MIDP application.**
- Its name come from the continuation of the naming **applets** and **servlets**.
- MIDLets are developed on regular desktop computers.

MIDLets Structure

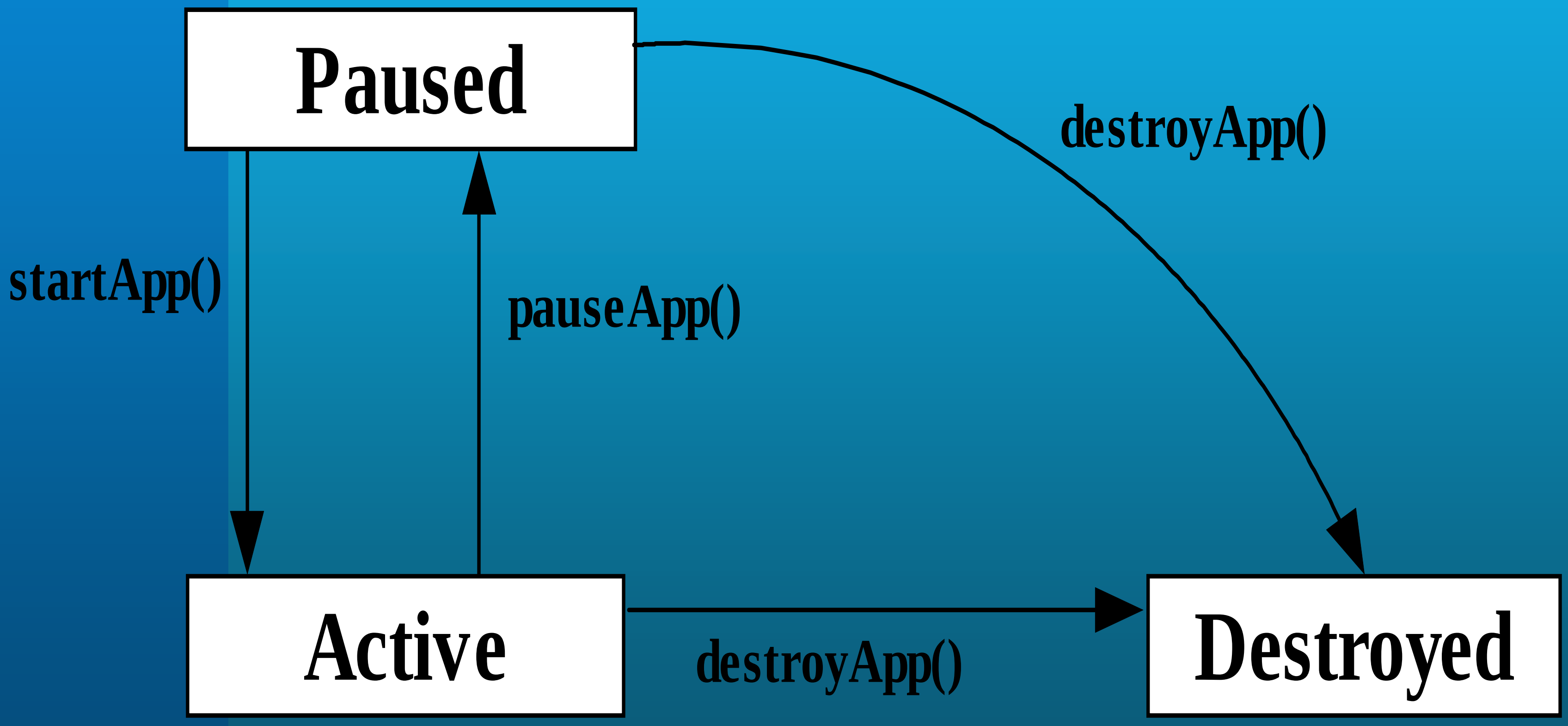
- **MIDLets must goes through these states**
  - **startApp() .**

    -MIDlet enter the Active state
  - **notifyPaused().**

    -put MIDLet back in Paused state
  - **notifyDestroyed().**

    -terminate MIDLet execution

MIDLets Structure



Source Code

- **import java.util.Hashtable;**

  -To use hashtable data structure.

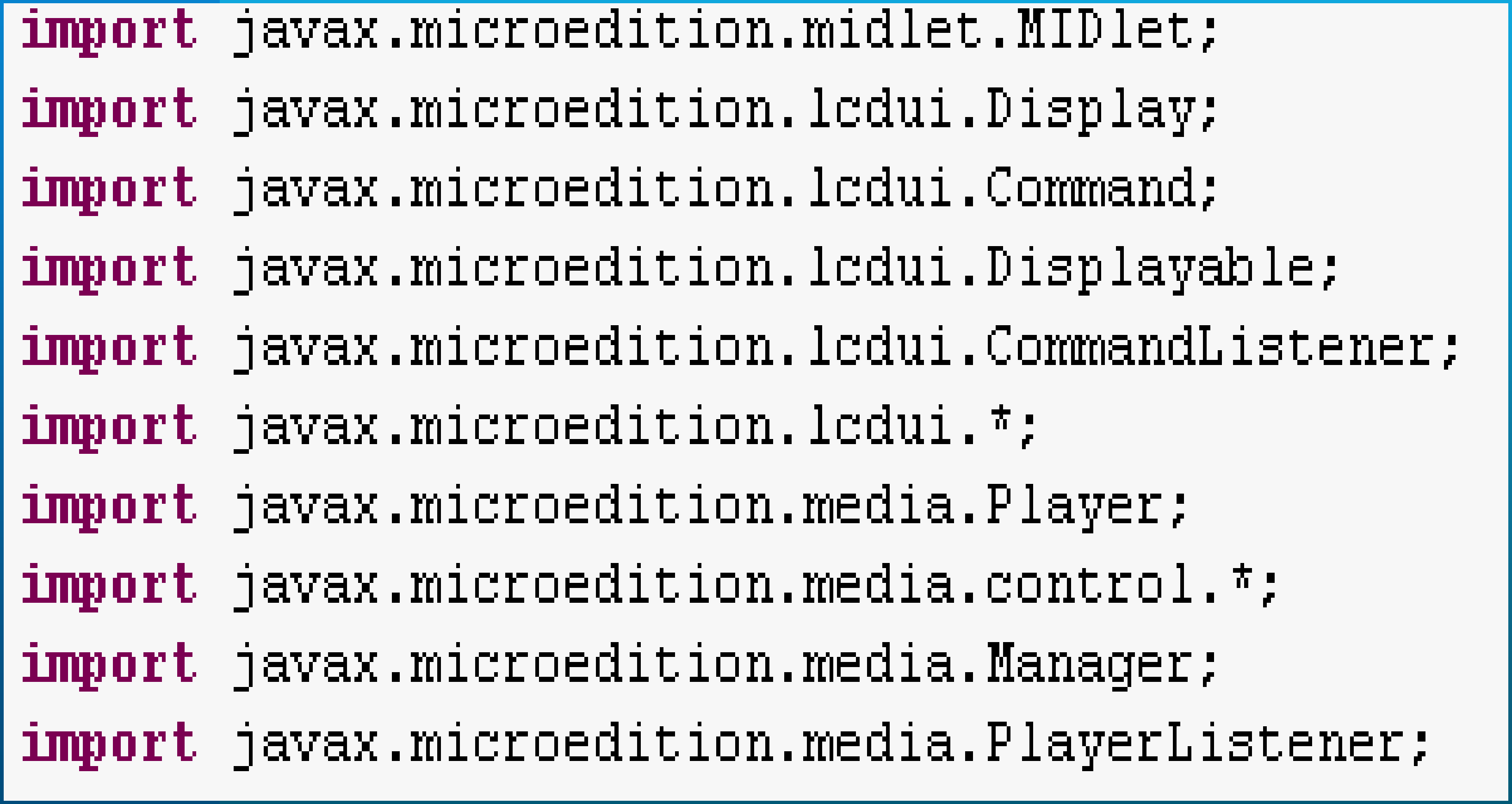
- **import java.util.Enumeration;**

  -to iterate through the elements of a container

- **import java.io.*;**

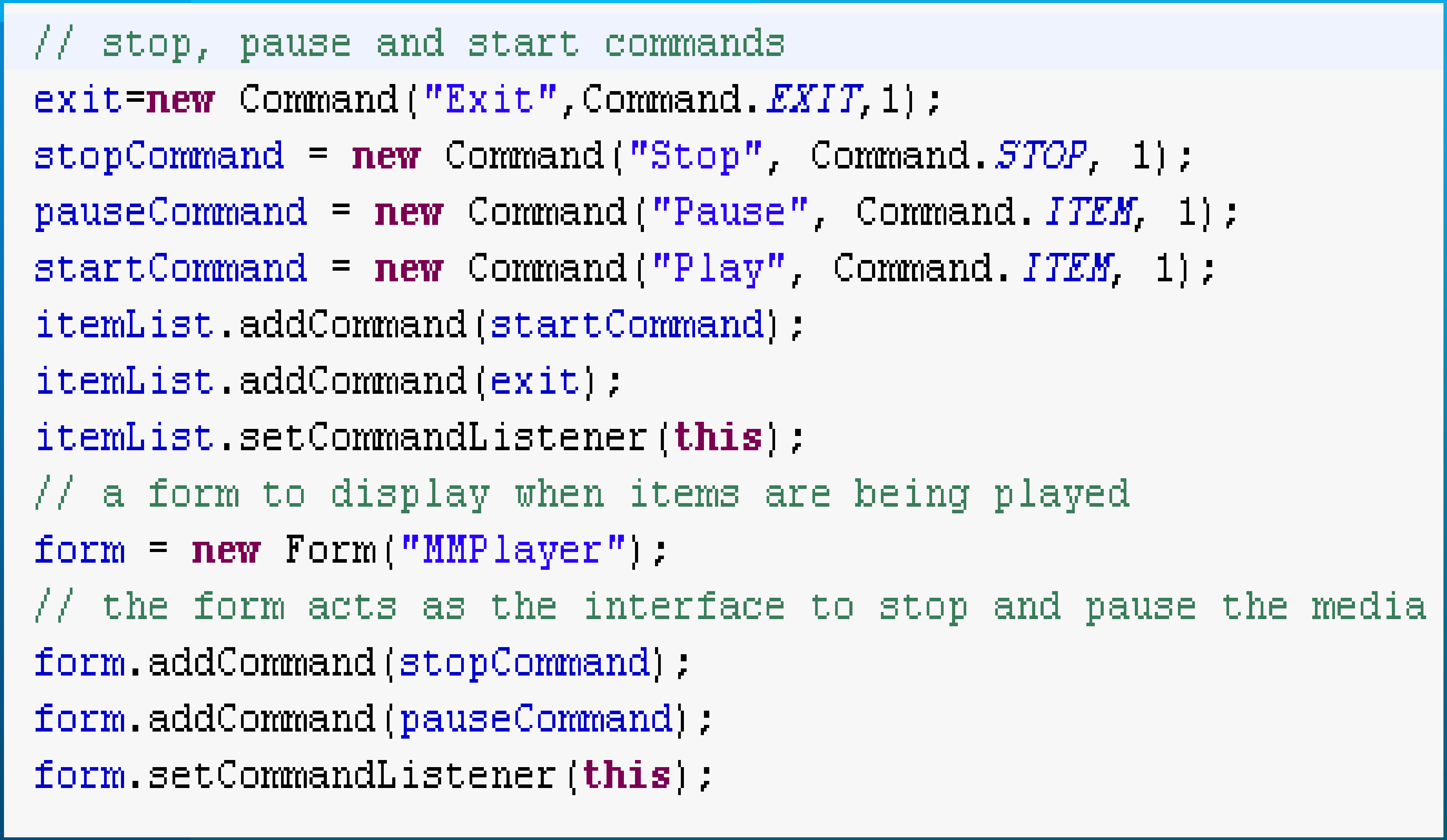
  -to load files.

Source Code Cont.



```
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.Display;
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.Displayable;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.*;
import javax.microedition.media.Player;
import javax.microedition.media.control.*;
import javax.microedition.media.Manager;
import javax.microedition.media.PlayerListener;
```

Source Code Cont.

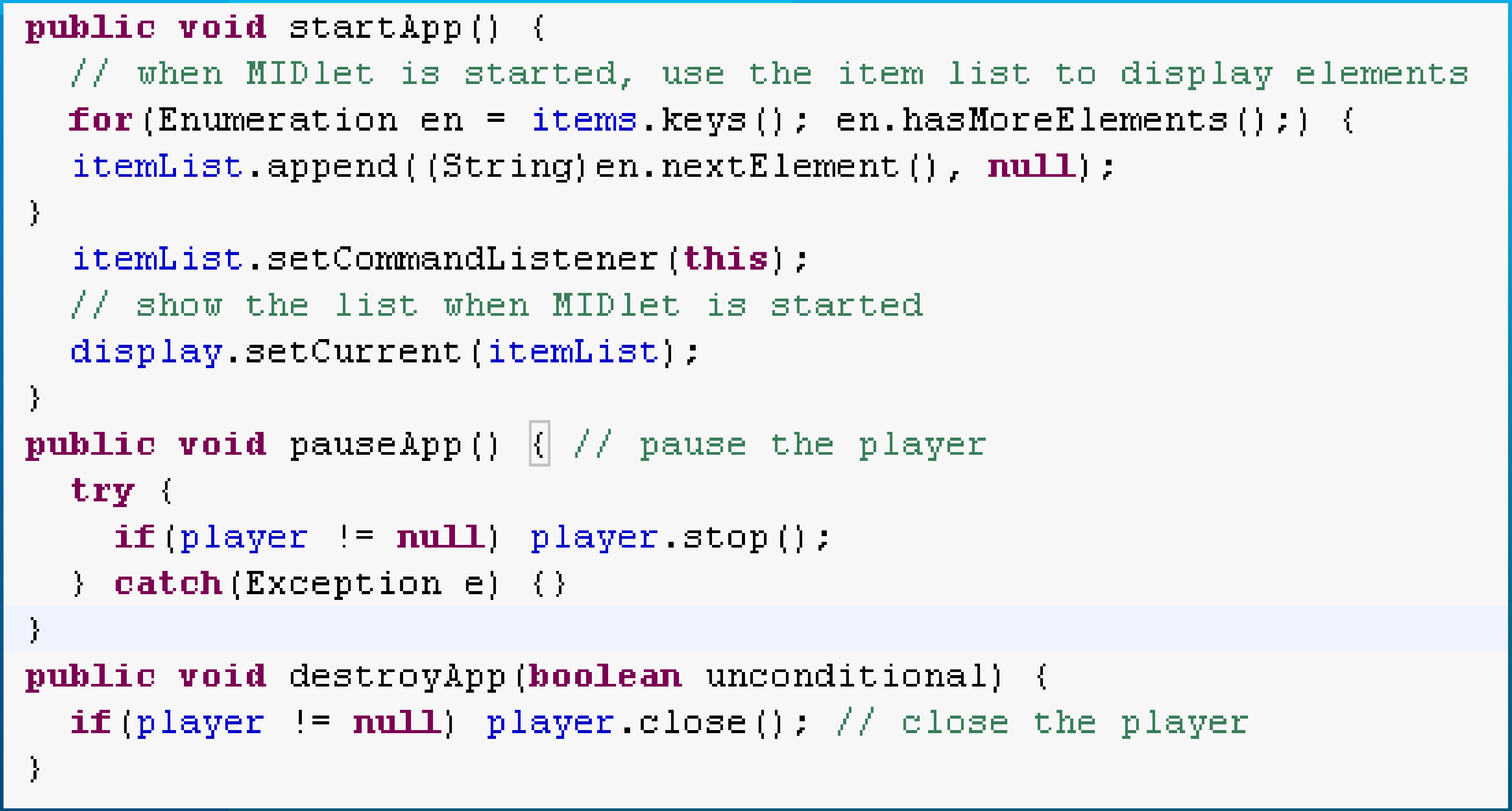


```
// stop, pause and start commands
exit=new Command("Exit",Command.EXIT,1);
stopCommand = new Command("Stop", Command.STOP, 1);
pauseCommand = new Command("Pause", Command.ITEM, 1);
startCommand = new Command("Play", Command.ITEM, 1);
itemList.addCommand(startCommand);
itemList.addCommand(exit);
itemList.setCommandListener(this);
// a form to display when items are being played
form = new Form("MMPlayer");
// the form acts as the interface to stop and pause the media
form.addCommand(stopCommand);
form.addCommand(pauseCommand);
form.setCommandListener(this);
```

Source Code Cont.

```
// create a hashtable of items
items = new Hashtable();
// and a hashtable to hold information about them
itemsInfo = new Hashtable();
// and populate both of them
items.put("Doaa", "file://doaa.wav");
itemsInfo.put("Doaa", "audio/x-wav");
 items.put("salah", "file://salah.wav");
itemsInfo.put("salah", "audio/x-wav");
items.put("Music", "file://Music.wav");
itemsInfo.put("Music", "audio/x-wav");
items.put("Song", "file://Song04.wav");
itemsInfo.put("Song", "audio/x-wav");
items.put("Video Promo", "file://promo.mpg");
itemsInfo.put("Video Promo", "video/mpeg");
items.put("Video sha3rawe", "file://2-11.mpg");
itemsInfo.put("Video sha3rawe", "video/mpeg");
```

Source Code Cont.



```
public void startApp() {
  // when MIDlet is started, use the item list to display elements
  for(Enumeration en = items.keys(); en.hasMoreElements();) {
  itemList.append((String)en.nextElement(), null);
}
  itemList.setCommandListener(this);
  // show the list when MIDlet is started
  display.setCurrent(itemList);
}
public void pauseApp() { // pause the player
  try {
    if(player != null) player.stop();
  } catch(Exception e) {}
}
public void destroyApp(boolean unconditional) {
  if(player != null) player.close(); // close the player
}
```

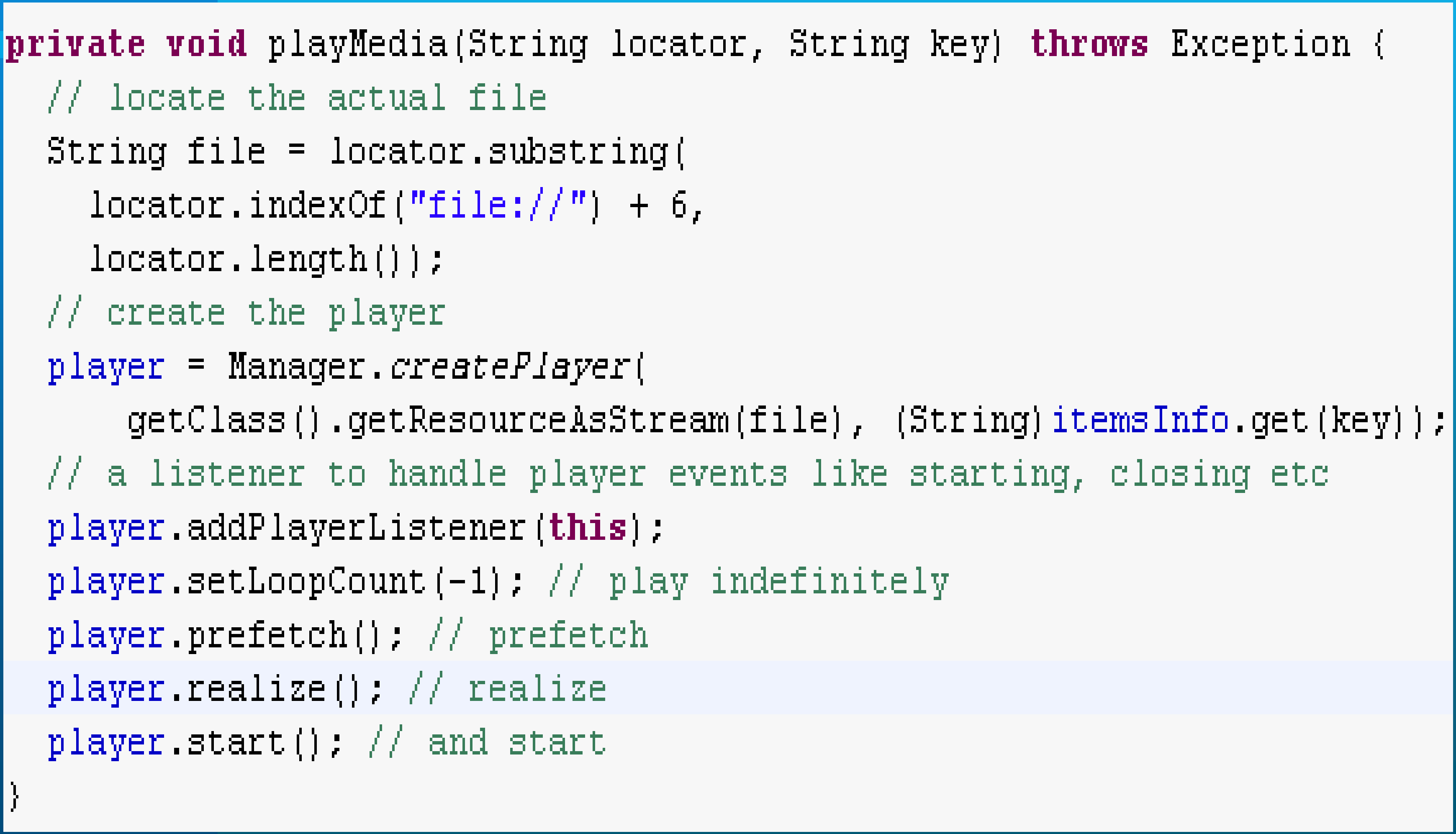
Source Code Cont.

```
public void commandAction(Command command, Displayable disp) {
     if(command==exit)
     {
          if(player!=null)
              player.close();
          notifyDestroyed();
          player.close();
          System.exit(0);
      }
  // if list is displayed, the user wants to play the item
  if(disp instanceof List) {
    List list = ((List)disp);
    String key = list.getString(list.getSelectedIndex());
    // try and play the selected file
    try {
      playMedia((String)items.get(key), key);
    } catch (Exception e) {
      System.err.println("Unable to play: " + e);
      e.printStackTrace();
    }
  }
```

Source Code Cont.

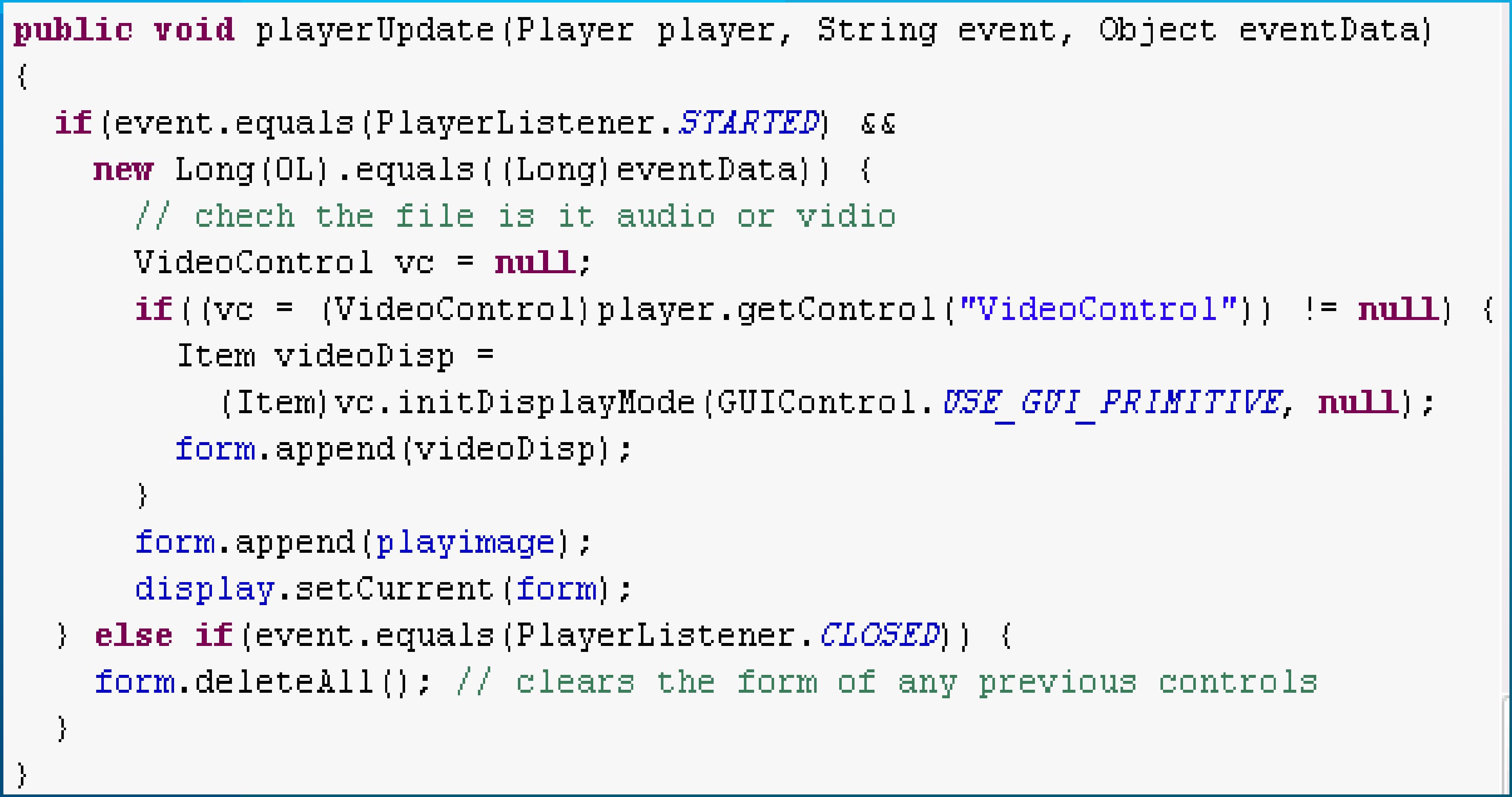
```
else if(disp instanceof Form) {
     form.append(gcontrol);
     vol=gcontrol.getValue();
     volcontrol.setLevel(vol);
  // if showing form, means the media is being played
  // and the user is trying to stop or pause the player
  try {
    if(command == stopCommand) { // if stopping the media play
      player.close(); // close the player
      display.setCurrent(itemList); // redisplay the list of media
      form.removeCommand(startCommand); // remove the start command
      form.addCommand(pauseCommand); // add the pause command
    } else if(command == pauseCommand) { // if pausing
      player.stop(); // pauses the media, note that it is called stop
      form.removeCommand(pauseCommand); // remove the pause command
      form.addCommand(startCommand); // add the start (restart) command
    } else if(command == startCommand) { // if restarting
      player.start(); // starts from where the last pause was called
      form.removeCommand(startCommand);
      form.addCommand(pauseCommand);
    }
  } catch(Exception e) {System.err.println(e);}
}
```

Source Code Cont.



```
private void playMedia(String locator, String key) throws Exception {
  // locate the actual file
  String file = locator.substring(
    locator.indexOf("file://") + 6,
    locator.length());
  // create the player
  player = Manager.createPlayer(
      getClass().getResourceAsStream(file), (String)itemsInfo.get(key));
  // a listener to handle player events like starting, closing etc
  player.addPlayerListener(this);
  player.setLoopCount(-1); // play indefinitely
  player.prefetch(); // prefetch
  player.realize(); // realize
  player.start(); // and start
}
```

Source Code Cont.

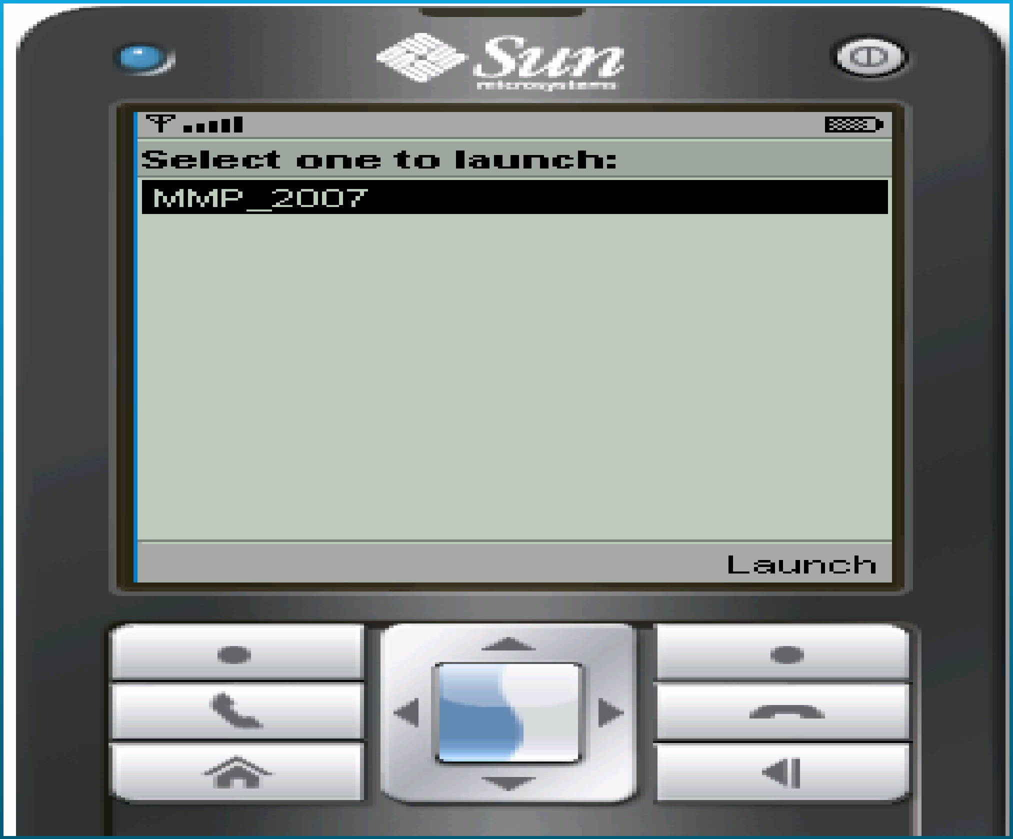


```
public void playerUpdate(Player player, String event, Object eventData)
{
  if(event.equals(PlayerListener.STARTED) &&
    new Long(0L).equals((Long)eventData)) {
      // chech the file is it audio or vidio
      VideoControl vc = null;
      if((vc = (VideoControl)player.getControl("VideoControl")) != null) {
        Item videoDisp =
          (Item)vc.initDisplayMode(GUIControl.USE_GUI_PRIMITIVE, null);
        form.append(videoDisp);
      }
      form.append(playimage);
      display.setCurrent(form);
  } else if(event.equals(PlayerListener.CLOSED)) {
    form.deleteAll(); // clears the form of any previous controls
  }
}
```

Source Code Cont.

```
private Image loadimage(String name)
{
        Image image=null;
        try
        {
                image=Image.createImage(name);
        }
        catch(IOException ioe)
        {
                System.out.println(ioe);
        }
        return image;
}
```

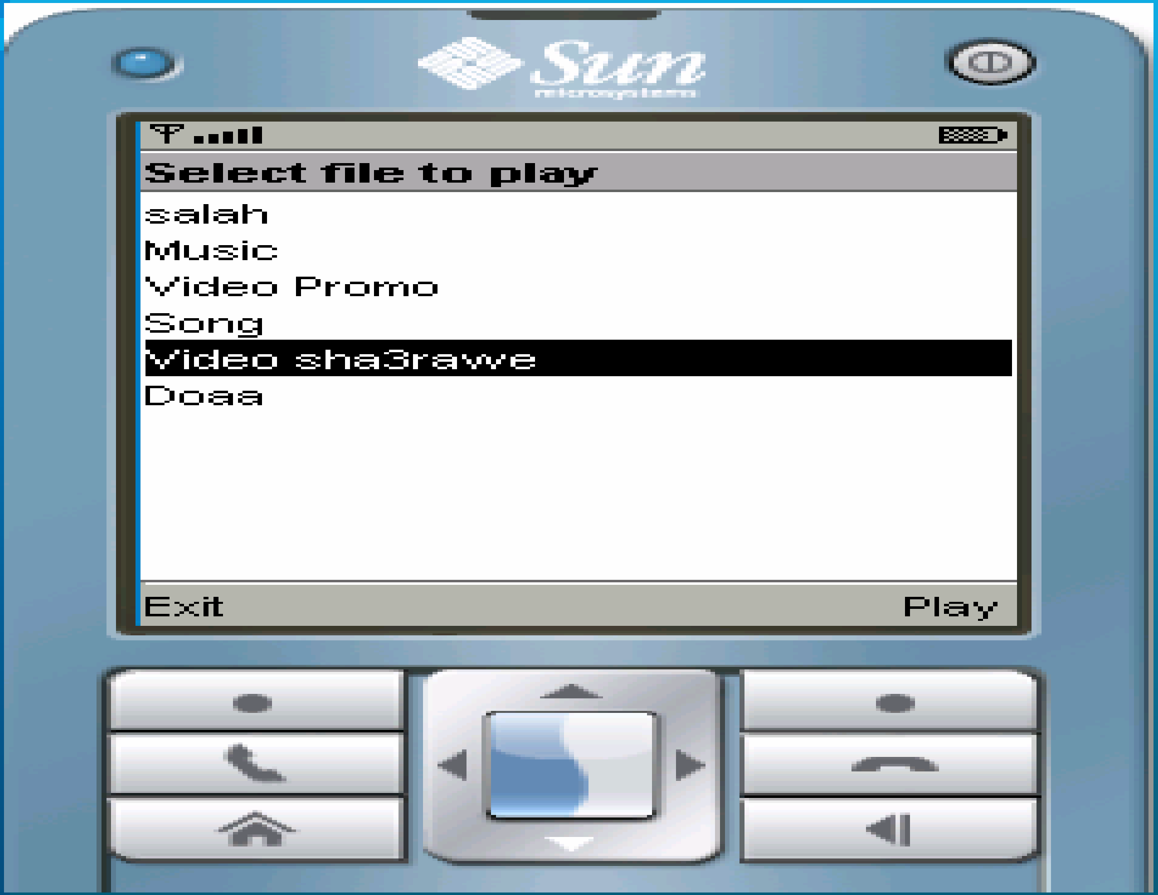
Running the application



Select one to launch:

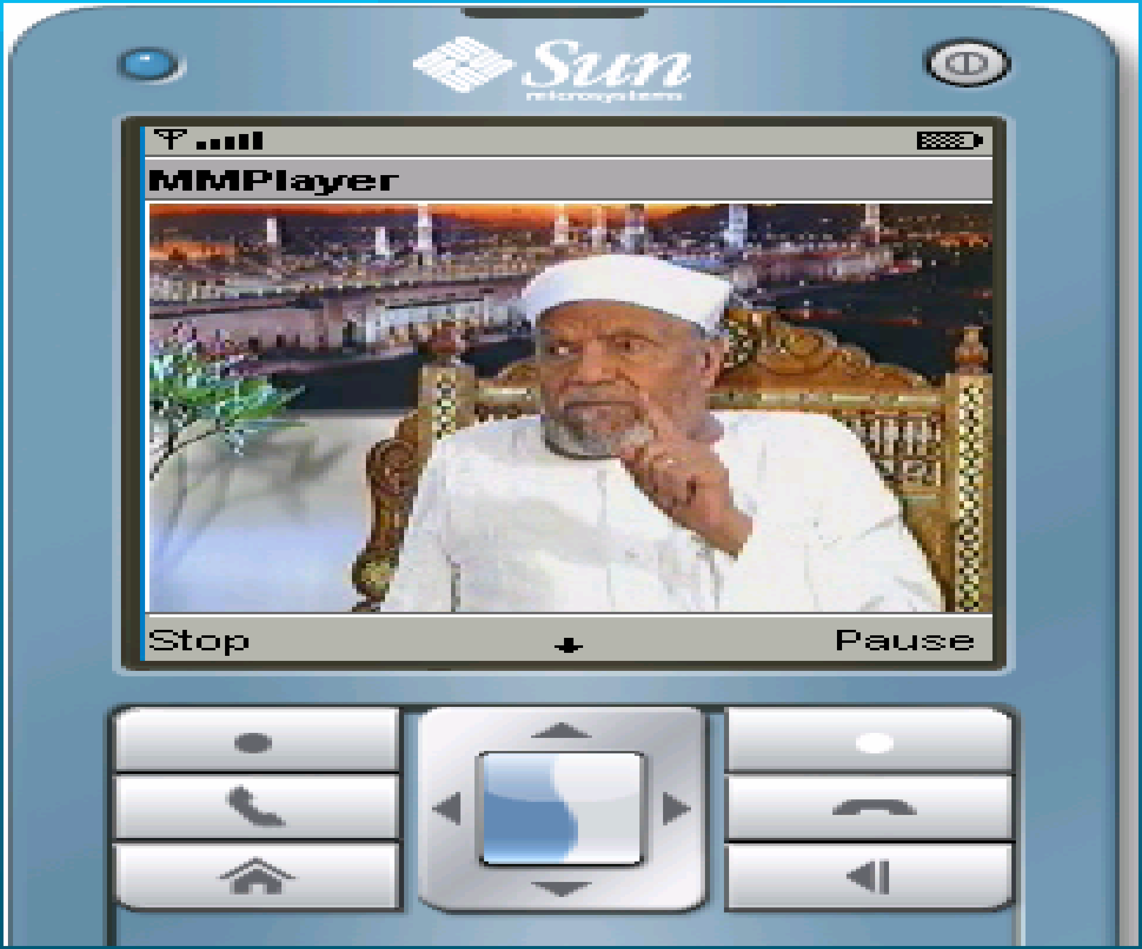
MMP_2007

Running the application


Select file to play
salah
Music
Video Promo
Song
Video sha3rawe
Doaa
Exit
Play

Running the application

Playing video file



Acknowledgment

- Thanks for eng\ Majd and eng\ Noha for their support and help.
- Thanks for every one who help me either by support or by directing me.

Resources

- **Java 2 Platform Micro Edition 2nd edition.**
- **www.java.net**

  J2ME Tutorial, Part 4: Multimedia and MIDP 2.0 by Vikram Goyal

Thanks