CS 683 Emerging Technologies
Fall Semester, 2004
Doc 19 MIDlet UI
Contents

References

J2ME in a Nutshell, Kim Topley, O'Reilly, 2002, Chapter 3 & 4

Examples in this lecture are based on examples in the above reference

Input From Files in Jar

Embedding UI text strings in code is a poor idea

Put them in files

Add the files to the application jar file

Files
- src/HelloMIDlet.java
- res/Text.txt

res/Text.txt

Hello World from a file

src/HelloMIDlet.java

```
import java.io.InputStream;
import java.io.InputStreamReader;
import java.io.IOException;
import javax.microedition.midlet.MIDlet;
import javax.microedition.lcdui.*;

public class HelloMIDlet extends MIDlet {

   public HelloMIDlet() {
      Form form = new Form( "Hi Mom"  );
      form.append( readFile()  );
      Display.getDisplay(this).setCurrent( form );
   }

   private String readFile()  {
      try {
         InputStream byteInput =
               getClass().getResourceAsStream("/Text.txt");
         InputStreamReader characterInput =
               new InputStreamReader(byteInput);
         char[] buffer = new char[32];
         StringBuffer stringBuffer = new StringBuffer();
         int count;
         while ((count = characterInput.read(buffer, 0, buffer.length)) > -1) {
            stringBuffer.append(buffer, 0, count);
         }
         return stringBuffer.toString();
      }
      catch (Exception ioProblem)  {
         return "Could not read file";
      }
   }
```

Source Continued

```
    public void startApp() {}
    public void pauseApp() { }

    public void destroyApp( boolean unconditional ) { }
  }
```

Two Ways to reference the file

Absolute

```
getClass().getResourceAsStream("/foo/bar/Text.txt");
```

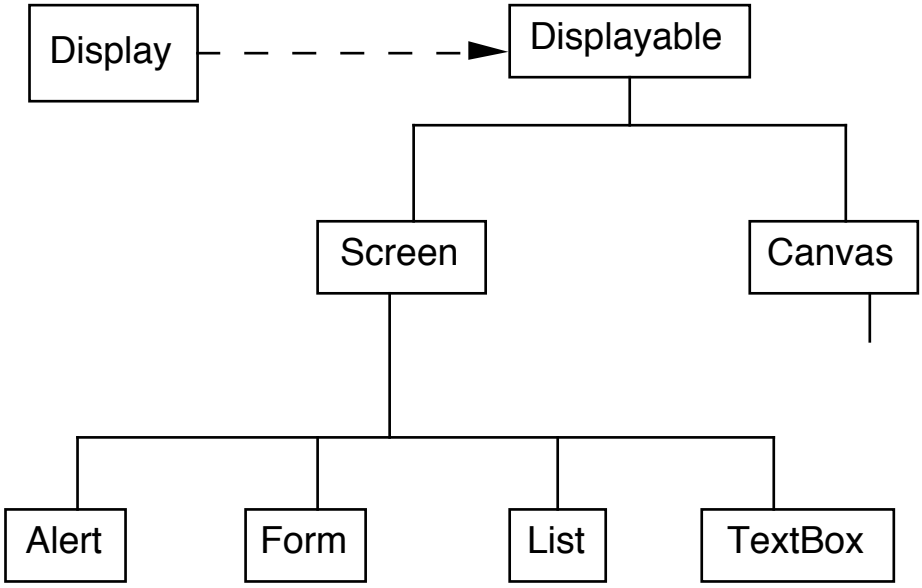
Path is from the top level of the jar file

Relative

```
getClass().getResourceAsStream("/foo/bar/Text.txt");
```

Path is from the location of the location of the class in the jar file

Hi-Level UI Components



TextBox

Textbox Documemtation

http://www.eli.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/TextBox.html

Input Constraints

http://www.eli.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/TextField.html

Basic TextBox

Based on example, pages 89-90, J2ME in a Nutshell, Kim Topley, O'Reilly

```
import java.io.*;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;

public class TextBoxMIDlet extends MIDlet  {
   private static final int MAX_TEXT_SIZE = 80;
   protected TextBox textBox;
   protected Display display;


   public TextBoxMIDlet() {
      textBox = new TextBox("TextBox Example", readFile("/Text.txt"),
                  MAX_TEXT_SIZE, TextField.ANY);

      Ticker ticker = new Ticker("This is a ticker...");
      textBox.setTicker(ticker);

      display = Display.getDisplay(this);
      display.setCurrent(textBox);
   }

   public void startApp() { }
   public void pauseApp() { }
   public void destroyApp( boolean unconditional ) { }
```

Basic TextBox Continued

```
    private String readFile(String filename)  {
        try {
            InputStream byteInput = getClass().getResourceAsStream(filename);
            InputStreamReader characterInput = new
  InputStreamReader(byteInput);
            char[] buffer = new char[32];
            StringBuffer stringBuffer = new StringBuffer();
            int count;
            while ((count = characterInput.read(buffer, 0, buffer.length)) > -1) {
                stringBuffer.append(buffer, 0, count);
            }
            return stringBuffer.toString();
        }
        catch (Exception ioProblem) {
            return "Could not read file";
        }
    }
}
```

TextBox with Timer

Based on examples, pages 73-74, 89-90, J2ME in a Nutshell

```
import java.io.*;
import javax.microedition.lcdui.*;
import javax.microedition.midlet.MIDlet;
import java.util.*;

public class ThreadTextBoxMIDlet extends MIDlet  {
   private static final int MAX_TEXT_SIZE = 80;
   protected TextBox textBox;
   private Timer timer;
   private int count = 0;

   public ThreadTextBoxMIDlet() {
      textBox = new TextBox("Timer Example", "Hello for now",
                     MAX_TEXT_SIZE, TextField.ANY);
      Display.getDisplay(this).setCurrent(textBox);
   }

   public void startApp()  {
      if (timer == null)
         startTimer();
      else
         count = 0;
   }
```

TextBox with Timer Continued

```
    public void pauseApp()  {
       textBox.setString("Here we go again");
    }

    public void destroyApp( boolean unconditional )  {
       stopTimer();
    }

    private void startTimer()  {
       TimerTask addCount = new TimerTask() {

          public void run() {
             textBox.setString( "" + count++);
             if (count == 10)
                notifyPaused();
             else if (count > 15)
                resumeRequest();
          }
       };

       timer = new Timer();
       timer.schedule(addCount, 2000, 1000);
    }

     private void stopTimer()  {
       if (timer != null)
          timer.cancel();
    }
  }
```

TextBox with Command

Based on example, pages 99-100, J2ME in a Nutshell

```
import javax.microedition.lcdui.Command;
import javax.microedition.lcdui.CommandListener;
import javax.microedition.lcdui.Displayable;

public class CommandTextBoxMIDlet extends TextBoxMIDlet
      implements CommandListener {

   private static final Command EXIT_COMMAND =
                  new Command("Exit", Command.EXIT, 0);

   private static final Command OK_COMMAND =
                  new Command("OK", Command.OK, 0);

   private static final Command CLEAR_COMMAND =
                  new Command("Clear", Command.SCREEN, 1);

   private static final Command REVERSE_COMMAND =
                  new Command("Reverse", Command.SCREEN, 1);

   public  CommandTextBoxMIDlet() {
      super();
      textBox.addCommand(OK_COMMAND);
      textBox.addCommand(EXIT_COMMAND);
      textBox.addCommand(CLEAR_COMMAND);
      textBox.addCommand(REVERSE_COMMAND);
      textBox.setCommandListener(this);
   }
```

TextBox with Command Continued

```
    public void commandAction(Command c, Displayable d) {
        if (c == EXIT_COMMAND) {
            destroyApp(true);
            notifyDestroyed();
        } else if (c == OK_COMMAND) {
            textBox.setString("OK pressed");
        } else if (c == CLEAR_COMMAND) {
            textBox.setString(null);
        } else if (c == REVERSE_COMMAND) {
            String str = textBox.getString();
            if (str != null) {
                StringBuffer sb = new StringBuffer(str);
                textBox.setString(sb.reverse().toString());
            }
        }
    }
}
```