

**CS 683 Emerging Technologies
Fall Semester, 2004
Doc 20 MIDlet UI pt 2
Contents**

References	2
Form & Items	3
StringItem	4
TextField.....	5
DateField	7
Gauge.....	9
List.....	11
Implicit List.....	13
ChoiceGroup	15

Copyright ©, All rights reserved. 2004 SDSU & Roger Whitney,
5500 Campanile Drive, San Diego, CA 92182-7700 USA.

OpenContent (<http://www.opencontent.org/opl.shtml>) license
defines the copyright on this document.

References

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

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

Form & Items

The following can be added to a form

ChoiceGroup	CustomItem
DateField	Gauge
ImageItem	Spacer
StringItem	TextField

StringItem

<http://www.elis.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/StringItem.html>

Based on an example from J2ME in a Nutshell

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

public class StringItems extends MIDlet {
    protected Display display;

    public StringItems() {
        Form form = new Form("StringItem");

        form.append(new StringItem("State ", "OK"));
        form.append(new StringItem(null, "No label\n"));
        form.append(new StringItem(null, "Line\nbreak"));
        form.append(new StringItem("Label", "Text."));
        form.append(new StringItem("Label2 ", "Text2."));
        form.append("\nImplicit String item");

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

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

TextField

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

Based on an example from J2ME in a Nutshell

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

public class TextFieldItems extends MIDlet implements
ItemStateListener {
    protected Display display;

    public TextFieldItems() {
        Form form = new Form("TextField");
        form.append("Some text fields");
        form.append(
            new TextField("Any", "Hi mom", 8, TextField.ANY));
        form.append(
            new TextField("Phone", "1234567890", 10,
                TextField.PHONENUMBER));
        form.append(
            new TextField("Number", "12345", 8,
                TextField.NUMERIC));
        form.append(
            new TextField("Password", null, 8, TextField.PASSWORD |
                TextField.NUMERIC));
        form.setItemStateListener(this);
        display = Display.getDisplay(this);
        display.setCurrent(form);
    }
}
```

TextField Continued

```
public void itemStateChanged(Item item) {  
    String textEntered =((TextField)item).getString();  
    Alert displayNewText = new Alert("Changes",textEntered, null,  
    AlertType.INFO);  
    displayNewText.setTimeout(2000);  
    display.setCurrent(displayNewText);  
}  
  
public void startApp() { }  
public void pauseApp() { }  
public void destroyApp( boolean unconditional ) { }  
}
```

DateField

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

Based on an example from J2ME in a Nutshell

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

public class DateFieldItems extends MIDlet implements ItemStateListener
{
    protected Display display;

    public void itemStateChanged(Item item)
    {
        Date newDate =((TextField)item).getDate();
        Calendar accessibleDate = Calendar.getInstance();
        accessibleDate.setTime( newDate );
        Alert displayNewText =
            new Alert("Date","" + accessibleDate, null, AlertType.INFO);
        displayNewText.setTimeout(2000);
        display.setCurrent(displayNewText);
    }

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

DateField

```
public DateFieldItems()
{
    Form form = new Form("Dates");

    DateField dateOnly = new DateField("Date", DateField.DATE);
    DateField both = new DateField("Both", DateField.DATE_TIME);
    Date now = new Date();
    dateOnly.setDate(now);
    both.setDate(now);
    form.append( both);
    form.append( dateOnly);

    //Time only requires date to be Jan 1. 1970, the start of time
    Date beginning = new Date(0);
    Calendar base = Calendar.getInstance();
    base.setTime(beginning);
    Calendar forTimeOnly = Calendar.getInstance();
    forTimeOnly.setTime(now);
    forTimeOnly.set(Calendar.YEAR, base.get(Calendar.YEAR));
    forTimeOnly.set(Calendar.MONTH, base.get(Calendar.MONTH));
    forTimeOnly.set(Calendar.DATE, base.get(Calendar.DATE));
    DateField timeOnly = new DateField("Time", DateField.TIME);
    timeOnly.setDate( base.getTime());
    form.append( timeOnly);

    form.setItemStateListener(this);
    display = Display.getDisplay(this);
    display.setCurrent(form);
}
```

Gauge

<http://www.elis.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/Gauge.html>

Based on an example from J2ME in a Nutshell

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

public class GaugeItems extends MIDlet implements ItemStateListener
{
    protected Display display;

    public GaugeItems()
    {
        Form form = new Form("Gauge");
        String label = null;
        int maxValue = 100;
        boolean editable = true;
        int initialValue = 50;
        form.append(new Gauge(label, editable, maxValue, initialValue));
        form.append(new Gauge(label, editable, maxValue, 25));
        form.append(new Gauge(label, false, maxValue, 10));

        form.setItemStateListener(this);
        display = Display.getDisplay(this);
        display.setCurrent(form);
    }
}
```

Gauge Continued

```
public void itemStateChanged(Item item)
{
    int value =((Gauge)item).getValue();
    Alert displayNewText =
        new Alert("New Value","" + value, null, AlertType.INFO);
    displayNewText.setTimeout(800);
    display.setCurrent(displayNewText);
}

public void startApp() { }

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

}
```

List

<http://www.elis.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/List.html>

Based on an example from J2ME in a Nutshell

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

public class Lists extends MIDlet implements CommandListener
{
    protected Display display;
    protected List list;
    private static final Command EXIT_COMMAND =
        new Command("Exit", Command.EXIT, 0);

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

    public Lists()
    {
        String[] options = { "A", "B", "C" };
        Image[] images = null;
        list = new List("Exclusive List", List.EXCLUSIVE, options, images);
        list.addCommand(OK_COMMAND);
        list.addCommand(EXIT_COMMAND);
        list.setCommandListener(this);

        display = Display.getDisplay(this);
        display.setCurrent(list);
    }
}
```

List Continued

```
public void commandAction(Command c, Displayable d)
{
    if (c == EXIT_COMMAND)
    {
        destroyApp(true);
        notifyDestroyed();
    }
    else if (c == OK_COMMAND)
    {
        int value =list.getSelectedIndex();
        Alert displayNewText =
            new Alert("Selected","Selected index: " + value, null,
AlertType.INFO);
        displayNewText.setTimeout(1000);
        display.setCurrent(displayNewText);
    }
}

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

}
```

Implicit List

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

public class ImplicitLists extends MIDlet implements CommandListener
{
    protected Display display;
    protected List list;
    private static final Command EXIT_COMMAND =
        new Command("Exit", Command.EXIT, 0);

    public ImplicitLists()
    {
        String[] options = { "A", "B", "C" };
        Image[] images = null;
        list = new List("Exclusive List", List.IMPLICIT, options, images);
        list.addCommand(EXIT_COMMAND);
        list.setCommandListener(this);

        display = Display.getDisplay(this);
        display.setCurrent(list);
    }
}
```

```
public void commandAction(Command c, Displayable d)
{
    if(c == EXIT_COMMAND)
    {
        destroyApp(true);
        notifyDestroyed();
    }
    else if(c == List.SELECT_COMMAND)
    {
        int value =list.getSelectedIndex();
        Alert displayNewText =
            new Alert("Selected","Selected index: " + value, null,
AlertType.INFO);
        displayNewText.setTimeout(1000);
        display.setCurrent(displayNewText);
    }
}

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

}
```

ChoiceGroup

<http://www.elis.sdsu.edu/courses/fall04/cs683/j2me/docs/api/midp/javax/microedition/lcdui/ChoiceGroup.html>

Based on an example from J2ME in a Nutshell

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

public class ChoiceItems extends MIDlet implements ItemStateListener
{
    protected Display display;

    public ChoiceItems()
    {
        Form form = new Form("Choice");
        String[] strings = new String[] { "Red", "Green", "Blue" };
        Image[] images = null;
        ChoiceGroup exGroup = new ChoiceGroup("Choose one",
                                              ChoiceGroup.EXCLUSIVE, strings, images);
        form.append(exGroup);

        ChoiceGroup multiGroup =
            new ChoiceGroup("Choose any", ChoiceGroup.MULTIPLE);
        form.append(multiGroup);
        multiGroup.append("A", null);
        multiGroup.append("B", null);
        multiGroup.append("C", null);

        form.setItemStateListener(this);
        display = Display.getDisplay(this);
        display.setCurrent(form);
    }
}
```

ChoiceGroup Continued

```
public void itemStateChanged(Item item)
{
    Choice choice =(Choice)item;
    int count = choice.size();
    boolean[] states = new boolean[count];
    int selCount = choice.getSelectedFlags(states);
    if (selCount > 0)
    {
        String selected = "Selected: ";
        for (int i = 0; i < count; i++)
            if (states[i]) selected += choice.getString(i) + " ";
        Alert displaySelected =
            new Alert("Selected",selected, null, AlertType.INFO);
        displaySelected.setTimeout(800);
        display.setCurrent(displaySelected);
    }

    int selectedIndex = choice.getSelectedIndex();
    Alert displaySelected =
        new Alert("Index ","" + selectedIndex, null, AlertType.INFO);
    displaySelected.setTimeout(1000);
    display.setCurrent(displaySelected);
}

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

Item Examples MANIFEST File

Manifest-Version: 1.0

MIDlet-7: ChoiceItems, , ChoiceItems

MIDlet-6: ImplicitLists, , ImplicitLists

MIDlet-5: Lists, , Lists

MIDlet-4: GaugeItems, , GaugeItems

MIDlet-3: DateFieldItems, , DateFieldItems

MIDlet-2: TextFieldItems, , TextFieldItems

MIDlet-1: StringItems, , StringItems

MIDlet-Name: StringItems

MIDlet-Version: 1.00.00

MIDlet-Vendor: mpowers LLC

MIDlet-Icon: /hello.png

MIDlet-Info-URL: http://mpowers.net/

MicroEdition-Configuration: CLDC-1.0

MicroEdition-Profile: MIDP-1.0

Item Examples JAD File

MIDlet-7: ChoiceItems, , ChoiceItems

MIDlet-6: ImplicitLists, , ImplicitLists

MIDlet-5: Lists, , Lists

MIDlet-4: GaugeItems, , GaugeItems

MIDlet-3: DateFieldItems, , DateFieldItems

MIDlet-2: TextFieldItems, , TextFieldItems

MIDlet-1: StringItems, , StringItems

MIDlet-Name: StringItems

MIDlet-Version: 1.00.00

MIDlet-Vendor: mpowers LLC

MIDlet-Jar-URL: MyProj.jar

MIDlet-Jar-Size: 12939

MIDlet-Description: A simple sample.

MIDlet-Icon: /hello.png

MIDlet-Info-URL: http://mpowers.net/

Title: Item Examples

Message: This is a simple sample midlet