

## Swing-a-majig user guide

### System requirements

Java 2 or later must be installed on the system.

The system has been tested successfully on Windows 98, ME, NT and UNIX platforms.

### Running instructions

To run program type:

```
Java -jar swing.jar
```

at the command prompt, or if Java 1.3 runtime is installed simply double click the swing.jar file.

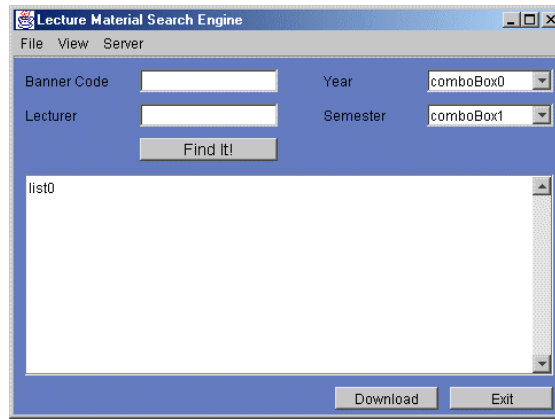
### Swing-a-majig tutorial

This tutorial explains how to create a basic user interface using the application.

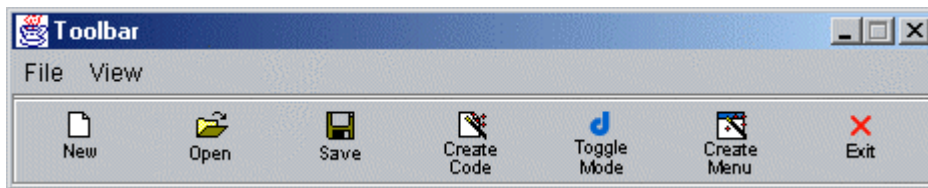
1. Follow the running instructions to load the program. The splash screen shown below should appear while the application is loading.



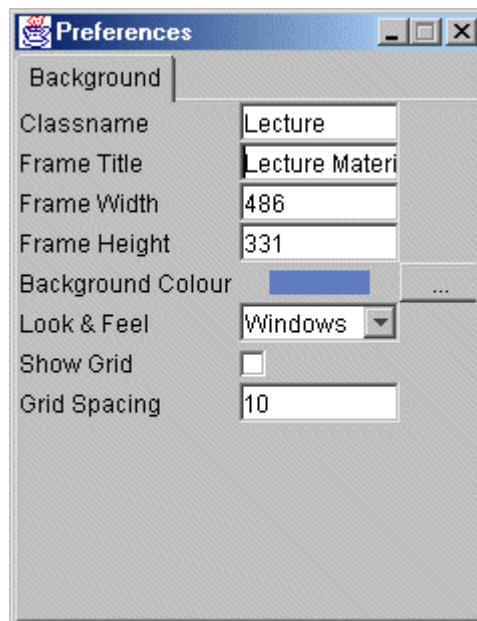
2. Once the application is loaded, five windows are displayed onscreen. For each of their uses refer back to the user interface chapter of this dissertation. The sample interface this tutorial teaches you to create is shown below. A copy of this interface can be found in the applications *examples* folder.



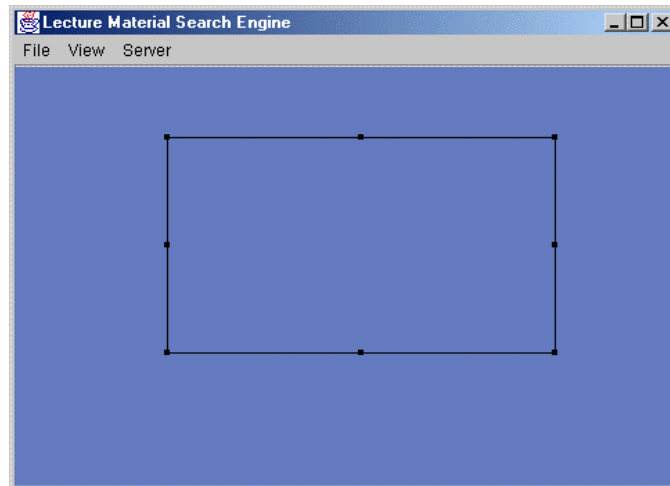
3. The interface combines many standard components as well as a more advanced menu bar structures. To begin creation of the interface, locate the toolbar and click the 'new' button to begin a new project. This will erase any work currently in the workspace.



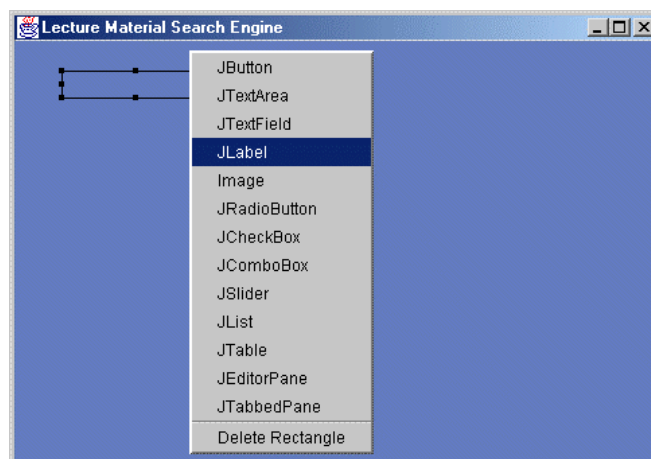
4. Resize the workspace window by dragging on of the corners so that it is around 180 pixels wider and 30 pixels higher than its current height. The size of the frame is shown in the preference window. Here you can simply type the new values in for height and width.
5. All properties of the frame are altered from the preferences window. To change the current title bar name click inside the title bar text box in preferences window and change the value to *Lecture Material Search Engine*. If you wish you can change the frames look and feel and background colour. It is a good idea to set the grid size to one that you are most comfortable working with. A size of ten is adequate for the time being.



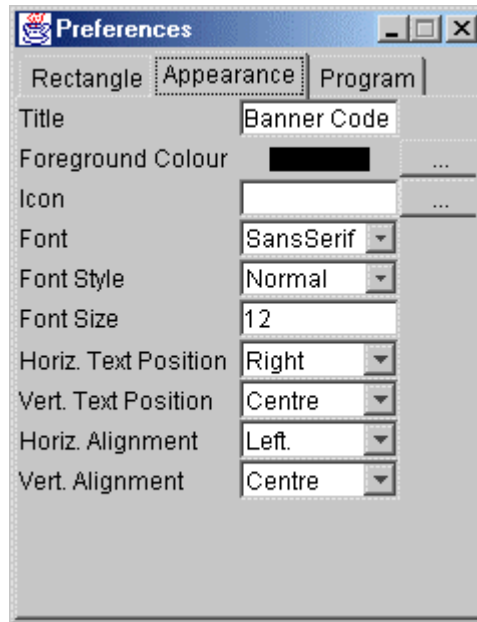
- Components are added to the workspace by first drawing placeholder rectangles then adding components to them. The rectangles serve as an extra way to plan your interface as well as an easy way to manipulate the components held inside them. Draw a rectangle by clicking the left mouse button and dragging the mouse to set its size. Clicking a drawn rectangle the object and allows you to manipulate it.



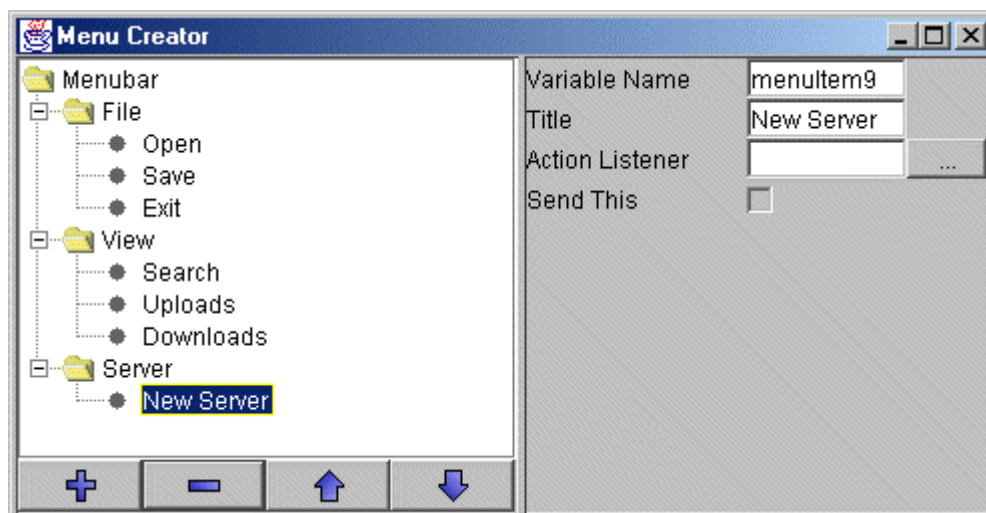
- Make the rectangle smaller by clicking and dragging the handles with the left mouse button. Clicking and dragging any other area of the rectangle moves it. You will notice the cursor change over resize areas and move areas. Left clicking the background deselects the rectangle for drawing other rectangles.
- Right click the rectangle and select a JLabel from the popup menu. This will add a JLabel Swing component to the rectangle.



- You can customise all Swing components added to your rectangle placeholders via the preference window. Select the JLabel component onscreen and alter its title to *Banner Code* in the preference window. Notice that when the JLabel is deselected its rectangle placeholder becomes invisible.



10. Now you have learnt the basics of adding and customising components draw the remainder of components onto the workspace that are shown in the sample GUI image. These components are JList, JComboBox, JLabel and JButton. Play with the preferences as you wish. You can delete unwanted rectangles by right clicking the rectangle and selecting the delete option. Components are removed in the same way.
11. Next a JMenuBar may be added. To do this click the 'menu creator' button on the toolbar. An empty menu creator will be displayed. By adding nodes using the plus button and removing them using the minus, create the menu structure shown below. The up and down arrows move the nodes up and down the branches. You can rename the menu components in the preference window opposite. Notices how the menu in the workspace is updated after each change you make.



12. You can save the project by clicking the 'save' button on the toolbar. Choose a filename and press ok. To create Java code for this interface click the 'create code' button on the toolbar. The class name used in the code generation is the set in the frame's preferences.

The code is compiled and run like any other class. Type:

```
javac classname.java
```

```
java classname
```

The saved file can be reloaded in at any point by clicking the 'load' button on the toolbar and selecting the appropriate file. You can also test out your design's Swing components by clicking the 'toggle mode' toolbar button. You can now click on the menu structure and buttons to get a feel for how the interface works. To get back into the design mode re-click the toggle button. The 'd' letter means you are in design mode, the 'p' means preview mode.

To add JTabbedPanels to the workspace the same process is performed for adding any component. Extra tabbed pane options are displayed when right clicking on the tabbed pane component. These allow you to display the next or previous tab as well as giving options to delete the tab or whole tabbed pane. Extra tabs are added by selecting 'add tab' from the popup menu.

Components are added to the tab by simply dragging them on. The current tab showing is the tab the component will be added to. To add the component to a currently hidden tab simply change the current tab displaying to the one you desire then drag the component on. Components inside the tabbed pane are moved too when the tabbed pane is moved. To remove a component from the tabbed pane just drag it out of the tabbed pane's area. Note, components cannot be drawn directly onto the tabbed pane. They must be drawn outside of it then dragged on.

When the tabbed pane's tab is changed all the components on the old tab are hidden and those on the new one are displayed. The preferences of each tab can be different. Individual tab preferences are altered in the preference window. The preferences shown at any one time are the preferences of the current tab. To alter another tabs preferences you must make that tab the currently visible one.

## Appendix A

### Example sections of the generated save file for the tutorial GUI

```
JFRAME
{
  classname: Lecture
  title: "Lecture Material Search Engine"
  width: 486
  height: 338
  background: 100 120 192
  grid: 5
  view_grid: false
  look_and_feel: Windows
}

JLABEL
{
  x: 10
  y: 33
  width: 90
  height: 20
  tab: -1
  selectable: true
  variable_name: label0
  text: "Banner Code"
  foreground: 0 0 0
  icon: null
  visible: true
  tooltip: "null"
  font_name: SansSerif
  font_size: 12
  font_style: 0
  horizontal_Text_Position: 4
  vertical_Text_Position: 0
  horizontal_Alignment: 10
  vertical_Alignment: 0
  set_method: false
  get_method: false
}

JTEXTFIELD
{
  x: 110
  y: 33
  width: 120
  height: 20
  tab: -1
  selectable: true
  variable_name: textField0
  text: ""
  action: null
  background: 255 255 255
  foreground: 0 0 0
  visible: true
  editable: true
  tooltip: "null"
  font_name: SansSerif
  font_size: 12
  font_style: 0
  set_method: false
  get_method: false
  scrollpane: no
  horizontal_policy: null
  vertical_policy: null
}

JLIST
{
  x: 10
  y: 125
  width: 460
  height: 175
  tab: -1
  selectable: true
  variable_name: list0
  visible: true
  tooltip: "null"
  set_method: false
  get_method: false
  scrollpane: yes
  horizontal_policy: 30
  vertical_policy: 22
}
```

```
JBUTTON
{
  x: 280
  y: 310
  width: 90
  height: 20
  tab: -1
  selectable: true
  variable_name: button3
  text: "Download"
  action: null
  background: 192 192 192
  foreground: 0 0 0
  icon: null
  paint_border: true
  selected: false
  visible: true
  enabled: true
  tooltip: "null"
  font_name: SansSerif
  font_size: 12
  font_style: 0
  horizontal_Text_Position: 0
  vertical_Text_Position: 0
  horizontal_Alignment: 0
  vertical_Alignment: 0
  set_method: false
  get_method: false
}

JCOMBOBOX
{
  x: 360
  y: 33
  width: 110
  height: 20
  tab: -1
  selectable: true
  variable_name: comboBox0
  action: null
  visible: true
  enabled: true
  tooltip: "null"
  font_name: Dialog
  font_size: 12
  font_style: 0
  set_method: false
  get_method: false
}

JMENUBAR
{
  variable_name: menubar

  JMENU
  {
    variable_name: menuItem0
    text: "File"
    children: 3

    JMENUITEM
    {
      variable_name: menuItem1
      text: "Open"
      action: null
      send_this: false
    }

    JMENUITEM
    {
      variable_name: menuItem2
      text: "Save"
      action: null
      send_this: false
    }

    :
    :
    :
  }
}
```

## Appendix B

### Example sections of the Java code generated for the tutorial GUI

```
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import javax.swing.event.*;

class Lecture
{
    Container container;
    JFrame frame;

    public Lecture()
    {
        addJFrame();
        addJMenuBar();
        addJLabel();
        addJTextField();
        addJComboBox();
        addJButton();
        addJList();
        frame.setVisible(true);
    }

    public static void main(String args[])
    {
        try
        {
            UIManager.setLookAndFeel(
                "com.sun.java.swing.plaf.windows.WindowsLookAndFeel");
        }
        catch (Exception e) { }
        Lecture lectureCLASS = new Lecture();
    }

    public void addJFrame()
    {
        frame = new JFrame("Lecture Material Search Engine");
        frame.setSize(486,361);
        container = frame.getContentPane();
        container.setBackground(new Color(100, 120, 192));
        container.setLayout(null);

        frame.addWindowListener(new WindowAdapter()
        {
            public void windowClosing(WindowEvent e)
            {
                System.exit(0);
            }
        });
    }

    public void addJMenuBar()
    {
        JMenuBar menubar = new JMenuBar();

        JMenu menuItem0 = new JMenu("File");
        menubar.add(menuItem0);

        JMenuItem menuItem1 = new JMenuItem("Open");
        menuItem0.add(menuItem1);

                :
                :

        frame.setJMenuBar(menubar);
    }
}
```

```

public void addJLabel()
{
    JLabel label0 = new JLabel();
    label0.setText("Banner Code");
    label0.setBounds(10,10,90,20);
    label0.setForeground(new Color(0, 0, 0));
    Font label0Font = new Font("SansSerif", 0, 12);
    label0.setFont(label0Font);
    label0.setHorizontalTextPosition(4);
    label0.setVerticalTextPosition(0);
    label0.setHorizontalAlignment(10);
    label0.setVerticalAlignment(0);
    container.add(label0);
    :
}

public void addJTextField()
{
    JTextField textField0 = new JTextField();
    textField0.setText("");
    textField0.setBounds(110,10,120,20);
    textField0.setBackground(new Color(255, 255, 255));
    textField0.setForeground(new Color(0, 0, 0));
    Font textField0Font = new Font("SansSerif", 0, 12);
    textField0.setFont(textField0Font);
    container.add(textField0);
    :
}

public void addJComboBox()
{
    JComboBox comboBox0 = new JComboBox();
    comboBox0.setBounds(360,10,110,20);
    Font comboBox0Font = new Font("Dialog", 0, 12);
    comboBox0.setFont(comboBox0Font);
    container.add(comboBox0);

    JComboBox comboBox1 = new JComboBox();
    comboBox1.setBounds(360,40,110,20);
    Font comboBox1Font = new Font("Dialog", 0, 12);
    comboBox1.setFont(comboBox1Font);
    container.add(comboBox1);
}

public void addJButton()
{
    JButton button0 = new JButton();
    button0.setText("Find It!");
    button0.setBounds(110,70,120,20);
    button0.setBackground(new Color(192, 192, 192));
    button0.setForeground(new Color(0, 0, 0));
    button0.setBorderPainted(true);
    button0.setEnabled(true);
    Font button0Font = new Font("SansSerif", 0, 14);
    button0.setFont(button0Font);
    button0.setHorizontalTextPosition(4);
    button0.setVerticalTextPosition(0);
    button0.setHorizontalAlignment(0);
    button0.setVerticalAlignment(0);
    container.add(button0);
    :
}

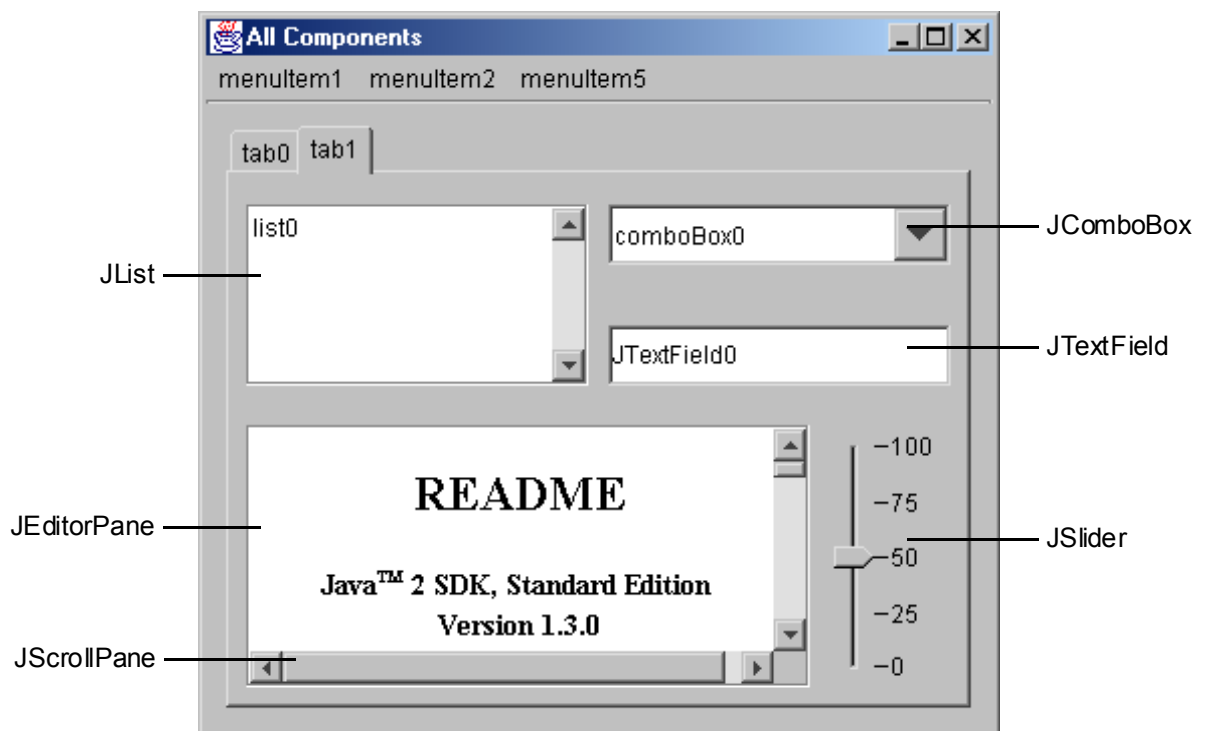
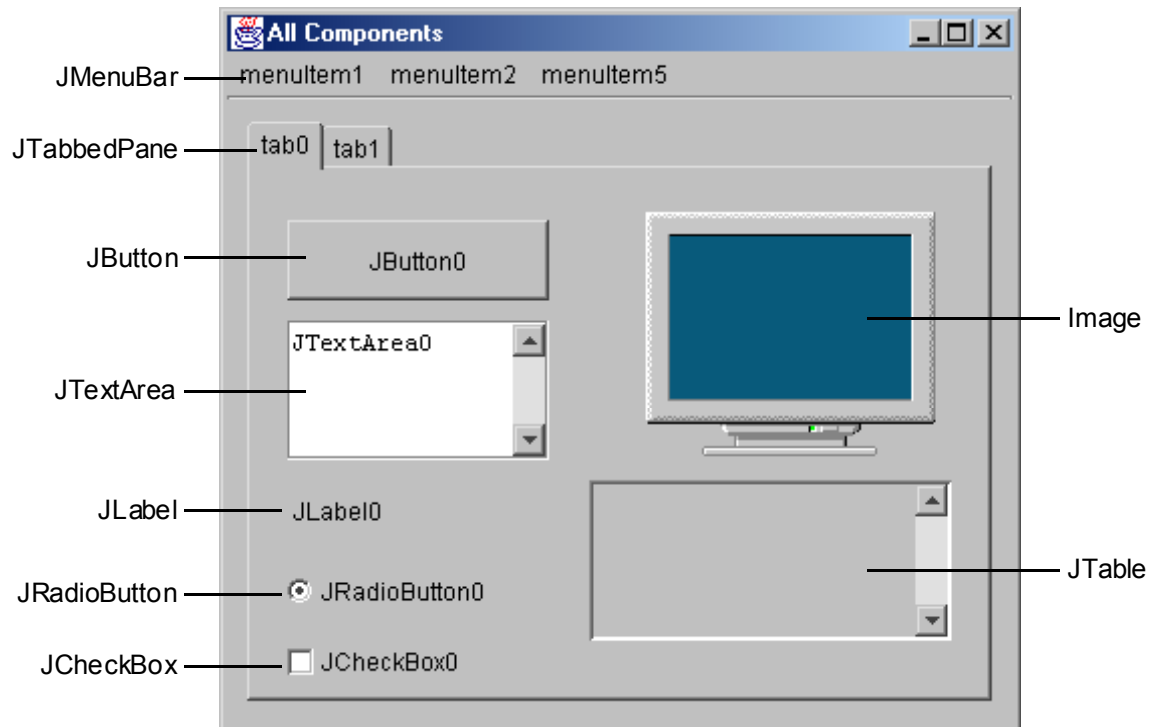
public void addJList()
{
    DefaultListModel list0Model = new DefaultListModel();
    list0Model.addElement("list0Model");
    JList list0 = new JList(list0Model);
    list0.setBounds(10,102,460,175);
    JScrollPane list0ScrollPane = new JScrollPane(list0);
    list0ScrollPane.setBounds(10,102,460,175);
    list0ScrollPane.setVerticalScrollBarPolicy(
        JScrollPane.VERTICAL_SCROLLBAR_ALWAYS);
    container.add(list0ScrollPane);
}
}

```

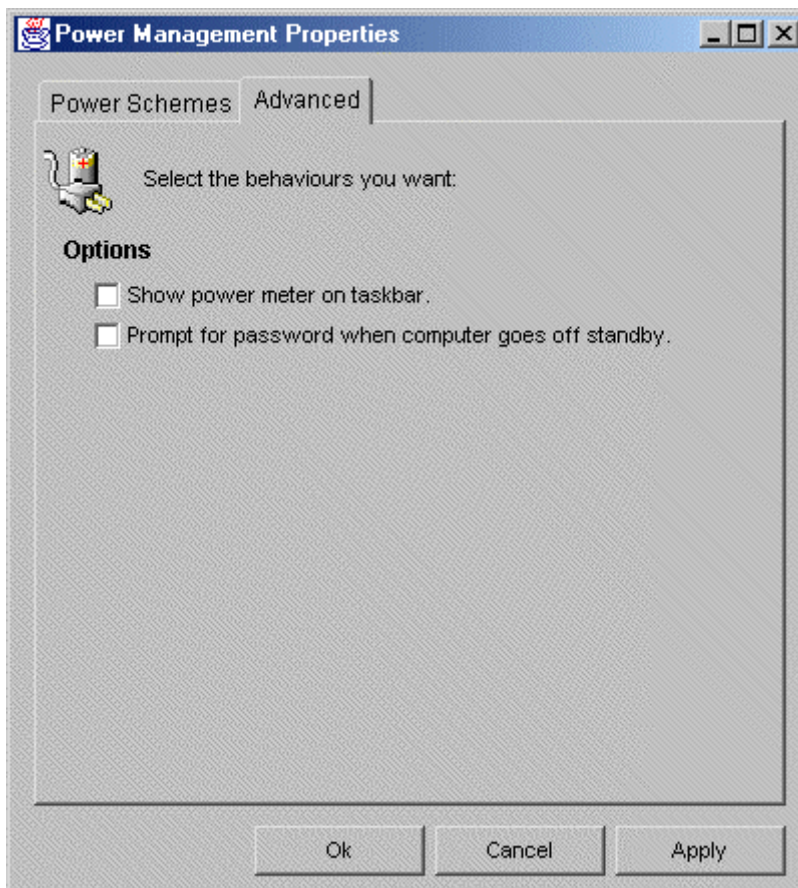
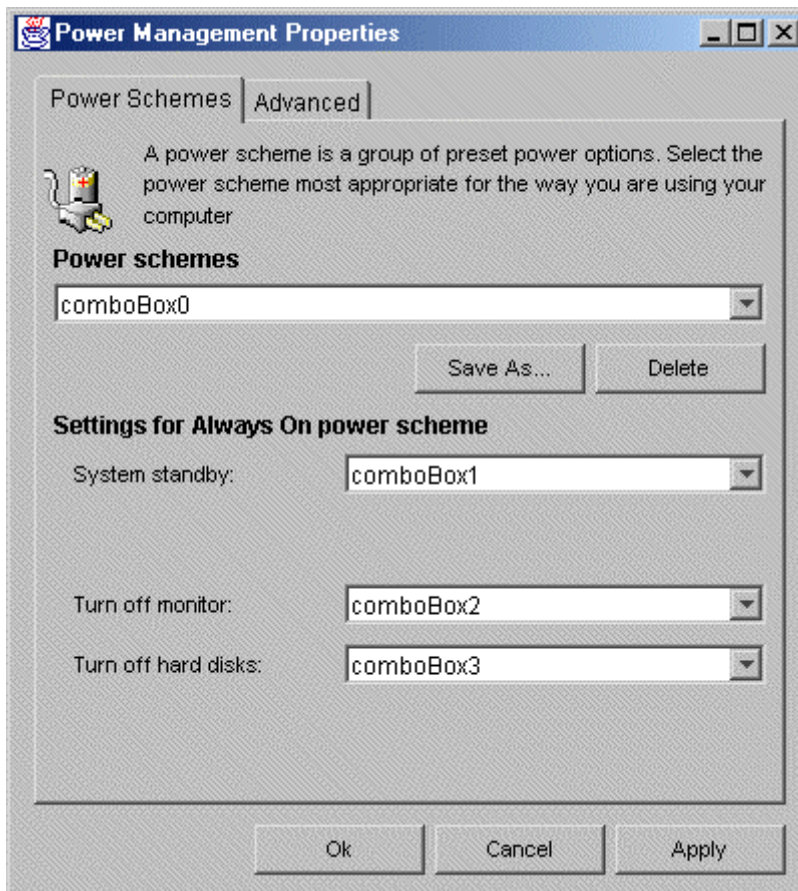
## Appendix C

### Example interfaces created using the developed application

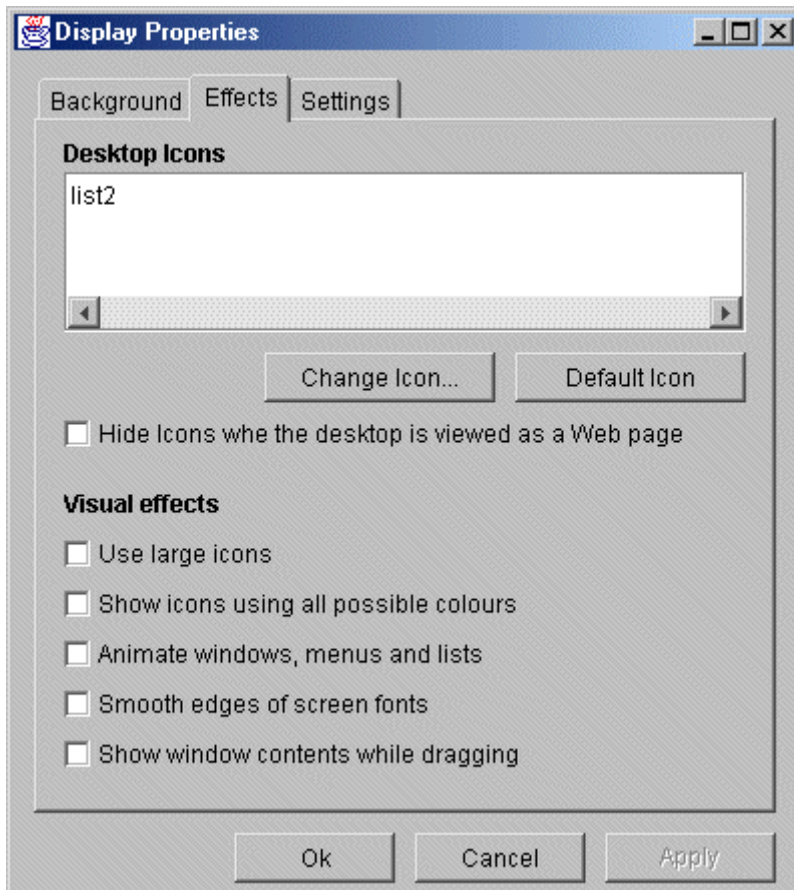
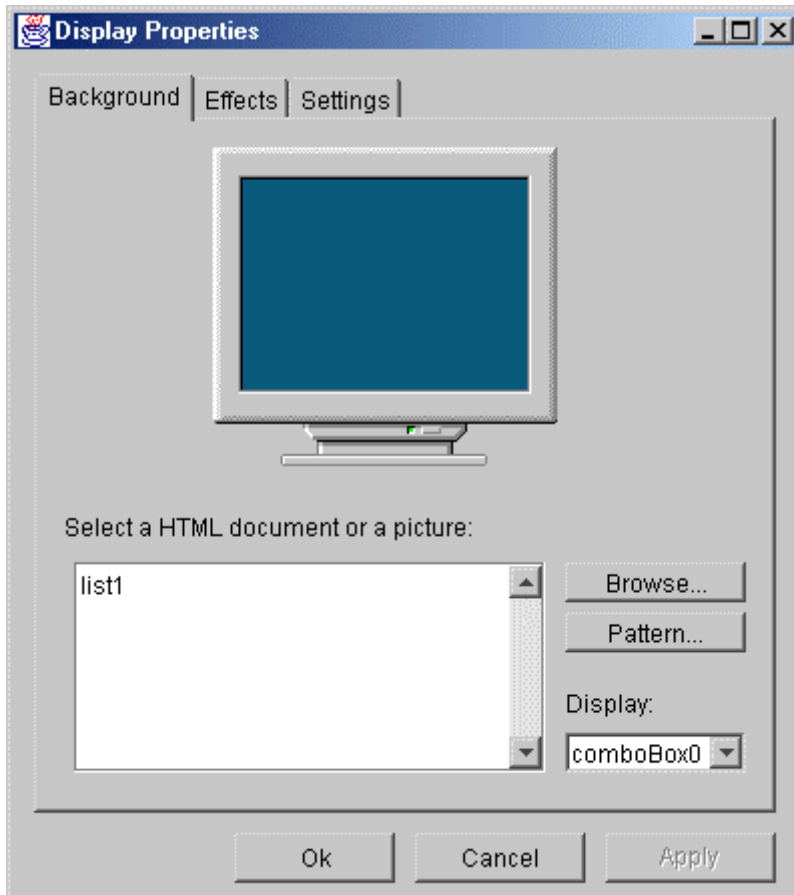
Visual guide to all Swing components available for use in the application



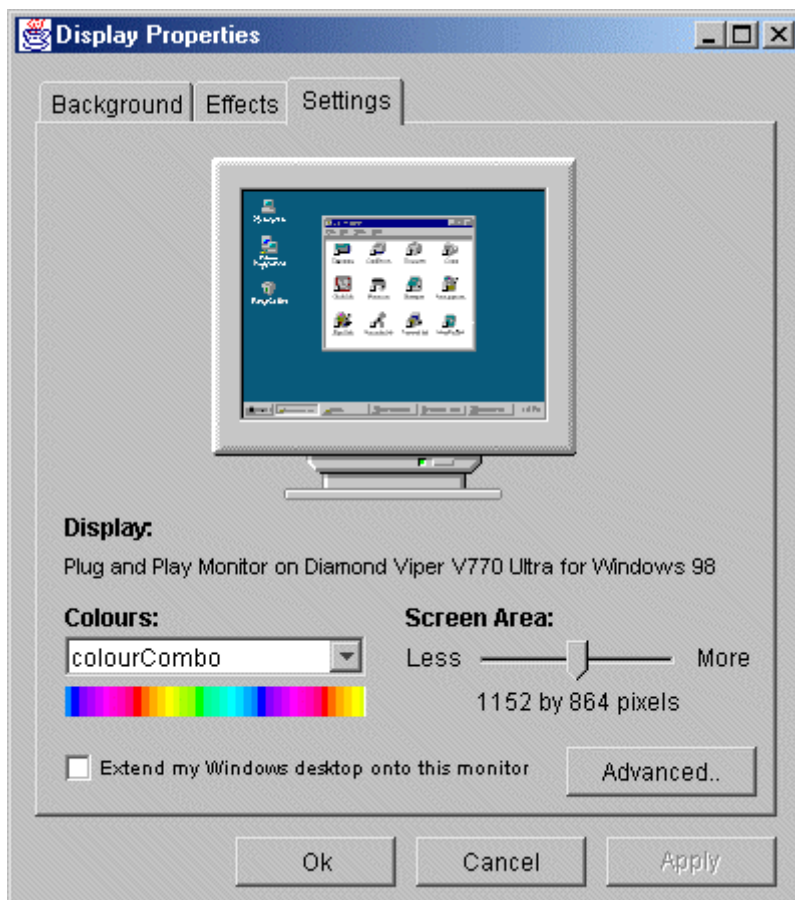
## Windows 98 power management GUI



## Windows 98 background properties GUI



Windows 98 background properties GUI cont.



Interface design for user tests

