

Tech Note 2.5 Running Interactive Programs

Recommended Startup Code

There is a significant amount of Java that is necessary to start up an interactive program.

Programs appearing after this point in the text will be assumed to contain the files listed below (in addition to any files specific to that program). There are three reasons for this:

1. To place this code where a programmer does not have to constantly scroll through it.
2. To avoid cluttering the example programs in the text.
3. To avoid getting mired in the distinction between applets and applications. The Panels section below discusses this in greater depth.

The contents of the 'html' and 'Java' files listed below appear on the following pages. These listings are provided for information only as the files will be available for downloading in project "templates" which will allow the user an easy way to create a new program by editing an existing template.

A program to be run as an applet is assumed to have available:

The file *PanelApplet.html*.

The file *PanelApplet.java*.

The file *EventPanel.java* or the file *EventPanel.class*.

A program to be run as an application is assumed to have available:

The file *PanelApplication.java*.

The file *EventPanel.java* or the file *EventPanel.class*.

At the present time complete sample programs using these project "templates" are available for download in zip files named *EventApplet.zip* and *EventApplication.zip* at

"<http://mathlab.nmu.edu/~bpeterso/Java Runtime/index.html>".

Applets and Applications

Starting up a program as an applet has the advantage that the program can be run on any computer with a Java 1.1 enabled browser. Nearly all browsers have such capability.

A program started up as an applet has the disadvantage that it generally cannot read or write disk files.

To be able to read and write disk files (as is done in chapter 7), a program is best started as an application.

Panels

A *Panel* is a class that represents a graphical sub-program. There is little penalty in filling the window of a graphical program with a *Panel* and treating the *Panel* as though it were the program itself. When this is done, the *Panel* effectively becomes a portable program — it can be started up as either an applet or an application, whichever is more convenient at the moment.

From the point of view of learning to program, starting each program with a *Panel* has the advantage that all classes in a program can have constructors written in the same way. When written in the usual way, an applet does not use its constructor to initialize instance variables; the surrogate *init* method is used instead. To an experienced programmer, this is a small, even trivial point. To a beginning student, it is just one more incomprehensible inconsistency in a situation filled with inconsistencies.

The file PanelApplet.html:

```
<applet code = PanelApplet width = 400 height = 300> </applet>
```

The file PanelApplet.java:

```
import java.awt.*;
import java.applet.Applet;

public class PanelApplet extends Applet
{
    private Panel panel;

    public void init()
    {
        setLayout(null);
        panel = new MyPanel(); // <-- your program name here
        add(panel);
        panel.setSize(getSize().width, getSize().height);
        panel.setLocation(0,0);
        panel.setVisible(true);
        panel.requestFocus();
        if (panel instanceof KeyListener)
            addKeyListener(panel);
    }
}
```

The file EventPanel.java:

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

public class EventPanel extends Panel
    implements MouseListener, MouseMotionListener, KeyListener
{
    public EventPanel()
    {
        addMouseListener(this);
        addMouseMotionListener(this);
        addKeyListener(this);
    }

    // Mouse Events
    public void mouseClicked      (MouseEvent e) {}
    public void mouseEntered      (MouseEvent e) {}
    public void mouseExited      (MouseEvent e) {}
    public void mousePressed      (MouseEvent e) {}
    public void mouseReleased     (MouseEvent e) {}

    // Mouse Motion Events
    public void mouseDragged      (MouseEvent e) {}
    public void mouseMoved        (MouseEvent e) {}

    // Key Events
    public void keyPressed        (KeyEvent e) {}
    public void keyReleased       (KeyEvent e) {}
    public void keyTyped          (KeyEvent e) {}
}
```

The file PanelApplication.java:

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

public class PanelApplication implements WindowListener
{
    public static final int PANEL_WIDTH = 400; // <-- put initial window size here
    public static final int PANEL_HEIGHT = 300;

    private static PanelApplication panelApplication;
    private Panel panel;
    private Frame frame;

    public static void main(String args[])
    {
        panelApplication = new PanelApplication();
    }

    private PanelApplication()
    {
        panel = new MyPanel(); // <-- your program name here
        panel.setLayout(null);
        panel.setSize(PANEL_WIDTH, PANEL_HEIGHT);

        frame = new Frame();
        frame.addWindowListener(this);
        frame.setTitle("My Program");
        frame.setLayout(new FlowLayout());
        frame.add(panel);
        frame.pack();
        frame.setVisible(true);
        if (panel instanceof KeyListener)
            frame.addKeyListener((KeyListener)panel);
    }

    public void windowActivated      (WindowEvent e) {}
    public void windowClosed        (WindowEvent e) {}
    public void windowClosing       (WindowEvent e)
    {
        frame.dispose();
        System.exit(0);
    }
    public void windowDeactivated   (WindowEvent e) {}
    public void windowDeiconified   (WindowEvent e) {}
    public void windowIconified     (WindowEvent e) {}
    public void windowOpened        (WindowEvent e) {}
}
```