

Supplement: Creating Internal Frames
For Introduction to Java Programming
By Y. Daniel Liang

<side remark: MDI>

You can create multiple windows, as discussed in §15.10, "Creating Multiple Windows." Java also allows you to use the JInternalFrame class to create windows within a window. This user interface is commonly known as a *multiple document interface* or *MDI*. It was once quite popular and was used in the earlier versions of many popular Windows software programs. Now, however, MDI is rarely used. That is why this section is marked optional.

The JInternalFrame class is almost the same as the external JFrame class. The components are added to the internal frame in the same way as they are added to the external frame. An internal frame can have menus, title, Close icon, Minimize icon, and Maximize icon just like an external frame. The following are the major differences:

- JInternalFrame extends JComponent, and JFrame extends the AWT Frame class. Therefore, JInternalFrame is a Swing lightweight component, and JFrame is a Swing heavyweight component.
- Both JInternalFrame and JFrame are used to hold other components. JFrame is a top-level window component, and JInternalFrame must be contained inside a JDesktopPane of a JFrame or a JApplet.

Here are the steps to create an internal frame inside another window:

1. Use a JFrame or a JApplet as the outer window.
2. Create a JDesktopPane and add it to the content pane of a JFrame or JApplet. Usually, the JDesktopPane is added to the center of the content pane.
3. Create a JInternalFrame and add it to the JDesktopPane using the add method.
4. Use the setVisible(true) method to display the internal frame.

Listing 1 gives an example that creates internal frames to display flags in an applet. You can select flags from the Flags menu. Clicking a menu item causes a flag to be displayed in an internal frame, as shown in Figure 1.



Figure 1

The flag image is displayed in an internal frame.

Listing 1 ShowInternalFrame.java

*****PD: Please add line numbers in the following code*****
*****Layout: Please layout exactly. Don't skip the space. This is true for all source code in the book. Thanks, AU.**

<Side Remark line 18: desktop pane>
<Side Remark line 20: internal frame>
<Side Remark line 23: add frame>
<Side Remark line 29: set frame icon>
<Side Remark line 31: add label>
<Side Remark line 59: main omitted>

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

public class ShowInternalFrame extends JApplet {
    // Create image icons
    private ImageIcon USIcon =
        new ImageIcon(getClass().getResource("image/usIcon.gif"));
    private ImageIcon CanadaIcon =
        new ImageIcon(getClass().getResource("image/caIcon.gif"));

    private JMenuBar jMenuBar1 = new JMenuBar();
    private JMenuItem jmiUS = new JMenuItem("US");
    private JMenuItem jmiCanada = new JMenuItem("Canada");
    private JLabel jlblImage = new JLabel(USIcon, JLabel.CENTER);

    // Create JDesktopPane to hold the internal frame
    private JDesktopPane desktop = new JDesktopPane();
    private JInternalFrame internalFrame =
        new JInternalFrame("US", true, true, true, true);

    public ShowInternalFrame() {
        desktop.add(internalFrame);

        this.setSize(new Dimension(400, 300));
        this.getContentPane().add(desktop, BorderLayout.CENTER);

        jlblImage.setIcon(USIcon);
        internalFrame.setFrameIcon(USIcon);

        internalFrame.add(jlblImage);
        internalFrame.setLocation(20, 20);
        internalFrame.setSize(100, 100);
        internalFrame.setVisible(true);

        JMenu jMenu1 = new JMenu("Flags");
        jMenuBar1.add(jMenu1);
        jMenu1.add(jmiUS);
        jMenu1.add(jmiCanada);
    }
}
```

```

        this.setJMenuBar(jMenuBar1);

        jmiUS.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jlblImage.setIcon(USIcon);
                internalFrame.setFrameIcon(USIcon);
                internalFrame.setTitle("US");
            }
        });

        jmiCanada.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                jlblImage.setIcon(CanadaIcon);
                internalFrame.setFrameIcon(CanadaIcon);
                internalFrame.setTitle("Canada");
            }
        });
    }
}

```

An image icon is displayed on a label (line 15). The label is placed inside an internal frame (line 31). As shown in Figure 34.22, an internal frame looks like an external frame. Internal frames can be used much the same way as external frames, except that internal frames are always placed inside a JDesktopPane. JDesktopPane is a subclass of JLayeredPane. Since JDesktopPane is also a subclass of JComponent, it can be placed into the content pane of a JFrame or a JApplet.

The properties of JInternalFrame and JFrame are very similar. You can set a title, an internal frame icon, size, and visible for an internal frame. You may modify this example to add menus to the internal frame too.