extent on the screen before it appears. The proxy can even delay loading the image at all until it receives a paint request and only then begin the loading process.

Sample Code

In this example program, we create a simple program to display an image on a JPanel when it is loaded. Rather than loading the image directly, we use a class we call ImageProxy to defer loading and draw a rectangle around the image area until loading is completed.

```
public class ProxyDisplay extends JxFrame {
    public ProxyDisplay() {
        super("Display proxied image");
        JPanel p = new JPanel();
        getContentPane().add(p);
        p.setLayout(new BorderLayout());
        ImageProxy image = new ImageProxy("elliott.jpg", 321, 271);

        p.add("Center", image);
        p.add("North", new Label("      "));
        p.add("West", new Label("    "));
        setSize(370, 350);
        setVisible(true);
    }
```

Note that we create the instance of the Image Proxy just as we would have for an Image and that we add it to the enclosing JPanel as we would an actual image.

The ImageProxy class sets up the image loading and creates a MediaTracker object to follow the loading process within the constructor.

```
public class ImageProxy extends JPanel
implements Runnable {
    private int height, width;
    private MediaTracker tracker;
    private Image img;
    private JFrame frame;
    private Thread imageCheck;     //to monitor loading
//-----------------
    public ImageProxy(String filename, int w, int h) {
        height = h;
        width = w;

        tracker = new MediaTracker(this);
        img =
            Toolkit.getDefaultToolkit().
```