July 2000

JavaWorld

FUELING INNOVATION

Search

GO!

Topical index
Net News Central
Developer Tools Guide
Book Catalog
Writers Guidelines
Privacy Policy
Copyright

Java Q&A

Build a better mouse trap

Take a look at the ins and outs of mouse events

By Tony Sintes

Q: **What is the difference between the `mouseClicked()` and `mousePressed()` methods of the `MouseListener` interface?**

A: Yours is one of the many questions I receive regarding mouse events -- obviously, it's a hot topic! Let's take a detailed look at the answer.

This month's Java Q&As

- Linux and Java: What's the scoop?

Want more? See the **Java Q&A** index for the full Q&A catalog.

Do you have a burning Java question? One whose answer would benefit not just you, but other *JavaWorld* readers? We've got the experts to help. Submit your **Java Q&A** questions to javaqa@javaworld.com.

Java's AWT library supplies two interfaces that allow listeners to receive mouse events. One is the `java.awt.event.MouseListener`:

```
public interface MouseListener extends EventListener
{
    public void mousePressed(MouseEvent e);
    public void mouseReleased(MouseEvent e);
    public void mouseClicked(MouseEvent e);
    public void mouseEntered(MouseEvent e);
    public void mouseExited(MouseEvent e);
}
```

The other, the `java.awt.event.MouseMotionListener`:

```
public interface MouseMotionListener extends EventListener
{
    public void mouseDragged(MouseEvent e);
    public void mouseMoved(MouseEvent e);
}
```

Swing supplies the `MouseInputListener`, an interface that extends both the `MouseMotionListener` and

`MouseListener` interfaces.

Let's look at each method found in these interfaces:

- **`mousePressed()`** occurs when the user presses the mouse button.
- **`mouseReleased()`** occurs when the user releases the mouse button.
- **`mouseClicked()`** occurs when the user presses and releases the mouse button. A user normally clicks the mouse button when selecting or double clicking an icon. (A double click is two mouse clicks in succession.) A mouse action will not result in a click if the user moves the mouse before releasing the button.
- Since a mouse click is the combination of pressing and releasing the mouse button, before the event is dispatched to the `mouseClicked()` method, the `mousePressed()` and `mouseReleased()` methods will both be called.
- **`mouseEntered()`** occurs when the mouse leaves its current component and enters the component you are listening to.
- **`mouseExited()`** occurs when the mouse leaves the component you are listening to. This event occurs the instant the mouse pointer no longer resides over the component.
- **`mouseDragged()`** occurs when the user presses the mouse button and moves the mouse before releasing the button. Releasing the mouse button after a `mouseDragged()` will not result in a `mouseClicked()`.
- **`mouseMoved()`** occurs when the mouse moves within the component without being dragged.

To listen to the mouse, one must either implement one or both of these interfaces, or extend a mouse adapter class. The AWT supplies two listener adapters: `java.awt.event.MouseAdapter` and `java.awt.event.MouseMotionAdapter`. Swing supplies an adapter for the `MouseInputListener` called `javax.swing.event.MouseInputAdapter`. Adapters eliminate the need to implement each method of the interface. Instead, you simply extend the adapter and override the methods you want to listen to.

If you are still confused, play around with the API and write a small test application or applet. Sometimes experimentation is the best way to understand the different events. I've written a small application that helps to visualize the differences between mouse events. Here's the full source (available for download in Resources):

```
import java.awt.*;
import java.awt.event.*;
public class MouseTest extends Frame
{
  public MouseTest()
  {
    Button b = new Button("JavaWorld JavaQ&A");
    b.addActionListener( new ActionListener() {
        public void actionPerformed(ActionEvent e)
        {
            System.exit(0);
        }
      }
    );
    add(b,BorderLayout.NORTH);
    addMouseListener(new MouseTest.MouseHandler());
    addMouseMotionListener(new MouseTest.MouseMotionHandler());
  }

  // MouseHandler is an inner class that implements the MouseListener.
  // Each method simply prints out a message to the command line.
  private class MouseHandler implements MouseListener
  {
    public void mousePressed(MouseEvent e)
    {
        System.out.println("mouse pressed");
    }
    public void mouseClicked(MouseEvent e)
    {
        System.out.println("moused clicked");
    }
    public void mouseReleased(MouseEvent e)
    {
        System.out.println("mouse released");
    }
    public void mouseEntered(MouseEvent e)
    {
        System.out.println("mouse entered");
    }
    public void mouseExited(MouseEvent e)
    {
        System.out.println("mouse exited");
    }
  }
  // MouseMotionHandler is an inner class that implements the MouseMotionListener.
  // Each method simply prints out a message to the command line.
  private class MouseMotionHandler implements MouseMotionListener
  {
    public void mouseMoved(MouseEvent e)
    {
        System.out.println("mouse moved");
    }
    public void mouseDragged(MouseEvent e)
    {
        System.out.println("mouse dragged");
    }
  }
  public static void main(String[] args)
  {
    new MouseTest().show();
  }
}
```

The application creates a frame and a button. As you move the mouse and press buttons, the application prints text messages to the command line, allowing you to see the mouse's actions map to events.

Writing a small test application such as the one above is a good way to experiment with most unfamiliar APIs.

Please see Resources for links to various interfaces' javadocs.

About the author
Tony Sintes is a principal consultant at BroadVision. Tony, a Sun-certified Java 1.1 programmer and Java 2 developer, has worked with Java since 1997.

Home | Mail this Story | Resources and Related Links

Resources

- To download the complete source code in zip format, go to: http://www.javaworld.com/javaqa/2000-07/mouse/02-qa-0714-mouse.zip
- For the `java.awt.event.MouseListener` javadoc, go to: http://java.sun.com/j2se/1.3/docs/api/java/awt/event/MouseListener.html
- For the `java.awt.event.MouseMotionListener` javadoc, go to: http://java.sun.com/j2se/1.3/docs/api/java/awt/event/MouseMotionListener.html
- For Swing's `MouseInputListener` interface javadoc, go to: http://java.sun.com/j2se/1.3/docs/api/javax/swing/event/MouseInputListener.html

Feedback: jweditors@javaworld.com
Technical difficulties: webmaster@javaworld.com
URL: http://www.javaworld.com/javaqa/2000-07/02-qa-0714-mouse.html
Last modified: Friday, July 14, 2000