

SWING - JWINDOW CLASS

http://www.tutorialspoint.com/swing/swing_jwindow.htm

Copyright © tutorialspoint.com

Introduction

The class **JWindow** is a container that can be displayed but does not have the title bar or window-management buttons.

Class declaration

Following is the declaration for **javax.swing.JWindow** class:

```
public class JWindow
    extends Window
    implements Accessible, RootPaneContainer
```

Field

Following are the fields for **java.awt.Component** class:

- **protected AccessibleContext accessibleContext** -- The accessible context property.
- **protected JRootPane rootPane** -- The JRootPane instance that manages the contentPane and optional menuBar for this frame, as well as the glassPane.
- **protected boolean rootPaneCheckingEnabled** -- If true then calls to add and setLayout will be forwarded to the contentPane.

Class constructors

S.N.	Constructor & Description
1	JWindow Creates a window with no specified owner.
2	JWindowFrameowner Creates a window with the specified owner frame.
3	JWindowGraphicsConfigurationgc Creates a window with the specified GraphicsConfiguration of a screen device.
4	JWindowWindowowner Creates a window with the specified owner window.
5	JWindowWindowowner, GraphicsConfigurationgc Creates a window with the specified owner window and GraphicsConfiguration of a screen device.

Class methods

S.N.	Method & Description
1	protected void addImpl <i>Componentcomp, Objectconstraints, intindex</i> Adds the specified child Component.
2	protected JRootPane createRootPane Called by the constructor methods to create the default rootPane.
3	AccessibleContext getAccessibleContext Gets the AccessibleContext associated with this JWindow.
4	Container getContentPane Returns the Container which is the contentPane for this window.
5	Component getGlassPane Returns the glassPane Component for this window.
6	Graphics getGraphics Creates a graphics context for this component.
7	JLayeredPane getLayeredPane Returns the layeredPane object for this window.
8	JRootPane getRootPane Returns the rootPane object for this window.
9	TransferHandler getTransferHandler Gets the transferHandler property.
10	protected boolean isRootPaneCheckingEnabled Returns whether calls to add and setLayout are forwarded to the contentPane.
11	protected String paramString Returns a string representation of this JWindow.
12	void remove <i>Componentcomp</i> Removes the specified component from the container.
13	void repaint <i>longtime, intx, inty, intwidth, intheight</i> Repaints the specified rectangle of this component within time milliseconds.
14	void setContentPane <i>ContainercontentPane</i>

	Sets the contentPane property for this window.
15	void setGlassPaneComponent <i>glassPane</i> Sets the glassPane property.
16	void setLayeredPaneJLayeredPanel <i>layeredPane</i> Sets the layeredPane property.
17	void setLayoutManager <i>manager</i> Sets the LayoutManager.
18	protected void setRootPaneJRootPane <i>root</i> Sets the new rootPane object for this window.
19	protected void setRootPaneCheckingEnabled <i>boolean enabled</i> Sets whether calls to add and setLayout are forwarded to the contentPane.
20	void setTransferHandlerTransferHandler <i>newHandler</i> Sets the transferHandler property, which is a mechanism to support transfer of data into this component.
21	void updateGraphics <i>g</i> Calls paintg.
22	protected void windowInit Called by the constructors to init the JWindow properly.

Methods inherited

This class inherits methods from the following classes:

- java.awt.Window
- java.awt.Container
- java.awt.Component
- java.lang.Object

JWindow Example

Create the following java program using any editor of your choice in say **D:/ > SWING > com > tutorialspoint > gui >**

SwingContainerDemo.java

```
package com.tutorialspoint.gui;

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

```

public class SwingContainerDemo {
    private JFrame mainFrame;
    private JLabel headerLabel;
    private JLabel statusLabel;
    private JPanel controlPanel;
    private JLabel msglabel;

    public SwingContainerDemo(){
        prepareGUI();
    }

    public static void main(String[] args){
        SwingContainerDemo swingContainerDemo = new SwingContainerDemo();
        swingContainerDemo.showJWindowDemo();
    }

    private void prepareGUI(){
        mainFrame = new JFrame("Java Swing Examples");
        mainFrame.setSize(400,400);
        mainFrame.setLayout(new GridLayout(3, 1));
        mainFrame.addWindowListener(new WindowAdapter() {
            public void windowClosing(WindowEvent windowEvent){
                System.exit(0);
            }
        });
        headerLabel = new JLabel("", JLabel.CENTER);
        statusLabel = new JLabel("", JLabel.CENTER);

        statusLabel.setSize(350,100);

        msglabel = new JLabel("Welcome to TutorialsPoint SWING Tutorial."
            , JLabel.CENTER);

        controlPanel = new JPanel();
        controlPanel.setLayout(new FlowLayout());

        mainFrame.add(headerLabel);
        mainFrame.add(controlPanel);
        mainFrame.add(statusLabel);
        mainFrame.setVisible(true);
    }

    private void showJWindowDemo(){
        headerLabel.setText("Container in action: JWindow");
        final MessageWindow window = new MessageWindow(mainFrame, "Welcome to
TutorialsPoint SWING Tutorial.");

        JButton okButton = new JButton("Open a Window");
        okButton.addActionListener(new ActionListener() {
            public void actionPerformed(ActionEvent e) {
                window.setVisible(true);
                statusLabel.setText("A Window shown to the user.");
            }
        });
        controlPanel.add(okButton);
        mainFrame.setVisible(true);
    }

    class MessageWindow extends JWindow{
        private String message;

        public MessageWindow(JFrame parent, String
            message) {
            super(parent);
            this.message = message;
            setSize(300, 300);
            setLocationRelativeTo(parent);
        }
    }
}

```

```
public void paint(Graphics g)
{
    super.paint(g);
    g.drawRect(0,0,getSize().width - 1,getSize().height - 1);
    g.drawString(message,50,150);
}
}
```

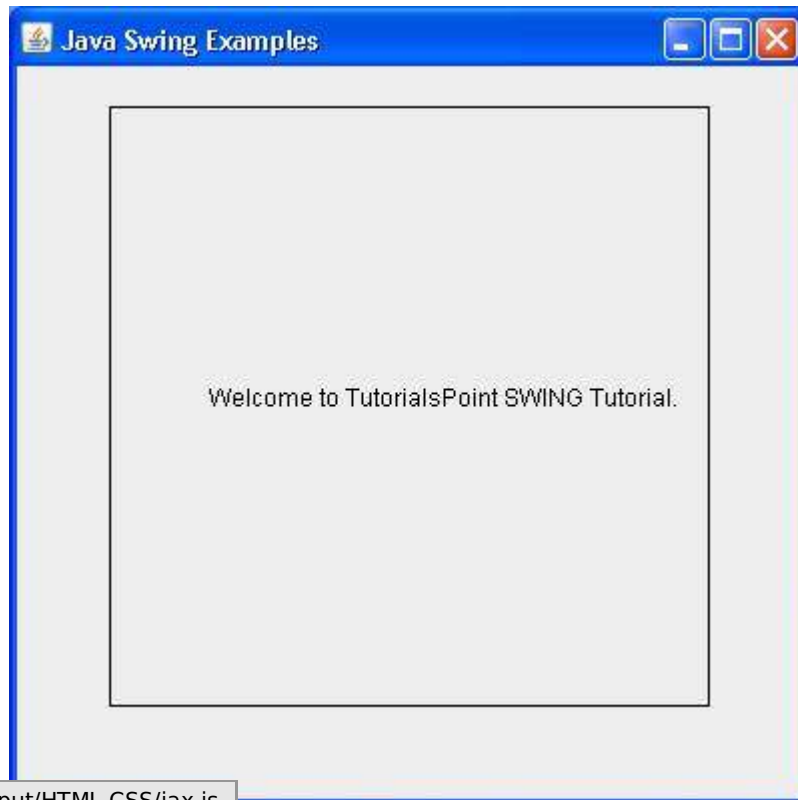
Compile the program using command prompt. Go to **D:/ > SWING** and type the following command.

```
D:\SWING>javac com\tutorialspoint\gui\SwingContainerDemo.java
```

If no error comes that means compilation is successful. Run the program using following command.

```
D:\SWING>java com.tutorialspoint.gui.SwingContainerDemo
```

Verify the following output



Loading [MathJax]/jax/output/HTML-CSS/jax.js