| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JRootPane.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JRadioButtonMenuItem.AccessibleJRadioButtonMenuItem.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JRootPane.AccessibleJRootPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JRootPane.html)    [**NO FRAMES**](http://docs.google.com/JRootPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#ihv636) | [METHOD](#1hmsyys) |

## **javax.swing**

Class JRootPane

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 [java.awt.Container](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 **javax.swing.JRootPane**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

public class **JRootPane**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

A lightweight container used behind the scenes by JFrame, JDialog, JWindow, JApplet, and JInternalFrame. For task-oriented information on functionality provided by root panes see [How to Use Root Panes](http://java.sun.com/docs/books/tutorial/uiswing/components/rootpane.html), a section in *The Java Tutorial*.

The following image shows the relationships between the classes that use root panes.



The "heavyweight" components (those that delegate to a peer, or native component on the host system) are shown with a darker, heavier box. The four heavyweight JFC/Swing containers (JFrame, JDialog, JWindow, and JApplet) are shown in relation to the AWT classes they extend. These four components are the only heavyweight containers in the Swing library. The lightweight container JInternalFrame is also shown. All five of these JFC/Swing containers implement the RootPaneContainer interface, and they all delegate their operations to a JRootPane (shown with a little "handle" on top).**Note:** The JComponent method getRootPane can be used to obtain the JRootPane that contains a given component.

|  |
| --- |

The diagram at right shows the structure of a JRootPane. A JRootpane is made up of a glassPane, an optional menuBar, and a contentPane. (The JLayeredPane manages the menuBar and the contentPane.) The glassPane sits over the top of everything, where it is in a position to intercept mouse movements. Since the glassPane (like the contentPane) can be an arbitrary component, it is also possible to set up the glassPane for drawing. Lines and images on the glassPane can then range over the frames underneath without being limited by their boundaries.

Although the menuBar component is optional, the layeredPane, contentPane, and glassPane always exist. Attempting to set them to null generates an exception.

To add components to the JRootPane (other than the optional menu bar), you add the object to the contentPane of the JRootPane, like this:

rootPane.getContentPane().add(child);

The same principle holds true for setting layout managers, removing components, listing children, etc. All these methods are invoked on the contentPane instead of on the JRootPane.**Note:** The default layout manager for the contentPane is a BorderLayout manager. However, the JRootPane uses a custom LayoutManager. So, when you want to change the layout manager for the components you added to a JRootPane, be sure to use code like this:

rootPane.getContentPane().setLayout(new BoxLayout());

If a JMenuBar component is set on the JRootPane, it is positioned along the upper edge of the frame. The contentPane is adjusted in location and size to fill the remaining area. (The JMenuBar and the contentPane are added to the layeredPane component at the JLayeredPane.FRAME\_CONTENT\_LAYER layer.)

The layeredPane is the parent of all children in the JRootPane -- both as the direct parent of the menu and the grandparent of all components added to the contentPane. It is an instance of JLayeredPane, which provides the ability to add components at several layers. This capability is very useful when working with menu popups, dialog boxes, and dragging -- situations in which you need to place a component on top of all other components in the pane.

The glassPane sits on top of all other components in the JRootPane. That provides a convenient place to draw above all other components, and makes it possible to intercept mouse events, which is useful both for dragging and for drawing. Developers can use setVisible on the glassPane to control when the glassPane displays over the other children. By default the glassPane is not visible.

The custom LayoutManager used by JRootPane ensures that:

1. The glassPane fills the entire viewable area of the JRootPane (bounds - insets).
2. The layeredPane fills the entire viewable area of the JRootPane. (bounds - insets)
3. The menuBar is positioned at the upper edge of the layeredPane.
4. The contentPane fills the entire viewable area, minus the menuBar, if present.

Any other views in the JRootPane view hierarchy are ignored.

If you replace the LayoutManager of the JRootPane, you are responsible for managing all of these views. So ordinarily you will want to be sure that you change the layout manager for the contentPane rather than for the JRootPane itself!

The painting architecture of Swing requires an opaque JComponent to exist in the containment hieararchy above all other components. This is typically provided by way of the content pane. If you replace the content pane, it is recommended that you make the content pane opaque by way of setOpaque(true). Additionally, if the content pane overrides paintComponent, it will need to completely fill in the background in an opaque color in paintComponent.

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html), [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html), [JWindow](http://docs.google.com/javax/swing/JWindow.html), [JFrame](http://docs.google.com/javax/swing/JFrame.html), [JDialog](http://docs.google.com/javax/swing/JDialog.html), [JApplet](http://docs.google.com/javax/swing/JApplet.html), [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html), [JComponent](http://docs.google.com/javax/swing/JComponent.html), [BoxLayout](http://docs.google.com/javax/swing/BoxLayout.html),  [Mixing Heavy and Light Components](http://java.sun.com/products/jfc/tsc/articles/mixing/)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JRootPane.AccessibleJRootPane**](http://docs.google.com/javax/swing/JRootPane.AccessibleJRootPane.html)            This class implements accessibility support for the JRootPane class. |
| protected  class | [**JRootPane.RootLayout**](http://docs.google.com/javax/swing/JRootPane.RootLayout.html)            A custom layout manager that is responsible for the layout of layeredPane, glassPane, and menuBar. |

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| static int | [**COLOR\_CHOOSER\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#COLOR_CHOOSER_DIALOG)            Constant used for the windowDecorationStyle property. |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**contentPane**](http://docs.google.com/javax/swing/JRootPane.html#contentPane)            The content pane. |
| protected  [JButton](http://docs.google.com/javax/swing/JButton.html) | [**defaultButton**](http://docs.google.com/javax/swing/JRootPane.html#defaultButton)            The button that gets activated when the pane has the focus and a UI-specific action like pressing the **Enter** key occurs. |
| protected  javax.swing.JRootPane.DefaultAction | [**defaultPressAction**](http://docs.google.com/javax/swing/JRootPane.html#defaultPressAction)  **Deprecated.** *As of Java 2 platform v1.3.* |
| protected  javax.swing.JRootPane.DefaultAction | [**defaultReleaseAction**](http://docs.google.com/javax/swing/JRootPane.html#defaultReleaseAction)  **Deprecated.** *As of Java 2 platform v1.3.* |
| static int | [**ERROR\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#ERROR_DIALOG)            Constant used for the windowDecorationStyle property. |
| static int | [**FILE\_CHOOSER\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#FILE_CHOOSER_DIALOG)            Constant used for the windowDecorationStyle property. |
| static int | [**FRAME**](http://docs.google.com/javax/swing/JRootPane.html#FRAME)            Constant used for the windowDecorationStyle property. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**glassPane**](http://docs.google.com/javax/swing/JRootPane.html#glassPane)            The glass pane that overlays the menu bar and content pane, so it can intercept mouse movements and such. |
| static int | [**INFORMATION\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#INFORMATION_DIALOG)            Constant used for the windowDecorationStyle property. |
| protected  [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**layeredPane**](http://docs.google.com/javax/swing/JRootPane.html#layeredPane)            The layered pane that manages the menu bar and content pane. |
| protected  [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**menuBar**](http://docs.google.com/javax/swing/JRootPane.html#menuBar)            The menu bar. |
| static int | [**NONE**](http://docs.google.com/javax/swing/JRootPane.html#NONE)            Constant used for the windowDecorationStyle property. |
| static int | [**PLAIN\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#PLAIN_DIALOG)            Constant used for the windowDecorationStyle property. |
| static int | [**QUESTION\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#QUESTION_DIALOG)            Constant used for the windowDecorationStyle property. |
| static int | [**WARNING\_DIALOG**](http://docs.google.com/javax/swing/JRootPane.html#WARNING_DIALOG)            Constant used for the windowDecorationStyle property. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**JRootPane**](http://docs.google.com/javax/swing/JRootPane.html#JRootPane())()            Creates a JRootPane, setting up its glassPane, layeredPane, and contentPane. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addImpl**](http://docs.google.com/javax/swing/JRootPane.html#addImpl(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Overridden to enforce the position of the glass component as the zero child. |
| void | [**addNotify**](http://docs.google.com/javax/swing/JRootPane.html#addNotify())()            Register ourselves with the SystemEventQueueUtils as a new root pane. |
| protected  [Container](http://docs.google.com/java/awt/Container.html) | [**createContentPane**](http://docs.google.com/javax/swing/JRootPane.html#createContentPane())()            Called by the constructor methods to create the default contentPane. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**createGlassPane**](http://docs.google.com/javax/swing/JRootPane.html#createGlassPane())()            Called by the constructor methods to create the default glassPane. |
| protected  [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**createLayeredPane**](http://docs.google.com/javax/swing/JRootPane.html#createLayeredPane())()            Called by the constructor methods to create the default layeredPane. |
| protected  [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) | [**createRootLayout**](http://docs.google.com/javax/swing/JRootPane.html#createRootLayout())()            Called by the constructor methods to create the default layoutManager. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JRootPane.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JRootPane. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getContentPane**](http://docs.google.com/javax/swing/JRootPane.html#getContentPane())()            Returns the content pane -- the container that holds the components parented by the root pane. |
| [JButton](http://docs.google.com/javax/swing/JButton.html) | [**getDefaultButton**](http://docs.google.com/javax/swing/JRootPane.html#getDefaultButton())()            Returns the value of the defaultButton property. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getGlassPane**](http://docs.google.com/javax/swing/JRootPane.html#getGlassPane())()            Returns the current glass pane for this JRootPane. |
| [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**getJMenuBar**](http://docs.google.com/javax/swing/JRootPane.html#getJMenuBar())()            Returns the menu bar from the layered pane. |
| [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**getLayeredPane**](http://docs.google.com/javax/swing/JRootPane.html#getLayeredPane())()            Gets the layered pane used by the root pane. |
| [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**getMenuBar**](http://docs.google.com/javax/swing/JRootPane.html#getMenuBar())()  **Deprecated.** *As of Swing version 1.0.3 replaced by getJMenuBar().* |
| [RootPaneUI](http://docs.google.com/javax/swing/plaf/RootPaneUI.html) | [**getUI**](http://docs.google.com/javax/swing/JRootPane.html#getUI())()            Returns the L&F object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JRootPane.html#getUIClassID())()            Returns a string that specifies the name of the L&F class that renders this component. |
| int | [**getWindowDecorationStyle**](http://docs.google.com/javax/swing/JRootPane.html#getWindowDecorationStyle())()            Returns a constant identifying the type of Window decorations the JRootPane is providing. |
| boolean | [**isOptimizedDrawingEnabled**](http://docs.google.com/javax/swing/JRootPane.html#isOptimizedDrawingEnabled())()            The glassPane and contentPane have the same bounds, which means JRootPane does not tiles its children and this should return false. |
| boolean | [**isValidateRoot**](http://docs.google.com/javax/swing/JRootPane.html#isValidateRoot())()            If a descendant of this JRootPane calls revalidate, validate from here on down. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JRootPane.html#paramString())()            Returns a string representation of this JRootPane. |
| void | [**removeNotify**](http://docs.google.com/javax/swing/JRootPane.html#removeNotify())()            Unregister ourselves from SystemEventQueueUtils. |
| void | [**setContentPane**](http://docs.google.com/javax/swing/JRootPane.html#setContentPane(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) content)            Sets the content pane -- the container that holds the components parented by the root pane. |
| void | [**setDefaultButton**](http://docs.google.com/javax/swing/JRootPane.html#setDefaultButton(javax.swing.JButton))([JButton](http://docs.google.com/javax/swing/JButton.html) defaultButton)            Sets the defaultButton property, which determines the current default button for this JRootPane. |
| void | [**setDoubleBuffered**](http://docs.google.com/javax/swing/JRootPane.html#setDoubleBuffered(boolean))(boolean aFlag)            Sets whether this component should use a buffer to paint. |
| void | [**setGlassPane**](http://docs.google.com/javax/swing/JRootPane.html#setGlassPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) glass)            Sets a specified Component to be the glass pane for this root pane. |
| void | [**setJMenuBar**](http://docs.google.com/javax/swing/JRootPane.html#setJMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menu)            Adds or changes the menu bar used in the layered pane. |
| void | [**setLayeredPane**](http://docs.google.com/javax/swing/JRootPane.html#setLayeredPane(javax.swing.JLayeredPane))([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layered)            Sets the layered pane for the root pane. |
| void | [**setMenuBar**](http://docs.google.com/javax/swing/JRootPane.html#setMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menu)  **Deprecated.** *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar menu).* |
| void | [**setUI**](http://docs.google.com/javax/swing/JRootPane.html#setUI(javax.swing.plaf.RootPaneUI))([RootPaneUI](http://docs.google.com/javax/swing/plaf/RootPaneUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**setWindowDecorationStyle**](http://docs.google.com/javax/swing/JRootPane.html#setWindowDecorationStyle(int))(int windowDecorationStyle)            Sets the type of Window decorations (such as borders, widgets for closing a Window, title ...) the JRootPane should provide. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JRootPane.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### NONE

public static final int **NONE**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should not provide any sort of Window decorations.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.NONE)

### FRAME

public static final int **FRAME**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Frame.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.FRAME)

### PLAIN\_DIALOG

public static final int **PLAIN\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.PLAIN_DIALOG)

### INFORMATION\_DIALOG

public static final int **INFORMATION\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to display an informational message.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.INFORMATION_DIALOG)

### ERROR\_DIALOG

public static final int **ERROR\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to display an error message.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.ERROR_DIALOG)

### COLOR\_CHOOSER\_DIALOG

public static final int **COLOR\_CHOOSER\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to display a JColorChooser.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.COLOR_CHOOSER_DIALOG)

### FILE\_CHOOSER\_DIALOG

public static final int **FILE\_CHOOSER\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to display a JFileChooser.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.FILE_CHOOSER_DIALOG)

### QUESTION\_DIALOG

public static final int **QUESTION\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to present a question to the user.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.QUESTION_DIALOG)

### WARNING\_DIALOG

public static final int **WARNING\_DIALOG**

Constant used for the windowDecorationStyle property. Indicates that the JRootPane should provide decorations appropriate for a Dialog used to display a warning message.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JRootPane.WARNING_DIALOG)

### menuBar

protected [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) **menuBar**

The menu bar.

### contentPane

protected [Container](http://docs.google.com/java/awt/Container.html) **contentPane**

The content pane.

### layeredPane

protected [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) **layeredPane**

The layered pane that manages the menu bar and content pane.

### glassPane

protected [Component](http://docs.google.com/java/awt/Component.html) **glassPane**

The glass pane that overlays the menu bar and content pane, so it can intercept mouse movements and such.

### defaultButton

protected [JButton](http://docs.google.com/javax/swing/JButton.html) **defaultButton**

The button that gets activated when the pane has the focus and a UI-specific action like pressing the **Enter** key occurs.

### defaultPressAction

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected javax.swing.JRootPane.DefaultAction **defaultPressAction**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this unusable field is no longer used. To override the default button you should replace the Action in the JRootPane's ActionMap. Please refer to the key bindings specification for further details.

**See Also:**[defaultButton](http://docs.google.com/javax/swing/JRootPane.html#defaultButton)

### defaultReleaseAction

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected javax.swing.JRootPane.DefaultAction **defaultReleaseAction**

**Deprecated.** *As of Java 2 platform v1.3.*As of Java 2 platform v1.3 this unusable field is no longer used. To override the default button you should replace the Action in the JRootPane's ActionMap. Please refer to the key bindings specification for further details.

**See Also:**[defaultButton](http://docs.google.com/javax/swing/JRootPane.html#defaultButton)

| **Constructor Detail** |
| --- |

### JRootPane

public **JRootPane**()

Creates a JRootPane, setting up its glassPane, layeredPane, and contentPane.

| **Method Detail** |
| --- |

### setDoubleBuffered

public void **setDoubleBuffered**(boolean aFlag)

Sets whether this component should use a buffer to paint. If set to true, all the drawing from this component will be done in an offscreen painting buffer. The offscreen painting buffer will the be copied onto the screen. If a Component is buffered and one of its ancestor is also buffered, the ancestor buffer will be used.

**Overrides:**[setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**aFlag - if true, set this component to be double buffered**Since:** 1.6

### getWindowDecorationStyle

public int **getWindowDecorationStyle**()

Returns a constant identifying the type of Window decorations the JRootPane is providing.

**Returns:**One of NONE, FRAME, PLAIN\_DIALOG, INFORMATION\_DIALOG, ERROR\_DIALOG, COLOR\_CHOOSER\_DIALOG, FILE\_CHOOSER\_DIALOG, QUESTION\_DIALOG or WARNING\_DIALOG.**Since:** 1.4 **See Also:**[setWindowDecorationStyle(int)](http://docs.google.com/javax/swing/JRootPane.html#setWindowDecorationStyle(int))

### setWindowDecorationStyle

public void **setWindowDecorationStyle**(int windowDecorationStyle)

Sets the type of Window decorations (such as borders, widgets for closing a Window, title ...) the JRootPane should provide. The default is to provide no Window decorations (NONE).

This is only a hint, and some look and feels may not support this. This is a bound property.

**Parameters:**windowDecorationStyle - Constant identifying Window decorations to provide. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if style is not one of: NONE, FRAME, PLAIN\_DIALOG, INFORMATION\_DIALOG, ERROR\_DIALOG, COLOR\_CHOOSER\_DIALOG, FILE\_CHOOSER\_DIALOG, QUESTION\_DIALOG, or WARNING\_DIALOG.**Since:** 1.4 **See Also:**[JDialog.setDefaultLookAndFeelDecorated(boolean)](http://docs.google.com/javax/swing/JDialog.html#setDefaultLookAndFeelDecorated(boolean)), [JFrame.setDefaultLookAndFeelDecorated(boolean)](http://docs.google.com/javax/swing/JFrame.html#setDefaultLookAndFeelDecorated(boolean)), [LookAndFeel.getSupportsWindowDecorations()](http://docs.google.com/javax/swing/LookAndFeel.html#getSupportsWindowDecorations())

### getUI

public [RootPaneUI](http://docs.google.com/javax/swing/plaf/RootPaneUI.html) **getUI**()

Returns the L&F object that renders this component.

**Returns:**LabelUI object**Since:** 1.3

### setUI

public void **setUI**([RootPaneUI](http://docs.google.com/javax/swing/plaf/RootPaneUI.html) ui)

Sets the L&F object that renders this component.

**Parameters:**ui - the LabelUI L&F object**Since:** 1.3 **See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### updateUI

public void **updateUI**()

Resets the UI property to a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

public [String](http://docs.google.com/java/lang/String.html) **getUIClassID**()

Returns a string that specifies the name of the L&F class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "RootPaneUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### createLayeredPane

protected [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) **createLayeredPane**()

Called by the constructor methods to create the default layeredPane. Bt default it creates a new JLayeredPane.

**Returns:**the default layeredPane

### createContentPane

protected [Container](http://docs.google.com/java/awt/Container.html) **createContentPane**()

Called by the constructor methods to create the default contentPane. By default this method creates a new JComponent add sets a BorderLayout as its LayoutManager.

**Returns:**the default contentPane

### createGlassPane

protected [Component](http://docs.google.com/java/awt/Component.html) **createGlassPane**()

Called by the constructor methods to create the default glassPane. By default this method creates a new JComponent with visibility set to false.

**Returns:**the default glassPane

### createRootLayout

protected [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **createRootLayout**()

Called by the constructor methods to create the default layoutManager.

**Returns:**the default layoutManager.

### setJMenuBar

public void **setJMenuBar**([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menu)

Adds or changes the menu bar used in the layered pane.

**Parameters:**menu - the JMenuBar to add

### setMenuBar

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **setMenuBar**([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menu)

**Deprecated.** *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar menu).*

Specifies the menu bar value.

**Parameters:**menu - the JMenuBar to add.

### getJMenuBar

public [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) **getJMenuBar**()

Returns the menu bar from the layered pane.

**Returns:**the JMenuBar used in the pane

### getMenuBar

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) **getMenuBar**()

**Deprecated.** *As of Swing version 1.0.3 replaced by getJMenuBar().*

Returns the menu bar value.

**Returns:**the JMenuBar used in the pane

### setContentPane

public void **setContentPane**([Container](http://docs.google.com/java/awt/Container.html) content)

Sets the content pane -- the container that holds the components parented by the root pane.

Swing's painting architecture requires an opaque JComponent in the containment hiearchy. This is typically provided by the content pane. If you replace the content pane it is recommended you replace it with an opaque JComponent.

**Parameters:**content - the Container to use for component-contents **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the content pane parameter is null

### getContentPane

public [Container](http://docs.google.com/java/awt/Container.html) **getContentPane**()

Returns the content pane -- the container that holds the components parented by the root pane.

**Returns:**the Container that holds the component-contents

### setLayeredPane

public void **setLayeredPane**([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layered)

Sets the layered pane for the root pane. The layered pane typically holds a content pane and an optional JMenuBar.

**Parameters:**layered - the JLayeredPane to use **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the layered pane parameter is null

### getLayeredPane

public [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) **getLayeredPane**()

Gets the layered pane used by the root pane. The layered pane typically holds a content pane and an optional JMenuBar.

**Returns:**the JLayeredPane currently in use

### setGlassPane

public void **setGlassPane**([Component](http://docs.google.com/java/awt/Component.html) glass)

Sets a specified Component to be the glass pane for this root pane. The glass pane should normally be a lightweight, transparent component, because it will be made visible when ever the root pane needs to grab input events.

The new glass pane's visibility is changed to match that of the current glass pane. An implication of this is that care must be taken when you want to replace the glass pane and make it visible. Either of the following will work:

root.setGlassPane(newGlassPane);  
 newGlassPane.setVisible(true);

or:

root.getGlassPane().setVisible(true);  
 root.setGlassPane(newGlassPane);

**Parameters:**glass - the Component to use as the glass pane for this JRootPane **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the glass parameter is null

### getGlassPane

public [Component](http://docs.google.com/java/awt/Component.html) **getGlassPane**()

Returns the current glass pane for this JRootPane.

**Returns:**the current glass pane**See Also:**[setGlassPane(java.awt.Component)](http://docs.google.com/javax/swing/JRootPane.html#setGlassPane(java.awt.Component))

### isValidateRoot

public boolean **isValidateRoot**()

If a descendant of this JRootPane calls revalidate, validate from here on down.

Deferred requests to layout a component and its descendents again. For example, calls to revalidate, are pushed upwards to either a JRootPane or a JScrollPane because both classes override isValidateRoot to return true.

**Overrides:**[isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**true**See Also:**[JComponent.isValidateRoot()](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot())

### isOptimizedDrawingEnabled

public boolean **isOptimizedDrawingEnabled**()

The glassPane and contentPane have the same bounds, which means JRootPane does not tiles its children and this should return false. On the other hand, the glassPane is normally not visible, and so this can return true if the glassPane isn't visible. Therefore, the return value here depends upon the visiblity of the glassPane.

**Overrides:**[isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**true if this component's children don't overlap

### addNotify

public void **addNotify**()

Register ourselves with the SystemEventQueueUtils as a new root pane.

**Overrides:**[addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.registerKeyboardAction(java.awt.event.ActionListener, java.lang.String, javax.swing.KeyStroke, int)](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int))

### removeNotify

public void **removeNotify**()

Unregister ourselves from SystemEventQueueUtils.

**Overrides:**[removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[addNotify()](http://docs.google.com/javax/swing/JRootPane.html#addNotify())

### setDefaultButton

public void **setDefaultButton**([JButton](http://docs.google.com/javax/swing/JButton.html) defaultButton)

Sets the defaultButton property, which determines the current default button for this JRootPane. The default button is the button which will be activated when a UI-defined activation event (typically the **Enter** key) occurs in the root pane regardless of whether or not the button has keyboard focus (unless there is another component within the root pane which consumes the activation event, such as a JTextPane). For default activation to work, the button must be an enabled descendent of the root pane when activation occurs. To remove a default button from this root pane, set this property to null.

**Parameters:**defaultButton - the JButton which is to be the default button**See Also:**[JButton.isDefaultButton()](http://docs.google.com/javax/swing/JButton.html#isDefaultButton())

### getDefaultButton

public [JButton](http://docs.google.com/javax/swing/JButton.html) **getDefaultButton**()

Returns the value of the defaultButton property.

**Returns:**the JButton which is currently the default button**See Also:**[setDefaultButton(javax.swing.JButton)](http://docs.google.com/javax/swing/JRootPane.html#setDefaultButton(javax.swing.JButton))

### addImpl

protected void **addImpl**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints,  
 int index)

Overridden to enforce the position of the glass component as the zero child.

**Overrides:**[addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**comp - the component to be enhancedconstraints - the constraints to be respectedindex - the index**See Also:**[Container.add(Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [Container.add(Component, int)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [Container.add(Component, java.lang.Object)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### paramString

protected [String](http://docs.google.com/java/lang/String.html) **paramString**()

Returns a string representation of this JRootPane. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Overrides:**[paramString](http://docs.google.com/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JRootPane.

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this JRootPane. For root panes, the AccessibleContext takes the form of an AccessibleJRootPane. A new AccessibleJRootPane instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJRootPane that serves as the AccessibleContext of this JRootPane

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JRootPane.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JRadioButtonMenuItem.AccessibleJRadioButtonMenuItem.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JRootPane.AccessibleJRootPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JRootPane.html)    [**NO FRAMES**](http://docs.google.com/JRootPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#ihv636) | [METHOD](#1hmsyys) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).