| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RootPaneContainer.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/RepaintManager.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/RowFilter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/RootPaneContainer.html)    [**NO FRAMES**](http://docs.google.com/RootPaneContainer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.swing**

Interface RootPaneContainer

**All Known Implementing Classes:** [JApplet](http://docs.google.com/javax/swing/JApplet.html), [JDialog](http://docs.google.com/javax/swing/JDialog.html), [JFrame](http://docs.google.com/javax/swing/JFrame.html), [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html), [JWindow](http://docs.google.com/javax/swing/JWindow.html)

public interface **RootPaneContainer**

This interface is implemented by components that have a single JRootPane child: JDialog, JFrame, JWindow, JApplet, JInternalFrame. The methods in this interface are just *covers* for the JRootPane properties, e.g. getContentPane() is generally implemented like this:

public Container getContentPane() {  
 return getRootPane().getContentPane();  
 }

This interface serves as a *marker* for Swing GUI builders that need to treat components like JFrame, that contain a single JRootPane, specially. For example in a GUI builder, dropping a component on a RootPaneContainer would be interpreted as frame.getContentPane().add(child).

For conveniance JFrame, JDialog, JWindow, JApplet and JInternalFrame, by default, forward, by default, all calls to the add, remove and setLayout methods, to the contentPane. This means you can call:

rootPaneContainer.add(component);

instead of:

rootPaneContainer.getContentPane().add(component);

The behavior of the add and setLayout methods for JFrame, JDialog, JWindow, JApplet and JInternalFrame is controlled by the rootPaneCheckingEnabled property. If this property is true (the default), then calls to these methods are forwarded to the contentPane; if false, these methods operate directly on the RootPaneContainer. This property is only intended for subclasses, and is therefore protected.

**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [JFrame](http://docs.google.com/javax/swing/JFrame.html), [JDialog](http://docs.google.com/javax/swing/JDialog.html), [JWindow](http://docs.google.com/javax/swing/JWindow.html), [JApplet](http://docs.google.com/javax/swing/JApplet.html), [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html)

| **Method Summary** | |
| --- | --- |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getContentPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#getContentPane())()            Returns the contentPane. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getGlassPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#getGlassPane())()            Returns the glassPane. |
| [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**getLayeredPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#getLayeredPane())()            Returns the layeredPane. |
| [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**getRootPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#getRootPane())()            Return this component's single JRootPane child. |
| void | [**setContentPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#setContentPane(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) contentPane)            The "contentPane" is the primary container for application specific components. |
| void | [**setGlassPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#setGlassPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) glassPane)            The glassPane is always the first child of the rootPane and the rootPanes layout manager ensures that it's always as big as the rootPane. |
| void | [**setLayeredPane**](http://docs.google.com/javax/swing/RootPaneContainer.html#setLayeredPane(javax.swing.JLayeredPane))([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layeredPane)            A Container that manages the contentPane and in some cases a menu bar. |

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

### getRootPane

[JRootPane](http://docs.google.com/javax/swing/JRootPane.html) **getRootPane**()

Return this component's single JRootPane child. A conventional implementation of this interface will have all of the other methods indirect through this one. The rootPane has two children: the glassPane and the layeredPane.

**Returns:**this components single JRootPane child.**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html)

### setContentPane

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

The "contentPane" is the primary container for application specific components. Applications should add children to the contentPane, set its layout manager, and so on.

The contentPane may not be null.

Generally implemented with getRootPane().setContentPane(contentPane);

**Parameters:**contentPane - the Container to use for the contents of this JRootPane **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the content pane parameter is null**See Also:**[JRootPane.getContentPane()](http://docs.google.com/javax/swing/JRootPane.html#getContentPane()), [getContentPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getContentPane())

### getContentPane

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

Returns the contentPane.

**Returns:**the value of the contentPane property.**See Also:**[setContentPane(java.awt.Container)](http://docs.google.com/javax/swing/RootPaneContainer.html#setContentPane(java.awt.Container))

### setLayeredPane

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

A Container that manages the contentPane and in some cases a menu bar. The layeredPane can be used by descendants that want to add a child to the RootPaneContainer that isn't layout managed. For example an internal dialog or a drag and drop effect component.

The layeredPane may not be null.

Generally implemented with

getRootPane().setLayeredPane(layeredPane);

**Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the layered pane parameter is null**See Also:**[getLayeredPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getLayeredPane()), [JRootPane.getLayeredPane()](http://docs.google.com/javax/swing/JRootPane.html#getLayeredPane())

### getLayeredPane

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

Returns the layeredPane.

**Returns:**the value of the layeredPane property.**See Also:**[setLayeredPane(javax.swing.JLayeredPane)](http://docs.google.com/javax/swing/RootPaneContainer.html#setLayeredPane(javax.swing.JLayeredPane))

### setGlassPane

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

The glassPane is always the first child of the rootPane and the rootPanes layout manager ensures that it's always as big as the rootPane. By default it's transparent and not visible. It can be used to temporarily grab all keyboard and mouse input by adding listeners and then making it visible. by default it's not visible.

The glassPane may not be null.

Generally implemented with getRootPane().setGlassPane(glassPane);

**See Also:**[getGlassPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getGlassPane()), [JRootPane.setGlassPane(java.awt.Component)](http://docs.google.com/javax/swing/JRootPane.html#setGlassPane(java.awt.Component))

### getGlassPane

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

Returns the glassPane.

**Returns:**the value of the glassPane property.**See Also:**[setGlassPane(java.awt.Component)](http://docs.google.com/javax/swing/RootPaneContainer.html#setGlassPane(java.awt.Component))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RootPaneContainer.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/RepaintManager.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/RowFilter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/RootPaneContainer.html)    [**NO FRAMES**](http://docs.google.com/RootPaneContainer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).