| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JScrollPane.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/JScrollBar.AccessibleJScrollBar.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JScrollPane.AccessibleJScrollPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JScrollPane.html)    [**NO FRAMES**](http://docs.google.com/JScrollPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#2p2csry) | [METHOD](#32hioqz) |

## **javax.swing**

Class JScrollPane

[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.JScrollPane**

**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), [ScrollPaneConstants](http://docs.google.com/javax/swing/ScrollPaneConstants.html)

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

Provides a scrollable view of a lightweight component. A JScrollPane manages a viewport, optional vertical and horizontal scroll bars, and optional row and column heading viewports. You can find task-oriented documentation of JScrollPane in [How to Use Scroll Panes](http://java.sun.com/docs/books/tutorial/uiswing/components/scrollpane.html), a section in *The Java Tutorial*. Note that JScrollPane does not support heavyweight components.

|  |
| --- |

The JViewport provides a window, or "viewport" onto a data source -- for example, a text file. That data source is the "scrollable client" (aka data model) displayed by the JViewport view. A JScrollPane basically consists of JScrollBars, a JViewport, and the wiring between them, as shown in the diagram at right.

In addition to the scroll bars and viewport, a JScrollPane can have a column header and a row header. Each of these is a JViewport object that you specify with setRowHeaderView, and setColumnHeaderView. The column header viewport automatically scrolls left and right, tracking the left-right scrolling of the main viewport. (It never scrolls vertically, however.) The row header acts in a similar fashion.

Where two scroll bars meet, the row header meets the column header, or a scroll bar meets one of the headers, both components stop short of the corner, leaving a rectangular space which is, by default, empty. These spaces can potentially exist in any number of the four corners. In the previous diagram, the top right space is present and identified by the label "corner component".

Any number of these empty spaces can be replaced by using the setCorner method to add a component to a particular corner. (Note: The same component cannot be added to multiple corners.) This is useful if there's some extra decoration or function you'd like to add to the scroll pane. The size of each corner component is entirely determined by the size of the headers and/or scroll bars that surround it.

A corner component will only be visible if there is an empty space in that corner for it to exist in. For example, consider a component set into the top right corner of a scroll pane with a column header. If the scroll pane's vertical scrollbar is not present, perhaps because the view component hasn't grown large enough to require it, then the corner component will not be shown (since there is no empty space in that corner created by the meeting of the header and vertical scroll bar). Forcing the scroll bar to always be shown, using setVerticalScrollBarPolicy(VERTICAL\_SCROLLBAR\_ALWAYS), will ensure that the space for the corner component always exists.

To add a border around the main viewport, you can use setViewportBorder. (Of course, you can also add a border around the whole scroll pane using setBorder.)

A common operation to want to do is to set the background color that will be used if the main viewport view is smaller than the viewport, or is not opaque. This can be accomplished by setting the background color of the viewport, via scrollPane.getViewport().setBackground(). The reason for setting the color of the viewport and not the scrollpane is that by default JViewport is opaque which, among other things, means it will completely fill in its background using its background color. Therefore when JScrollPane draws its background the viewport will usually draw over it.

By default JScrollPane uses ScrollPaneLayout to handle the layout of its child Components. ScrollPaneLayout determines the size to make the viewport view in one of two ways:

1. If the view implements Scrollable a combination of getPreferredScrollableViewportSize, getScrollableTracksViewportWidth and getScrollableTracksViewportHeightis used, otherwise
2. getPreferredSize is used.

**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:**[JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html), [JViewport](http://docs.google.com/javax/swing/JViewport.html), [ScrollPaneLayout](http://docs.google.com/javax/swing/ScrollPaneLayout.html), [Scrollable](http://docs.google.com/javax/swing/Scrollable.html), [Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [setViewportView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component)), [setRowHeaderView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeaderView(java.awt.Component)), [setColumnHeaderView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeaderView(java.awt.Component)), [setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component)), [setViewportBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportBorder(javax.swing.border.Border))

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JScrollPane.AccessibleJScrollPane**](http://docs.google.com/javax/swing/JScrollPane.AccessibleJScrollPane.html)            This class implements accessibility support for the JScrollPane class. |
| protected  class | [**JScrollPane.ScrollBar**](http://docs.google.com/javax/swing/JScrollPane.ScrollBar.html)            By default JScrollPane creates scrollbars that are instances of this class. |

| **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** | |
| --- | --- |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**columnHeader**](http://docs.google.com/javax/swing/JScrollPane.html#columnHeader)            The column header child. |
| protected  [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**horizontalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#horizontalScrollBar)            The scrollpane's horizontal scrollbar child. |
| protected  int | [**horizontalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#horizontalScrollBarPolicy)            The display policy for the horizontal scrollbar. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**lowerLeft**](http://docs.google.com/javax/swing/JScrollPane.html#lowerLeft)            The component to display in the lower left corner. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**lowerRight**](http://docs.google.com/javax/swing/JScrollPane.html#lowerRight)            The component to display in the lower right corner. |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**rowHeader**](http://docs.google.com/javax/swing/JScrollPane.html#rowHeader)            The row header child. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**upperLeft**](http://docs.google.com/javax/swing/JScrollPane.html#upperLeft)            The component to display in the upper left corner. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**upperRight**](http://docs.google.com/javax/swing/JScrollPane.html#upperRight)            The component to display in the upper right corner. |
| protected  [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**verticalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#verticalScrollBar)            The scrollpane's vertical scrollbar child. |
| protected  int | [**verticalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#verticalScrollBarPolicy)            The display policy for the vertical scrollbar. |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**viewport**](http://docs.google.com/javax/swing/JScrollPane.html#viewport)            The scrollpane's viewport child. |

| **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 javax.swing.**[**ScrollPaneConstants**](http://docs.google.com/javax/swing/ScrollPaneConstants.html) |
| --- |
| [COLUMN\_HEADER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#COLUMN_HEADER), [HORIZONTAL\_SCROLLBAR](http://docs.google.com/javax/swing/ScrollPaneConstants.html#HORIZONTAL_SCROLLBAR), [HORIZONTAL\_SCROLLBAR\_ALWAYS](http://docs.google.com/javax/swing/ScrollPaneConstants.html#HORIZONTAL_SCROLLBAR_ALWAYS), [HORIZONTAL\_SCROLLBAR\_AS\_NEEDED](http://docs.google.com/javax/swing/ScrollPaneConstants.html#HORIZONTAL_SCROLLBAR_AS_NEEDED), [HORIZONTAL\_SCROLLBAR\_NEVER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#HORIZONTAL_SCROLLBAR_NEVER), [HORIZONTAL\_SCROLLBAR\_POLICY](http://docs.google.com/javax/swing/ScrollPaneConstants.html#HORIZONTAL_SCROLLBAR_POLICY), [LOWER\_LEADING\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#LOWER_LEADING_CORNER), [LOWER\_LEFT\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#LOWER_LEFT_CORNER), [LOWER\_RIGHT\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#LOWER_RIGHT_CORNER), [LOWER\_TRAILING\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#LOWER_TRAILING_CORNER), [ROW\_HEADER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#ROW_HEADER), [UPPER\_LEADING\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#UPPER_LEADING_CORNER), [UPPER\_LEFT\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#UPPER_LEFT_CORNER), [UPPER\_RIGHT\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#UPPER_RIGHT_CORNER), [UPPER\_TRAILING\_CORNER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#UPPER_TRAILING_CORNER), [VERTICAL\_SCROLLBAR](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VERTICAL_SCROLLBAR), [VERTICAL\_SCROLLBAR\_ALWAYS](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VERTICAL_SCROLLBAR_ALWAYS), [VERTICAL\_SCROLLBAR\_AS\_NEEDED](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VERTICAL_SCROLLBAR_AS_NEEDED), [VERTICAL\_SCROLLBAR\_NEVER](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VERTICAL_SCROLLBAR_NEVER), [VERTICAL\_SCROLLBAR\_POLICY](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VERTICAL_SCROLLBAR_POLICY), [VIEWPORT](http://docs.google.com/javax/swing/ScrollPaneConstants.html#VIEWPORT) |

| **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** | |
| --- | --- |
| [**JScrollPane**](http://docs.google.com/javax/swing/JScrollPane.html#JScrollPane())()            Creates an empty (no viewport view) JScrollPane where both horizontal and vertical scrollbars appear when needed. |
| [**JScrollPane**](http://docs.google.com/javax/swing/JScrollPane.html#JScrollPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) view)            Creates a JScrollPane that displays the contents of the specified component, where both horizontal and vertical scrollbars appear whenever the component's contents are larger than the view. |
| [**JScrollPane**](http://docs.google.com/javax/swing/JScrollPane.html#JScrollPane(java.awt.Component,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) view, int vsbPolicy, int hsbPolicy)            Creates a JScrollPane that displays the view component in a viewport whose view position can be controlled with a pair of scrollbars. |
| [**JScrollPane**](http://docs.google.com/javax/swing/JScrollPane.html#JScrollPane(int,%20int))(int vsbPolicy, int hsbPolicy)            Creates an empty (no viewport view) JScrollPane with specified scrollbar policies. |

| **Method Summary** | |
| --- | --- |
| [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**createHorizontalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#createHorizontalScrollBar())()            Returns a JScrollPane.ScrollBar by default. |
| [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**createVerticalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#createVerticalScrollBar())()            Returns a JScrollPane.ScrollBar by default. |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**createViewport**](http://docs.google.com/javax/swing/JScrollPane.html#createViewport())()            Returns a new JViewport by default. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JScrollPane.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JScrollPane. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getColumnHeader**](http://docs.google.com/javax/swing/JScrollPane.html#getColumnHeader())()            Returns the column header. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getCorner**](http://docs.google.com/javax/swing/JScrollPane.html#getCorner(java.lang.String))([String](http://docs.google.com/java/lang/String.html) key)            Returns the component at the specified corner. |
| [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**getHorizontalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#getHorizontalScrollBar())()            Returns the horizontal scroll bar that controls the viewport's horizontal view position. |
| int | [**getHorizontalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#getHorizontalScrollBarPolicy())()            Returns the horizontal scroll bar policy value. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getRowHeader**](http://docs.google.com/javax/swing/JScrollPane.html#getRowHeader())()            Returns the row header. |
| [ScrollPaneUI](http://docs.google.com/javax/swing/plaf/ScrollPaneUI.html) | [**getUI**](http://docs.google.com/javax/swing/JScrollPane.html#getUI())()            Returns the look and feel (L&F) object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JScrollPane.html#getUIClassID())()            Returns the suffix used to construct the name of the L&F class used to render this component. |
| [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**getVerticalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#getVerticalScrollBar())()            Returns the vertical scroll bar that controls the viewports vertical view position. |
| int | [**getVerticalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#getVerticalScrollBarPolicy())()            Returns the vertical scroll bar policy value. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getViewport**](http://docs.google.com/javax/swing/JScrollPane.html#getViewport())()            Returns the current JViewport. |
| [Border](http://docs.google.com/javax/swing/border/Border.html) | [**getViewportBorder**](http://docs.google.com/javax/swing/JScrollPane.html#getViewportBorder())()            Returns the Border object that surrounds the viewport. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getViewportBorderBounds**](http://docs.google.com/javax/swing/JScrollPane.html#getViewportBorderBounds())()            Returns the bounds of the viewport's border. |
| boolean | [**isValidateRoot**](http://docs.google.com/javax/swing/JScrollPane.html#isValidateRoot())()            Overridden to return true so that any calls to revalidate on any descendants of this JScrollPane will cause the entire tree beginning with this JScrollPane to be validated. |
| boolean | [**isWheelScrollingEnabled**](http://docs.google.com/javax/swing/JScrollPane.html#isWheelScrollingEnabled())()            Indicates whether or not scrolling will take place in response to the mouse wheel. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JScrollPane.html#paramString())()            Returns a string representation of this JScrollPane. |
| void | [**setColumnHeader**](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeader(javax.swing.JViewport))([JViewport](http://docs.google.com/javax/swing/JViewport.html) columnHeader)            Removes the old columnHeader, if it exists; if the new columnHeader isn't null, syncs the x coordinate of its viewPosition with the viewport (if there is one) and then adds it to the scroll pane. |
| void | [**setColumnHeaderView**](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeaderView(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) view)            Creates a column-header viewport if necessary, sets its view, and then adds the column-header viewport to the scrollpane. |
| void | [**setComponentOrientation**](http://docs.google.com/javax/swing/JScrollPane.html#setComponentOrientation(java.awt.ComponentOrientation))([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) co)            Sets the orientation for the vertical and horizontal scrollbars as determined by the ComponentOrientation argument. |
| void | [**setCorner**](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) key, [Component](http://docs.google.com/java/awt/Component.html) corner)            Adds a child that will appear in one of the scroll panes corners, if there's room. |
| void | [**setHorizontalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBar(javax.swing.JScrollBar))([JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) horizontalScrollBar)            Adds the scrollbar that controls the viewport's horizontal view position to the scrollpane. |
| void | [**setHorizontalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBarPolicy(int))(int policy)            Determines when the horizontal scrollbar appears in the scrollpane. |
| void | [**setLayout**](http://docs.google.com/javax/swing/JScrollPane.html#setLayout(java.awt.LayoutManager))([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) layout)            Sets the layout manager for this JScrollPane. |
| void | [**setRowHeader**](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeader(javax.swing.JViewport))([JViewport](http://docs.google.com/javax/swing/JViewport.html) rowHeader)            Removes the old rowHeader, if it exists; if the new rowHeader isn't null, syncs the y coordinate of its viewPosition with the viewport (if there is one) and then adds it to the scroll pane. |
| void | [**setRowHeaderView**](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeaderView(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) view)            Creates a row-header viewport if necessary, sets its view and then adds the row-header viewport to the scrollpane. |
| void | [**setUI**](http://docs.google.com/javax/swing/JScrollPane.html#setUI(javax.swing.plaf.ScrollPaneUI))([ScrollPaneUI](http://docs.google.com/javax/swing/plaf/ScrollPaneUI.html) ui)            Sets the ScrollPaneUI object that provides the look and feel (L&F) for this component. |
| void | [**setVerticalScrollBar**](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBar(javax.swing.JScrollBar))([JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) verticalScrollBar)            Adds the scrollbar that controls the viewports vertical view position to the scrollpane. |
| void | [**setVerticalScrollBarPolicy**](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBarPolicy(int))(int policy)            Determines when the vertical scrollbar appears in the scrollpane. |
| void | [**setViewport**](http://docs.google.com/javax/swing/JScrollPane.html#setViewport(javax.swing.JViewport))([JViewport](http://docs.google.com/javax/swing/JViewport.html) viewport)            Removes the old viewport (if there is one); forces the viewPosition of the new viewport to be in the +x,+y quadrant; syncs up the row and column headers (if there are any) with the new viewport; and finally syncs the scrollbars and headers with the new viewport. |
| void | [**setViewportBorder**](http://docs.google.com/javax/swing/JScrollPane.html#setViewportBorder(javax.swing.border.Border))([Border](http://docs.google.com/javax/swing/border/Border.html) viewportBorder)            Adds a border around the viewport. |
| void | [**setViewportView**](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) view)            Creates a viewport if necessary and then sets its view. |
| void | [**setWheelScrollingEnabled**](http://docs.google.com/javax/swing/JScrollPane.html#setWheelScrollingEnabled(boolean))(boolean handleWheel)            Enables/disables scrolling in response to movement of the mouse wheel. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JScrollPane.html#updateUI())()            Replaces the current ScrollPaneUI object with a version from the current default 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)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [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()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [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)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [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)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [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)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [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)), [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)), [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** |
| --- |

### verticalScrollBarPolicy

protected int **verticalScrollBarPolicy**

The display policy for the vertical scrollbar. The default is ScrollPaneConstants.VERTICAL\_SCROLLBAR\_AS\_NEEDED.

**See Also:**[setVerticalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBarPolicy(int))

### horizontalScrollBarPolicy

protected int **horizontalScrollBarPolicy**

The display policy for the horizontal scrollbar. The default is ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED.

**See Also:**[setHorizontalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBarPolicy(int))

### viewport

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

The scrollpane's viewport child. Default is an empty JViewport.

**See Also:**[setViewport(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setViewport(javax.swing.JViewport))

### verticalScrollBar

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

The scrollpane's vertical scrollbar child. Default is a JScrollBar.

**See Also:**[setVerticalScrollBar(javax.swing.JScrollBar)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBar(javax.swing.JScrollBar))

### horizontalScrollBar

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

The scrollpane's horizontal scrollbar child. Default is a JScrollBar.

**See Also:**[setHorizontalScrollBar(javax.swing.JScrollBar)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBar(javax.swing.JScrollBar))

### rowHeader

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

The row header child. Default is null.

**See Also:**[setRowHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeader(javax.swing.JViewport))

### columnHeader

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

The column header child. Default is null.

**See Also:**[setColumnHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeader(javax.swing.JViewport))

### lowerLeft

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

The component to display in the lower left corner. Default is null.

**See Also:**[setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))

### lowerRight

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

The component to display in the lower right corner. Default is null.

**See Also:**[setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))

### upperLeft

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

The component to display in the upper left corner. Default is null.

**See Also:**[setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))

### upperRight

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

The component to display in the upper right corner. Default is null.

**See Also:**[setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))

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

### JScrollPane

public **JScrollPane**([Component](http://docs.google.com/java/awt/Component.html) view,  
 int vsbPolicy,  
 int hsbPolicy)

Creates a JScrollPane that displays the view component in a viewport whose view position can be controlled with a pair of scrollbars. The scrollbar policies specify when the scrollbars are displayed, For example, if vsbPolicy is VERTICAL\_SCROLLBAR\_AS\_NEEDED then the vertical scrollbar only appears if the view doesn't fit vertically. The available policy settings are listed at [setVerticalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBarPolicy(int)) and [setHorizontalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBarPolicy(int)).

**Parameters:**view - the component to display in the scrollpanes viewportvsbPolicy - an integer that specifies the vertical scrollbar policyhsbPolicy - an integer that specifies the horizontal scrollbar policy**See Also:**[setViewportView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component))

### JScrollPane

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

Creates a JScrollPane that displays the contents of the specified component, where both horizontal and vertical scrollbars appear whenever the component's contents are larger than the view.

**Parameters:**view - the component to display in the scrollpane's viewport**See Also:**[setViewportView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component))

### JScrollPane

public **JScrollPane**(int vsbPolicy,  
 int hsbPolicy)

Creates an empty (no viewport view) JScrollPane with specified scrollbar policies. The available policy settings are listed at [setVerticalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBarPolicy(int)) and [setHorizontalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBarPolicy(int)).

**Parameters:**vsbPolicy - an integer that specifies the vertical scrollbar policyhsbPolicy - an integer that specifies the horizontal scrollbar policy**See Also:**[setViewportView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component))

### JScrollPane

public **JScrollPane**()

Creates an empty (no viewport view) JScrollPane where both horizontal and vertical scrollbars appear when needed.

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

### getUI

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

Returns the look and feel (L&F) object that renders this component.

**Returns:**the ScrollPaneUI object that renders this component**See Also:**[setUI(javax.swing.plaf.ScrollPaneUI)](http://docs.google.com/javax/swing/JScrollPane.html#setUI(javax.swing.plaf.ScrollPaneUI))

### setUI

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

Sets the ScrollPaneUI object that provides the look and feel (L&F) for this component.

**Parameters:**ui - the ScrollPaneUI L&F object**See Also:**[getUI()](http://docs.google.com/javax/swing/JScrollPane.html#getUI())

### updateUI

public void **updateUI**()

Replaces the current ScrollPaneUI object with a version from the current default look and feel. To be called when the default look and feel changes.

**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()), [UIManager.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIManager.html#getUI(javax.swing.JComponent))

### getUIClassID

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

Returns the suffix used to construct the name of the L&F class used to render 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 "ScrollPaneUI"**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))

### setLayout

public void **setLayout**([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) layout)

Sets the layout manager for this JScrollPane. This method overrides setLayout in java.awt.Container to ensure that only LayoutManagers which are subclasses of ScrollPaneLayout can be used in a JScrollPane. If layout is non-null, this will invoke syncWithScrollPane on it.

**Overrides:**[setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**layout - the specified layout manager **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if layout is not a ScrollPaneLayout**See Also:**[Container.getLayout()](http://docs.google.com/java/awt/Container.html#getLayout()), [Container.setLayout(java.awt.LayoutManager)](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager))

### isValidateRoot

public boolean **isValidateRoot**()

Overridden to return true so that any calls to revalidate on any descendants of this JScrollPane will cause the entire tree beginning with this JScrollPane to be validated.

**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:**[Container.validate()](http://docs.google.com/java/awt/Container.html#validate()), [JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [JComponent.isValidateRoot()](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot())

### getVerticalScrollBarPolicy

public int **getVerticalScrollBarPolicy**()

Returns the vertical scroll bar policy value.

**Returns:**the verticalScrollBarPolicy property**See Also:**[setVerticalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBarPolicy(int))

### setVerticalScrollBarPolicy

public void **setVerticalScrollBarPolicy**(int policy)

Determines when the vertical scrollbar appears in the scrollpane. Legal values are:

* ScrollPaneConstants.VERTICAL\_SCROLLBAR\_AS\_NEEDED
* ScrollPaneConstants.VERTICAL\_SCROLLBAR\_NEVER
* ScrollPaneConstants.VERTICAL\_SCROLLBAR\_ALWAYS

**Parameters:**policy - one of the three values listed above **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if policy is not one of the legal values shown above**See Also:**[getVerticalScrollBarPolicy()](http://docs.google.com/javax/swing/JScrollPane.html#getVerticalScrollBarPolicy())

### getHorizontalScrollBarPolicy

public int **getHorizontalScrollBarPolicy**()

Returns the horizontal scroll bar policy value.

**Returns:**the horizontalScrollBarPolicy property**See Also:**[setHorizontalScrollBarPolicy(int)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBarPolicy(int))

### setHorizontalScrollBarPolicy

public void **setHorizontalScrollBarPolicy**(int policy)

Determines when the horizontal scrollbar appears in the scrollpane. The options are:

* ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_AS\_NEEDED
* ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_NEVER
* ScrollPaneConstants.HORIZONTAL\_SCROLLBAR\_ALWAYS

**Parameters:**policy - one of the three values listed above **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if policy is not one of the legal values shown above**See Also:**[getHorizontalScrollBarPolicy()](http://docs.google.com/javax/swing/JScrollPane.html#getHorizontalScrollBarPolicy())

### getViewportBorder

public [Border](http://docs.google.com/javax/swing/border/Border.html) **getViewportBorder**()

Returns the Border object that surrounds the viewport.

**Returns:**the viewportBorder property**See Also:**[setViewportBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportBorder(javax.swing.border.Border))

### setViewportBorder

public void **setViewportBorder**([Border](http://docs.google.com/javax/swing/border/Border.html) viewportBorder)

Adds a border around the viewport. Note that the border isn't set on the viewport directly, JViewport doesn't support the JComponent border property. Similarly setting the JScrollPanes viewport doesn't affect the viewportBorder property.

The default value of this property is computed by the look and feel implementation.

**Parameters:**viewportBorder - the border to be added**See Also:**[getViewportBorder()](http://docs.google.com/javax/swing/JScrollPane.html#getViewportBorder()), [setViewport(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setViewport(javax.swing.JViewport))

### getViewportBorderBounds

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

Returns the bounds of the viewport's border.

**Returns:**a Rectangle object specifying the viewport border

### createHorizontalScrollBar

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

Returns a JScrollPane.ScrollBar by default. Subclasses may override this method to force ScrollPaneUI implementations to use a JScrollBar subclass. Used by ScrollPaneUI implementations to create the horizontal scrollbar.

**Returns:**a JScrollBar with a horizontal orientation**See Also:**[JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html)

### getHorizontalScrollBar

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

Returns the horizontal scroll bar that controls the viewport's horizontal view position.

**Returns:**the horizontalScrollBar property**See Also:**[setHorizontalScrollBar(javax.swing.JScrollBar)](http://docs.google.com/javax/swing/JScrollPane.html#setHorizontalScrollBar(javax.swing.JScrollBar))

### setHorizontalScrollBar

public void **setHorizontalScrollBar**([JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) horizontalScrollBar)

Adds the scrollbar that controls the viewport's horizontal view position to the scrollpane. This is usually unnecessary, as JScrollPane creates horizontal and vertical scrollbars by default.

**Parameters:**horizontalScrollBar - the horizontal scrollbar to be added**See Also:**[createHorizontalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#createHorizontalScrollBar()), [getHorizontalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#getHorizontalScrollBar())

### createVerticalScrollBar

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

Returns a JScrollPane.ScrollBar by default. Subclasses may override this method to force ScrollPaneUI implementations to use a JScrollBar subclass. Used by ScrollPaneUI implementations to create the vertical scrollbar.

**Returns:**a JScrollBar with a vertical orientation**See Also:**[JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html)

### getVerticalScrollBar

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

Returns the vertical scroll bar that controls the viewports vertical view position.

**Returns:**the verticalScrollBar property**See Also:**[setVerticalScrollBar(javax.swing.JScrollBar)](http://docs.google.com/javax/swing/JScrollPane.html#setVerticalScrollBar(javax.swing.JScrollBar))

### setVerticalScrollBar

public void **setVerticalScrollBar**([JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) verticalScrollBar)

Adds the scrollbar that controls the viewports vertical view position to the scrollpane. This is usually unnecessary, as JScrollPane creates vertical and horizontal scrollbars by default.

**Parameters:**verticalScrollBar - the new vertical scrollbar to be added**See Also:**[createVerticalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#createVerticalScrollBar()), [getVerticalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#getVerticalScrollBar())

### createViewport

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

Returns a new JViewport by default. Used to create the viewport (as needed) in setViewportView, setRowHeaderView, and setColumnHeaderView. Subclasses may override this method to return a subclass of JViewport.

**Returns:**a new JViewport

### getViewport

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

Returns the current JViewport.

**Returns:**the viewport property**See Also:**[setViewport(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setViewport(javax.swing.JViewport))

### setViewport

public void **setViewport**([JViewport](http://docs.google.com/javax/swing/JViewport.html) viewport)

Removes the old viewport (if there is one); forces the viewPosition of the new viewport to be in the +x,+y quadrant; syncs up the row and column headers (if there are any) with the new viewport; and finally syncs the scrollbars and headers with the new viewport.

Most applications will find it more convenient to use setViewportView to add a viewport and a view to the scrollpane.

**Parameters:**viewport - the new viewport to be used; if viewport is null, the old viewport is still removed and the new viewport is set to null**See Also:**[createViewport()](http://docs.google.com/javax/swing/JScrollPane.html#createViewport()), [getViewport()](http://docs.google.com/javax/swing/JScrollPane.html#getViewport()), [setViewportView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setViewportView(java.awt.Component))

### setViewportView

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

Creates a viewport if necessary and then sets its view. Applications that don't provide the view directly to the JScrollPane constructor should use this method to specify the scrollable child that's going to be displayed in the scrollpane. For example:

JScrollPane scrollpane = new JScrollPane();  
 scrollpane.setViewportView(myBigComponentToScroll);

Applications should not add children directly to the scrollpane.

**Parameters:**view - the component to add to the viewport**See Also:**[setViewport(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setViewport(javax.swing.JViewport)), [JViewport.setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### getRowHeader

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

Returns the row header.

**Returns:**the rowHeader property**See Also:**[setRowHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeader(javax.swing.JViewport))

### setRowHeader

public void **setRowHeader**([JViewport](http://docs.google.com/javax/swing/JViewport.html) rowHeader)

Removes the old rowHeader, if it exists; if the new rowHeader isn't null, syncs the y coordinate of its viewPosition with the viewport (if there is one) and then adds it to the scroll pane.

Most applications will find it more convenient to use setRowHeaderView to add a row header component and its viewport to the scroll pane.

**Parameters:**rowHeader - the new row header to be used; if null the old row header is still removed and the new rowHeader is set to null**See Also:**[getRowHeader()](http://docs.google.com/javax/swing/JScrollPane.html#getRowHeader()), [setRowHeaderView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeaderView(java.awt.Component))

### setRowHeaderView

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

Creates a row-header viewport if necessary, sets its view and then adds the row-header viewport to the scrollpane. For example:

JScrollPane scrollpane = new JScrollPane();  
 scrollpane.setViewportView(myBigComponentToScroll);  
 scrollpane.setRowHeaderView(myBigComponentsRowHeader);

**Parameters:**view - the component to display as the row header**See Also:**[setRowHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setRowHeader(javax.swing.JViewport)), [JViewport.setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### getColumnHeader

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

Returns the column header.

**Returns:**the columnHeader property**See Also:**[setColumnHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeader(javax.swing.JViewport))

### setColumnHeader

public void **setColumnHeader**([JViewport](http://docs.google.com/javax/swing/JViewport.html) columnHeader)

Removes the old columnHeader, if it exists; if the new columnHeader isn't null, syncs the x coordinate of its viewPosition with the viewport (if there is one) and then adds it to the scroll pane.

Most applications will find it more convenient to use setColumnHeaderView to add a column header component and its viewport to the scroll pane.

**See Also:**[getColumnHeader()](http://docs.google.com/javax/swing/JScrollPane.html#getColumnHeader()), [setColumnHeaderView(java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeaderView(java.awt.Component))

### setColumnHeaderView

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

Creates a column-header viewport if necessary, sets its view, and then adds the column-header viewport to the scrollpane. For example:

JScrollPane scrollpane = new JScrollPane();  
 scrollpane.setViewportView(myBigComponentToScroll);  
 scrollpane.setColumnHeaderView(myBigComponentsColumnHeader);

**Parameters:**view - the component to display as the column header**See Also:**[setColumnHeader(javax.swing.JViewport)](http://docs.google.com/javax/swing/JScrollPane.html#setColumnHeader(javax.swing.JViewport)), [JViewport.setView(java.awt.Component)](http://docs.google.com/javax/swing/JViewport.html#setView(java.awt.Component))

### getCorner

public [Component](http://docs.google.com/java/awt/Component.html) **getCorner**([String](http://docs.google.com/java/lang/String.html) key)

Returns the component at the specified corner. The key value specifying the corner is one of:

* ScrollPaneConstants.LOWER\_LEFT\_CORNER
* ScrollPaneConstants.LOWER\_RIGHT\_CORNER
* ScrollPaneConstants.UPPER\_LEFT\_CORNER
* ScrollPaneConstants.UPPER\_RIGHT\_CORNER
* ScrollPaneConstants.LOWER\_LEADING\_CORNER
* ScrollPaneConstants.LOWER\_TRAILING\_CORNER
* ScrollPaneConstants.UPPER\_LEADING\_CORNER
* ScrollPaneConstants.UPPER\_TRAILING\_CORNER

**Parameters:**key - one of the values as shown above **Returns:**the corner component (which may be null) identified by the given key, or null if the key is invalid**See Also:**[setCorner(java.lang.String, java.awt.Component)](http://docs.google.com/javax/swing/JScrollPane.html#setCorner(java.lang.String,%20java.awt.Component))

### setCorner

public void **setCorner**([String](http://docs.google.com/java/lang/String.html) key,  
 [Component](http://docs.google.com/java/awt/Component.html) corner)

Adds a child that will appear in one of the scroll panes corners, if there's room. For example with both scrollbars showing (on the right and bottom edges of the scrollpane) the lower left corner component will be shown in the space between ends of the two scrollbars. Legal values for the **key** are:

* ScrollPaneConstants.LOWER\_LEFT\_CORNER
* ScrollPaneConstants.LOWER\_RIGHT\_CORNER
* ScrollPaneConstants.UPPER\_LEFT\_CORNER
* ScrollPaneConstants.UPPER\_RIGHT\_CORNER
* ScrollPaneConstants.LOWER\_LEADING\_CORNER
* ScrollPaneConstants.LOWER\_TRAILING\_CORNER
* ScrollPaneConstants.UPPER\_LEADING\_CORNER
* ScrollPaneConstants.UPPER\_TRAILING\_CORNER

Although "corner" doesn't match any beans property signature, PropertyChange events are generated with the property name set to the corner key.

**Parameters:**key - identifies which corner the component will appear incorner - one of the following components:

* lowerLeft
* lowerRight
* upperLeft
* upperRight

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if corner key is invalid

### setComponentOrientation

public void **setComponentOrientation**([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) co)

Sets the orientation for the vertical and horizontal scrollbars as determined by the ComponentOrientation argument.

**Overrides:**[setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**co - one of the following values:

* java.awt.ComponentOrientation.LEFT\_TO\_RIGHT
* java.awt.ComponentOrientation.RIGHT\_TO\_LEFT
* java.awt.ComponentOrientation.UNKNOWN

**See Also:**[ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html)

### isWheelScrollingEnabled

public boolean **isWheelScrollingEnabled**()

Indicates whether or not scrolling will take place in response to the mouse wheel. Wheel scrolling is enabled by default.

**Since:** 1.4 **See Also:**[setWheelScrollingEnabled(boolean)](http://docs.google.com/javax/swing/JScrollPane.html#setWheelScrollingEnabled(boolean))

### setWheelScrollingEnabled

public void **setWheelScrollingEnabled**(boolean handleWheel)

Enables/disables scrolling in response to movement of the mouse wheel. Wheel scrolling is enabled by default.

**Parameters:**handleWheel - true if scrolling should be done automatically for a MouseWheelEvent, false otherwise.**Since:** 1.4 **See Also:**[isWheelScrollingEnabled()](http://docs.google.com/javax/swing/JScrollPane.html#isWheelScrollingEnabled()), [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html)

### paramString

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

Returns a string representation of this JScrollPane. 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 JScrollPane.

### getAccessibleContext

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

Gets the AccessibleContext associated with this JScrollPane. For scroll panes, the AccessibleContext takes the form of an AccessibleJScrollPane. A new AccessibleJScrollPane 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 AccessibleJScrollPane that serves as the AccessibleContext of this JScrollPane

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JScrollPane.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/JScrollBar.AccessibleJScrollBar.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JScrollPane.AccessibleJScrollPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JScrollPane.html)    [**NO FRAMES**](http://docs.google.com/JScrollPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#2p2csry) | [METHOD](#32hioqz) |

[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).