| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ScrollPaneLayout.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/ScrollPaneConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ScrollPaneLayout.UIResource.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ScrollPaneLayout.html)    [**NO FRAMES**](http://docs.google.com/ScrollPaneLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#4i7ojhp) | [METHOD](#1ci93xb) |

## **javax.swing**

Class ScrollPaneLayout

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.ScrollPaneLayout**

**All Implemented Interfaces:** [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [ScrollPaneConstants](http://docs.google.com/javax/swing/ScrollPaneConstants.html) **Direct Known Subclasses:** [ScrollPaneLayout.UIResource](http://docs.google.com/javax/swing/ScrollPaneLayout.UIResource.html)

public class **ScrollPaneLayout**extends [Object](http://docs.google.com/java/lang/Object.html)implements [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [ScrollPaneConstants](http://docs.google.com/javax/swing/ScrollPaneConstants.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

The layout manager used by JScrollPane. JScrollPaneLayout is responsible for nine components: a viewport, two scrollbars, a row header, a column header, and four "corner" components.

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

| **Nested Class Summary** | |
| --- | --- |
| static class | [**ScrollPaneLayout.UIResource**](http://docs.google.com/javax/swing/ScrollPaneLayout.UIResource.html)            The UI resource version of ScrollPaneLayout. |

| **Field Summary** | |
| --- | --- |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**colHead**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#colHead)            The column header child. |
| protected  [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**hsb**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#hsb)            The scrollpane's horizontal scrollbar child. |
| protected  int | [**hsbPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#hsbPolicy)            The display policy for the horizontal scrollbar. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**lowerLeft**](http://docs.google.com/javax/swing/ScrollPaneLayout.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/ScrollPaneLayout.html#lowerRight)            The component to display in the lower right corner. |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**rowHead**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#rowHead)            The row header child. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**upperLeft**](http://docs.google.com/javax/swing/ScrollPaneLayout.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/ScrollPaneLayout.html#upperRight)            The component to display in the upper right corner. |
| protected  [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**viewport**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#viewport)            The scrollpane's viewport child. |
| protected  [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**vsb**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#vsb)            The scrollpane's vertical scrollbar child. |
| protected  int | [**vsbPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#vsbPolicy)            The display policy for the vertical scrollbar. |

| **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) |

| **Constructor Summary** | |
| --- | --- |
| [**ScrollPaneLayout**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#ScrollPaneLayout())() |

| **Method Summary** | |
| --- | --- |
| void | [**addLayoutComponent**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#addLayoutComponent(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) s, [Component](http://docs.google.com/java/awt/Component.html) c)            Adds the specified component to the layout. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**addSingletonComponent**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#addSingletonComponent(java.awt.Component,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) oldC, [Component](http://docs.google.com/java/awt/Component.html) newC)            Removes an existing component. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getColumnHeader**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getColumnHeader())()            Returns the JViewport object that is the column header. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getCorner**](http://docs.google.com/javax/swing/ScrollPaneLayout.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/ScrollPaneLayout.html#getHorizontalScrollBar())()            Returns the JScrollBar object that handles horizontal scrolling. |
| int | [**getHorizontalScrollBarPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getHorizontalScrollBarPolicy())()            Returns the horizontal scrollbar-display policy. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getRowHeader**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getRowHeader())()            Returns the JViewport object that is the row header. |
| [JScrollBar](http://docs.google.com/javax/swing/JScrollBar.html) | [**getVerticalScrollBar**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getVerticalScrollBar())()            Returns the JScrollBar object that handles vertical scrolling. |
| int | [**getVerticalScrollBarPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getVerticalScrollBarPolicy())()            Returns the vertical scrollbar-display policy. |
| [JViewport](http://docs.google.com/javax/swing/JViewport.html) | [**getViewport**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getViewport())()            Returns the JViewport object that displays the scrollable contents. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getViewportBorderBounds**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#getViewportBorderBounds(javax.swing.JScrollPane))([JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html) scrollpane)  **Deprecated.** *As of JDK version Swing1.1 replaced by JScrollPane.getViewportBorderBounds().* |
| void | [**layoutContainer**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#layoutContainer(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            Lays out the scrollpane. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumLayoutSize**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#minimumLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            The minimum size of a ScrollPane is the size of the insets plus minimum size of the viewport, plus the scrollpane's viewportBorder insets, plus the minimum size of the visible headers, plus the minimum size of the scrollbars whose displayPolicy isn't NEVER. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredLayoutSize**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#preferredLayoutSize(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) parent)            The preferred size of a ScrollPane is the size of the insets, plus the preferred size of the viewport, plus the preferred size of the visible headers, plus the preferred size of the scrollbars that will appear given the current view and the current scrollbar displayPolicies. |
| void | [**removeLayoutComponent**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#removeLayoutComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Removes the specified component from the layout. |
| void | [**setHorizontalScrollBarPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#setHorizontalScrollBarPolicy(int))(int x)            Sets the horizontal scrollbar-display policy. |
| void | [**setVerticalScrollBarPolicy**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#setVerticalScrollBarPolicy(int))(int x)            Sets the vertical scrollbar-display policy. |
| void | [**syncWithScrollPane**](http://docs.google.com/javax/swing/ScrollPaneLayout.html#syncWithScrollPane(javax.swing.JScrollPane))([JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html) sp)            This method is invoked after the ScrollPaneLayout is set as the LayoutManager of a JScrollPane. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### viewport

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

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

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

### vsb

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

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

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

### hsb

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

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

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

### rowHead

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

The row header child. Default is null.

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

### colHead

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

The column header child. Default is null.

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

### vsbPolicy

protected int **vsbPolicy**

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

This field is obsolete, please use the JScrollPane field instead.

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

### hsbPolicy

protected int **hsbPolicy**

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

This field is obsolete, please use the JScrollPane field instead.

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

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

### ScrollPaneLayout

public **ScrollPaneLayout**()

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

### syncWithScrollPane

public void **syncWithScrollPane**([JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html) sp)

This method is invoked after the ScrollPaneLayout is set as the LayoutManager of a JScrollPane. It initializes all of the internal fields that are ordinarily set by addLayoutComponent. For example:

ScrollPaneLayout mySPLayout = new ScrollPanelLayout() {  
 public void layoutContainer(Container p) {  
 super.layoutContainer(p);  
 // do some extra work here ...  
 }  
 };  
 scrollpane.setLayout(mySPLayout):

### addSingletonComponent

protected [Component](http://docs.google.com/java/awt/Component.html) **addSingletonComponent**([Component](http://docs.google.com/java/awt/Component.html) oldC,  
 [Component](http://docs.google.com/java/awt/Component.html) newC)

Removes an existing component. When a new component, such as the left corner, or vertical scrollbar, is added, the old one, if it exists, must be removed.

This method returns newC. If oldC is not equal to newC and is non-null, it will be removed from its parent.

**Parameters:**oldC - the Component to replacenewC - the Component to add **Returns:**the newC

### addLayoutComponent

public void **addLayoutComponent**([String](http://docs.google.com/java/lang/String.html) s,  
 [Component](http://docs.google.com/java/awt/Component.html) c)

Adds the specified component to the layout. The layout is identified using one of:

* ScrollPaneConstants.VIEWPORT
* ScrollPaneConstants.VERTICAL\_SCROLLBAR
* ScrollPaneConstants.HORIZONTAL\_SCROLLBAR
* ScrollPaneConstants.ROW\_HEADER
* ScrollPaneConstants.COLUMN\_HEADER
* ScrollPaneConstants.LOWER\_LEFT\_CORNER
* ScrollPaneConstants.LOWER\_RIGHT\_CORNER
* ScrollPaneConstants.UPPER\_LEFT\_CORNER
* ScrollPaneConstants.UPPER\_RIGHT\_CORNER

**Specified by:**[addLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#addLayoutComponent(java.lang.String,%20java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**s - the component identifierc - the the component to be added **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if s is an invalid key

### removeLayoutComponent

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

Removes the specified component from the layout.

**Specified by:**[removeLayoutComponent](http://docs.google.com/java/awt/LayoutManager.html#removeLayoutComponent(java.awt.Component)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**c - the component to remove

### getVerticalScrollBarPolicy

public int **getVerticalScrollBarPolicy**()

Returns the vertical scrollbar-display policy.

**Returns:**an integer giving the display policy**See Also:**[setVerticalScrollBarPolicy(int)](http://docs.google.com/javax/swing/ScrollPaneLayout.html#setVerticalScrollBarPolicy(int))

### setVerticalScrollBarPolicy

public void **setVerticalScrollBarPolicy**(int x)

Sets the vertical scrollbar-display policy. The options are:

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

Note: Applications should use the JScrollPane version of this method. It only exists for backwards compatibility with the Swing 1.0.2 (and earlier) versions of this class.

**Parameters:**x - an integer giving the display policy **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if x is an invalid vertical scroll bar policy, as listed above

### getHorizontalScrollBarPolicy

public int **getHorizontalScrollBarPolicy**()

Returns the horizontal scrollbar-display policy.

**Returns:**an integer giving the display policy**See Also:**[setHorizontalScrollBarPolicy(int)](http://docs.google.com/javax/swing/ScrollPaneLayout.html#setHorizontalScrollBarPolicy(int))

### setHorizontalScrollBarPolicy

public void **setHorizontalScrollBarPolicy**(int x)

Sets the horizontal scrollbar-display policy. The options are:

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

Note: Applications should use the JScrollPane version of this method. It only exists for backwards compatibility with the Swing 1.0.2 (and earlier) versions of this class.

**Parameters:**x - an int giving the display policy **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if x is not a valid horizontal scrollbar policy, as listed above

### getViewport

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

Returns the JViewport object that displays the scrollable contents.

**Returns:**the JViewport object that displays the scrollable contents**See Also:**[JScrollPane.getViewport()](http://docs.google.com/javax/swing/JScrollPane.html#getViewport())

### getHorizontalScrollBar

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

Returns the JScrollBar object that handles horizontal scrolling.

**Returns:**the JScrollBar object that handles horizontal scrolling**See Also:**[JScrollPane.getHorizontalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#getHorizontalScrollBar())

### getVerticalScrollBar

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

Returns the JScrollBar object that handles vertical scrolling.

**Returns:**the JScrollBar object that handles vertical scrolling**See Also:**[JScrollPane.getVerticalScrollBar()](http://docs.google.com/javax/swing/JScrollPane.html#getVerticalScrollBar())

### getRowHeader

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

Returns the JViewport object that is the row header.

**Returns:**the JViewport object that is the row header**See Also:**[JScrollPane.getRowHeader()](http://docs.google.com/javax/swing/JScrollPane.html#getRowHeader())

### getColumnHeader

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

Returns the JViewport object that is the column header.

**Returns:**the JViewport object that is the column header**See Also:**[JScrollPane.getColumnHeader()](http://docs.google.com/javax/swing/JScrollPane.html#getColumnHeader())

### 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.

**Parameters:**key - the String specifying the corner **Returns:**the Component at the specified corner, as defined in [ScrollPaneConstants](http://docs.google.com/javax/swing/ScrollPaneConstants.html); if key is not one of the four corners, null is returned**See Also:**[JScrollPane.getCorner(java.lang.String)](http://docs.google.com/javax/swing/JScrollPane.html#getCorner(java.lang.String))

### preferredLayoutSize

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

The preferred size of a ScrollPane is the size of the insets, plus the preferred size of the viewport, plus the preferred size of the visible headers, plus the preferred size of the scrollbars that will appear given the current view and the current scrollbar displayPolicies.

Note that the rowHeader is calculated as part of the preferred width and the colHeader is calculated as part of the preferred size.

**Specified by:**[preferredLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**parent - the Container that will be laid out **Returns:**a Dimension object specifying the preferred size of the viewport and any scrollbars**See Also:**[ViewportLayout](http://docs.google.com/javax/swing/ViewportLayout.html), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### minimumLayoutSize

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

The minimum size of a ScrollPane is the size of the insets plus minimum size of the viewport, plus the scrollpane's viewportBorder insets, plus the minimum size of the visible headers, plus the minimum size of the scrollbars whose displayPolicy isn't NEVER.

**Specified by:**[minimumLayoutSize](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**parent - the Container that will be laid out **Returns:**a Dimension object specifying the minimum size**See Also:**[LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### layoutContainer

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

Lays out the scrollpane. The positioning of components depends on the following constraints:

* The row header, if present and visible, gets its preferred width and the viewport's height.
* The column header, if present and visible, gets its preferred height and the viewport's width.
* If a vertical scrollbar is needed, i.e. if the viewport's extent height is smaller than its view height or if the displayPolicy is ALWAYS, it's treated like the row header with respect to its dimensions and is made visible.
* If a horizontal scrollbar is needed, it is treated like the column header (see the paragraph above regarding the vertical scrollbar).
* If the scrollpane has a non-null viewportBorder, then space is allocated for that.
* The viewport gets the space available after accounting for the previous constraints.
* The corner components, if provided, are aligned with the ends of the scrollbars and headers. If there is a vertical scrollbar, the right corners appear; if there is a horizontal scrollbar, the lower corners appear; a row header gets left corners, and a column header gets upper corners.

**Specified by:**[layoutContainer](http://docs.google.com/java/awt/LayoutManager.html#layoutContainer(java.awt.Container)) in interface [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) **Parameters:**parent - the Container to lay out

### getViewportBorderBounds

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getViewportBorderBounds**([JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html) scrollpane)

**Deprecated.** *As of JDK version Swing1.1 replaced by JScrollPane.getViewportBorderBounds().*

Returns the bounds of the border around the specified scroll pane's viewport.

**Returns:**the size and position of the viewport border

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ScrollPaneLayout.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/ScrollPaneConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ScrollPaneLayout.UIResource.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ScrollPaneLayout.html)    [**NO FRAMES**](http://docs.google.com/ScrollPaneLayout.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#4i7ojhp) | [METHOD](#1ci93xb) |

[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).