| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ScrollPane.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/java/awt/Scrollbar.AccessibleAWTScrollBar.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/ScrollPane.AccessibleAWTScrollPane.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/ScrollPane.html)    [**NO FRAMES**](http://docs.google.com/ScrollPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#35nkun2) | [CONSTR](#z337ya) | [METHOD](#4i7ojhp) |

## **java.awt**

Class ScrollPane

[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)  
 **java.awt.ScrollPane**

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

public class **ScrollPane**extends [Container](http://docs.google.com/java/awt/Container.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

A container class which implements automatic horizontal and/or vertical scrolling for a single child component. The display policy for the scrollbars can be set to:

1. as needed: scrollbars created and shown only when needed by scrollpane
2. always: scrollbars created and always shown by the scrollpane
3. never: scrollbars never created or shown by the scrollpane

The state of the horizontal and vertical scrollbars is represented by two ScrollPaneAdjustable objects (one for each dimension) which implement the Adjustable interface. The API provides methods to access those objects such that the attributes on the Adjustable object (such as unitIncrement, value, etc.) can be manipulated.

Certain adjustable properties (minimum, maximum, blockIncrement, and visibleAmount) are set internally by the scrollpane in accordance with the geometry of the scrollpane and its child and these should not be set by programs using the scrollpane.

If the scrollbar display policy is defined as "never", then the scrollpane can still be programmatically scrolled using the setScrollPosition() method and the scrollpane will move and clip the child's contents appropriately. This policy is useful if the program needs to create and manage its own adjustable controls.

The placement of the scrollbars is controlled by platform-specific properties set by the user outside of the program.

The initial size of this container is set to 100x100, but can be reset using setSize().

Scrolling with the wheel on a wheel-equipped mouse is enabled by default. This can be disabled using setWheelScrollingEnabled. Wheel scrolling can be customized by setting the block and unit increment of the horizontal and vertical Adjustables. For information on how mouse wheel events are dispatched, see the class description for [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html).

Insets are used to define any space used by scrollbars and any borders created by the scroll pane. getInsets() can be used to get the current value for the insets. If the value of scrollbarsAlwaysVisible is false, then the value of the insets will change dynamically depending on whether the scrollbars are currently visible or not.

**See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.ScrollPane)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**ScrollPane.AccessibleAWTScrollPane**](http://docs.google.com/java/awt/ScrollPane.AccessibleAWTScrollPane.html)            This class implements accessibility support for the ScrollPane class. |

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

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

| **Field Summary** | |
| --- | --- |
| static int | [**SCROLLBARS\_ALWAYS**](http://docs.google.com/java/awt/ScrollPane.html#SCROLLBARS_ALWAYS)            Specifies that horizontal/vertical scrollbars should always be shown regardless of the respective sizes of the scrollpane and child. |
| static int | [**SCROLLBARS\_AS\_NEEDED**](http://docs.google.com/java/awt/ScrollPane.html#SCROLLBARS_AS_NEEDED)            Specifies that horizontal/vertical scrollbar should be shown only when the size of the child exceeds the size of the scrollpane in the horizontal/vertical dimension. |
| static int | [**SCROLLBARS\_NEVER**](http://docs.google.com/java/awt/ScrollPane.html#SCROLLBARS_NEVER)            Specifies that horizontal/vertical scrollbars should never be shown regardless of the respective sizes of the scrollpane and child. |

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

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

| **Constructor Summary** | |
| --- | --- |
| [**ScrollPane**](http://docs.google.com/java/awt/ScrollPane.html#ScrollPane())()            Create a new scrollpane container with a scrollbar display policy of "as needed". |
| [**ScrollPane**](http://docs.google.com/java/awt/ScrollPane.html#ScrollPane(int))(int scrollbarDisplayPolicy)            Create a new scrollpane container. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addImpl**](http://docs.google.com/java/awt/ScrollPane.html#addImpl(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Adds the specified component to this scroll pane container. |
| void | [**addNotify**](http://docs.google.com/java/awt/ScrollPane.html#addNotify())()            Creates the scroll pane's peer. |
| void | [**doLayout**](http://docs.google.com/java/awt/ScrollPane.html#doLayout())()            Lays out this container by resizing its child to its preferred size. |
| protected  boolean | [**eventTypeEnabled**](http://docs.google.com/java/awt/ScrollPane.html#eventTypeEnabled(int))(int type)            If wheel scrolling is enabled, we return true for MouseWheelEvents |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/ScrollPane.html#getAccessibleContext())()            Gets the AccessibleContext associated with this ScrollPane. |
| [Adjustable](http://docs.google.com/java/awt/Adjustable.html) | [**getHAdjustable**](http://docs.google.com/java/awt/ScrollPane.html#getHAdjustable())()            Returns the ScrollPaneAdjustable object which represents the state of the horizontal scrollbar. |
| int | [**getHScrollbarHeight**](http://docs.google.com/java/awt/ScrollPane.html#getHScrollbarHeight())()            Returns the height that would be occupied by a horizontal scrollbar, which is independent of whether it is currently displayed by the scroll pane or not. |
| int | [**getScrollbarDisplayPolicy**](http://docs.google.com/java/awt/ScrollPane.html#getScrollbarDisplayPolicy())()            Returns the display policy for the scrollbars. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getScrollPosition**](http://docs.google.com/java/awt/ScrollPane.html#getScrollPosition())()            Returns the current x,y position within the child which is displayed at the 0,0 location of the scrolled panel's view port. |
| [Adjustable](http://docs.google.com/java/awt/Adjustable.html) | [**getVAdjustable**](http://docs.google.com/java/awt/ScrollPane.html#getVAdjustable())()            Returns the ScrollPaneAdjustable object which represents the state of the vertical scrollbar. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getViewportSize**](http://docs.google.com/java/awt/ScrollPane.html#getViewportSize())()            Returns the current size of the scroll pane's view port. |
| int | [**getVScrollbarWidth**](http://docs.google.com/java/awt/ScrollPane.html#getVScrollbarWidth())()            Returns the width that would be occupied by a vertical scrollbar, which is independent of whether it is currently displayed by the scroll pane or not. |
| boolean | [**isWheelScrollingEnabled**](http://docs.google.com/java/awt/ScrollPane.html#isWheelScrollingEnabled())()            Indicates whether or not scrolling will take place in response to the mouse wheel. |
| void | [**layout**](http://docs.google.com/java/awt/ScrollPane.html#layout())()  **Deprecated.** *As of JDK version 1.1, replaced by doLayout().* |
| [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/ScrollPane.html#paramString())()            Returns a string representing the state of this ScrollPane. |
| void | [**printComponents**](http://docs.google.com/java/awt/ScrollPane.html#printComponents(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Prints the component in this scroll pane. |
| protected  void | [**processMouseWheelEvent**](http://docs.google.com/java/awt/ScrollPane.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent))([MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html) e)            Process mouse wheel events that are delivered to this ScrollPane by scrolling an appropriate amount. |
| void | [**setLayout**](http://docs.google.com/java/awt/ScrollPane.html#setLayout(java.awt.LayoutManager))([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) mgr)            Sets the layout manager for this container. |
| void | [**setScrollPosition**](http://docs.google.com/java/awt/ScrollPane.html#setScrollPosition(int,%20int))(int x, int y)            Scrolls to the specified position within the child component. |
| void | [**setScrollPosition**](http://docs.google.com/java/awt/ScrollPane.html#setScrollPosition(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Scrolls to the specified position within the child component. |
| void | [**setWheelScrollingEnabled**](http://docs.google.com/java/awt/ScrollPane.html#setWheelScrollingEnabled(boolean))(boolean handleWheel)            Enables/disables scrolling in response to movement of the mouse wheel. |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [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)), [getAlignmentX](http://docs.google.com/java/awt/Container.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Container.html#getAlignmentY()), [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()), [getInsets](http://docs.google.com/java/awt/Container.html#getInsets()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getListeners](http://docs.google.com/java/awt/Container.html#getListeners(java.lang.Class)), [getMaximumSize](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/java/awt/Container.html#getMinimumSize()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [getPreferredSize](http://docs.google.com/java/awt/Container.html#getPreferredSize()), [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()), [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()), [paint](http://docs.google.com/java/awt/Container.html#paint(java.awt.Graphics)), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [print](http://docs.google.com/java/awt/Container.html#print(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)), [removeNotify](http://docs.google.com/java/awt/Container.html#removeNotify()), [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)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setFont](http://docs.google.com/java/awt/Container.html#setFont(java.awt.Font)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [update](http://docs.google.com/java/awt/Container.html#update(java.awt.Graphics)), [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(int,%20int)), [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)), [disable](http://docs.google.com/java/awt/Component.html#disable()), [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()), [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,%20boolean,%20boolean)), [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,%20char,%20char)), [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,%20int,%20int)), [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()), [getBaseline](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle)), [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()), [getFontMetrics](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font)), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHeight](http://docs.google.com/java/awt/Component.html#getHeight()), [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()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point)), [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()), [getSize](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension)), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [getWidth](http://docs.google.com/java/awt/Component.html#getWidth()), [getX](http://docs.google.com/java/awt/Component.html#getX()), [getY](http://docs.google.com/java/awt/Component.html#getY()), [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()), [isDoubleBuffered](http://docs.google.com/java/awt/Component.html#isDoubleBuffered()), [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()), [isOpaque](http://docs.google.com/java/awt/Component.html#isOpaque()), [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)), [printAll](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics)), [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)), [processKeyEvent](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [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)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus()), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean)), [reshape](http://docs.google.com/java/awt/Component.html#reshape(int,%20int,%20int,%20int)), [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)), [setBackground](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setEnabled](http://docs.google.com/java/awt/Component.html#setEnabled(boolean)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setForeground](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color)), [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)), [setMaximumSize](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setPreferredSize](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)), [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)), [setVisible](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [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** |
| --- |

### SCROLLBARS\_AS\_NEEDED

public static final int **SCROLLBARS\_AS\_NEEDED**

Specifies that horizontal/vertical scrollbar should be shown only when the size of the child exceeds the size of the scrollpane in the horizontal/vertical dimension.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.ScrollPane.SCROLLBARS_AS_NEEDED)

### SCROLLBARS\_ALWAYS

public static final int **SCROLLBARS\_ALWAYS**

Specifies that horizontal/vertical scrollbars should always be shown regardless of the respective sizes of the scrollpane and child.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.ScrollPane.SCROLLBARS_ALWAYS)

### SCROLLBARS\_NEVER

public static final int **SCROLLBARS\_NEVER**

Specifies that horizontal/vertical scrollbars should never be shown regardless of the respective sizes of the scrollpane and child.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.ScrollPane.SCROLLBARS_NEVER)

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

### ScrollPane

public **ScrollPane**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Create a new scrollpane container with a scrollbar display policy of "as needed".

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### ScrollPane

public **ScrollPane**(int scrollbarDisplayPolicy)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Create a new scrollpane container.

**Parameters:**scrollbarDisplayPolicy - policy for when scrollbars should be shown **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the specified scrollbar display policy is invalid [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

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

### addImpl

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

Adds the specified component to this scroll pane container. If the scroll pane has an existing child component, that component is removed and the new one is added.

**Overrides:**[addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**comp - the component to be addedconstraints - not applicableindex - position of child component (must be <= 0)**See Also:**[Container.add(Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [Container.add(Component, int)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [Container.add(Component, java.lang.Object)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [LayoutManager2](http://docs.google.com/java/awt/LayoutManager2.html)

### getScrollbarDisplayPolicy

public int **getScrollbarDisplayPolicy**()

Returns the display policy for the scrollbars.

**Returns:**the display policy for the scrollbars

### getViewportSize

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

Returns the current size of the scroll pane's view port.

**Returns:**the size of the view port in pixels

### getHScrollbarHeight

public int **getHScrollbarHeight**()

Returns the height that would be occupied by a horizontal scrollbar, which is independent of whether it is currently displayed by the scroll pane or not.

**Returns:**the height of a horizontal scrollbar in pixels

### getVScrollbarWidth

public int **getVScrollbarWidth**()

Returns the width that would be occupied by a vertical scrollbar, which is independent of whether it is currently displayed by the scroll pane or not.

**Returns:**the width of a vertical scrollbar in pixels

### getVAdjustable

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

Returns the ScrollPaneAdjustable object which represents the state of the vertical scrollbar. The declared return type of this method is Adjustable to maintain backward compatibility.

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

### getHAdjustable

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

Returns the ScrollPaneAdjustable object which represents the state of the horizontal scrollbar. The declared return type of this method is Adjustable to maintain backward compatibility.

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

### setScrollPosition

public void **setScrollPosition**(int x,  
 int y)

Scrolls to the specified position within the child component. A call to this method is only valid if the scroll pane contains a child. Specifying a position outside of the legal scrolling bounds of the child will scroll to the closest legal position. Legal bounds are defined to be the rectangle: x = 0, y = 0, width = (child width - view port width), height = (child height - view port height). This is a convenience method which interfaces with the Adjustable objects which represent the state of the scrollbars.

**Parameters:**x - the x position to scroll toy - the y position to scroll to **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the scrollpane does not contain a child

### setScrollPosition

public void **setScrollPosition**([Point](http://docs.google.com/java/awt/Point.html) p)

Scrolls to the specified position within the child component. A call to this method is only valid if the scroll pane contains a child and the specified position is within legal scrolling bounds of the child. Specifying a position outside of the legal scrolling bounds of the child will scroll to the closest legal position. Legal bounds are defined to be the rectangle: x = 0, y = 0, width = (child width - view port width), height = (child height - view port height). This is a convenience method which interfaces with the Adjustable objects which represent the state of the scrollbars.

**Parameters:**p - the Point representing the position to scroll to

### getScrollPosition

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

Returns the current x,y position within the child which is displayed at the 0,0 location of the scrolled panel's view port. This is a convenience method which interfaces with the adjustable objects which represent the state of the scrollbars.

**Returns:**the coordinate position for the current scroll position **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if the scrollpane does not contain a child

### setLayout

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

Sets the layout manager for this container. This method is overridden to prevent the layout mgr from being set.

**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:**mgr - the specified layout manager**See Also:**[Container.doLayout()](http://docs.google.com/java/awt/Container.html#doLayout()), [Container.getLayout()](http://docs.google.com/java/awt/Container.html#getLayout())

### doLayout

public void **doLayout**()

Lays out this container by resizing its child to its preferred size. If the new preferred size of the child causes the current scroll position to be invalid, the scroll position is set to the closest valid position.

**Overrides:**[doLayout](http://docs.google.com/java/awt/Container.html#doLayout()) in class [Container](http://docs.google.com/java/awt/Container.html) **See Also:**[Component.validate()](http://docs.google.com/java/awt/Component.html#validate())

### layout

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **layout**()

**Deprecated.** *As of JDK version 1.1, replaced by doLayout().*

**Overrides:**[layout](http://docs.google.com/java/awt/Container.html#layout()) in class [Container](http://docs.google.com/java/awt/Container.html)

### printComponents

public void **printComponents**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Prints the component in this scroll pane.

**Overrides:**[printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**g - the specified Graphics window**See Also:**[Component.print(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)), [Component.printAll(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics))

### addNotify

public void **addNotify**()

Creates the scroll pane's peer.

**Overrides:**[addNotify](http://docs.google.com/java/awt/Container.html#addNotify()) in class [Container](http://docs.google.com/java/awt/Container.html) **See Also:**[Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [Container.removeNotify()](http://docs.google.com/java/awt/Container.html#removeNotify())

### paramString

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

Returns a string representing the state of this ScrollPane. 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/java/awt/Container.html#paramString()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**the parameter string of this scroll pane

### processMouseWheelEvent

protected void **processMouseWheelEvent**([MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html) e)

Process mouse wheel events that are delivered to this ScrollPane by scrolling an appropriate amount.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Overrides:**[processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**e - the mouse wheel event**Since:** 1.4 **See Also:**[MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [Component.addMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### eventTypeEnabled

protected boolean **eventTypeEnabled**(int type)

If wheel scrolling is enabled, we return true for MouseWheelEvents

**Since:** 1.4

### 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/java/awt/ScrollPane.html#isWheelScrollingEnabled()), [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.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/java/awt/ScrollPane.html#setWheelScrollingEnabled(boolean))

### getAccessibleContext

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

Gets the AccessibleContext associated with this ScrollPane. For scroll panes, the AccessibleContext takes the form of an AccessibleAWTScrollPane. A new AccessibleAWTScrollPane 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/java/awt/Component.html#getAccessibleContext()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an AccessibleAWTScrollPane that serves as the AccessibleContext of this ScrollPane**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ScrollPane.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/java/awt/Scrollbar.AccessibleAWTScrollBar.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/ScrollPane.AccessibleAWTScrollPane.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/ScrollPane.html)    [**NO FRAMES**](http://docs.google.com/ScrollPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#35nkun2) | [CONSTR](#z337ya) | [METHOD](#4i7ojhp) |

[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).