| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JScrollBar.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/JRootPane.RootLayout.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JScrollBar.AccessibleJScrollBar.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JScrollBar.html)    [**NO FRAMES**](http://docs.google.com/JScrollBar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#1ci93xb) | [METHOD](#3as4poj) |

## **javax.swing**

Class JScrollBar

[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.JScrollBar**

**All Implemented Interfaces:** [Adjustable](http://docs.google.com/java/awt/Adjustable.html), [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) **Direct Known Subclasses:** [JScrollPane.ScrollBar](http://docs.google.com/javax/swing/JScrollPane.ScrollBar.html)

public class **JScrollBar**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [Adjustable](http://docs.google.com/java/awt/Adjustable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

An implementation of a scrollbar. The user positions the knob in the scrollbar to determine the contents of the viewing area. The program typically adjusts the display so that the end of the scrollbar represents the end of the displayable contents, or 100% of the contents. The start of the scrollbar is the beginning of the displayable contents, or 0%. The position of the knob within those bounds then translates to the corresponding percentage of the displayable contents.

Typically, as the position of the knob in the scrollbar changes a corresponding change is made to the position of the JViewport on the underlying view, changing the contents of the JViewport.

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

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JScrollBar.AccessibleJScrollBar**](http://docs.google.com/javax/swing/JScrollBar.AccessibleJScrollBar.html)            This class implements accessibility support for the JScrollBar 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  int | [**blockIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#blockIncrement) |
| protected  [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**model**](http://docs.google.com/javax/swing/JScrollBar.html#model)            The model that represents the scrollbar's minimum, maximum, extent (aka "visibleAmount") and current value. |
| protected  int | [**orientation**](http://docs.google.com/javax/swing/JScrollBar.html#orientation) |
| protected  int | [**unitIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#unitIncrement) |

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

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

| **Fields inherited from interface java.awt.**[**Adjustable**](http://docs.google.com/java/awt/Adjustable.html) |
| --- |
| [HORIZONTAL](http://docs.google.com/java/awt/Adjustable.html#HORIZONTAL), [NO\_ORIENTATION](http://docs.google.com/java/awt/Adjustable.html#NO_ORIENTATION), [VERTICAL](http://docs.google.com/java/awt/Adjustable.html#VERTICAL) |

| **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** | |
| --- | --- |
| [**JScrollBar**](http://docs.google.com/javax/swing/JScrollBar.html#JScrollBar())()            Creates a vertical scrollbar with the following initial values: |
| [**JScrollBar**](http://docs.google.com/javax/swing/JScrollBar.html#JScrollBar(int))(int orientation)            Creates a scrollbar with the specified orientation and the following initial values: |
| [**JScrollBar**](http://docs.google.com/javax/swing/JScrollBar.html#JScrollBar(int,%20int,%20int,%20int,%20int))(int orientation, int value, int extent, int min, int max)            Creates a scrollbar with the specified orientation, value, extent, minimum, and maximum. |

| **Method Summary** | |
| --- | --- |
| void | [**addAdjustmentListener**](http://docs.google.com/javax/swing/JScrollBar.html#addAdjustmentListener(java.awt.event.AdjustmentListener))([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)            Adds an AdjustmentListener. |
| protected  void | [**fireAdjustmentValueChanged**](http://docs.google.com/javax/swing/JScrollBar.html#fireAdjustmentValueChanged(int,%20int,%20int))(int id, int type, int value)            Notify listeners that the scrollbar's model has changed. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JScrollBar.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JScrollBar. |
| [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)[] | [**getAdjustmentListeners**](http://docs.google.com/javax/swing/JScrollBar.html#getAdjustmentListeners())()            Returns an array of all the AdjustmentListeners added to this JScrollBar with addAdjustmentListener(). |
| int | [**getBlockIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#getBlockIncrement())()            For backwards compatibility with java.awt.Scrollbar. |
| int | [**getBlockIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#getBlockIncrement(int))(int direction)            Returns the amount to change the scrollbar's value by, given a block (usually "page") up/down request. |
| int | [**getMaximum**](http://docs.google.com/javax/swing/JScrollBar.html#getMaximum())()            The maximum value of the scrollbar is maximum - extent. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/JScrollBar.html#getMaximumSize())()            The scrollbar is flexible along it's scrolling axis and rigid along the other axis. |
| int | [**getMinimum**](http://docs.google.com/javax/swing/JScrollBar.html#getMinimum())()            Returns the minimum value supported by the scrollbar (usually zero). |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/JScrollBar.html#getMinimumSize())()            The scrollbar is flexible along it's scrolling axis and rigid along the other axis. |
| [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**getModel**](http://docs.google.com/javax/swing/JScrollBar.html#getModel())()            Returns data model that handles the scrollbar's four fundamental properties: minimum, maximum, value, extent. |
| int | [**getOrientation**](http://docs.google.com/javax/swing/JScrollBar.html#getOrientation())()            Returns the component's orientation (horizontal or vertical). |
| [ScrollBarUI](http://docs.google.com/javax/swing/plaf/ScrollBarUI.html) | [**getUI**](http://docs.google.com/javax/swing/JScrollBar.html#getUI())()            Returns the delegate that implements the look and feel for this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JScrollBar.html#getUIClassID())()            Returns the name of the LookAndFeel class for this component. |
| int | [**getUnitIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#getUnitIncrement())()            For backwards compatibility with java.awt.Scrollbar. |
| int | [**getUnitIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#getUnitIncrement(int))(int direction)            Returns the amount to change the scrollbar's value by, given a unit up/down request. |
| int | [**getValue**](http://docs.google.com/javax/swing/JScrollBar.html#getValue())()            Returns the scrollbar's value. |
| boolean | [**getValueIsAdjusting**](http://docs.google.com/javax/swing/JScrollBar.html#getValueIsAdjusting())()            True if the scrollbar knob is being dragged. |
| int | [**getVisibleAmount**](http://docs.google.com/javax/swing/JScrollBar.html#getVisibleAmount())()            Returns the scrollbar's extent, aka its "visibleAmount". |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JScrollBar.html#paramString())()            Returns a string representation of this JScrollBar. |
| void | [**removeAdjustmentListener**](http://docs.google.com/javax/swing/JScrollBar.html#removeAdjustmentListener(java.awt.event.AdjustmentListener))([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)            Removes an AdjustmentEvent listener. |
| void | [**setBlockIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#setBlockIncrement(int))(int blockIncrement)            Sets the blockIncrement property. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/JScrollBar.html#setEnabled(boolean))(boolean x)            Enables the component so that the knob position can be changed. |
| void | [**setMaximum**](http://docs.google.com/javax/swing/JScrollBar.html#setMaximum(int))(int maximum)            Sets the model's maximum property. |
| void | [**setMinimum**](http://docs.google.com/javax/swing/JScrollBar.html#setMinimum(int))(int minimum)            Sets the model's minimum property. |
| void | [**setModel**](http://docs.google.com/javax/swing/JScrollBar.html#setModel(javax.swing.BoundedRangeModel))([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)            Sets the model that handles the scrollbar's four fundamental properties: minimum, maximum, value, extent. |
| void | [**setOrientation**](http://docs.google.com/javax/swing/JScrollBar.html#setOrientation(int))(int orientation)            Set the scrollbar's orientation to either VERTICAL or HORIZONTAL. |
| void | [**setUI**](http://docs.google.com/javax/swing/JScrollBar.html#setUI(javax.swing.plaf.ScrollBarUI))([ScrollBarUI](http://docs.google.com/javax/swing/plaf/ScrollBarUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**setUnitIncrement**](http://docs.google.com/javax/swing/JScrollBar.html#setUnitIncrement(int))(int unitIncrement)            Sets the unitIncrement property. |
| void | [**setValue**](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int))(int value)            Sets the scrollbar's value. |
| void | [**setValueIsAdjusting**](http://docs.google.com/javax/swing/JScrollBar.html#setValueIsAdjusting(boolean))(boolean b)            Sets the model's valueIsAdjusting property. |
| void | [**setValues**](http://docs.google.com/javax/swing/JScrollBar.html#setValues(int,%20int,%20int,%20int))(int newValue, int newExtent, int newMin, int newMax)            Sets the four BoundedRangeModel properties after forcing the arguments to obey the usual constraints: |
| void | [**setVisibleAmount**](http://docs.google.com/javax/swing/JScrollBar.html#setVisibleAmount(int))(int extent)            Set the model's extent property. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JScrollBar.html#updateUI())()            Overrides JComponent.updateUI. |

| **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)), [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()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [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)), [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)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

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

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

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

### model

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

The model that represents the scrollbar's minimum, maximum, extent (aka "visibleAmount") and current value.

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

### orientation

protected int **orientation**

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

### unitIncrement

protected int **unitIncrement**

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

### blockIncrement

protected int **blockIncrement**

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

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

### JScrollBar

public **JScrollBar**(int orientation,  
 int value,  
 int extent,  
 int min,  
 int max)

Creates a scrollbar with the specified orientation, value, extent, minimum, and maximum. The "extent" is the size of the viewable area. It is also known as the "visible amount".

Note: Use setBlockIncrement to set the block increment to a size slightly smaller than the view's extent. That way, when the user jumps the knob to an adjacent position, one or two lines of the original contents remain in view.

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if orientation is not one of VERTICAL, HORIZONTAL**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JScrollBar.html#setOrientation(int)), [setValue(int)](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int)), [setVisibleAmount(int)](http://docs.google.com/javax/swing/JScrollBar.html#setVisibleAmount(int)), [setMinimum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMaximum(int))

### JScrollBar

public **JScrollBar**(int orientation)

Creates a scrollbar with the specified orientation and the following initial values:

minimum = 0   
 maximum = 100   
 value = 0  
 extent = 10

### JScrollBar

public **JScrollBar**()

Creates a vertical scrollbar with the following initial values:

minimum = 0   
 maximum = 100   
 value = 0  
 extent = 10

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

### setUI

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

Sets the L&F object that renders this component.

**Parameters:**ui - the ScrollBarUI L&F object**Since:** 1.4 **See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### getUI

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

Returns the delegate that implements the look and feel for this component.

**See Also:**[JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI))

### updateUI

public void **updateUI**()

Overrides JComponent.updateUI.

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

### getUIClassID

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

Returns the name of the LookAndFeel class for 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:**"ScrollBarUI"**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))

### getOrientation

public int **getOrientation**()

Returns the component's orientation (horizontal or vertical).

**Specified by:**[getOrientation](http://docs.google.com/java/awt/Adjustable.html#getOrientation()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**VERTICAL or HORIZONTAL**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JScrollBar.html#setOrientation(int)), [Adjustable.getOrientation()](http://docs.google.com/java/awt/Adjustable.html#getOrientation())

### setOrientation

public void **setOrientation**(int orientation)

Set the scrollbar's orientation to either VERTICAL or HORIZONTAL.

**Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if orientation is not one of VERTICAL, HORIZONTAL**See Also:**[getOrientation()](http://docs.google.com/javax/swing/JScrollBar.html#getOrientation())

### getModel

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

Returns data model that handles the scrollbar's four fundamental properties: minimum, maximum, value, extent.

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

### setModel

public void **setModel**([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)

Sets the model that handles the scrollbar's four fundamental properties: minimum, maximum, value, extent.

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

### getUnitIncrement

public int **getUnitIncrement**(int direction)

Returns the amount to change the scrollbar's value by, given a unit up/down request. A ScrollBarUI implementation typically calls this method when the user clicks on a scrollbar up/down arrow and uses the result to update the scrollbar's value. Subclasses my override this method to compute a value, e.g. the change required to scroll up or down one (variable height) line text or one row in a table.

The JScrollPane component creates scrollbars (by default) that override this method and delegate to the viewports Scrollable view, if it has one. The Scrollable interface provides a more specialized version of this method.

**Parameters:**direction - is -1 or 1 for up/down respectively **Returns:**the value of the unitIncrement property**See Also:**[setUnitIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#setUnitIncrement(int)), [setValue(int)](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int)), [Scrollable.getScrollableUnitIncrement(java.awt.Rectangle, int, int)](http://docs.google.com/javax/swing/Scrollable.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int))

### setUnitIncrement

public void **setUnitIncrement**(int unitIncrement)

Sets the unitIncrement property.

Note, that if the argument is equal to the value of Integer.MIN\_VALUE, the most look and feels will not provide the scrolling to the right/down.

**Specified by:**[setUnitIncrement](http://docs.google.com/java/awt/Adjustable.html#setUnitIncrement(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**unitIncrement - the unit increment**See Also:**[getUnitIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#getUnitIncrement(int))

### getBlockIncrement

public int **getBlockIncrement**(int direction)

Returns the amount to change the scrollbar's value by, given a block (usually "page") up/down request. A ScrollBarUI implementation typically calls this method when the user clicks above or below the scrollbar "knob" to change the value up or down by large amount. Subclasses my override this method to compute a value, e.g. the change required to scroll up or down one paragraph in a text document.

The JScrollPane component creates scrollbars (by default) that override this method and delegate to the viewports Scrollable view, if it has one. The Scrollable interface provides a more specialized version of this method.

**Parameters:**direction - is -1 or 1 for up/down respectively **Returns:**the value of the blockIncrement property**See Also:**[setBlockIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#setBlockIncrement(int)), [setValue(int)](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int)), [Scrollable.getScrollableBlockIncrement(java.awt.Rectangle, int, int)](http://docs.google.com/javax/swing/Scrollable.html#getScrollableBlockIncrement(java.awt.Rectangle,%20int,%20int))

### setBlockIncrement

public void **setBlockIncrement**(int blockIncrement)

Sets the blockIncrement property.

Note, that if the argument is equal to the value of Integer.MIN\_VALUE, the most look and feels will not provide the scrolling to the right/down.

**Specified by:**[setBlockIncrement](http://docs.google.com/java/awt/Adjustable.html#setBlockIncrement(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**blockIncrement - the block increment**See Also:**[getBlockIncrement()](http://docs.google.com/javax/swing/JScrollBar.html#getBlockIncrement())

### getUnitIncrement

public int **getUnitIncrement**()

For backwards compatibility with java.awt.Scrollbar.

**Specified by:**[getUnitIncrement](http://docs.google.com/java/awt/Adjustable.html#getUnitIncrement()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the unit value increment for the adjustable object**See Also:**[Adjustable.getUnitIncrement()](http://docs.google.com/java/awt/Adjustable.html#getUnitIncrement()), [getUnitIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#getUnitIncrement(int))

### getBlockIncrement

public int **getBlockIncrement**()

For backwards compatibility with java.awt.Scrollbar.

**Specified by:**[getBlockIncrement](http://docs.google.com/java/awt/Adjustable.html#getBlockIncrement()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the block value increment for the adjustable object**See Also:**[Adjustable.getBlockIncrement()](http://docs.google.com/java/awt/Adjustable.html#getBlockIncrement()), [getBlockIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#getBlockIncrement(int))

### getValue

public int **getValue**()

Returns the scrollbar's value.

**Specified by:**[getValue](http://docs.google.com/java/awt/Adjustable.html#getValue()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the model's value property**See Also:**[setValue(int)](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int))

### setValue

public void **setValue**(int value)

Sets the scrollbar's value. This method just forwards the value to the model.

**Specified by:**[setValue](http://docs.google.com/java/awt/Adjustable.html#setValue(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**value - the current value, between minimum and maximum - visibleAmount**See Also:**[getValue()](http://docs.google.com/javax/swing/JScrollBar.html#getValue()), [BoundedRangeModel.setValue(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setValue(int))

### getVisibleAmount

public int **getVisibleAmount**()

Returns the scrollbar's extent, aka its "visibleAmount". In many scrollbar look and feel implementations the size of the scrollbar "knob" or "thumb" is proportional to the extent.

**Specified by:**[getVisibleAmount](http://docs.google.com/java/awt/Adjustable.html#getVisibleAmount()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the value of the model's extent property**See Also:**[setVisibleAmount(int)](http://docs.google.com/javax/swing/JScrollBar.html#setVisibleAmount(int))

### setVisibleAmount

public void **setVisibleAmount**(int extent)

Set the model's extent property.

**Specified by:**[setVisibleAmount](http://docs.google.com/java/awt/Adjustable.html#setVisibleAmount(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**extent - the length of the indicator**See Also:**[getVisibleAmount()](http://docs.google.com/javax/swing/JScrollBar.html#getVisibleAmount()), [BoundedRangeModel.setExtent(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setExtent(int))

### getMinimum

public int **getMinimum**()

Returns the minimum value supported by the scrollbar (usually zero).

**Specified by:**[getMinimum](http://docs.google.com/java/awt/Adjustable.html#getMinimum()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the value of the model's minimum property**See Also:**[setMinimum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMinimum(int))

### setMinimum

public void **setMinimum**(int minimum)

Sets the model's minimum property.

**Specified by:**[setMinimum](http://docs.google.com/java/awt/Adjustable.html#setMinimum(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**minimum - the minimum value**See Also:**[getMinimum()](http://docs.google.com/javax/swing/JScrollBar.html#getMinimum()), [BoundedRangeModel.setMinimum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMinimum(int))

### getMaximum

public int **getMaximum**()

The maximum value of the scrollbar is maximum - extent.

**Specified by:**[getMaximum](http://docs.google.com/java/awt/Adjustable.html#getMaximum()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the value of the model's maximum property**See Also:**[setMaximum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMaximum(int))

### setMaximum

public void **setMaximum**(int maximum)

Sets the model's maximum property. Note that the scrollbar's value can only be set to maximum - extent.

**Specified by:**[setMaximum](http://docs.google.com/java/awt/Adjustable.html#setMaximum(int)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**maximum - the maximum value**See Also:**[getMaximum()](http://docs.google.com/javax/swing/JScrollBar.html#getMaximum()), [BoundedRangeModel.setMaximum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMaximum(int))

### getValueIsAdjusting

public boolean **getValueIsAdjusting**()

True if the scrollbar knob is being dragged.

**Returns:**the value of the model's valueIsAdjusting property**See Also:**[setValueIsAdjusting(boolean)](http://docs.google.com/javax/swing/JScrollBar.html#setValueIsAdjusting(boolean))

### setValueIsAdjusting

public void **setValueIsAdjusting**(boolean b)

Sets the model's valueIsAdjusting property. Scrollbar look and feel implementations should set this property to true when a knob drag begins, and to false when the drag ends. The scrollbar model will not generate ChangeEvents while valueIsAdjusting is true.

**See Also:**[getValueIsAdjusting()](http://docs.google.com/javax/swing/JScrollBar.html#getValueIsAdjusting()), [BoundedRangeModel.setValueIsAdjusting(boolean)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setValueIsAdjusting(boolean))

### setValues

public void **setValues**(int newValue,  
 int newExtent,  
 int newMin,  
 int newMax)

Sets the four BoundedRangeModel properties after forcing the arguments to obey the usual constraints:

minimum <= value <= value+extent <= maximum

**See Also:**[BoundedRangeModel.setRangeProperties(int, int, int, int, boolean)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setRangeProperties(int,%20int,%20int,%20int,%20boolean)), [setValue(int)](http://docs.google.com/javax/swing/JScrollBar.html#setValue(int)), [setVisibleAmount(int)](http://docs.google.com/javax/swing/JScrollBar.html#setVisibleAmount(int)), [setMinimum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/javax/swing/JScrollBar.html#setMaximum(int))

### addAdjustmentListener

public void **addAdjustmentListener**([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)

Adds an AdjustmentListener. Adjustment listeners are notified each time the scrollbar's model changes. Adjustment events are provided for backwards compatibility with java.awt.Scrollbar.

Note that the AdjustmentEvents type property will always have a placeholder value of AdjustmentEvent.TRACK because all changes to a BoundedRangeModels value are considered equivalent. To change the value of a BoundedRangeModel one just sets its value property, i.e. model.setValue(123). No information about the origin of the change, e.g. it's a block decrement, is provided. We don't try fabricate the origin of the change here.

**Specified by:**[addAdjustmentListener](http://docs.google.com/java/awt/Adjustable.html#addAdjustmentListener(java.awt.event.AdjustmentListener)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**l - the AdjustmentLister to add**See Also:**[removeAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/javax/swing/JScrollBar.html#removeAdjustmentListener(java.awt.event.AdjustmentListener)), [BoundedRangeModel.addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/BoundedRangeModel.html#addChangeListener(javax.swing.event.ChangeListener))

### removeAdjustmentListener

public void **removeAdjustmentListener**([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)

Removes an AdjustmentEvent listener.

**Specified by:**[removeAdjustmentListener](http://docs.google.com/java/awt/Adjustable.html#removeAdjustmentListener(java.awt.event.AdjustmentListener)) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Parameters:**l - the AdjustmentLister to remove**See Also:**[addAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/javax/swing/JScrollBar.html#addAdjustmentListener(java.awt.event.AdjustmentListener))

### getAdjustmentListeners

public [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)[] **getAdjustmentListeners**()

Returns an array of all the AdjustmentListeners added to this JScrollBar with addAdjustmentListener().

**Returns:**all of the AdjustmentListeners added or an empty array if no listeners have been added**Since:** 1.4

### fireAdjustmentValueChanged

protected void **fireAdjustmentValueChanged**(int id,  
 int type,  
 int value)

Notify listeners that the scrollbar's model has changed.

**See Also:**[addAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/javax/swing/JScrollBar.html#addAdjustmentListener(java.awt.event.AdjustmentListener)), [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getMinimumSize

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

The scrollbar is flexible along it's scrolling axis and rigid along the other axis.

**Overrides:**[getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the value of the minimumSize property**See Also:**[JComponent.setMinimumSize(java.awt.Dimension)](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)

### getMaximumSize

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

The scrollbar is flexible along it's scrolling axis and rigid along the other axis.

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the value of the maximumSize property**See Also:**[JComponent.setMaximumSize(java.awt.Dimension)](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)

### setEnabled

public void **setEnabled**(boolean x)

Enables the component so that the knob position can be changed. When the disabled, the knob position cannot be changed.

**Overrides:**[setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**x - a boolean value, where true enables the component and false disables it**See Also:**[Component.isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [Component.isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### paramString

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

Returns a string representation of this JScrollBar. 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 JScrollBar.

### getAccessibleContext

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

Gets the AccessibleContext associated with this JScrollBar. For JScrollBar, the AccessibleContext takes the form of an AccessibleJScrollBar. A new AccessibleJScrollBar 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 AccessibleJScrollBar that serves as the AccessibleContext of this JScrollBar

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JScrollBar.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/JRootPane.RootLayout.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JScrollBar.AccessibleJScrollBar.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JScrollBar.html)    [**NO FRAMES**](http://docs.google.com/JScrollBar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#1ci93xb) | [METHOD](#3as4poj) |

[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).