| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Scrollbar.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/Robot.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Scrollbar.AccessibleAWTScrollBar.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Scrollbar.html)    [**NO FRAMES**](http://docs.google.com/Scrollbar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

## **java.awt**

Class Scrollbar

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

**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)

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

The Scrollbar class embodies a scroll bar, a familiar user-interface object. A scroll bar provides a convenient means for allowing a user to select from a range of values. The following three vertical scroll bars could be used as slider controls to pick the red, green, and blue components of a color:



Each scroll bar in this example could be created with code similar to the following:

redSlider=new Scrollbar(Scrollbar.VERTICAL, 0, 1, 0, 255);  
 add(redSlider);

Alternatively, a scroll bar can represent a range of values. For example, if a scroll bar is used for scrolling through text, the width of the "bubble" (also called the "thumb" or "scroll box") can be used to represent the amount of text that is visible. Here is an example of a scroll bar that represents a range:



The value range represented by the bubble in this example is the *visible amount*. The horizontal scroll bar in this example could be created with code like the following:

ranger = new Scrollbar(Scrollbar.HORIZONTAL, 0, 60, 0, 300);  
 add(ranger);

Note that the actual maximum value of the scroll bar is the maximum minus the visible amount. In the previous example, because the maximum is 300 and the visible amount is 60, the actual maximum value is 240. The range of the scrollbar track is 0 - 300. The left side of the bubble indicates the value of the scroll bar.

Normally, the user changes the value of the scroll bar by making a gesture with the mouse. For example, the user can drag the scroll bar's bubble up and down, or click in the scroll bar's unit increment or block increment areas. Keyboard gestures can also be mapped to the scroll bar. By convention, the **Page Up** and **Page Down** keys are equivalent to clicking in the scroll bar's block increment and block decrement areas.

When the user changes the value of the scroll bar, the scroll bar receives an instance of AdjustmentEvent. The scroll bar processes this event, passing it along to any registered listeners.

Any object that wishes to be notified of changes to the scroll bar's value should implement AdjustmentListener, an interface defined in the package java.awt.event. Listeners can be added and removed dynamically by calling the methods addAdjustmentListener and removeAdjustmentListener.

The AdjustmentEvent class defines five types of adjustment event, listed here:

* AdjustmentEvent.TRACK is sent out when the user drags the scroll bar's bubble.
* AdjustmentEvent.UNIT\_INCREMENT is sent out when the user clicks in the left arrow of a horizontal scroll bar, or the top arrow of a vertical scroll bar, or makes the equivalent gesture from the keyboard.
* AdjustmentEvent.UNIT\_DECREMENT is sent out when the user clicks in the right arrow of a horizontal scroll bar, or the bottom arrow of a vertical scroll bar, or makes the equivalent gesture from the keyboard.
* AdjustmentEvent.BLOCK\_INCREMENT is sent out when the user clicks in the track, to the left of the bubble on a horizontal scroll bar, or above the bubble on a vertical scroll bar. By convention, the **Page Up** key is equivalent, if the user is using a keyboard that defines a **Page Up** key.
* AdjustmentEvent.BLOCK\_DECREMENT is sent out when the user clicks in the track, to the right of the bubble on a horizontal scroll bar, or below the bubble on a vertical scroll bar. By convention, the **Page Down** key is equivalent, if the user is using a keyboard that defines a **Page Down** key.

The JDK 1.0 event system is supported for backwards compatibility, but its use with newer versions of the platform is discouraged. The five types of adjustment events introduced with JDK 1.1 correspond to the five event types that are associated with scroll bars in previous platform versions. The following list gives the adjustment event type, and the corresponding JDK 1.0 event type it replaces.

* AdjustmentEvent.TRACK replaces Event.SCROLL\_ABSOLUTE
* AdjustmentEvent.UNIT\_INCREMENT replaces Event.SCROLL\_LINE\_UP
* AdjustmentEvent.UNIT\_DECREMENT replaces Event.SCROLL\_LINE\_DOWN
* AdjustmentEvent.BLOCK\_INCREMENT replaces Event.SCROLL\_PAGE\_UP
* AdjustmentEvent.BLOCK\_DECREMENT replaces Event.SCROLL\_PAGE\_DOWN

**Note**: We recommend using a Scrollbar for value selection only. If you want to implement a scrollable component inside a container, we recommend you use a [ScrollPane](http://docs.google.com/java/awt/ScrollPane.html). If you use a Scrollbar for this purpose, you are likely to encounter issues with painting, key handling, sizing and positioning.

**Since:** JDK1.0 **See Also:**[AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Scrollbar)

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

| **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 | [**HORIZONTAL**](http://docs.google.com/java/awt/Scrollbar.html#HORIZONTAL)            A constant that indicates a horizontal scroll bar. |
| static int | [**VERTICAL**](http://docs.google.com/java/awt/Scrollbar.html#VERTICAL)            A constant that indicates a vertical scroll bar. |

| **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) |
| --- |
| [NO\_ORIENTATION](http://docs.google.com/java/awt/Adjustable.html#NO_ORIENTATION) |

| **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** | |
| --- | --- |
| [**Scrollbar**](http://docs.google.com/java/awt/Scrollbar.html#Scrollbar())()            Constructs a new vertical scroll bar. |
| [**Scrollbar**](http://docs.google.com/java/awt/Scrollbar.html#Scrollbar(int))(int orientation)            Constructs a new scroll bar with the specified orientation. |
| [**Scrollbar**](http://docs.google.com/java/awt/Scrollbar.html#Scrollbar(int,%20int,%20int,%20int,%20int))(int orientation, int value, int visible, int minimum, int maximum)            Constructs a new scroll bar with the specified orientation, initial value, visible amount, and minimum and maximum values. |

| **Method Summary** | |
| --- | --- |
| void | [**addAdjustmentListener**](http://docs.google.com/java/awt/Scrollbar.html#addAdjustmentListener(java.awt.event.AdjustmentListener))([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)            Adds the specified adjustment listener to receive instances of AdjustmentEvent from this scroll bar. |
| void | [**addNotify**](http://docs.google.com/java/awt/Scrollbar.html#addNotify())()            Creates the Scrollbar's peer. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/Scrollbar.html#getAccessibleContext())()            Gets the AccessibleContext associated with this Scrollbar. |
| [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)[] | [**getAdjustmentListeners**](http://docs.google.com/java/awt/Scrollbar.html#getAdjustmentListeners())()            Returns an array of all the adjustment listeners registered on this scrollbar. |
| int | [**getBlockIncrement**](http://docs.google.com/java/awt/Scrollbar.html#getBlockIncrement())()            Gets the block increment of this scroll bar. |
| int | [**getLineIncrement**](http://docs.google.com/java/awt/Scrollbar.html#getLineIncrement())()  **Deprecated.** *As of JDK version 1.1, replaced by getUnitIncrement().* |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/Scrollbar.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this Scrollbar. |
| int | [**getMaximum**](http://docs.google.com/java/awt/Scrollbar.html#getMaximum())()            Gets the maximum value of this scroll bar. |
| int | [**getMinimum**](http://docs.google.com/java/awt/Scrollbar.html#getMinimum())()            Gets the minimum value of this scroll bar. |
| int | [**getOrientation**](http://docs.google.com/java/awt/Scrollbar.html#getOrientation())()            Returns the orientation of this scroll bar. |
| int | [**getPageIncrement**](http://docs.google.com/java/awt/Scrollbar.html#getPageIncrement())()  **Deprecated.** *As of JDK version 1.1, replaced by getBlockIncrement().* |
| int | [**getUnitIncrement**](http://docs.google.com/java/awt/Scrollbar.html#getUnitIncrement())()            Gets the unit increment for this scrollbar. |
| int | [**getValue**](http://docs.google.com/java/awt/Scrollbar.html#getValue())()            Gets the current value of this scroll bar. |
| boolean | [**getValueIsAdjusting**](http://docs.google.com/java/awt/Scrollbar.html#getValueIsAdjusting())()            Returns true if the value is in the process of changing as a result of actions being taken by the user. |
| int | [**getVisible**](http://docs.google.com/java/awt/Scrollbar.html#getVisible())()  **Deprecated.** *As of JDK version 1.1, replaced by getVisibleAmount().* |
| int | [**getVisibleAmount**](http://docs.google.com/java/awt/Scrollbar.html#getVisibleAmount())()            Gets the visible amount of this scroll bar. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Scrollbar.html#paramString())()            Returns a string representing the state of this Scrollbar. |
| protected  void | [**processAdjustmentEvent**](http://docs.google.com/java/awt/Scrollbar.html#processAdjustmentEvent(java.awt.event.AdjustmentEvent))([AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html) e)            Processes adjustment events occurring on this scrollbar by dispatching them to any registered AdjustmentListener objects. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/Scrollbar.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events on this scroll bar. |
| void | [**removeAdjustmentListener**](http://docs.google.com/java/awt/Scrollbar.html#removeAdjustmentListener(java.awt.event.AdjustmentListener))([AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html) l)            Removes the specified adjustment listener so that it no longer receives instances of AdjustmentEvent from this scroll bar. |
| void | [**setBlockIncrement**](http://docs.google.com/java/awt/Scrollbar.html#setBlockIncrement(int))(int v)            Sets the block increment for this scroll bar. |
| void | [**setLineIncrement**](http://docs.google.com/java/awt/Scrollbar.html#setLineIncrement(int))(int v)  **Deprecated.** *As of JDK version 1.1, replaced by setUnitIncrement(int).* |
| void | [**setMaximum**](http://docs.google.com/java/awt/Scrollbar.html#setMaximum(int))(int newMaximum)            Sets the maximum value of this scroll bar. |
| void | [**setMinimum**](http://docs.google.com/java/awt/Scrollbar.html#setMinimum(int))(int newMinimum)            Sets the minimum value of this scroll bar. |
| void | [**setOrientation**](http://docs.google.com/java/awt/Scrollbar.html#setOrientation(int))(int orientation)            Sets the orientation for this scroll bar. |
| void | [**setPageIncrement**](http://docs.google.com/java/awt/Scrollbar.html#setPageIncrement(int))(int v)  **Deprecated.** *As of JDK version 1.1, replaced by setBlockIncrement().* |
| void | [**setUnitIncrement**](http://docs.google.com/java/awt/Scrollbar.html#setUnitIncrement(int))(int v)            Sets the unit increment for this scroll bar. |
| void | [**setValue**](http://docs.google.com/java/awt/Scrollbar.html#setValue(int))(int newValue)            Sets the value of this scroll bar to the specified value. |
| void | [**setValueIsAdjusting**](http://docs.google.com/java/awt/Scrollbar.html#setValueIsAdjusting(boolean))(boolean b)            Sets the valueIsAdjusting property. |
| void | [**setValues**](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int))(int value, int visible, int minimum, int maximum)            Sets the values of four properties for this scroll bar: value, visibleAmount, minimum, and maximum. |
| void | [**setVisibleAmount**](http://docs.google.com/java/awt/Scrollbar.html#setVisibleAmount(int))(int newAmount)            Sets the visible amount of this scroll bar. |

| **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)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Component.html#areFocusTraversalKeysSet(int)), [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)), [deliverEvent](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event)), [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)), [doLayout](http://docs.google.com/java/awt/Component.html#doLayout()), [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)), [getAlignmentX](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [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()), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point)), [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()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int)), [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()), [getMaximumSize](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [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()), [getPreferredSize](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [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)), [invalidate](http://docs.google.com/java/awt/Component.html#invalidate()), [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()), [isFocusCycleRoot](http://docs.google.com/java/awt/Component.html#isFocusCycleRoot(java.awt.Container)), [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)), [layout](http://docs.google.com/java/awt/Component.html#layout()), [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.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Component.html#locate(int,%20int)), [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)), [minimumSize](http://docs.google.com/java/awt/Component.html#minimumSize()), [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()), [paint](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [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)), [preferredSize](http://docs.google.com/java/awt/Component.html#preferredSize()), [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)), [print](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)), [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)), [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)), [removeNotify](http://docs.google.com/java/awt/Component.html#removeNotify()), [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)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setFont](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font)), [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()), [transferFocusBackward](http://docs.google.com/java/awt/Component.html#transferFocusBackward()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()), [update](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics)), [validate](http://docs.google.com/java/awt/Component.html#validate()) |

| **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** |
| --- |

### HORIZONTAL

public static final int **HORIZONTAL**

A constant that indicates a horizontal scroll bar.

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

### VERTICAL

public static final int **VERTICAL**

A constant that indicates a vertical scroll bar.

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

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

### Scrollbar

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

Constructs a new vertical scroll bar. The default properties of the scroll bar are listed in the following table:

| Property | Description | Default Value |
| --- | --- | --- |
| orientation | indicates whether the scroll bar is vertical  or horizontal | Scrollbar.VERTICAL |
| value | value which controls the location  of the scroll bar's bubble | 0 |
| visible amount | visible amount of the scroll bar's range,  typically represented by the size of the  scroll bar's bubble | 10 |
| minimum | minimum value of the scroll bar | 0 |
| maximum | maximum value of the scroll bar | 100 |
| unit increment | amount the value changes when the  Line Up or Line Down key is pressed,  or when the end arrows of the scrollbar  are clicked | 1 |
| block increment | amount the value changes when the  Page Up or Page Down key is pressed,  or when the scrollbar track is clicked  on either side of the bubble | 10 |

**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())

### Scrollbar

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

Constructs a new scroll bar with the specified orientation.

The orientation argument must take one of the two values Scrollbar.HORIZONTAL, or Scrollbar.VERTICAL, indicating a horizontal or vertical scroll bar, respectively.

**Parameters:**orientation - indicates the orientation of the scroll bar **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - when an illegal value for the orientation argument is supplied [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())

### Scrollbar

public **Scrollbar**(int orientation,  
 int value,  
 int visible,  
 int minimum,  
 int maximum)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new scroll bar with the specified orientation, initial value, visible amount, and minimum and maximum values.

The orientation argument must take one of the two values Scrollbar.HORIZONTAL, or Scrollbar.VERTICAL, indicating a horizontal or vertical scroll bar, respectively.

The parameters supplied to this constructor are subject to the constraints described in [setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int)).

**Parameters:**orientation - indicates the orientation of the scroll bar.value - the initial value of the scroll barvisible - the visible amount of the scroll bar, typically represented by the size of the bubbleminimum - the minimum value of the scroll barmaximum - the maximum value of the scroll bar **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - when an illegal value for the orientation argument is supplied [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

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

### addNotify

public void **addNotify**()

Creates the Scrollbar's peer. The peer allows you to modify the appearance of the Scrollbar without changing any of its functionality.

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

### getOrientation

public int **getOrientation**()

Returns the orientation of this scroll bar.

**Specified by:**[getOrientation](http://docs.google.com/java/awt/Adjustable.html#getOrientation()) in interface [Adjustable](http://docs.google.com/java/awt/Adjustable.html) **Returns:**the orientation of this scroll bar, either Scrollbar.HORIZONTAL or Scrollbar.VERTICAL**See Also:**[setOrientation(int)](http://docs.google.com/java/awt/Scrollbar.html#setOrientation(int))

### setOrientation

public void **setOrientation**(int orientation)

Sets the orientation for this scroll bar.

**Parameters:**orientation - the orientation of this scroll bar, either Scrollbar.HORIZONTAL or Scrollbar.VERTICAL **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the value supplied for orientation is not a legal value**Since:** JDK1.1 **See Also:**[getOrientation()](http://docs.google.com/java/awt/Scrollbar.html#getOrientation())

### getValue

public int **getValue**()

Gets the current value of this scroll bar.

**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 current value of this scroll bar**See Also:**[getMinimum()](http://docs.google.com/java/awt/Scrollbar.html#getMinimum()), [getMaximum()](http://docs.google.com/java/awt/Scrollbar.html#getMaximum())

### setValue

public void **setValue**(int newValue)

Sets the value of this scroll bar to the specified value.

If the value supplied is less than the current minimum or greater than the current maximum - visibleAmount, then either minimum or maximum - visibleAmount is substituted, as appropriate.

Normally, a program should change a scroll bar's value only by calling setValues. The setValues method simultaneously and synchronously sets the minimum, maximum, visible amount, and value properties of a scroll bar, so that they are mutually consistent.

Calling this method does not fire an AdjustmentEvent.

**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:**newValue - the new value of the scroll bar**See Also:**[setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int)), [getValue()](http://docs.google.com/java/awt/Scrollbar.html#getValue()), [getMinimum()](http://docs.google.com/java/awt/Scrollbar.html#getMinimum()), [getMaximum()](http://docs.google.com/java/awt/Scrollbar.html#getMaximum())

### getMinimum

public int **getMinimum**()

Gets the minimum value of this scroll bar.

**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 minimum value of this scroll bar**See Also:**[getValue()](http://docs.google.com/java/awt/Scrollbar.html#getValue()), [getMaximum()](http://docs.google.com/java/awt/Scrollbar.html#getMaximum())

### setMinimum

public void **setMinimum**(int newMinimum)

Sets the minimum value of this scroll bar.

When setMinimum is called, the minimum value is changed, and other values (including the maximum, the visible amount, and the current scroll bar value) are changed to be consistent with the new minimum.

Normally, a program should change a scroll bar's minimum value only by calling setValues. The setValues method simultaneously and synchronously sets the minimum, maximum, visible amount, and value properties of a scroll bar, so that they are mutually consistent.

Note that setting the minimum value to Integer.MAX\_VALUE will result in the new minimum value being set to Integer.MAX\_VALUE - 1.

**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:**newMinimum - the new minimum value for this scroll bar**Since:** JDK1.1 **See Also:**[setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int)), [setMaximum(int)](http://docs.google.com/java/awt/Scrollbar.html#setMaximum(int))

### getMaximum

public int **getMaximum**()

Gets the maximum value of this scroll bar.

**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 maximum value of this scroll bar**See Also:**[getValue()](http://docs.google.com/java/awt/Scrollbar.html#getValue()), [getMinimum()](http://docs.google.com/java/awt/Scrollbar.html#getMinimum())

### setMaximum

public void **setMaximum**(int newMaximum)

Sets the maximum value of this scroll bar.

When setMaximum is called, the maximum value is changed, and other values (including the minimum, the visible amount, and the current scroll bar value) are changed to be consistent with the new maximum.

Normally, a program should change a scroll bar's maximum value only by calling setValues. The setValues method simultaneously and synchronously sets the minimum, maximum, visible amount, and value properties of a scroll bar, so that they are mutually consistent.

Note that setting the maximum value to Integer.MIN\_VALUE will result in the new maximum value being set to Integer.MIN\_VALUE + 1.

**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:**newMaximum - the new maximum value for this scroll bar**Since:** JDK1.1 **See Also:**[setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int)), [setMinimum(int)](http://docs.google.com/java/awt/Scrollbar.html#setMinimum(int))

### getVisibleAmount

public int **getVisibleAmount**()

Gets the visible amount of this scroll bar.

When a scroll bar is used to select a range of values, the visible amount is used to represent the range of values that are currently visible. The size of the scroll bar's bubble (also called a thumb or scroll box), usually gives a visual representation of the relationship of the visible amount to the range of the scroll bar.

The scroll bar's bubble may not be displayed when it is not moveable (e.g. when it takes up the entire length of the scroll bar's track, or when the scroll bar is disabled). Whether the bubble is displayed or not will not affect the value returned by getVisibleAmount.

**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 visible amount of this scroll bar**Since:** JDK1.1 **See Also:**[setVisibleAmount(int)](http://docs.google.com/java/awt/Scrollbar.html#setVisibleAmount(int))

### getVisible

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

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

### setVisibleAmount

public void **setVisibleAmount**(int newAmount)

Sets the visible amount of this scroll bar.

When a scroll bar is used to select a range of values, the visible amount is used to represent the range of values that are currently visible. The size of the scroll bar's bubble (also called a thumb or scroll box), usually gives a visual representation of the relationship of the visible amount to the range of the scroll bar.

The scroll bar's bubble may not be displayed when it is not moveable (e.g. when it takes up the entire length of the scroll bar's track, or when the scroll bar is disabled). Whether the bubble is displayed or not will not affect the value returned by getVisibleAmount.

If the visible amount supplied is less than one or greater than the current maximum - minimum, then either one or maximum - minimum is substituted, as appropriate.

Normally, a program should change a scroll bar's value only by calling setValues. The setValues method simultaneously and synchronously sets the minimum, maximum, visible amount, and value properties of a scroll bar, so that they are mutually consistent.

**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:**newAmount - the new visible amount**Since:** JDK1.1 **See Also:**[getVisibleAmount()](http://docs.google.com/java/awt/Scrollbar.html#getVisibleAmount()), [setValues(int, int, int, int)](http://docs.google.com/java/awt/Scrollbar.html#setValues(int,%20int,%20int,%20int))

### setUnitIncrement

public void **setUnitIncrement**(int v)

Sets the unit increment for this scroll bar.

The unit increment is the value that is added or subtracted when the user activates the unit increment area of the scroll bar, generally through a mouse or keyboard gesture that the scroll bar receives as an adjustment event. The unit increment must be greater than zero. Attepts to set the unit increment to a value lower than 1 will result in a value of 1 being set.

**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:**v - the amount by which to increment or decrement the scroll bar's value**Since:** JDK1.1 **See Also:**[getUnitIncrement()](http://docs.google.com/java/awt/Scrollbar.html#getUnitIncrement())

### setLineIncrement

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

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

### getUnitIncrement

public int **getUnitIncrement**()

Gets the unit increment for this scrollbar.

The unit increment is the value that is added or subtracted when the user activates the unit increment area of the scroll bar, generally through a mouse or keyboard gesture that the scroll bar receives as an adjustment event. The unit increment must be greater than zero.

**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 increment of this scroll bar**Since:** JDK1.1 **See Also:**[setUnitIncrement(int)](http://docs.google.com/java/awt/Scrollbar.html#setUnitIncrement(int))

### getLineIncrement

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

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

### setBlockIncrement

public void **setBlockIncrement**(int v)

Sets the block increment for this scroll bar.

The block increment is the value that is added or subtracted when the user activates the block increment area of the scroll bar, generally through a mouse or keyboard gesture that the scroll bar receives as an adjustment event. The block increment must be greater than zero. Attepts to set the block increment to a value lower than 1 will result in a value of 1 being set.

**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:**v - the amount by which to increment or decrement the scroll bar's value**Since:** JDK1.1 **See Also:**[getBlockIncrement()](http://docs.google.com/java/awt/Scrollbar.html#getBlockIncrement())

### setPageIncrement

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

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

### getBlockIncrement

public int **getBlockIncrement**()

Gets the block increment of this scroll bar.

The block increment is the value that is added or subtracted when the user activates the block increment area of the scroll bar, generally through a mouse or keyboard gesture that the scroll bar receives as an adjustment event. The block increment must be greater than zero.

**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 increment of this scroll bar**Since:** JDK1.1 **See Also:**[setBlockIncrement(int)](http://docs.google.com/java/awt/Scrollbar.html#setBlockIncrement(int))

### getPageIncrement

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

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

### setValues

public void **setValues**(int value,  
 int visible,  
 int minimum,  
 int maximum)

Sets the values of four properties for this scroll bar: value, visibleAmount, minimum, and maximum. If the values supplied for these properties are inconsistent or incorrect, they will be changed to ensure consistency.

This method simultaneously and synchronously sets the values of four scroll bar properties, assuring that the values of these properties are mutually consistent. It enforces the following constraints: maximum must be greater than minimum, maximum - minimum must not be greater than Integer.MAX\_VALUE, visibleAmount must be greater than zero. visibleAmount must not be greater than maximum - minimum, value must not be less than minimum, and value must not be greater than maximum - visibleAmount

Calling this method does not fire an AdjustmentEvent.

**Parameters:**value - is the position in the current windowvisible - is the visible amount of the scroll barminimum - is the minimum value of the scroll barmaximum - is the maximum value of the scroll bar**See Also:**[setMinimum(int)](http://docs.google.com/java/awt/Scrollbar.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/java/awt/Scrollbar.html#setMaximum(int)), [setVisibleAmount(int)](http://docs.google.com/java/awt/Scrollbar.html#setVisibleAmount(int)), [setValue(int)](http://docs.google.com/java/awt/Scrollbar.html#setValue(int))

### getValueIsAdjusting

public boolean **getValueIsAdjusting**()

Returns true if the value is in the process of changing as a result of actions being taken by the user.

**Returns:**the value of the valueIsAdjusting property**Since:** 1.4 **See Also:**[setValueIsAdjusting(boolean)](http://docs.google.com/java/awt/Scrollbar.html#setValueIsAdjusting(boolean))

### setValueIsAdjusting

public void **setValueIsAdjusting**(boolean b)

Sets the valueIsAdjusting property.

**Parameters:**b - new adjustment-in-progress status**Since:** 1.4 **See Also:**[getValueIsAdjusting()](http://docs.google.com/java/awt/Scrollbar.html#getValueIsAdjusting())

### addAdjustmentListener

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

Adds the specified adjustment listener to receive instances of AdjustmentEvent from this scroll bar. If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**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 adjustment listener**Since:** JDK1.1 **See Also:**[removeAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/java/awt/Scrollbar.html#removeAdjustmentListener(java.awt.event.AdjustmentListener)), [getAdjustmentListeners()](http://docs.google.com/java/awt/Scrollbar.html#getAdjustmentListeners()), [AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)

### removeAdjustmentListener

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

Removes the specified adjustment listener so that it no longer receives instances of AdjustmentEvent from this scroll bar. If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**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 adjustment listener**Since:** JDK1.1 **See Also:**[addAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/java/awt/Scrollbar.html#addAdjustmentListener(java.awt.event.AdjustmentListener)), [getAdjustmentListeners()](http://docs.google.com/java/awt/Scrollbar.html#getAdjustmentListeners()), [AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)

### getAdjustmentListeners

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

Returns an array of all the adjustment listeners registered on this scrollbar.

**Returns:**all of this scrollbar's AdjustmentListeners or an empty array if no adjustment listeners are currently registered**Since:** 1.4 **See Also:**[addAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/java/awt/Scrollbar.html#addAdjustmentListener(java.awt.event.AdjustmentListener)), [removeAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/java/awt/Scrollbar.html#removeAdjustmentListener(java.awt.event.AdjustmentListener)), [AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this Scrollbar. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a Scrollbar c for its mouse listeners with the following code:

MouseListener[] mls = (MouseListener[])(c.getListeners(MouseListener.class));

If no such listeners exist, this method returns an empty array.

**Overrides:**[getListeners](http://docs.google.com/java/awt/Component.html#getListeners(java.lang.Class)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this component, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[Component.getComponentListeners()](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [Component.getFocusListeners()](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [Component.getHierarchyListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [Component.getHierarchyBoundsListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [Component.getKeyListeners()](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [Component.getMouseListeners()](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [Component.getMouseMotionListeners()](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [Component.getMouseWheelListeners()](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [Component.getInputMethodListeners()](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [Component.getPropertyChangeListeners()](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners())

### processEvent

protected void **processEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)

Processes events on this scroll bar. If the event is an instance of AdjustmentEvent, it invokes the processAdjustmentEvent method. Otherwise, it invokes its superclass's processEvent method.

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

**Overrides:**[processEvent](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**e - the event**Since:** JDK1.1 **See Also:**[AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [processAdjustmentEvent(java.awt.event.AdjustmentEvent)](http://docs.google.com/java/awt/Scrollbar.html#processAdjustmentEvent(java.awt.event.AdjustmentEvent))

### processAdjustmentEvent

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

Processes adjustment events occurring on this scrollbar by dispatching them to any registered AdjustmentListener objects.

This method is not called unless adjustment events are enabled for this component. Adjustment events are enabled when one of the following occurs:

* An AdjustmentListener object is registered via addAdjustmentListener.
* Adjustment events are enabled via enableEvents.

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

**Parameters:**e - the adjustment event**Since:** JDK1.1 **See Also:**[AdjustmentEvent](http://docs.google.com/java/awt/event/AdjustmentEvent.html), [AdjustmentListener](http://docs.google.com/java/awt/event/AdjustmentListener.html), [addAdjustmentListener(java.awt.event.AdjustmentListener)](http://docs.google.com/java/awt/Scrollbar.html#addAdjustmentListener(java.awt.event.AdjustmentListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### paramString

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

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

### getAccessibleContext

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

Gets the AccessibleContext associated with this Scrollbar. For scrollbars, the AccessibleContext takes the form of an AccessibleAWTScrollBar. A new AccessibleAWTScrollBar 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 AccessibleAWTScrollBar that serves as the AccessibleContext of this ScrollBar**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/Scrollbar.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/Robot.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Scrollbar.AccessibleAWTScrollBar.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Scrollbar.html)    [**NO FRAMES**](http://docs.google.com/Scrollbar.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#lnxbz9) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

[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).