| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JSlider.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/JSeparator.AccessibleJSeparator.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JSlider.AccessibleJSlider.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JSlider.html)    [**NO FRAMES**](http://docs.google.com/JSlider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#qsh70q) | [METHOD](#23ckvvd) |

## **javax.swing**

Class JSlider

[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.JSlider**

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

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

A component that lets the user graphically select a value by sliding a knob within a bounded interval.

The slider can show both major tick marks, and minor tick marks between the major ones. The number of values between the tick marks is controlled with setMajorTickSpacing and setMinorTickSpacing. Painting of tick marks is controlled by setPaintTicks.

Sliders can also print text labels at regular intervals (or at arbitrary locations) along the slider track. Painting of labels is controlled by setLabelTable and setPaintLabels.

For further information and examples see [How to Use Sliders](http://java.sun.com/docs/books/tutorial/uiswing/components/slider.html), a section in *The Java Tutorial.*

**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).

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JSlider.AccessibleJSlider**](http://docs.google.com/javax/swing/JSlider.AccessibleJSlider.html)            This class implements accessibility support for the JSlider 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  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | [**changeEvent**](http://docs.google.com/javax/swing/JSlider.html#changeEvent)            Only one ChangeEvent is needed per slider instance since the event's only (read-only) state is the source property. |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**changeListener**](http://docs.google.com/javax/swing/JSlider.html#changeListener)            The changeListener (no suffix) is the listener we add to the slider's model. |
| protected  int | [**majorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#majorTickSpacing)            The number of values between the major tick marks -- the larger marks that break up the minor tick marks. |
| protected  int | [**minorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#minorTickSpacing)            The number of values between the minor tick marks -- the smaller marks that occur between the major tick marks. |
| protected  int | [**orientation**](http://docs.google.com/javax/swing/JSlider.html#orientation)            Whether the slider is horizontal or vertical The default is horizontal. |
| protected  [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**sliderModel**](http://docs.google.com/javax/swing/JSlider.html#sliderModel)            The data model that handles the numeric maximum value, minimum value, and current-position value for the slider. |
| protected  boolean | [**snapToTicks**](http://docs.google.com/javax/swing/JSlider.html#snapToTicks)            If true, the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob. |

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

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

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **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** | |
| --- | --- |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider())()            Creates a horizontal slider with the range 0 to 100 and an initial value of 50. |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider(javax.swing.BoundedRangeModel))([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) brm)            Creates a horizontal slider using the specified BoundedRangeModel. |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider(int))(int orientation)            Creates a slider using the specified orientation with the range 0 to 100 and an initial value of 50. |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider(int,%20int))(int min, int max)            Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max. |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider(int,%20int,%20int))(int min, int max, int value)            Creates a horizontal slider using the specified min, max and value. |
| [**JSlider**](http://docs.google.com/javax/swing/JSlider.html#JSlider(int,%20int,%20int,%20int))(int orientation, int min, int max, int value)            Creates a slider with the specified orientation and the specified minimum, maximum, and initial values. |

| **Method Summary** | |
| --- | --- |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds a ChangeListener to the slider. |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**createChangeListener**](http://docs.google.com/javax/swing/JSlider.html#createChangeListener())()            Subclasses that want to handle ChangeEvents from the model differently can override this to return an instance of a custom ChangeListener implementation. |
| [Hashtable](http://docs.google.com/java/util/Hashtable.html) | [**createStandardLabels**](http://docs.google.com/javax/swing/JSlider.html#createStandardLabels(int))(int increment)            Creates a Hashtable of numerical text labels, starting at the slider minimum, and using the increment specified. |
| [Hashtable](http://docs.google.com/java/util/Hashtable.html) | [**createStandardLabels**](http://docs.google.com/javax/swing/JSlider.html#createStandardLabels(int,%20int))(int increment, int start)            Creates a Hashtable of numerical text labels, starting at the starting point specified, and using the increment specified. |
| protected  void | [**fireStateChanged**](http://docs.google.com/javax/swing/JSlider.html#fireStateChanged())()            Send a ChangeEvent, whose source is this JSlider, to all ChangeListeners that have registered interest in ChangeEvents. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JSlider.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JSlider. |
| [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] | [**getChangeListeners**](http://docs.google.com/javax/swing/JSlider.html#getChangeListeners())()            Returns an array of all the ChangeListeners added to this JSlider with addChangeListener(). |
| int | [**getExtent**](http://docs.google.com/javax/swing/JSlider.html#getExtent())()            Returns the "extent" from the BoundedRangeModel. |
| boolean | [**getInverted**](http://docs.google.com/javax/swing/JSlider.html#getInverted())()            Returns true if the value-range shown for the slider is reversed, |
| [Dictionary](http://docs.google.com/java/util/Dictionary.html) | [**getLabelTable**](http://docs.google.com/javax/swing/JSlider.html#getLabelTable())()            Returns the dictionary of what labels to draw at which values. |
| int | [**getMajorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#getMajorTickSpacing())()            This method returns the major tick spacing. |
| int | [**getMaximum**](http://docs.google.com/javax/swing/JSlider.html#getMaximum())()            Returns the maximum value supported by the slider from the BoundedRangeModel. |
| int | [**getMinimum**](http://docs.google.com/javax/swing/JSlider.html#getMinimum())()            Returns the minimum value supported by the slider from the BoundedRangeModel. |
| int | [**getMinorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#getMinorTickSpacing())()            This method returns the minor tick spacing. |
| [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**getModel**](http://docs.google.com/javax/swing/JSlider.html#getModel())()            Returns the BoundedRangeModel that handles the slider's three fundamental properties: minimum, maximum, value. |
| int | [**getOrientation**](http://docs.google.com/javax/swing/JSlider.html#getOrientation())()            Return this slider's vertical or horizontal orientation. |
| boolean | [**getPaintLabels**](http://docs.google.com/javax/swing/JSlider.html#getPaintLabels())()            Tells if labels are to be painted. |
| boolean | [**getPaintTicks**](http://docs.google.com/javax/swing/JSlider.html#getPaintTicks())()            Tells if tick marks are to be painted. |
| boolean | [**getPaintTrack**](http://docs.google.com/javax/swing/JSlider.html#getPaintTrack())()            Tells if the track (area the slider slides in) is to be painted. |
| boolean | [**getSnapToTicks**](http://docs.google.com/javax/swing/JSlider.html#getSnapToTicks())()            Returns true if the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob. |
| [SliderUI](http://docs.google.com/javax/swing/plaf/SliderUI.html) | [**getUI**](http://docs.google.com/javax/swing/JSlider.html#getUI())()            Gets the UI object which implements the L&F for this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JSlider.html#getUIClassID())()            Returns the name of the L&F class that renders this component. |
| int | [**getValue**](http://docs.google.com/javax/swing/JSlider.html#getValue())()            Returns the slider's current value from the BoundedRangeModel. |
| boolean | [**getValueIsAdjusting**](http://docs.google.com/javax/swing/JSlider.html#getValueIsAdjusting())()            Returns the valueIsAdjusting property from the model. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JSlider.html#paramString())()            Returns a string representation of this JSlider. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/JSlider.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a ChangeListener from the slider. |
| void | [**setExtent**](http://docs.google.com/javax/swing/JSlider.html#setExtent(int))(int extent)            Sets the size of the range "covered" by the knob. |
| void | [**setFont**](http://docs.google.com/javax/swing/JSlider.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) font)            Sets the font for this component. |
| void | [**setInverted**](http://docs.google.com/javax/swing/JSlider.html#setInverted(boolean))(boolean b)            Specify true to reverse the value-range shown for the slider and false to put the value range in the normal order. |
| void | [**setLabelTable**](http://docs.google.com/javax/swing/JSlider.html#setLabelTable(java.util.Dictionary))([Dictionary](http://docs.google.com/java/util/Dictionary.html) labels)            Used to specify what label will be drawn at any given value. |
| void | [**setMajorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#setMajorTickSpacing(int))(int n)            This method sets the major tick spacing. |
| void | [**setMaximum**](http://docs.google.com/javax/swing/JSlider.html#setMaximum(int))(int maximum)            Sets the slider's maximum value to maximum. |
| void | [**setMinimum**](http://docs.google.com/javax/swing/JSlider.html#setMinimum(int))(int minimum)            Sets the slider's minimum value to minimum. |
| void | [**setMinorTickSpacing**](http://docs.google.com/javax/swing/JSlider.html#setMinorTickSpacing(int))(int n)            This method sets the minor tick spacing. |
| void | [**setModel**](http://docs.google.com/javax/swing/JSlider.html#setModel(javax.swing.BoundedRangeModel))([BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) newModel)            Sets the BoundedRangeModel that handles the slider's three fundamental properties: minimum, maximum, value. |
| void | [**setOrientation**](http://docs.google.com/javax/swing/JSlider.html#setOrientation(int))(int orientation)            Set the slider's orientation to either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL. |
| void | [**setPaintLabels**](http://docs.google.com/javax/swing/JSlider.html#setPaintLabels(boolean))(boolean b)            Determines whether labels are painted on the slider. |
| void | [**setPaintTicks**](http://docs.google.com/javax/swing/JSlider.html#setPaintTicks(boolean))(boolean b)            Determines whether tick marks are painted on the slider. |
| void | [**setPaintTrack**](http://docs.google.com/javax/swing/JSlider.html#setPaintTrack(boolean))(boolean b)            Determines whether the track is painted on the slider. |
| void | [**setSnapToTicks**](http://docs.google.com/javax/swing/JSlider.html#setSnapToTicks(boolean))(boolean b)            Specifying true makes the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob. |
| void | [**setUI**](http://docs.google.com/javax/swing/JSlider.html#setUI(javax.swing.plaf.SliderUI))([SliderUI](http://docs.google.com/javax/swing/plaf/SliderUI.html) ui)            Sets the UI object which implements the L&F for this component. |
| void | [**setValue**](http://docs.google.com/javax/swing/JSlider.html#setValue(int))(int n)            Sets the slider's current value to n. |
| void | [**setValueIsAdjusting**](http://docs.google.com/javax/swing/JSlider.html#setValueIsAdjusting(boolean))(boolean b)            Sets the model's valueIsAdjusting property. |
| protected  void | [**updateLabelUIs**](http://docs.google.com/javax/swing/JSlider.html#updateLabelUIs())()            Updates the UIs for the labels in the label table by calling updateUI on each label. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JSlider.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [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)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [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** |
| --- |

### sliderModel

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

The data model that handles the numeric maximum value, minimum value, and current-position value for the slider.

### majorTickSpacing

protected int **majorTickSpacing**

The number of values between the major tick marks -- the larger marks that break up the minor tick marks.

### minorTickSpacing

protected int **minorTickSpacing**

The number of values between the minor tick marks -- the smaller marks that occur between the major tick marks.

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

### snapToTicks

protected boolean **snapToTicks**

If true, the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob. The default is false.

**See Also:**[setSnapToTicks(boolean)](http://docs.google.com/javax/swing/JSlider.html#setSnapToTicks(boolean))

### orientation

protected int **orientation**

Whether the slider is horizontal or vertical The default is horizontal.

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

### changeListener

protected [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) **changeListener**

The changeListener (no suffix) is the listener we add to the slider's model. This listener is initialized to the ChangeListener returned from createChangeListener, which by default just forwards events to ChangeListeners (if any) added directly to the slider.

**See Also:**[addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener)), [createChangeListener()](http://docs.google.com/javax/swing/JSlider.html#createChangeListener())

### changeEvent

protected transient [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) **changeEvent**

Only one ChangeEvent is needed per slider instance since the event's only (read-only) state is the source property. The source of events generated here is always "this". The event is lazily created the first time that an event notification is fired.

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

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

### JSlider

public **JSlider**()

Creates a horizontal slider with the range 0 to 100 and an initial value of 50.

### JSlider

public **JSlider**(int orientation)

Creates a slider using the specified orientation with the range 0 to 100 and an initial value of 50. The orientation can be either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.

**Parameters:**orientation - the orientation of the slider **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/JSlider.html#setOrientation(int))

### JSlider

public **JSlider**(int min,  
 int max)

Creates a horizontal slider using the specified min and max with an initial value equal to the average of the min plus max.

The BoundedRangeModel that holds the slider's data handles any issues that may arise from improperly setting the minimum and maximum values on the slider. See the BoundedRangeModel documentation for details.

**Parameters:**min - the minimum value of the slidermax - the maximum value of the slider**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [setMinimum(int)](http://docs.google.com/javax/swing/JSlider.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/javax/swing/JSlider.html#setMaximum(int))

### JSlider

public **JSlider**(int min,  
 int max,  
 int value)

Creates a horizontal slider using the specified min, max and value.

The BoundedRangeModel that holds the slider's data handles any issues that may arise from improperly setting the minimum, initial, and maximum values on the slider. See the BoundedRangeModel documentation for details.

**Parameters:**min - the minimum value of the slidermax - the maximum value of the slidervalue - the initial value of the slider**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [setMinimum(int)](http://docs.google.com/javax/swing/JSlider.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/javax/swing/JSlider.html#setMaximum(int)), [setValue(int)](http://docs.google.com/javax/swing/JSlider.html#setValue(int))

### JSlider

public **JSlider**(int orientation,  
 int min,  
 int max,  
 int value)

Creates a slider with the specified orientation and the specified minimum, maximum, and initial values. The orientation can be either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.

The BoundedRangeModel that holds the slider's data handles any issues that may arise from improperly setting the minimum, initial, and maximum values on the slider. See the BoundedRangeModel documentation for details.

**Parameters:**orientation - the orientation of the slidermin - the minimum value of the slidermax - the maximum value of the slidervalue - the initial value of the slider **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if orientation is not one of VERTICAL, HORIZONTAL**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html), [setOrientation(int)](http://docs.google.com/javax/swing/JSlider.html#setOrientation(int)), [setMinimum(int)](http://docs.google.com/javax/swing/JSlider.html#setMinimum(int)), [setMaximum(int)](http://docs.google.com/javax/swing/JSlider.html#setMaximum(int)), [setValue(int)](http://docs.google.com/javax/swing/JSlider.html#setValue(int))

### JSlider

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

Creates a horizontal slider using the specified BoundedRangeModel.

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

### getUI

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

Gets the UI object which implements the L&F for this component.

**Returns:**the SliderUI object that implements the Slider L&F

### setUI

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

Sets the UI object which implements the L&F for this component.

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

### updateUI

public void **updateUI**()

Resets the UI property to a value from the current look and feel.

**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 L&F class that renders 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:**"SliderUI"**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))

### createChangeListener

protected [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) **createChangeListener**()

Subclasses that want to handle ChangeEvents from the model differently can override this to return an instance of a custom ChangeListener implementation. The default ChangeListener simply calls the fireStateChanged method to forward ChangeEvents to the ChangeListeners that have been added directly to the slider.

**See Also:**[changeListener](http://docs.google.com/javax/swing/JSlider.html#changeListener), [fireStateChanged()](http://docs.google.com/javax/swing/JSlider.html#fireStateChanged()), [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### addChangeListener

public void **addChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Adds a ChangeListener to the slider.

**Parameters:**l - the ChangeListener to add**See Also:**[fireStateChanged()](http://docs.google.com/javax/swing/JSlider.html#fireStateChanged()), [removeChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#removeChangeListener(javax.swing.event.ChangeListener))

### removeChangeListener

public void **removeChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Removes a ChangeListener from the slider.

**Parameters:**l - the ChangeListener to remove**See Also:**[fireStateChanged()](http://docs.google.com/javax/swing/JSlider.html#fireStateChanged()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener))

### getChangeListeners

public [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] **getChangeListeners**()

Returns an array of all the ChangeListeners added to this JSlider with addChangeListener().

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

### fireStateChanged

protected void **fireStateChanged**()

Send a ChangeEvent, whose source is this JSlider, to all ChangeListeners that have registered interest in ChangeEvents. This method is called each time a ChangeEvent is received from the model.

The event instance is created if necessary, and stored in changeEvent.

**See Also:**[addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener)), [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getModel

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

Returns the BoundedRangeModel that handles the slider's three fundamental properties: minimum, maximum, value.

**Returns:**the data model for this component**See Also:**[setModel(javax.swing.BoundedRangeModel)](http://docs.google.com/javax/swing/JSlider.html#setModel(javax.swing.BoundedRangeModel)), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### setModel

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

Sets the BoundedRangeModel that handles the slider's three fundamental properties: minimum, maximum, value.

Attempts to pass a null model to this method result in undefined behavior, and, most likely, exceptions.

**Parameters:**newModel - the new, non-null BoundedRangeModel to use**See Also:**[getModel()](http://docs.google.com/javax/swing/JSlider.html#getModel()), [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### getValue

public int **getValue**()

Returns the slider's current value from the BoundedRangeModel.

**Returns:**the current value of the slider**See Also:**[setValue(int)](http://docs.google.com/javax/swing/JSlider.html#setValue(int)), [BoundedRangeModel.getValue()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getValue())

### setValue

public void **setValue**(int n)

Sets the slider's current value to n. This method forwards the new value to the model.

The data model (an instance of BoundedRangeModel) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the new value is different from the previous value, all change listeners are notified.

**Parameters:**n - the new value**See Also:**[getValue()](http://docs.google.com/javax/swing/JSlider.html#getValue()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setValue(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setValue(int))

### getMinimum

public int **getMinimum**()

Returns the minimum value supported by the slider from the BoundedRangeModel.

**Returns:**the value of the model's minimum property**See Also:**[setMinimum(int)](http://docs.google.com/javax/swing/JSlider.html#setMinimum(int)), [BoundedRangeModel.getMinimum()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getMinimum())

### setMinimum

public void **setMinimum**(int minimum)

Sets the slider's minimum value to minimum. This method forwards the new minimum value to the model.

The data model (an instance of BoundedRangeModel) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the new minimum value is different from the previous minimum value, all change listeners are notified.

**Parameters:**n - the new minimum**See Also:**[getMinimum()](http://docs.google.com/javax/swing/JSlider.html#getMinimum()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setMinimum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMinimum(int))

### getMaximum

public int **getMaximum**()

Returns the maximum value supported by the slider from the BoundedRangeModel.

**Returns:**the value of the model's maximum property**See Also:**[setMaximum(int)](http://docs.google.com/javax/swing/JSlider.html#setMaximum(int)), [BoundedRangeModel.getMaximum()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getMaximum())

### setMaximum

public void **setMaximum**(int maximum)

Sets the slider's maximum value to maximum. This method forwards the new maximum value to the model.

The data model (an instance of BoundedRangeModel) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the new maximum value is different from the previous maximum value, all change listeners are notified.

**Parameters:**n - the new maximum**See Also:**[getMaximum()](http://docs.google.com/javax/swing/JSlider.html#getMaximum()), [addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/JSlider.html#addChangeListener(javax.swing.event.ChangeListener)), [BoundedRangeModel.setMaximum(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setMaximum(int))

### getValueIsAdjusting

public boolean **getValueIsAdjusting**()

Returns the valueIsAdjusting property from the model. For details on how this is used, see the setValueIsAdjusting documentation.

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

### setValueIsAdjusting

public void **setValueIsAdjusting**(boolean b)

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

**Parameters:**b - the new value for the valueIsAdjusting property**See Also:**[getValueIsAdjusting()](http://docs.google.com/javax/swing/JSlider.html#getValueIsAdjusting()), [BoundedRangeModel.setValueIsAdjusting(boolean)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setValueIsAdjusting(boolean))

### getExtent

public int **getExtent**()

Returns the "extent" from the BoundedRangeModel. This respresents the range of values "covered" by the knob.

**Returns:**an int representing the extent**See Also:**[setExtent(int)](http://docs.google.com/javax/swing/JSlider.html#setExtent(int)), [BoundedRangeModel.getExtent()](http://docs.google.com/javax/swing/BoundedRangeModel.html#getExtent())

### setExtent

public void **setExtent**(int extent)

Sets the size of the range "covered" by the knob. Most look and feel implementations will change the value by this amount if the user clicks on either side of the knob. This method just forwards the new extent value to the model.

The data model (an instance of BoundedRangeModel) handles any mathematical issues arising from assigning faulty values. See the BoundedRangeModel documentation for details.

If the new extent value is different from the previous extent value, all change listeners are notified.

**Parameters:**extent - the new extent**See Also:**[getExtent()](http://docs.google.com/javax/swing/JSlider.html#getExtent()), [BoundedRangeModel.setExtent(int)](http://docs.google.com/javax/swing/BoundedRangeModel.html#setExtent(int))

### getOrientation

public int **getOrientation**()

Return this slider's vertical or horizontal orientation.

**Returns:**SwingConstants.VERTICAL or SwingConstants.HORIZONTAL**See Also:**[setOrientation(int)](http://docs.google.com/javax/swing/JSlider.html#setOrientation(int))

### setOrientation

public void **setOrientation**(int orientation)

Set the slider's orientation to either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.

**Parameters:**orientation - HORIZONTAL or VERTICAL **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/JSlider.html#getOrientation())

### setFont

public void **setFont**([Font](http://docs.google.com/java/awt/Font.html) font)

Sets the font for this component.

**Overrides:**[setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**font - the desired Font for this component**Since:** 1.6 **See Also:**[Component.getFont()](http://docs.google.com/java/awt/Component.html#getFont())

### getLabelTable

public [Dictionary](http://docs.google.com/java/util/Dictionary.html) **getLabelTable**()

Returns the dictionary of what labels to draw at which values.

**Returns:**the Dictionary containing labels and where to draw them

### setLabelTable

public void **setLabelTable**([Dictionary](http://docs.google.com/java/util/Dictionary.html) labels)

Used to specify what label will be drawn at any given value. The key-value pairs are of this format: { Integer value, java.swing.JComponent label }.

An easy way to generate a standard table of value labels is by using the createStandardLabels method.

Once the labels have been set, this method calls [updateLabelUIs()](http://docs.google.com/javax/swing/JSlider.html#updateLabelUIs()). Note that the labels are only painted if the paintLabels property is true.

**Parameters:**labels - new Dictionary of labels, or null to remove all labels**See Also:**[createStandardLabels(int)](http://docs.google.com/javax/swing/JSlider.html#createStandardLabels(int)), [getLabelTable()](http://docs.google.com/javax/swing/JSlider.html#getLabelTable()), [setPaintLabels(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintLabels(boolean))

### updateLabelUIs

protected void **updateLabelUIs**()

Updates the UIs for the labels in the label table by calling updateUI on each label. The UIs are updated from the current look and feel. The labels are also set to their preferred size.

**See Also:**[setLabelTable(java.util.Dictionary)](http://docs.google.com/javax/swing/JSlider.html#setLabelTable(java.util.Dictionary)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### createStandardLabels

public [Hashtable](http://docs.google.com/java/util/Hashtable.html) **createStandardLabels**(int increment)

Creates a Hashtable of numerical text labels, starting at the slider minimum, and using the increment specified. For example, if you call createStandardLabels( 10 ) and the slider minimum is zero, then labels will be created for the values 0, 10, 20, 30, and so on.

For the labels to be drawn on the slider, the returned Hashtable must be passed into setLabelTable, and setPaintLabels must be set to true.

For further details on the makeup of the returned Hashtable, see the setLabelTable documentation.

**Parameters:**increment - distance between labels in the generated hashtable **Returns:**a new Hashtable of labels **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if increment is less than or equal to zero**See Also:**[setLabelTable(java.util.Dictionary)](http://docs.google.com/javax/swing/JSlider.html#setLabelTable(java.util.Dictionary)), [setPaintLabels(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintLabels(boolean))

### createStandardLabels

public [Hashtable](http://docs.google.com/java/util/Hashtable.html) **createStandardLabels**(int increment,  
 int start)

Creates a Hashtable of numerical text labels, starting at the starting point specified, and using the increment specified. For example, if you call createStandardLabels( 10, 2 ), then labels will be created for the values 2, 12, 22, 32, and so on.

For the labels to be drawn on the slider, the returned Hashtable must be passed into setLabelTable, and setPaintLabels must be set to true.

For further details on the makeup of the returned Hashtable, see the setLabelTable documentation.

**Parameters:**increment - distance between labels in the generated hashtablestart - value at which the labels will begin **Returns:**a new Hashtable of labels **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start is out of range, or if increment is less than or equal to zero**See Also:**[setLabelTable(java.util.Dictionary)](http://docs.google.com/javax/swing/JSlider.html#setLabelTable(java.util.Dictionary)), [setPaintLabels(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintLabels(boolean))

### getInverted

public boolean **getInverted**()

Returns true if the value-range shown for the slider is reversed,

**Returns:**true if the slider values are reversed from their normal order**See Also:**[setInverted(boolean)](http://docs.google.com/javax/swing/JSlider.html#setInverted(boolean))

### setInverted

public void **setInverted**(boolean b)

Specify true to reverse the value-range shown for the slider and false to put the value range in the normal order. The order depends on the slider's ComponentOrientation property. Normal (non-inverted) horizontal sliders with a ComponentOrientation value of LEFT\_TO\_RIGHT have their maximum on the right. Normal horizontal sliders with a ComponentOrientation value of RIGHT\_TO\_LEFT have their maximum on the left. Normal vertical sliders have their maximum on the top. These labels are reversed when the slider is inverted.

By default, the value of this property is false.

**Parameters:**b - true to reverse the slider values from their normal order

### getMajorTickSpacing

public int **getMajorTickSpacing**()

This method returns the major tick spacing. The number that is returned represents the distance, measured in values, between each major tick mark. If you have a slider with a range from 0 to 50 and the major tick spacing is set to 10, you will get major ticks next to the following values: 0, 10, 20, 30, 40, 50.

**Returns:**the number of values between major ticks**See Also:**[setMajorTickSpacing(int)](http://docs.google.com/javax/swing/JSlider.html#setMajorTickSpacing(int))

### setMajorTickSpacing

public void **setMajorTickSpacing**(int n)

This method sets the major tick spacing. The number that is passed in represents the distance, measured in values, between each major tick mark. If you have a slider with a range from 0 to 50 and the major tick spacing is set to 10, you will get major ticks next to the following values: 0, 10, 20, 30, 40, 50.

In order for major ticks to be painted, setPaintTicks must be set to true.

This method will also set up a label table for you. If there is not already a label table, and the major tick spacing is > 0, and getPaintLabels returns true, a standard label table will be generated (by calling createStandardLabels) with labels at the major tick marks. For the example above, you would get text labels: "0", "10", "20", "30", "40", "50". The label table is then set on the slider by calling setLabelTable.

**Parameters:**n - new value for the majorTickSpacing property**See Also:**[getMajorTickSpacing()](http://docs.google.com/javax/swing/JSlider.html#getMajorTickSpacing()), [setPaintTicks(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintTicks(boolean)), [setLabelTable(java.util.Dictionary)](http://docs.google.com/javax/swing/JSlider.html#setLabelTable(java.util.Dictionary)), [createStandardLabels(int)](http://docs.google.com/javax/swing/JSlider.html#createStandardLabels(int))

### getMinorTickSpacing

public int **getMinorTickSpacing**()

This method returns the minor tick spacing. The number that is returned represents the distance, measured in values, between each minor tick mark. If you have a slider with a range from 0 to 50 and the minor tick spacing is set to 10, you will get minor ticks next to the following values: 0, 10, 20, 30, 40, 50.

**Returns:**the number of values between minor ticks**See Also:**[getMinorTickSpacing()](http://docs.google.com/javax/swing/JSlider.html#getMinorTickSpacing())

### setMinorTickSpacing

public void **setMinorTickSpacing**(int n)

This method sets the minor tick spacing. The number that is passed in represents the distance, measured in values, between each minor tick mark. If you have a slider with a range from 0 to 50 and the minor tick spacing is set to 10, you will get minor ticks next to the following values: 0, 10, 20, 30, 40, 50.

In order for minor ticks to be painted, setPaintTicks must be set to true.

**Parameters:**n - new value for the minorTickSpacing property**See Also:**[getMinorTickSpacing()](http://docs.google.com/javax/swing/JSlider.html#getMinorTickSpacing()), [setPaintTicks(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintTicks(boolean))

### getSnapToTicks

public boolean **getSnapToTicks**()

Returns true if the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob.

**Returns:**true if the value snaps to the nearest tick mark, else false**See Also:**[setSnapToTicks(boolean)](http://docs.google.com/javax/swing/JSlider.html#setSnapToTicks(boolean))

### setSnapToTicks

public void **setSnapToTicks**(boolean b)

Specifying true makes the knob (and the data value it represents) resolve to the closest tick mark next to where the user positioned the knob. By default, this property is false.

**Parameters:**b - true to snap the knob to the nearest tick mark**See Also:**[getSnapToTicks()](http://docs.google.com/javax/swing/JSlider.html#getSnapToTicks())

### getPaintTicks

public boolean **getPaintTicks**()

Tells if tick marks are to be painted.

**Returns:**true if tick marks are painted, else false**See Also:**[setPaintTicks(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintTicks(boolean))

### setPaintTicks

public void **setPaintTicks**(boolean b)

Determines whether tick marks are painted on the slider. By default, this property is false.

**Parameters:**b - whether or not tick marks should be painted**See Also:**[getPaintTicks()](http://docs.google.com/javax/swing/JSlider.html#getPaintTicks())

### getPaintTrack

public boolean **getPaintTrack**()

Tells if the track (area the slider slides in) is to be painted.

**Returns:**true if track is painted, else false**See Also:**[setPaintTrack(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintTrack(boolean))

### setPaintTrack

public void **setPaintTrack**(boolean b)

Determines whether the track is painted on the slider. By default, this property is true.

**Parameters:**b - whether or not to paint the slider track**See Also:**[getPaintTrack()](http://docs.google.com/javax/swing/JSlider.html#getPaintTrack())

### getPaintLabels

public boolean **getPaintLabels**()

Tells if labels are to be painted.

**Returns:**true if labels are painted, else false**See Also:**[setPaintLabels(boolean)](http://docs.google.com/javax/swing/JSlider.html#setPaintLabels(boolean))

### setPaintLabels

public void **setPaintLabels**(boolean b)

Determines whether labels are painted on the slider.

This method will also set up a label table for you. If there is not already a label table, and the major tick spacing is > 0, a standard label table will be generated (by calling createStandardLabels) with labels at the major tick marks. The label table is then set on the slider by calling setLabelTable.

By default, this property is false.

**Parameters:**b - whether or not to paint labels**See Also:**[getPaintLabels()](http://docs.google.com/javax/swing/JSlider.html#getPaintLabels()), [getLabelTable()](http://docs.google.com/javax/swing/JSlider.html#getLabelTable()), [createStandardLabels(int)](http://docs.google.com/javax/swing/JSlider.html#createStandardLabels(int))

### paramString

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

Returns a string representation of this JSlider. 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 JSlider.

### getAccessibleContext

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

Gets the AccessibleContext associated with this JSlider. For sliders, the AccessibleContext takes the form of an AccessibleJSlider. A new AccessibleJSlider 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 AccessibleJSlider that serves as the AccessibleContext of this JSlider

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JSlider.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/JSeparator.AccessibleJSeparator.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JSlider.AccessibleJSlider.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JSlider.html)    [**NO FRAMES**](http://docs.google.com/JSlider.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#qsh70q) | [METHOD](#23ckvvd) |

[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).