| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSliderUI.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/plaf/basic/BasicSeparatorUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ActionScroller.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSliderUI.html)    [**NO FRAMES**](http://docs.google.com/BasicSliderUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3o7alnk) | [METHOD](#ihv636) |

## **javax.swing.plaf.basic**

Class BasicSliderUI

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.plaf.ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)  
 [javax.swing.plaf.SliderUI](http://docs.google.com/javax/swing/plaf/SliderUI.html)  
 **javax.swing.plaf.basic.BasicSliderUI**

**Direct Known Subclasses:** [MetalSliderUI](http://docs.google.com/javax/swing/plaf/metal/MetalSliderUI.html)

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

A Basic L&F implementation of SliderUI.

| **Nested Class Summary** | |
| --- | --- |
| class | [**BasicSliderUI.ActionScroller**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ActionScroller.html)            As of Java 2 platform v1.3 this undocumented class is no longer used. |
| class | [**BasicSliderUI.ChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ChangeHandler.html)            Data model listener. |
| class | [**BasicSliderUI.ComponentHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ComponentHandler.html)            Listener for resizing events. |
| class | [**BasicSliderUI.FocusHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.FocusHandler.html)            Focus-change listener. |
| class | [**BasicSliderUI.PropertyChangeHandler**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.PropertyChangeHandler.html) |
| class | [**BasicSliderUI.ScrollListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ScrollListener.html)            Scroll-event listener. |
| class | [**BasicSliderUI.TrackListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.TrackListener.html)            Track mouse movements. |

| **Field Summary** | |
| --- | --- |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**changeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#changeListener) |
| protected  [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) | [**componentListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#componentListener) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**contentRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#contentRect) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**focusInsets**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#focusInsets) |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**focusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#focusListener) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**focusRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#focusRect) |
| protected  [Insets](http://docs.google.com/java/awt/Insets.html) | [**insetCache**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#insetCache) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**labelRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#labelRect) |
| protected  boolean | [**leftToRightCache**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#leftToRightCache) |
| static int | [**MAX\_SCROLL**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#MAX_SCROLL) |
| static int | [**MIN\_SCROLL**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#MIN_SCROLL) |
| static int | [**NEGATIVE\_SCROLL**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#NEGATIVE_SCROLL) |
| static int | [**POSITIVE\_SCROLL**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#POSITIVE_SCROLL) |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**propertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#propertyChangeListener) |
| protected  [BasicSliderUI.ScrollListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ScrollListener.html) | [**scrollListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#scrollListener) |
| protected  [Timer](http://docs.google.com/javax/swing/Timer.html) | [**scrollTimer**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#scrollTimer) |
| protected  [JSlider](http://docs.google.com/javax/swing/JSlider.html) | [**slider**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#slider) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**thumbRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#thumbRect) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**tickRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#tickRect) |
| protected  int | [**trackBuffer**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#trackBuffer) |
| protected  [BasicSliderUI.TrackListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.TrackListener.html) | [**trackListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#trackListener) |
| protected  [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**trackRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#trackRect) |

| **Constructor Summary** | |
| --- | --- |
| [**BasicSliderUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#BasicSliderUI(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) b) |

| **Method Summary** | |
| --- | --- |
| protected  void | [**calculateContentRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateContentRect())() |
| protected  void | [**calculateFocusRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateFocusRect())() |
| protected  void | [**calculateGeometry**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateGeometry())() |
| protected  void | [**calculateLabelRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateLabelRect())() |
| protected  void | [**calculateThumbLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateThumbLocation())() |
| protected  void | [**calculateThumbSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateThumbSize())() |
| protected  void | [**calculateTickRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateTickRect())() |
| protected  void | [**calculateTrackBuffer**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateTrackBuffer())() |
| protected  void | [**calculateTrackRect**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#calculateTrackRect())() |
| protected  [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) | [**createChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createChangeListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) | [**createComponentListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createComponentListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) | [**createFocusListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createFocusListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createPropertyChangeListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createPropertyChangeListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  [BasicSliderUI.ScrollListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ScrollListener.html) | [**createScrollListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createScrollListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  [BasicSliderUI.TrackListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.TrackListener.html) | [**createTrackListener**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createTrackListener(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) | [**createUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#createUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) b) |
| protected  boolean | [**drawInverted**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#drawInverted())() |
| int | [**getBaseline**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getBaseline(javax.swing.JComponent,%20int,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, int width, int height)            Returns the baseline. |
| [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | [**getBaselineResizeBehavior**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getBaselineResizeBehavior(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns an enum indicating how the baseline of the component changes as the size changes. |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**getFocusColor**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getFocusColor())() |
| protected  int | [**getHeightOfHighValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHeightOfHighValueLabel())() |
| protected  int | [**getHeightOfLowValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHeightOfLowValueLabel())() |
| protected  int | [**getHeightOfTallestLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHeightOfTallestLabel())() |
| protected  [Integer](http://docs.google.com/java/lang/Integer.html) | [**getHighestValue**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHighestValue())()            Returns the biggest value that has an entry in the label table. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**getHighestValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHighestValueLabel())()            Returns the label that corresponds to the lowest slider value in the label table. |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**getHighlightColor**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getHighlightColor())() |
| protected  [Integer](http://docs.google.com/java/lang/Integer.html) | [**getLowestValue**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getLowestValue())()            Returns the smallest value that has an entry in the label table. |
| protected  [Component](http://docs.google.com/java/awt/Component.html) | [**getLowestValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getLowestValueLabel())()            Returns the label that corresponds to the highest slider value in the label table. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getMaximumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's maximum size appropriate for the look and feel. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumHorizontalSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getMinimumHorizontalSize())() |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getMinimumSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's minimum size appropriate for the look and feel. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumVerticalSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getMinimumVerticalSize())() |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredHorizontalSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getPreferredHorizontalSize())() |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getPreferredSize(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Returns the specified component's preferred size appropriate for the look and feel. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredVerticalSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getPreferredVerticalSize())() |
| protected  [Color](http://docs.google.com/java/awt/Color.html) | [**getShadowColor**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getShadowColor())() |
| protected  [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getThumbSize**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getThumbSize())() |
| protected  int | [**getTickLength**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getTickLength())()            Gets the height of the tick area for horizontal sliders and the width of the tick area for vertical sliders. |
| protected  int | [**getWidthOfHighValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getWidthOfHighValueLabel())() |
| protected  int | [**getWidthOfLowValueLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getWidthOfLowValueLabel())() |
| protected  int | [**getWidthOfWidestLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#getWidthOfWidestLabel())() |
| protected  void | [**installDefaults**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#installDefaults(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  void | [**installKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#installKeyboardActions(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  void | [**installListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#installListeners(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| void | [**installUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#installUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Configures the specified component appropriate for the look and feel. |
| protected  boolean | [**isDragging**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#isDragging())()            Returns true if the user is dragging the slider. |
| protected  boolean | [**labelsHaveSameBaselines**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#labelsHaveSameBaselines())()            Returns true if all the labels from the label table have the same baseline. |
| void | [**paint**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Paints the specified component appropriate for the look and feel. |
| void | [**paintFocus**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintFocus(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g) |
| protected  void | [**paintHorizontalLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintHorizontalLabel(java.awt.Graphics,%20int,%20java.awt.Component))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int value, [Component](http://docs.google.com/java/awt/Component.html) label)            Called for every label in the label table. |
| void | [**paintLabels**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintLabels(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g) |
| protected  void | [**paintMajorTickForHorizSlider**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintMajorTickForHorizSlider(java.awt.Graphics,%20java.awt.Rectangle,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds, int x) |
| protected  void | [**paintMajorTickForVertSlider**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintMajorTickForVertSlider(java.awt.Graphics,%20java.awt.Rectangle,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds, int y) |
| protected  void | [**paintMinorTickForHorizSlider**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintMinorTickForHorizSlider(java.awt.Graphics,%20java.awt.Rectangle,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds, int x) |
| protected  void | [**paintMinorTickForVertSlider**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintMinorTickForVertSlider(java.awt.Graphics,%20java.awt.Rectangle,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds, int y) |
| void | [**paintThumb**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintThumb(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g) |
| void | [**paintTicks**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintTicks(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g) |
| void | [**paintTrack**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintTrack(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g) |
| protected  void | [**paintVerticalLabel**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#paintVerticalLabel(java.awt.Graphics,%20int,%20java.awt.Component))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, int value, [Component](http://docs.google.com/java/awt/Component.html) label)            Called for every label in the label table. |
| protected  void | [**recalculateIfInsetsChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#recalculateIfInsetsChanged())() |
| protected  void | [**recalculateIfOrientationChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#recalculateIfOrientationChanged())() |
| void | [**scrollByBlock**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#scrollByBlock(int))(int direction) |
| void | [**scrollByUnit**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#scrollByUnit(int))(int direction) |
| protected  void | [**scrollDueToClickInTrack**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#scrollDueToClickInTrack(int))(int dir)            This function is called when a mousePressed was detected in the track, not in the thumb. |
| void | [**setThumbLocation**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#setThumbLocation(int,%20int))(int x, int y) |
| protected  void | [**uninstallKeyboardActions**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#uninstallKeyboardActions(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| protected  void | [**uninstallListeners**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#uninstallListeners(javax.swing.JSlider))([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider) |
| void | [**uninstallUI**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#uninstallUI(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)            Reverses configuration which was done on the specified component during installUI. |
| int | [**valueForXPosition**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#valueForXPosition(int))(int xPos)            Returns a value give an x position. |
| int | [**valueForYPosition**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#valueForYPosition(int))(int yPos)            Returns a value give a y position. |
| protected  int | [**xPositionForValue**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#xPositionForValue(int))(int value) |
| protected  int | [**yPositionForValue**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#yPositionForValue(int))(int value) |
| protected  int | [**yPositionForValue**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.html#yPositionForValue(int,%20int,%20int))(int value, int trackY, int trackHeight)            Returns the y location for the specified value. |

| **Methods inherited from class javax.swing.plaf.**[**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html) |
| --- |
| [contains](http://docs.google.com/javax/swing/plaf/ComponentUI.html#contains(javax.swing.JComponent,%20int,%20int)), [getAccessibleChild](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChild(javax.swing.JComponent,%20int)), [getAccessibleChildrenCount](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getAccessibleChildrenCount(javax.swing.JComponent)), [update](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent)) |

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

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

### POSITIVE\_SCROLL

public static final int **POSITIVE\_SCROLL**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicSliderUI.POSITIVE_SCROLL)

### NEGATIVE\_SCROLL

public static final int **NEGATIVE\_SCROLL**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicSliderUI.NEGATIVE_SCROLL)

### MIN\_SCROLL

public static final int **MIN\_SCROLL**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicSliderUI.MIN_SCROLL)

### MAX\_SCROLL

public static final int **MAX\_SCROLL**

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.plaf.basic.BasicSliderUI.MAX_SCROLL)

### scrollTimer

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

### slider

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

### focusInsets

protected [Insets](http://docs.google.com/java/awt/Insets.html) **focusInsets**

### insetCache

protected [Insets](http://docs.google.com/java/awt/Insets.html) **insetCache**

### leftToRightCache

protected boolean **leftToRightCache**

### focusRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **focusRect**

### contentRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **contentRect**

### labelRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **labelRect**

### tickRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **tickRect**

### trackRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **trackRect**

### thumbRect

protected [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **thumbRect**

### trackBuffer

protected int **trackBuffer**

### trackListener

protected [BasicSliderUI.TrackListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.TrackListener.html) **trackListener**

### changeListener

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

### componentListener

protected [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) **componentListener**

### focusListener

protected [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) **focusListener**

### scrollListener

protected [BasicSliderUI.ScrollListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ScrollListener.html) **scrollListener**

### propertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **propertyChangeListener**

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

### BasicSliderUI

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

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

### getShadowColor

protected [Color](http://docs.google.com/java/awt/Color.html) **getShadowColor**()

### getHighlightColor

protected [Color](http://docs.google.com/java/awt/Color.html) **getHighlightColor**()

### getFocusColor

protected [Color](http://docs.google.com/java/awt/Color.html) **getFocusColor**()

### isDragging

protected boolean **isDragging**()

Returns true if the user is dragging the slider.

**Returns:**true if the user is dragging the slider**Since:** 1.5

### createUI

public static [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **createUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) b)

### installUI

public void **installUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) Configures the specified component appropriate for the look and feel. This method is invoked when the ComponentUI instance is being installed as the UI delegate on the specified component. This method should completely configure the component for the look and feel, including the following:

1. Install any default property values for color, fonts, borders, icons, opacity, etc. on the component. Whenever possible, property values initialized by the client program should *not* be overridden.
2. Install a LayoutManager on the component if necessary.
3. Create/add any required sub-components to the component.
4. Create/install event listeners on the component.
5. Create/install a PropertyChangeListener on the component in order to detect and respond to component property changes appropriately.
6. Install keyboard UI (mnemonics, traversal, etc.) on the component.
7. Initialize any appropriate instance data.

**Overrides:**[installUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component where this UI delegate is being installed**See Also:**[ComponentUI.uninstallUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)), [JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### uninstallUI

public void **uninstallUI**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) Reverses configuration which was done on the specified component during installUI. This method is invoked when this UIComponent instance is being removed as the UI delegate for the specified component. This method should undo the configuration performed in installUI, being careful to leave the JComponent instance in a clean state (no extraneous listeners, look-and-feel-specific property objects, etc.). This should include the following:

1. Remove any UI-set borders from the component.
2. Remove any UI-set layout managers on the component.
3. Remove any UI-added sub-components from the component.
4. Remove any UI-added event/property listeners from the component.
5. Remove any UI-installed keyboard UI from the component.
6. Nullify any allocated instance data objects to allow for GC.

**Overrides:**[uninstallUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html#uninstallUI(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component from which this UI delegate is being removed; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.installUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#installUI(javax.swing.JComponent)), [JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### installDefaults

protected void **installDefaults**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### createTrackListener

protected [BasicSliderUI.TrackListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.TrackListener.html) **createTrackListener**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### createChangeListener

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

### createComponentListener

protected [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) **createComponentListener**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### createFocusListener

protected [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) **createFocusListener**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### createScrollListener

protected [BasicSliderUI.ScrollListener](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ScrollListener.html) **createScrollListener**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### createPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createPropertyChangeListener**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### installListeners

protected void **installListeners**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### uninstallListeners

protected void **uninstallListeners**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### installKeyboardActions

protected void **installKeyboardActions**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### uninstallKeyboardActions

protected void **uninstallKeyboardActions**([JSlider](http://docs.google.com/javax/swing/JSlider.html) slider)

### getBaseline

public int **getBaseline**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 int width,  
 int height)

Returns the baseline.

**Overrides:**[getBaseline](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaseline(javax.swing.JComponent,%20int,%20int)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent baseline is being requested forwidth - the width to get the baseline forheight - the height to get the baseline for **Returns:**baseline or a value < 0 indicating there is no reasonable baseline **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is < 0**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### getBaselineResizeBehavior

public [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) **getBaselineResizeBehavior**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

Returns an enum indicating how the baseline of the component changes as the size changes.

**Overrides:**[getBaselineResizeBehavior](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getBaselineResizeBehavior(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - JComponent to return baseline resize behavior for **Returns:**an enum indicating how the baseline changes as the component size changes **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if c is null**Since:** 1.6 **See Also:**[JComponent.getBaseline(int, int)](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int))

### labelsHaveSameBaselines

protected boolean **labelsHaveSameBaselines**()

Returns true if all the labels from the label table have the same baseline.

**Returns:**true if all the labels from the label table have the same baseline**Since:** 1.6

### getPreferredHorizontalSize

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

### getPreferredVerticalSize

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

### getMinimumHorizontalSize

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

### getMinimumVerticalSize

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

### getPreferredSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getPreferredSize**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) Returns the specified component's preferred size appropriate for the look and feel. If null is returned, the preferred size will be calculated by the component's layout manager instead (this is the preferred approach for any component with a specific layout manager installed). The default implementation of this method returns null.

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose preferred size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [LayoutManager.preferredLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#preferredLayoutSize(java.awt.Container))

### getMinimumSize

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) Returns the specified component's minimum size appropriate for the look and feel. If null is returned, the minimum size will be calculated by the component's layout manager instead (this is the preferred approach for any component with a specific layout manager installed). The default implementation of this method invokes getPreferredSize and returns that value.

**Overrides:**[getMinimumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMinimumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose minimum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMinimumSize()](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [LayoutManager.minimumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager.html#minimumLayoutSize(java.awt.Container)), [ComponentUI.getPreferredSize(javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getPreferredSize(javax.swing.JComponent))

### getMaximumSize

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

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) Returns the specified component's maximum size appropriate for the look and feel. If null is returned, the maximum size will be calculated by the component's layout manager instead (this is the preferred approach for any component with a specific layout manager installed). The default implementation of this method invokes getPreferredSize and returns that value.

**Overrides:**[getMaximumSize](http://docs.google.com/javax/swing/plaf/ComponentUI.html#getMaximumSize(javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**c - the component whose maximum size is being queried; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components **Returns:**a Dimension object or null**See Also:**[JComponent.getMaximumSize()](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [LayoutManager2.maximumLayoutSize(java.awt.Container)](http://docs.google.com/java/awt/LayoutManager2.html#maximumLayoutSize(java.awt.Container))

### calculateGeometry

protected void **calculateGeometry**()

### calculateFocusRect

protected void **calculateFocusRect**()

### calculateThumbSize

protected void **calculateThumbSize**()

### calculateContentRect

protected void **calculateContentRect**()

### calculateThumbLocation

protected void **calculateThumbLocation**()

### calculateTrackBuffer

protected void **calculateTrackBuffer**()

### calculateTrackRect

protected void **calculateTrackRect**()

### getTickLength

protected int **getTickLength**()

Gets the height of the tick area for horizontal sliders and the width of the tick area for vertical sliders. BasicSliderUI uses the returned value to determine the tick area rectangle. If you want to give your ticks some room, make this larger than you need and paint your ticks away from the sides in paintTicks().

### calculateTickRect

protected void **calculateTickRect**()

### calculateLabelRect

protected void **calculateLabelRect**()

### getThumbSize

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

### getWidthOfWidestLabel

protected int **getWidthOfWidestLabel**()

### getHeightOfTallestLabel

protected int **getHeightOfTallestLabel**()

### getWidthOfHighValueLabel

protected int **getWidthOfHighValueLabel**()

### getWidthOfLowValueLabel

protected int **getWidthOfLowValueLabel**()

### getHeightOfHighValueLabel

protected int **getHeightOfHighValueLabel**()

### getHeightOfLowValueLabel

protected int **getHeightOfLowValueLabel**()

### drawInverted

protected boolean **drawInverted**()

### getHighestValue

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

Returns the biggest value that has an entry in the label table.

**Returns:**biggest value that has an entry in the label table, or null.**Since:** 1.6

### getLowestValue

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

Returns the smallest value that has an entry in the label table.

**Returns:**smallest value that has an entry in the label table, or null.**Since:** 1.6

### getLowestValueLabel

protected [Component](http://docs.google.com/java/awt/Component.html) **getLowestValueLabel**()

Returns the label that corresponds to the highest slider value in the label table.

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

### getHighestValueLabel

protected [Component](http://docs.google.com/java/awt/Component.html) **getHighestValueLabel**()

Returns the label that corresponds to the lowest slider value in the label table.

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

### paint

public void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [JComponent](http://docs.google.com/javax/swing/JComponent.html) c)

**Description copied from class:** [**ComponentUI**](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) Paints the specified component appropriate for the look and feel. This method is invoked from the ComponentUI.update method when the specified component is being painted. Subclasses should override this method and use the specified Graphics object to render the content of the component.

**Overrides:**[paint](http://docs.google.com/javax/swing/plaf/ComponentUI.html#paint(java.awt.Graphics,%20javax.swing.JComponent)) in class [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html) **Parameters:**g - the Graphics context in which to paintc - the component being painted; this argument is often ignored, but might be used if the UI object is stateless and shared by multiple components**See Also:**[ComponentUI.update(java.awt.Graphics, javax.swing.JComponent)](http://docs.google.com/javax/swing/plaf/ComponentUI.html#update(java.awt.Graphics,%20javax.swing.JComponent))

### recalculateIfInsetsChanged

protected void **recalculateIfInsetsChanged**()

### recalculateIfOrientationChanged

protected void **recalculateIfOrientationChanged**()

### paintFocus

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

### paintTrack

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

### paintTicks

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

### paintMinorTickForHorizSlider

protected void **paintMinorTickForHorizSlider**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds,  
 int x)

### paintMajorTickForHorizSlider

protected void **paintMajorTickForHorizSlider**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds,  
 int x)

### paintMinorTickForVertSlider

protected void **paintMinorTickForVertSlider**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds,  
 int y)

### paintMajorTickForVertSlider

protected void **paintMajorTickForVertSlider**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) tickBounds,  
 int y)

### paintLabels

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

### paintHorizontalLabel

protected void **paintHorizontalLabel**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int value,  
 [Component](http://docs.google.com/java/awt/Component.html) label)

Called for every label in the label table. Used to draw the labels for horizontal sliders. The graphics have been translated to labelRect.y already.

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

### paintVerticalLabel

protected void **paintVerticalLabel**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 int value,  
 [Component](http://docs.google.com/java/awt/Component.html) label)

Called for every label in the label table. Used to draw the labels for vertical sliders. The graphics have been translated to labelRect.x already.

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

### paintThumb

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

### setThumbLocation

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

### scrollByBlock

public void **scrollByBlock**(int direction)

### scrollByUnit

public void **scrollByUnit**(int direction)

### scrollDueToClickInTrack

protected void **scrollDueToClickInTrack**(int dir)

This function is called when a mousePressed was detected in the track, not in the thumb. The default behavior is to scroll by block. You can override this method to stop it from scrolling or to add additional behavior.

### xPositionForValue

protected int **xPositionForValue**(int value)

### yPositionForValue

protected int **yPositionForValue**(int value)

### yPositionForValue

protected int **yPositionForValue**(int value,  
 int trackY,  
 int trackHeight)

Returns the y location for the specified value. No checking is done on the arguments. In particular if trackHeight is negative undefined results may occur.

**Parameters:**value - the slider value to get the location fortrackY - y-origin of the tracktrackHeight - the height of the track**Since:** 1.6

### valueForYPosition

public int **valueForYPosition**(int yPos)

Returns a value give a y position. If yPos is past the track at the top or the bottom it will set the value to the min or max of the slider, depending if the slider is inverted or not.

### valueForXPosition

public int **valueForXPosition**(int xPos)

Returns a value give an x position. If xPos is past the track at the left or the right it will set the value to the min or max of the slider, depending if the slider is inverted or not.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicSliderUI.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/plaf/basic/BasicSeparatorUI.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ActionScroller.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/plaf/basic/BasicSliderUI.html)    [**NO FRAMES**](http://docs.google.com/BasicSliderUI.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3o7alnk) | [METHOD](#ihv636) |

[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).