| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LabelView.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/text/Keymap.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/LayeredHighlighter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/LabelView.html)    [**NO FRAMES**](http://docs.google.com/LabelView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#35nkun2) | [METHOD](#44sinio) |

## **javax.swing.text**

Class LabelView

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.swing.text.View](http://docs.google.com/javax/swing/text/View.html)  
 [javax.swing.text.GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html)  
 **javax.swing.text.LabelView**

**All Implemented Interfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [TabableView](http://docs.google.com/javax/swing/text/TabableView.html) **Direct Known Subclasses:** [InlineView](http://docs.google.com/javax/swing/text/html/InlineView.html)

public class **LabelView**extends [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html)implements [TabableView](http://docs.google.com/javax/swing/text/TabableView.html)

A LabelView is a styled chunk of text that represents a view mapped over an element in the text model. It caches the character level attributes used for rendering.

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**GlyphView**](http://docs.google.com/javax/swing/text/GlyphView.html) |
| --- |
| [GlyphView.GlyphPainter](http://docs.google.com/javax/swing/text/GlyphView.GlyphPainter.html) |

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [BadBreakWeight](http://docs.google.com/javax/swing/text/View.html#BadBreakWeight), [ExcellentBreakWeight](http://docs.google.com/javax/swing/text/View.html#ExcellentBreakWeight), [ForcedBreakWeight](http://docs.google.com/javax/swing/text/View.html#ForcedBreakWeight), [GoodBreakWeight](http://docs.google.com/javax/swing/text/View.html#GoodBreakWeight), [X\_AXIS](http://docs.google.com/javax/swing/text/View.html#X_AXIS), [Y\_AXIS](http://docs.google.com/javax/swing/text/View.html#Y_AXIS) |

| **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) |

| **Constructor Summary** | |
| --- | --- |
| [**LabelView**](http://docs.google.com/javax/swing/text/LabelView.html#LabelView(javax.swing.text.Element))([Element](http://docs.google.com/javax/swing/text/Element.html) elem)            Constructs a new view wrapped on an element. |

| **Method Summary** | |
| --- | --- |
| void | [**changedUpdate**](http://docs.google.com/javax/swing/text/LabelView.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e, [Shape](http://docs.google.com/java/awt/Shape.html) a, [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)            Gives notification from the document that attributes were changed in a location that this view is responsible for. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/javax/swing/text/LabelView.html#getBackground())()            Fetches the background color to use to render the glyphs. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/javax/swing/text/LabelView.html#getFont())()            Fetches the font that the glyphs should be based upon. |
| protected  [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/javax/swing/text/LabelView.html#getFontMetrics())()  **Deprecated.** *FontMetrics are not used for glyph rendering when running in the JDK.* |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getForeground**](http://docs.google.com/javax/swing/text/LabelView.html#getForeground())()            Fetches the foreground color to use to render the glyphs. |
| boolean | [**isStrikeThrough**](http://docs.google.com/javax/swing/text/LabelView.html#isStrikeThrough())()            Determines if the glyphs should have a strikethrough line. |
| boolean | [**isSubscript**](http://docs.google.com/javax/swing/text/LabelView.html#isSubscript())()            Determines if the glyphs should be rendered as superscript. |
| boolean | [**isSuperscript**](http://docs.google.com/javax/swing/text/LabelView.html#isSuperscript())()            Determines if the glyphs should be rendered as subscript. |
| boolean | [**isUnderline**](http://docs.google.com/javax/swing/text/LabelView.html#isUnderline())()            Determines if the glyphs should be underlined. |
| protected  void | [**setBackground**](http://docs.google.com/javax/swing/text/LabelView.html#setBackground(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) bg)            Sets the background color for the view. |
| protected  void | [**setPropertiesFromAttributes**](http://docs.google.com/javax/swing/text/LabelView.html#setPropertiesFromAttributes())()            Sets the cached properties from the attributes. |
| protected  void | [**setStrikeThrough**](http://docs.google.com/javax/swing/text/LabelView.html#setStrikeThrough(boolean))(boolean s)            Sets whether or not the view has a strike/line through it. |
| protected  void | [**setSubscript**](http://docs.google.com/javax/swing/text/LabelView.html#setSubscript(boolean))(boolean s)            Sets whether or not the view represents a subscript. |
| protected  void | [**setSuperscript**](http://docs.google.com/javax/swing/text/LabelView.html#setSuperscript(boolean))(boolean s)            Sets whether or not the view represents a superscript. |
| protected  void | [**setUnderline**](http://docs.google.com/javax/swing/text/LabelView.html#setUnderline(boolean))(boolean u)            Sets whether or not the view is underlined. |

| **Methods inherited from class javax.swing.text.**[**GlyphView**](http://docs.google.com/javax/swing/text/GlyphView.html) |
| --- |
| [breakView](http://docs.google.com/javax/swing/text/GlyphView.html#breakView(int,%20int,%20float,%20float)), [checkPainter](http://docs.google.com/javax/swing/text/GlyphView.html#checkPainter()), [clone](http://docs.google.com/javax/swing/text/GlyphView.html#clone()), [createFragment](http://docs.google.com/javax/swing/text/GlyphView.html#createFragment(int,%20int)), [getAlignment](http://docs.google.com/javax/swing/text/GlyphView.html#getAlignment(int)), [getBreakWeight](http://docs.google.com/javax/swing/text/GlyphView.html#getBreakWeight(int,%20float,%20float)), [getEndOffset](http://docs.google.com/javax/swing/text/GlyphView.html#getEndOffset()), [getGlyphPainter](http://docs.google.com/javax/swing/text/GlyphView.html#getGlyphPainter()), [getNextVisualPositionFrom](http://docs.google.com/javax/swing/text/GlyphView.html#getNextVisualPositionFrom(int,%20javax.swing.text.Position.Bias,%20java.awt.Shape,%20int,%20javax.swing.text.Position.Bias%5B%5D)), [getPartialSpan](http://docs.google.com/javax/swing/text/GlyphView.html#getPartialSpan(int,%20int)), [getPreferredSpan](http://docs.google.com/javax/swing/text/GlyphView.html#getPreferredSpan(int)), [getStartOffset](http://docs.google.com/javax/swing/text/GlyphView.html#getStartOffset()), [getTabbedSpan](http://docs.google.com/javax/swing/text/GlyphView.html#getTabbedSpan(float,%20javax.swing.text.TabExpander)), [getTabExpander](http://docs.google.com/javax/swing/text/GlyphView.html#getTabExpander()), [getText](http://docs.google.com/javax/swing/text/GlyphView.html#getText(int,%20int)), [insertUpdate](http://docs.google.com/javax/swing/text/GlyphView.html#insertUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [modelToView](http://docs.google.com/javax/swing/text/GlyphView.html#modelToView(int,%20java.awt.Shape,%20javax.swing.text.Position.Bias)), [paint](http://docs.google.com/javax/swing/text/GlyphView.html#paint(java.awt.Graphics,%20java.awt.Shape)), [removeUpdate](http://docs.google.com/javax/swing/text/GlyphView.html#removeUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [setGlyphPainter](http://docs.google.com/javax/swing/text/GlyphView.html#setGlyphPainter(javax.swing.text.GlyphView.GlyphPainter)), [viewToModel](http://docs.google.com/javax/swing/text/GlyphView.html#viewToModel(float,%20float,%20java.awt.Shape,%20javax.swing.text.Position.Bias%5B%5D)) |

| **Methods inherited from class javax.swing.text.**[**View**](http://docs.google.com/javax/swing/text/View.html) |
| --- |
| [append](http://docs.google.com/javax/swing/text/View.html#append(javax.swing.text.View)), [forwardUpdate](http://docs.google.com/javax/swing/text/View.html#forwardUpdate(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [forwardUpdateToView](http://docs.google.com/javax/swing/text/View.html#forwardUpdateToView(javax.swing.text.View,%20javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)), [getAttributes](http://docs.google.com/javax/swing/text/View.html#getAttributes()), [getChildAllocation](http://docs.google.com/javax/swing/text/View.html#getChildAllocation(int,%20java.awt.Shape)), [getContainer](http://docs.google.com/javax/swing/text/View.html#getContainer()), [getDocument](http://docs.google.com/javax/swing/text/View.html#getDocument()), [getElement](http://docs.google.com/javax/swing/text/View.html#getElement()), [getGraphics](http://docs.google.com/javax/swing/text/View.html#getGraphics()), [getMaximumSpan](http://docs.google.com/javax/swing/text/View.html#getMaximumSpan(int)), [getMinimumSpan](http://docs.google.com/javax/swing/text/View.html#getMinimumSpan(int)), [getParent](http://docs.google.com/javax/swing/text/View.html#getParent()), [getResizeWeight](http://docs.google.com/javax/swing/text/View.html#getResizeWeight(int)), [getToolTipText](http://docs.google.com/javax/swing/text/View.html#getToolTipText(float,%20float,%20java.awt.Shape)), [getView](http://docs.google.com/javax/swing/text/View.html#getView(int)), [getViewCount](http://docs.google.com/javax/swing/text/View.html#getViewCount()), [getViewFactory](http://docs.google.com/javax/swing/text/View.html#getViewFactory()), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(float,%20float,%20java.awt.Shape)), [getViewIndex](http://docs.google.com/javax/swing/text/View.html#getViewIndex(int,%20javax.swing.text.Position.Bias)), [insert](http://docs.google.com/javax/swing/text/View.html#insert(int,%20javax.swing.text.View)), [isVisible](http://docs.google.com/javax/swing/text/View.html#isVisible()), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20javax.swing.text.Position.Bias,%20int,%20javax.swing.text.Position.Bias,%20java.awt.Shape)), [modelToView](http://docs.google.com/javax/swing/text/View.html#modelToView(int,%20java.awt.Shape)), [preferenceChanged](http://docs.google.com/javax/swing/text/View.html#preferenceChanged(javax.swing.text.View,%20boolean,%20boolean)), [remove](http://docs.google.com/javax/swing/text/View.html#remove(int)), [removeAll](http://docs.google.com/javax/swing/text/View.html#removeAll()), [replace](http://docs.google.com/javax/swing/text/View.html#replace(int,%20int,%20javax.swing.text.View%5B%5D)), [setParent](http://docs.google.com/javax/swing/text/View.html#setParent(javax.swing.text.View)), [setSize](http://docs.google.com/javax/swing/text/View.html#setSize(float,%20float)), [updateChildren](http://docs.google.com/javax/swing/text/View.html#updateChildren(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20javax.swing.text.ViewFactory)), [updateLayout](http://docs.google.com/javax/swing/text/View.html#updateLayout(javax.swing.event.DocumentEvent.ElementChange,%20javax.swing.event.DocumentEvent,%20java.awt.Shape)), [viewToModel](http://docs.google.com/javax/swing/text/View.html#viewToModel(float,%20float,%20java.awt.Shape)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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)) |

| **Methods inherited from interface javax.swing.text.**[**TabableView**](http://docs.google.com/javax/swing/text/TabableView.html) |
| --- |
| [getPartialSpan](http://docs.google.com/javax/swing/text/TabableView.html#getPartialSpan(int,%20int)), [getTabbedSpan](http://docs.google.com/javax/swing/text/TabableView.html#getTabbedSpan(float,%20javax.swing.text.TabExpander)) |

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

### LabelView

public **LabelView**([Element](http://docs.google.com/javax/swing/text/Element.html) elem)

Constructs a new view wrapped on an element.

**Parameters:**elem - the element

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

### setUnderline

protected void **setUnderline**(boolean u)

Sets whether or not the view is underlined. Note that this setter is protected and is really only meant if you need to update some additional state when set.

**Parameters:**u - true if the view is underlined, otherwise false**See Also:**[isUnderline()](http://docs.google.com/javax/swing/text/LabelView.html#isUnderline())

### setStrikeThrough

protected void **setStrikeThrough**(boolean s)

Sets whether or not the view has a strike/line through it. Note that this setter is protected and is really only meant if you need to update some additional state when set.

**Parameters:**s - true if the view has a strike/line through it, otherwise false**See Also:**[isStrikeThrough()](http://docs.google.com/javax/swing/text/LabelView.html#isStrikeThrough())

### setSuperscript

protected void **setSuperscript**(boolean s)

Sets whether or not the view represents a superscript. Note that this setter is protected and is really only meant if you need to update some additional state when set.

**Parameters:**s - true if the view represents a superscript, otherwise false**See Also:**[isSuperscript()](http://docs.google.com/javax/swing/text/LabelView.html#isSuperscript())

### setSubscript

protected void **setSubscript**(boolean s)

Sets whether or not the view represents a subscript. Note that this setter is protected and is really only meant if you need to update some additional state when set.

**Parameters:**s - true if the view represents a subscript, otherwise false**See Also:**[isSubscript()](http://docs.google.com/javax/swing/text/LabelView.html#isSubscript())

### setBackground

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

Sets the background color for the view. This method is typically invoked as part of configuring this View. If you need to customize the background color you should override setPropertiesFromAttributes and invoke this method. A value of null indicates no background should be rendered, so that the background of the parent View will show through.

**Parameters:**bg - background color, or null**Since:** 1.5 **See Also:**[setPropertiesFromAttributes()](http://docs.google.com/javax/swing/text/LabelView.html#setPropertiesFromAttributes())

### setPropertiesFromAttributes

protected void **setPropertiesFromAttributes**()

Sets the cached properties from the attributes.

### getFontMetrics

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
protected [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**()

**Deprecated.** *FontMetrics are not used for glyph rendering when running in the JDK.*

Fetches the FontMetrics used for this view.

### getBackground

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

Fetches the background color to use to render the glyphs. This is implemented to return a cached background color, which defaults to null.

**Overrides:**[getBackground](http://docs.google.com/javax/swing/text/GlyphView.html#getBackground()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the cached background color**Since:** 1.3

### getForeground

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

Fetches the foreground color to use to render the glyphs. This is implemented to return a cached foreground color, which defaults to null.

**Overrides:**[getForeground](http://docs.google.com/javax/swing/text/GlyphView.html#getForeground()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the cached foreground color**Since:** 1.3

### getFont

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

Fetches the font that the glyphs should be based upon. This is implemented to return a cached font.

**Overrides:**[getFont](http://docs.google.com/javax/swing/text/GlyphView.html#getFont()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the cached font

### isUnderline

public boolean **isUnderline**()

Determines if the glyphs should be underlined. If true, an underline should be drawn through the baseline. This is implemented to return the cached underline property.

When you request this property, LabelView re-syncs its state with the properties of the Element's AttributeSet. If Element's AttributeSet does not have this property set, it will revert to false.

**Overrides:**[isUnderline](http://docs.google.com/javax/swing/text/GlyphView.html#isUnderline()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the value of the cached underline property**Since:** 1.3

### isStrikeThrough

public boolean **isStrikeThrough**()

Determines if the glyphs should have a strikethrough line. If true, a line should be drawn through the center of the glyphs. This is implemented to return the cached strikeThrough property.

When you request this property, LabelView re-syncs its state with the properties of the Element's AttributeSet. If Element's AttributeSet does not have this property set, it will revert to false.

**Overrides:**[isStrikeThrough](http://docs.google.com/javax/swing/text/GlyphView.html#isStrikeThrough()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the value of the cached strikeThrough property**Since:** 1.3

### isSubscript

public boolean **isSubscript**()

Determines if the glyphs should be rendered as superscript.

**Overrides:**[isSubscript](http://docs.google.com/javax/swing/text/GlyphView.html#isSubscript()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the value of the cached subscript property

When you request this property, LabelView re-syncs its state with the properties of the Element's AttributeSet. If Element's AttributeSet does not have this property set, it will revert to false.

**Since:** 1.3

### isSuperscript

public boolean **isSuperscript**()

Determines if the glyphs should be rendered as subscript.

When you request this property, LabelView re-syncs its state with the properties of the Element's AttributeSet. If Element's AttributeSet does not have this property set, it will revert to false.

**Overrides:**[isSuperscript](http://docs.google.com/javax/swing/text/GlyphView.html#isSuperscript()) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Returns:**the value of the cached superscript property**Since:** 1.3

### changedUpdate

public void **changedUpdate**([DocumentEvent](http://docs.google.com/javax/swing/event/DocumentEvent.html) e,  
 [Shape](http://docs.google.com/java/awt/Shape.html) a,  
 [ViewFactory](http://docs.google.com/javax/swing/text/ViewFactory.html) f)

Gives notification from the document that attributes were changed in a location that this view is responsible for.

**Overrides:**[changedUpdate](http://docs.google.com/javax/swing/text/GlyphView.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory)) in class [GlyphView](http://docs.google.com/javax/swing/text/GlyphView.html) **Parameters:**e - the change information from the associated documenta - the current allocation of the viewf - the factory to use to rebuild if the view has children**See Also:**[View.changedUpdate(javax.swing.event.DocumentEvent, java.awt.Shape, javax.swing.text.ViewFactory)](http://docs.google.com/javax/swing/text/View.html#changedUpdate(javax.swing.event.DocumentEvent,%20java.awt.Shape,%20javax.swing.text.ViewFactory))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LabelView.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/text/Keymap.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/text/LayeredHighlighter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/text/LabelView.html)    [**NO FRAMES**](http://docs.google.com/LabelView.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#3dy6vkm) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: FIELD | [CONSTR](#35nkun2) | [METHOD](#44sinio) |

[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).