| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JLabel.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/JInternalFrame.JDesktopIcon.AccessibleJDesktopIcon.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JLabel.AccessibleJLabel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JLabel.html)    [**NO FRAMES**](http://docs.google.com/JLabel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#1y810tw) | [METHOD](#3as4poj) |

## **javax.swing**

Class JLabel

[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.JLabel**

**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) **Direct Known Subclasses:** [BasicComboBoxRenderer](http://docs.google.com/javax/swing/plaf/basic/BasicComboBoxRenderer.html), [DefaultListCellRenderer](http://docs.google.com/javax/swing/DefaultListCellRenderer.html), [DefaultTableCellRenderer](http://docs.google.com/javax/swing/table/DefaultTableCellRenderer.html), [DefaultTreeCellRenderer](http://docs.google.com/javax/swing/tree/DefaultTreeCellRenderer.html)

public class **JLabel**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 display area for a short text string or an image, or both. A label does not react to input events. As a result, it cannot get the keyboard focus. A label can, however, display a keyboard alternative as a convenience for a nearby component that has a keyboard alternative but can't display it.

A JLabel object can display either text, an image, or both. You can specify where in the label's display area the label's contents are aligned by setting the vertical and horizontal alignment. By default, labels are vertically centered in their display area. Text-only labels are leading edge aligned, by default; image-only labels are horizontally centered, by default.

You can also specify the position of the text relative to the image. By default, text is on the trailing edge of the image, with the text and image vertically aligned.

A label's leading and trailing edge are determined from the value of its [ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) property. At present, the default ComponentOrientation setting maps the leading edge to left and the trailing edge to right.

Finally, you can use the setIconTextGap method to specify how many pixels should appear between the text and the image. The default is 4 pixels.

See [How to Use Labels](http://java.sun.com/docs/books/tutorial/uiswing/components/label.html) in *The Java Tutorial* for further documentation.

**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 | [**JLabel.AccessibleJLabel**](http://docs.google.com/javax/swing/JLabel.AccessibleJLabel.html)            The class used to obtain the accessible role for this object. |

| **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  [Component](http://docs.google.com/java/awt/Component.html) | [**labelFor**](http://docs.google.com/javax/swing/JLabel.html#labelFor) |

| **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** | |
| --- | --- |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel())()            Creates a JLabel instance with no image and with an empty string for the title. |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) image)            Creates a JLabel instance with the specified image. |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel(javax.swing.Icon,%20int))([Icon](http://docs.google.com/javax/swing/Icon.html) image, int horizontalAlignment)            Creates a JLabel instance with the specified image and horizontal alignment. |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Creates a JLabel instance with the specified text. |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel(java.lang.String,%20javax.swing.Icon,%20int))([String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int horizontalAlignment)            Creates a JLabel instance with the specified text, image, and horizontal alignment. |
| [**JLabel**](http://docs.google.com/javax/swing/JLabel.html#JLabel(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) text, int horizontalAlignment)            Creates a JLabel instance with the specified text and horizontal alignment. |

| **Method Summary** | |
| --- | --- |
| protected  int | [**checkHorizontalKey**](http://docs.google.com/javax/swing/JLabel.html#checkHorizontalKey(int,%20java.lang.String))(int key, [String](http://docs.google.com/java/lang/String.html) message)            Verify that key is a legal value for the horizontalAlignment properties. |
| protected  int | [**checkVerticalKey**](http://docs.google.com/javax/swing/JLabel.html#checkVerticalKey(int,%20java.lang.String))(int key, [String](http://docs.google.com/java/lang/String.html) message)            Verify that key is a legal value for the verticalAlignment or verticalTextPosition properties. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JLabel.html#getAccessibleContext())()            Get the AccessibleContext of this object |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getDisabledIcon**](http://docs.google.com/javax/swing/JLabel.html#getDisabledIcon())()            Returns the icon used by the label when it's disabled. |
| int | [**getDisplayedMnemonic**](http://docs.google.com/javax/swing/JLabel.html#getDisplayedMnemonic())()            Return the keycode that indicates a mnemonic key. |
| int | [**getDisplayedMnemonicIndex**](http://docs.google.com/javax/swing/JLabel.html#getDisplayedMnemonicIndex())()            Returns the character, as an index, that the look and feel should provide decoration for as representing the mnemonic character. |
| int | [**getHorizontalAlignment**](http://docs.google.com/javax/swing/JLabel.html#getHorizontalAlignment())()            Returns the alignment of the label's contents along the X axis. |
| int | [**getHorizontalTextPosition**](http://docs.google.com/javax/swing/JLabel.html#getHorizontalTextPosition())()            Returns the horizontal position of the label's text, relative to its image. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIcon**](http://docs.google.com/javax/swing/JLabel.html#getIcon())()            Returns the graphic image (glyph, icon) that the label displays. |
| int | [**getIconTextGap**](http://docs.google.com/javax/swing/JLabel.html#getIconTextGap())()            Returns the amount of space between the text and the icon displayed in this label. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getLabelFor**](http://docs.google.com/javax/swing/JLabel.html#getLabelFor())()            Get the component this is labelling. |
| [String](http://docs.google.com/java/lang/String.html) | [**getText**](http://docs.google.com/javax/swing/JLabel.html#getText())()            Returns the text string that the label displays. |
| [LabelUI](http://docs.google.com/javax/swing/plaf/LabelUI.html) | [**getUI**](http://docs.google.com/javax/swing/JLabel.html#getUI())()            Returns the L&F object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JLabel.html#getUIClassID())()            Returns a string that specifies the name of the l&f class that renders this component. |
| int | [**getVerticalAlignment**](http://docs.google.com/javax/swing/JLabel.html#getVerticalAlignment())()            Returns the alignment of the label's contents along the Y axis. |
| int | [**getVerticalTextPosition**](http://docs.google.com/javax/swing/JLabel.html#getVerticalTextPosition())()            Returns the vertical position of the label's text, relative to its image. |
| boolean | [**imageUpdate**](http://docs.google.com/javax/swing/JLabel.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))([Image](http://docs.google.com/java/awt/Image.html) img, int infoflags, int x, int y, int w, int h)            This is overridden to return false if the current Icon's Image is not equal to the passed in Image img. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JLabel.html#paramString())()            Returns a string representation of this JLabel. |
| void | [**setDisabledIcon**](http://docs.google.com/javax/swing/JLabel.html#setDisabledIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) disabledIcon)            Set the icon to be displayed if this JLabel is "disabled" (JLabel.setEnabled(false)). |
| void | [**setDisplayedMnemonic**](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonic(char))(char aChar)            Specifies the displayedMnemonic as a char value. |
| void | [**setDisplayedMnemonic**](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonic(int))(int key)            Specify a keycode that indicates a mnemonic key. |
| void | [**setDisplayedMnemonicIndex**](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonicIndex(int))(int index)            Provides a hint to the look and feel as to which character in the text should be decorated to represent the mnemonic. |
| void | [**setHorizontalAlignment**](http://docs.google.com/javax/swing/JLabel.html#setHorizontalAlignment(int))(int alignment)            Sets the alignment of the label's contents along the X axis. |
| void | [**setHorizontalTextPosition**](http://docs.google.com/javax/swing/JLabel.html#setHorizontalTextPosition(int))(int textPosition)            Sets the horizontal position of the label's text, relative to its image. |
| void | [**setIcon**](http://docs.google.com/javax/swing/JLabel.html#setIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Defines the icon this component will display. |
| void | [**setIconTextGap**](http://docs.google.com/javax/swing/JLabel.html#setIconTextGap(int))(int iconTextGap)            If both the icon and text properties are set, this property defines the space between them. |
| void | [**setLabelFor**](http://docs.google.com/javax/swing/JLabel.html#setLabelFor(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Set the component this is labelling. |
| void | [**setText**](http://docs.google.com/javax/swing/JLabel.html#setText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Defines the single line of text this component will display. |
| void | [**setUI**](http://docs.google.com/javax/swing/JLabel.html#setUI(javax.swing.plaf.LabelUI))([LabelUI](http://docs.google.com/javax/swing/plaf/LabelUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**setVerticalAlignment**](http://docs.google.com/javax/swing/JLabel.html#setVerticalAlignment(int))(int alignment)            Sets the alignment of the label's contents along the Y axis. |
| void | [**setVerticalTextPosition**](http://docs.google.com/javax/swing/JLabel.html#setVerticalTextPosition(int))(int textPosition)            Sets the vertical position of the label's text, relative to its image. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JLabel.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)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [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** |
| --- |

### labelFor

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

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

### JLabel

public **JLabel**([String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int horizontalAlignment)

Creates a JLabel instance with the specified text, image, and horizontal alignment. The label is centered vertically in its display area. The text is on the trailing edge of the image.

**Parameters:**text - The text to be displayed by the label.icon - The image to be displayed by the label.horizontalAlignment - One of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING or TRAILING.

### JLabel

public **JLabel**([String](http://docs.google.com/java/lang/String.html) text,  
 int horizontalAlignment)

Creates a JLabel instance with the specified text and horizontal alignment. The label is centered vertically in its display area.

**Parameters:**text - The text to be displayed by the label.horizontalAlignment - One of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING or TRAILING.

### JLabel

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

Creates a JLabel instance with the specified text. The label is aligned against the leading edge of its display area, and centered vertically.

**Parameters:**text - The text to be displayed by the label.

### JLabel

public **JLabel**([Icon](http://docs.google.com/javax/swing/Icon.html) image,  
 int horizontalAlignment)

Creates a JLabel instance with the specified image and horizontal alignment. The label is centered vertically in its display area.

**Parameters:**image - The image to be displayed by the label.horizontalAlignment - One of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING or TRAILING.

### JLabel

public **JLabel**([Icon](http://docs.google.com/javax/swing/Icon.html) image)

Creates a JLabel instance with the specified image. The label is centered vertically and horizontally in its display area.

**Parameters:**image - The image to be displayed by the label.

### JLabel

public **JLabel**()

Creates a JLabel instance with no image and with an empty string for the title. The label is centered vertically in its display area. The label's contents, once set, will be displayed on the leading edge of the label's display area.

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

### getUI

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

Returns the L&F object that renders this component.

**Returns:**LabelUI object

### setUI

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

Sets the L&F object that renders this component.

**Parameters:**ui - the LabelUI 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 a string that specifies 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:**String "LabelUI"**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))

### getText

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

Returns the text string that the label displays.

**Returns:**a String**See Also:**[setText(java.lang.String)](http://docs.google.com/javax/swing/JLabel.html#setText(java.lang.String))

### setText

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

Defines the single line of text this component will display. If the value of text is null or empty string, nothing is displayed.

The default value of this property is null.

This is a JavaBeans bound property.

**See Also:**[setVerticalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalTextPosition(int)), [setHorizontalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setHorizontalTextPosition(int)), [setIcon(javax.swing.Icon)](http://docs.google.com/javax/swing/JLabel.html#setIcon(javax.swing.Icon))

### getIcon

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

Returns the graphic image (glyph, icon) that the label displays.

**Returns:**an Icon**See Also:**[setIcon(javax.swing.Icon)](http://docs.google.com/javax/swing/JLabel.html#setIcon(javax.swing.Icon))

### setIcon

public void **setIcon**([Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Defines the icon this component will display. If the value of icon is null, nothing is displayed.

The default value of this property is null.

This is a JavaBeans bound property.

**See Also:**[setVerticalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalTextPosition(int)), [setHorizontalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setHorizontalTextPosition(int)), [getIcon()](http://docs.google.com/javax/swing/JLabel.html#getIcon())

### getDisabledIcon

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

Returns the icon used by the label when it's disabled. If no disabled icon has been set this will forward the call to the look and feel to construct an appropriate disabled Icon.

Some look and feels might not render the disabled Icon, in which case they will ignore this.

**Returns:**the disabledIcon property**See Also:**[setDisabledIcon(javax.swing.Icon)](http://docs.google.com/javax/swing/JLabel.html#setDisabledIcon(javax.swing.Icon)), [LookAndFeel.getDisabledIcon(javax.swing.JComponent, javax.swing.Icon)](http://docs.google.com/javax/swing/LookAndFeel.html#getDisabledIcon(javax.swing.JComponent,%20javax.swing.Icon)), [ImageIcon](http://docs.google.com/javax/swing/ImageIcon.html)

### setDisabledIcon

public void **setDisabledIcon**([Icon](http://docs.google.com/javax/swing/Icon.html) disabledIcon)

Set the icon to be displayed if this JLabel is "disabled" (JLabel.setEnabled(false)).

The default value of this property is null.

**Parameters:**disabledIcon - the Icon to display when the component is disabled**See Also:**[getDisabledIcon()](http://docs.google.com/javax/swing/JLabel.html#getDisabledIcon()), [JComponent.setEnabled(boolean)](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean))

### setDisplayedMnemonic

public void **setDisplayedMnemonic**(int key)

Specify a keycode that indicates a mnemonic key. This property is used when the label is part of a larger component. If the labelFor property of the label is not null, the label will call the requestFocus method of the component specified by the labelFor property when the mnemonic is activated.

**See Also:**[getLabelFor()](http://docs.google.com/javax/swing/JLabel.html#getLabelFor()), [setLabelFor(java.awt.Component)](http://docs.google.com/javax/swing/JLabel.html#setLabelFor(java.awt.Component))

### setDisplayedMnemonic

public void **setDisplayedMnemonic**(char aChar)

Specifies the displayedMnemonic as a char value.

**Parameters:**aChar - a char specifying the mnemonic to display**See Also:**[setDisplayedMnemonic(int)](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonic(int))

### getDisplayedMnemonic

public int **getDisplayedMnemonic**()

Return the keycode that indicates a mnemonic key. This property is used when the label is part of a larger component. If the labelFor property of the label is not null, the label will call the requestFocus method of the component specified by the labelFor property when the mnemonic is activated.

**Returns:**int value for the mnemonic key**See Also:**[getLabelFor()](http://docs.google.com/javax/swing/JLabel.html#getLabelFor()), [setLabelFor(java.awt.Component)](http://docs.google.com/javax/swing/JLabel.html#setLabelFor(java.awt.Component))

### setDisplayedMnemonicIndex

public void **setDisplayedMnemonicIndex**(int index)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Provides a hint to the look and feel as to which character in the text should be decorated to represent the mnemonic. Not all look and feels may support this. A value of -1 indicates either there is no mnemonic, the mnemonic character is not contained in the string, or the developer does not wish the mnemonic to be displayed.

The value of this is updated as the properties relating to the mnemonic change (such as the mnemonic itself, the text...). You should only ever have to call this if you do not wish the default character to be underlined. For example, if the text was 'Save As', with a mnemonic of 'a', and you wanted the 'A' to be decorated, as 'Save As', you would have to invoke setDisplayedMnemonicIndex(5) after invoking setDisplayedMnemonic(KeyEvent.VK\_A).

**Parameters:**index - Index into the String to underline **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - will be thrown if index= length of the text, or < -1**Since:** 1.4

### getDisplayedMnemonicIndex

public int **getDisplayedMnemonicIndex**()

Returns the character, as an index, that the look and feel should provide decoration for as representing the mnemonic character.

**Returns:**index representing mnemonic character**Since:** 1.4 **See Also:**[setDisplayedMnemonicIndex(int)](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonicIndex(int))

### checkHorizontalKey

protected int **checkHorizontalKey**(int key,  
 [String](http://docs.google.com/java/lang/String.html) message)

Verify that key is a legal value for the horizontalAlignment properties.

**Parameters:**key - the property value to checkmessage - the IllegalArgumentException detail message **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if key isn't LEFT, CENTER, RIGHT, LEADING or TRAILING.**See Also:**[setHorizontalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setHorizontalTextPosition(int)), [setHorizontalAlignment(int)](http://docs.google.com/javax/swing/JLabel.html#setHorizontalAlignment(int))

### checkVerticalKey

protected int **checkVerticalKey**(int key,  
 [String](http://docs.google.com/java/lang/String.html) message)

Verify that key is a legal value for the verticalAlignment or verticalTextPosition properties.

**Parameters:**key - the property value to checkmessage - the IllegalArgumentException detail message **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if key isn't TOP, CENTER, or BOTTOM.**See Also:**[setVerticalAlignment(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalAlignment(int)), [setVerticalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalTextPosition(int))

### getIconTextGap

public int **getIconTextGap**()

Returns the amount of space between the text and the icon displayed in this label.

**Returns:**an int equal to the number of pixels between the text and the icon.**See Also:**[setIconTextGap(int)](http://docs.google.com/javax/swing/JLabel.html#setIconTextGap(int))

### setIconTextGap

public void **setIconTextGap**(int iconTextGap)

If both the icon and text properties are set, this property defines the space between them.

The default value of this property is 4 pixels.

This is a JavaBeans bound property.

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

### getVerticalAlignment

public int **getVerticalAlignment**()

Returns the alignment of the label's contents along the Y axis.

**Returns:**The value of the verticalAlignment property, one of the following constants defined in SwingConstants: TOP, CENTER, or BOTTOM.**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [setVerticalAlignment(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalAlignment(int))

### setVerticalAlignment

public void **setVerticalAlignment**(int alignment)

Sets the alignment of the label's contents along the Y axis.

The default value of this property is CENTER.

**Parameters:**alignment - One of the following constants defined in SwingConstants: TOP, CENTER (the default), or BOTTOM.**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [getVerticalAlignment()](http://docs.google.com/javax/swing/JLabel.html#getVerticalAlignment())

### getHorizontalAlignment

public int **getHorizontalAlignment**()

Returns the alignment of the label's contents along the X axis.

**Returns:**The value of the horizontalAlignment property, one of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING or TRAILING.**See Also:**[setHorizontalAlignment(int)](http://docs.google.com/javax/swing/JLabel.html#setHorizontalAlignment(int)), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

### setHorizontalAlignment

public void **setHorizontalAlignment**(int alignment)

Sets the alignment of the label's contents along the X axis.

This is a JavaBeans bound property.

**Parameters:**alignment - One of the following constants defined in SwingConstants: LEFT, CENTER (the default for image-only labels), RIGHT, LEADING (the default for text-only labels) or TRAILING.**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [getHorizontalAlignment()](http://docs.google.com/javax/swing/JLabel.html#getHorizontalAlignment())

### getVerticalTextPosition

public int **getVerticalTextPosition**()

Returns the vertical position of the label's text, relative to its image.

**Returns:**One of the following constants defined in SwingConstants: TOP, CENTER, or BOTTOM.**See Also:**[setVerticalTextPosition(int)](http://docs.google.com/javax/swing/JLabel.html#setVerticalTextPosition(int)), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

### setVerticalTextPosition

public void **setVerticalTextPosition**(int textPosition)

Sets the vertical position of the label's text, relative to its image.

The default value of this property is CENTER.

This is a JavaBeans bound property.

**Parameters:**textPosition - One of the following constants defined in SwingConstants: TOP, CENTER (the default), or BOTTOM.**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html), [getVerticalTextPosition()](http://docs.google.com/javax/swing/JLabel.html#getVerticalTextPosition())

### getHorizontalTextPosition

public int **getHorizontalTextPosition**()

Returns the horizontal position of the label's text, relative to its image.

**Returns:**One of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING or TRAILING.**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

### setHorizontalTextPosition

public void **setHorizontalTextPosition**(int textPosition)

Sets the horizontal position of the label's text, relative to its image.

**Parameters:**textPosition - One of the following constants defined in SwingConstants: LEFT, CENTER, RIGHT, LEADING, or TRAILING (the default). **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)**See Also:**[SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

### imageUpdate

public boolean **imageUpdate**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int infoflags,  
 int x,  
 int y,  
 int w,  
 int h)

This is overridden to return false if the current Icon's Image is not equal to the passed in Image img.

**Specified by:**[imageUpdate](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)) in interface [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)**Overrides:**[imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**img - the image being observedinfoflags - see imageUpdate for more informationx - the *x* coordinatey - the *y* coordinatew - the widthh - the height **Returns:**false if the infoflags indicate that the image is completely loaded; true otherwise.**See Also:**[ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [Component.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### paramString

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

Returns a string representation of this JLabel. 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 JLabel.

### getLabelFor

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

Get the component this is labelling.

**Returns:**the Component this is labelling. Can be null if this does not label a Component. If the displayedMnemonic property is set and the labelFor property is also set, the label will call the requestFocus method of the component specified by the labelFor property when the mnemonic is activated.**See Also:**[getDisplayedMnemonic()](http://docs.google.com/javax/swing/JLabel.html#getDisplayedMnemonic()), [setDisplayedMnemonic(int)](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonic(int))

### setLabelFor

public void **setLabelFor**([Component](http://docs.google.com/java/awt/Component.html) c)

Set the component this is labelling. Can be null if this does not label a Component. If the displayedMnemonic property is set and the labelFor property is also set, the label will call the requestFocus method of the component specified by the labelFor property when the mnemonic is activated.

**Parameters:**c - the Component this label is for, or null if the label is not the label for a component**See Also:**[getDisplayedMnemonic()](http://docs.google.com/javax/swing/JLabel.html#getDisplayedMnemonic()), [setDisplayedMnemonic(int)](http://docs.google.com/javax/swing/JLabel.html#setDisplayedMnemonic(int))

### getAccessibleContext

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

Get the AccessibleContext of this object

**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:**the AccessibleContext of this object

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JLabel.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/JInternalFrame.JDesktopIcon.AccessibleJDesktopIcon.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JLabel.AccessibleJLabel.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JLabel.html)    [**NO FRAMES**](http://docs.google.com/JLabel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#1y810tw) | [METHOD](#3as4poj) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).