| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TextComponent.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/TextArea.AccessibleAWTTextArea.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextComponent.AccessibleAWTTextComponent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextComponent.html)    [**NO FRAMES**](http://docs.google.com/TextComponent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | CONSTR | [METHOD](#4d34og8) | DETAIL: [FIELD](#3rdcrjn) | CONSTR | [METHOD](#lnxbz9) |

## **java.awt**

Class TextComponent

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

**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) **Direct Known Subclasses:** [TextArea](http://docs.google.com/java/awt/TextArea.html), [TextField](http://docs.google.com/java/awt/TextField.html)

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

The TextComponent class is the superclass of any component that allows the editing of some text.

A text component embodies a string of text. The TextComponent class defines a set of methods that determine whether or not this text is editable. If the component is editable, it defines another set of methods that supports a text insertion caret.

In addition, the class defines methods that are used to maintain a current *selection* from the text. The text selection, a substring of the component's text, is the target of editing operations. It is also referred to as the *selected text*.

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

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

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| protected  [TextListener](http://docs.google.com/java/awt/event/TextListener.html) | [**textListener**](http://docs.google.com/java/awt/TextComponent.html#textListener) |

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

| **Fields inherited from interface java.awt.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) |

| **Method Summary** | |
| --- | --- |
| void | [**addNotify**](http://docs.google.com/java/awt/TextComponent.html#addNotify())()            Makes this Component displayable by connecting it to a native screen resource. |
| void | [**addTextListener**](http://docs.google.com/java/awt/TextComponent.html#addTextListener(java.awt.event.TextListener))([TextListener](http://docs.google.com/java/awt/event/TextListener.html) l)            Adds the specified text event listener to receive text events from this text component. |
| void | [**enableInputMethods**](http://docs.google.com/java/awt/TextComponent.html#enableInputMethods(boolean))(boolean enable)            Enables or disables input method support for this text component. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/TextComponent.html#getAccessibleContext())()            Gets the AccessibleContext associated with this TextComponent. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/java/awt/TextComponent.html#getBackground())()            Gets the background color of this text component. |
| int | [**getCaretPosition**](http://docs.google.com/java/awt/TextComponent.html#getCaretPosition())()            Returns the position of the text insertion caret. |
| [InputMethodRequests](http://docs.google.com/java/awt/im/InputMethodRequests.html) | [**getInputMethodRequests**](http://docs.google.com/java/awt/TextComponent.html#getInputMethodRequests())()            Gets the input method request handler which supports requests from input methods for this component. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/TextComponent.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this TextComponent. |
| [String](http://docs.google.com/java/lang/String.html) | [**getSelectedText**](http://docs.google.com/java/awt/TextComponent.html#getSelectedText())()            Returns the selected text from the text that is presented by this text component. |
| int | [**getSelectionEnd**](http://docs.google.com/java/awt/TextComponent.html#getSelectionEnd())()            Gets the end position of the selected text in this text component. |
| int | [**getSelectionStart**](http://docs.google.com/java/awt/TextComponent.html#getSelectionStart())()            Gets the start position of the selected text in this text component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getText**](http://docs.google.com/java/awt/TextComponent.html#getText())()            Returns the text that is presented by this text component. |
| [TextListener](http://docs.google.com/java/awt/event/TextListener.html)[] | [**getTextListeners**](http://docs.google.com/java/awt/TextComponent.html#getTextListeners())()            Returns an array of all the text listeners registered on this text component. |
| boolean | [**isEditable**](http://docs.google.com/java/awt/TextComponent.html#isEditable())()            Indicates whether or not this text component is editable. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/TextComponent.html#paramString())()            Returns a string representing the state of this TextComponent. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/TextComponent.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events on this text component. |
| protected  void | [**processTextEvent**](http://docs.google.com/java/awt/TextComponent.html#processTextEvent(java.awt.event.TextEvent))([TextEvent](http://docs.google.com/java/awt/event/TextEvent.html) e)            Processes text events occurring on this text component by dispatching them to any registered TextListener objects. |
| void | [**removeNotify**](http://docs.google.com/java/awt/TextComponent.html#removeNotify())()            Removes the TextComponent's peer. |
| void | [**removeTextListener**](http://docs.google.com/java/awt/TextComponent.html#removeTextListener(java.awt.event.TextListener))([TextListener](http://docs.google.com/java/awt/event/TextListener.html) l)            Removes the specified text event listener so that it no longer receives text events from this text component If l is null, no exception is thrown and no action is performed. |
| void | [**select**](http://docs.google.com/java/awt/TextComponent.html#select(int,%20int))(int selectionStart, int selectionEnd)            Selects the text between the specified start and end positions. |
| void | [**selectAll**](http://docs.google.com/java/awt/TextComponent.html#selectAll())()            Selects all the text in this text component. |
| void | [**setBackground**](http://docs.google.com/java/awt/TextComponent.html#setBackground(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) c)            Sets the background color of this text component. |
| void | [**setCaretPosition**](http://docs.google.com/java/awt/TextComponent.html#setCaretPosition(int))(int position)            Sets the position of the text insertion caret. |
| void | [**setEditable**](http://docs.google.com/java/awt/TextComponent.html#setEditable(boolean))(boolean b)            Sets the flag that determines whether or not this text component is editable. |
| void | [**setSelectionEnd**](http://docs.google.com/java/awt/TextComponent.html#setSelectionEnd(int))(int selectionEnd)            Sets the selection end for this text component to the specified position. |
| void | [**setSelectionStart**](http://docs.google.com/java/awt/TextComponent.html#setSelectionStart(int))(int selectionStart)            Sets the selection start for this text component to the specified position. |
| void | [**setText**](http://docs.google.com/java/awt/TextComponent.html#setText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) t)            Sets the text that is presented by this text component to be the specified text. |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [applyComponentOrientation](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Component.html#areFocusTraversalKeysSet(int)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [deliverEvent](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event)), [disable](http://docs.google.com/java/awt/Component.html#disable()), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [doLayout](http://docs.google.com/java/awt/Component.html#doLayout()), [enable](http://docs.google.com/java/awt/Component.html#enable()), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20int,%20int)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getAlignmentX](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [getBaseline](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle)), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point)), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int)), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getFontMetrics](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font)), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHeight](http://docs.google.com/java/awt/Component.html#getHeight()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point)), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMaximumSize](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPreferredSize](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getSize](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension)), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [getWidth](http://docs.google.com/java/awt/Component.html#getWidth()), [getX](http://docs.google.com/java/awt/Component.html#getX()), [getY](http://docs.google.com/java/awt/Component.html#getY()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [invalidate](http://docs.google.com/java/awt/Component.html#invalidate()), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isDoubleBuffered](http://docs.google.com/java/awt/Component.html#isDoubleBuffered()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusCycleRoot](http://docs.google.com/java/awt/Component.html#isFocusCycleRoot(java.awt.Container)), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isOpaque](http://docs.google.com/java/awt/Component.html#isOpaque()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [layout](http://docs.google.com/java/awt/Component.html#layout()), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Component.html#locate(int,%20int)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [minimumSize](http://docs.google.com/java/awt/Component.html#minimumSize()), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paint](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [preferredSize](http://docs.google.com/java/awt/Component.html#preferredSize()), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [print](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processKeyEvent](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus()), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean)), [reshape](http://docs.google.com/java/awt/Component.html#reshape(int,%20int,%20int,%20int)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setEnabled](http://docs.google.com/java/awt/Component.html#setEnabled(boolean)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setFont](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setMaximumSize](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setPreferredSize](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [setVisible](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusBackward](http://docs.google.com/java/awt/Component.html#transferFocusBackward()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()), [update](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics)), [validate](http://docs.google.com/java/awt/Component.html#validate()) |

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

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

### textListener

protected transient [TextListener](http://docs.google.com/java/awt/event/TextListener.html) **textListener**

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

### enableInputMethods

public void **enableInputMethods**(boolean enable)

Enables or disables input method support for this text component. If input method support is enabled and the text component also processes key events, incoming events are offered to the current input method and will only be processed by the component or dispatched to its listeners if the input method does not consume them. Whether and how input method support for this text component is enabled or disabled by default is implementation dependent.

**Overrides:**[enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**enable - true to enable, false to disable**Since:** 1.2 **See Also:**[Component.processKeyEvent(java.awt.event.KeyEvent)](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent))

### getInputMethodRequests

public [InputMethodRequests](http://docs.google.com/java/awt/im/InputMethodRequests.html) **getInputMethodRequests**()

**Description copied from class:** [**Component**](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()) Gets the input method request handler which supports requests from input methods for this component. A component that supports on-the-spot text input must override this method to return an InputMethodRequests instance. At the same time, it also has to handle input method events.

**Overrides:**[getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**the input method request handler for this component, null by default**See Also:**[Component.addInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener))

### addNotify

public void **addNotify**()

Makes this Component displayable by connecting it to a native screen resource. This method is called internally by the toolkit and should not be called directly by programs.

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

### removeNotify

public void **removeNotify**()

Removes the TextComponent's peer. The peer allows us to modify the appearance of the TextComponent without changing its functionality.

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

### setText

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

Sets the text that is presented by this text component to be the specified text.

**Parameters:**t - the new text; if this parameter is null then the text is set to the empty string ""**See Also:**[getText()](http://docs.google.com/java/awt/TextComponent.html#getText())

### getText

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

Returns the text that is presented by this text component. By default, this is an empty string.

**Returns:**the value of this TextComponent**See Also:**[setText(java.lang.String)](http://docs.google.com/java/awt/TextComponent.html#setText(java.lang.String))

### getSelectedText

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

Returns the selected text from the text that is presented by this text component.

**Returns:**the selected text of this text component**See Also:**[select(int, int)](http://docs.google.com/java/awt/TextComponent.html#select(int,%20int))

### isEditable

public boolean **isEditable**()

Indicates whether or not this text component is editable.

**Returns:**true if this text component is editable; false otherwise.**Since:** JDK1.0 **See Also:**[setEditable(boolean)](http://docs.google.com/java/awt/TextComponent.html#setEditable(boolean))

### setEditable

public void **setEditable**(boolean b)

Sets the flag that determines whether or not this text component is editable.

If the flag is set to true, this text component becomes user editable. If the flag is set to false, the user cannot change the text of this text component. By default, non-editable text components have a background color of SystemColor.control. This default can be overridden by calling setBackground.

**Parameters:**b - a flag indicating whether this text component is user editable.**Since:** JDK1.0 **See Also:**[isEditable()](http://docs.google.com/java/awt/TextComponent.html#isEditable())

### getBackground

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

Gets the background color of this text component. By default, non-editable text components have a background color of SystemColor.control. This default can be overridden by calling setBackground.

**Overrides:**[getBackground](http://docs.google.com/java/awt/Component.html#getBackground()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**This text component's background color. If this text component does not have a background color, the background color of its parent is returned.**Since:** JDK1.0 **See Also:**[setBackground(Color)](http://docs.google.com/java/awt/TextComponent.html#setBackground(java.awt.Color))

### setBackground

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

Sets the background color of this text component.

**Overrides:**[setBackground](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**c - The color to become this text component's color. If this parameter is null then this text component will inherit the background color of its parent.**Since:** JDK1.0 **See Also:**[getBackground()](http://docs.google.com/java/awt/TextComponent.html#getBackground())

### getSelectionStart

public int **getSelectionStart**()

Gets the start position of the selected text in this text component.

**Returns:**the start position of the selected text**See Also:**[setSelectionStart(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionStart(int)), [getSelectionEnd()](http://docs.google.com/java/awt/TextComponent.html#getSelectionEnd())

### setSelectionStart

public void **setSelectionStart**(int selectionStart)

Sets the selection start for this text component to the specified position. The new start point is constrained to be at or before the current selection end. It also cannot be set to less than zero, the beginning of the component's text. If the caller supplies a value for selectionStart that is out of bounds, the method enforces these constraints silently, and without failure.

**Parameters:**selectionStart - the start position of the selected text**Since:** JDK1.1 **See Also:**[getSelectionStart()](http://docs.google.com/java/awt/TextComponent.html#getSelectionStart()), [setSelectionEnd(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionEnd(int))

### getSelectionEnd

public int **getSelectionEnd**()

Gets the end position of the selected text in this text component.

**Returns:**the end position of the selected text**See Also:**[setSelectionEnd(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionEnd(int)), [getSelectionStart()](http://docs.google.com/java/awt/TextComponent.html#getSelectionStart())

### setSelectionEnd

public void **setSelectionEnd**(int selectionEnd)

Sets the selection end for this text component to the specified position. The new end point is constrained to be at or after the current selection start. It also cannot be set beyond the end of the component's text. If the caller supplies a value for selectionEnd that is out of bounds, the method enforces these constraints silently, and without failure.

**Parameters:**selectionEnd - the end position of the selected text**Since:** JDK1.1 **See Also:**[getSelectionEnd()](http://docs.google.com/java/awt/TextComponent.html#getSelectionEnd()), [setSelectionStart(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionStart(int))

### select

public void **select**(int selectionStart,  
 int selectionEnd)

Selects the text between the specified start and end positions.

This method sets the start and end positions of the selected text, enforcing the restriction that the start position must be greater than or equal to zero. The end position must be greater than or equal to the start position, and less than or equal to the length of the text component's text. The character positions are indexed starting with zero. The length of the selection is endPosition - startPosition, so the character at endPosition is not selected. If the start and end positions of the selected text are equal, all text is deselected.

If the caller supplies values that are inconsistent or out of bounds, the method enforces these constraints silently, and without failure. Specifically, if the start position or end position is greater than the length of the text, it is reset to equal the text length. If the start position is less than zero, it is reset to zero, and if the end position is less than the start position, it is reset to the start position.

**Parameters:**selectionStart - the zero-based index of the first character (char value) to be selectedselectionEnd - the zero-based end position of the text to be selected; the character (char value) at selectionEnd is not selected**See Also:**[setSelectionStart(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionStart(int)), [setSelectionEnd(int)](http://docs.google.com/java/awt/TextComponent.html#setSelectionEnd(int)), [selectAll()](http://docs.google.com/java/awt/TextComponent.html#selectAll())

### selectAll

public void **selectAll**()

Selects all the text in this text component.

**See Also:**[select(int, int)](http://docs.google.com/java/awt/TextComponent.html#select(int,%20int))

### setCaretPosition

public void **setCaretPosition**(int position)

Sets the position of the text insertion caret. The caret position is constrained to be between 0 and the last character of the text, inclusive. If the passed-in value is greater than this range, the value is set to the last character (or 0 if the TextComponent contains no text) and no error is returned. If the passed-in value is less than 0, an IllegalArgumentException is thrown.

**Parameters:**position - the position of the text insertion caret **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if position is less than zero**Since:** JDK1.1

### getCaretPosition

public int **getCaretPosition**()

Returns the position of the text insertion caret. The caret position is constrained to be between 0 and the last character of the text, inclusive. If the text or caret have not been set, the default caret position is 0.

**Returns:**the position of the text insertion caret**Since:** JDK1.1 **See Also:**[setCaretPosition(int)](http://docs.google.com/java/awt/TextComponent.html#setCaretPosition(int))

### addTextListener

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

Adds the specified text event listener to receive text events from this text component. If l is null, no exception is thrown and no action is performed.

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

**Parameters:**l - the text event listener**See Also:**[removeTextListener(java.awt.event.TextListener)](http://docs.google.com/java/awt/TextComponent.html#removeTextListener(java.awt.event.TextListener)), [getTextListeners()](http://docs.google.com/java/awt/TextComponent.html#getTextListeners()), [TextListener](http://docs.google.com/java/awt/event/TextListener.html)

### removeTextListener

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

Removes the specified text event listener so that it no longer receives text events from this text component If l is null, no exception is thrown and no action is performed.

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

**Parameters:**l - the text listener**Since:** JDK1.1 **See Also:**[addTextListener(java.awt.event.TextListener)](http://docs.google.com/java/awt/TextComponent.html#addTextListener(java.awt.event.TextListener)), [getTextListeners()](http://docs.google.com/java/awt/TextComponent.html#getTextListeners()), [TextListener](http://docs.google.com/java/awt/event/TextListener.html)

### getTextListeners

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

Returns an array of all the text listeners registered on this text component.

**Returns:**all of this text component's TextListeners or an empty array if no text listeners are currently registered**Since:** 1.4 **See Also:**[addTextListener(java.awt.event.TextListener)](http://docs.google.com/java/awt/TextComponent.html#addTextListener(java.awt.event.TextListener)), [removeTextListener(java.awt.event.TextListener)](http://docs.google.com/java/awt/TextComponent.html#removeTextListener(java.awt.event.TextListener))

### getListeners

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

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

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

TextListener[] tls = (TextListener[])(t.getListeners(TextListener.class));

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

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

### processEvent

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

Processes events on this text component. If the event is a TextEvent, it invokes the processTextEvent method else it invokes its superclass's processEvent.

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

**Overrides:**[processEvent](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**e - the event**See Also:**[Component.processComponentEvent(java.awt.event.ComponentEvent)](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [Component.processFocusEvent(java.awt.event.FocusEvent)](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [Component.processKeyEvent(java.awt.event.KeyEvent)](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [Component.processMouseEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [Component.processMouseMotionEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [Component.processInputMethodEvent(java.awt.event.InputMethodEvent)](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [Component.processHierarchyEvent(java.awt.event.HierarchyEvent)](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [Component.processMouseWheelEvent(java.awt.event.MouseWheelEvent)](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent))

### processTextEvent

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

Processes text events occurring on this text component by dispatching them to any registered TextListener objects.

NOTE: This method will not be called unless text events are enabled for this component. This happens when one of the following occurs:

* A TextListener object is registered via addTextListener
* Text events are enabled via enableEvents

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

**Parameters:**e - the text event**See Also:**[Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### paramString

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

Returns a string representing the state of this TextComponent. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Overrides:**[paramString](http://docs.google.com/java/awt/Component.html#paramString()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**the parameter string of this text component

### getAccessibleContext

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

Gets the AccessibleContext associated with this TextComponent. For text components, the AccessibleContext takes the form of an AccessibleAWTTextComponent. A new AccessibleAWTTextComponent instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/java/awt/Component.html#getAccessibleContext()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**an AccessibleAWTTextComponent that serves as the AccessibleContext of this TextComponent**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TextComponent.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/TextArea.AccessibleAWTTextArea.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextComponent.AccessibleAWTTextComponent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextComponent.html)    [**NO FRAMES**](http://docs.google.com/TextComponent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#tyjcwt) | CONSTR | [METHOD](#4d34og8) | DETAIL: [FIELD](#3rdcrjn) | CONSTR | [METHOD](#lnxbz9) |

[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).