| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTextArea.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/JTable.PrintMode.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTextArea.AccessibleJTextArea.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTextArea.html)    [**NO FRAMES**](http://docs.google.com/JTextArea.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: FIELD | [CONSTR](#1y810tw) | [METHOD](#3as4poj) |

## **javax.swing**

Class JTextArea

[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.text.JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html)  
 **javax.swing.JTextArea**

**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), [Scrollable](http://docs.google.com/javax/swing/Scrollable.html)

public class **JTextArea**extends [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html)

A JTextArea is a multi-line area that displays plain text. It is intended to be a lightweight component that provides source compatibility with the java.awt.TextArea class where it can reasonably do so. You can find information and examples of using all the text components in [Using Text Components](http://java.sun.com/docs/books/tutorial/uiswing/components/text.html), a section in *The Java Tutorial.*

This component has capabilities not found in the java.awt.TextArea class. The superclass should be consulted for additional capabilities. Alternative multi-line text classes with more capabilities are JTextPane and JEditorPane.

The java.awt.TextArea internally handles scrolling. JTextArea is different in that it doesn't manage scrolling, but implements the swing Scrollable interface. This allows it to be placed inside a JScrollPane if scrolling behavior is desired, and used directly if scrolling is not desired.

The java.awt.TextArea has the ability to do line wrapping. This was controlled by the horizontal scrolling policy. Since scrolling is not done by JTextArea directly, backward compatibility must be provided another way. JTextArea has a bound property for line wrapping that controls whether or not it will wrap lines. By default, the line wrapping property is set to false (not wrapped).

java.awt.TextArea has two properties rows and columns that are used to determine the preferred size. JTextArea uses these properties to indicate the preferred size of the viewport when placed inside a JScrollPane to match the functionality provided by java.awt.TextArea. JTextArea has a preferred size of what is needed to display all of the text, so that it functions properly inside of a JScrollPane. If the value for rows or columns is equal to zero, the preferred size along that axis is used for the viewport preferred size along the same axis.

The java.awt.TextArea could be monitored for changes by adding a TextListener for TextEvents. In the JTextComponent based components, changes are broadcasted from the model via a DocumentEvent to DocumentListeners. The DocumentEvent gives the location of the change and the kind of change if desired. The code fragment might look something like:

DocumentListener myListener = ??;  
 JTextArea myArea = ??;  
 myArea.getDocument().addDocumentListener(myListener);

**Newlines** For a discussion on how newlines are handled, see [DefaultEditorKit](http://docs.google.com/text/DefaultEditorKit.html).

**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).

**See Also:**[JTextPane](http://docs.google.com/javax/swing/JTextPane.html), [JEditorPane](http://docs.google.com/javax/swing/JEditorPane.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JTextArea.AccessibleJTextArea**](http://docs.google.com/javax/swing/JTextArea.AccessibleJTextArea.html)            This class implements accessibility support for the JTextArea class. |

| **Nested classes/interfaces inherited from class javax.swing.text.**[**JTextComponent**](http://docs.google.com/javax/swing/text/JTextComponent.html) |
| --- |
| [JTextComponent.AccessibleJTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.AccessibleJTextComponent.html), [JTextComponent.DropLocation](http://docs.google.com/javax/swing/text/JTextComponent.DropLocation.html), [JTextComponent.KeyBinding](http://docs.google.com/javax/swing/text/JTextComponent.KeyBinding.html) |

| **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** | |
| --- | --- |

| **Fields inherited from class javax.swing.text.**[**JTextComponent**](http://docs.google.com/javax/swing/text/JTextComponent.html) |
| --- |
| [DEFAULT\_KEYMAP](http://docs.google.com/javax/swing/text/JTextComponent.html#DEFAULT_KEYMAP), [FOCUS\_ACCELERATOR\_KEY](http://docs.google.com/javax/swing/text/JTextComponent.html#FOCUS_ACCELERATOR_KEY) |

| **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 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** | |
| --- | --- |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea())()            Constructs a new TextArea. |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea(javax.swing.text.Document))([Document](http://docs.google.com/javax/swing/text/Document.html) doc)            Constructs a new JTextArea with the given document model, and defaults for all of the other arguments (null, 0, 0). |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea(javax.swing.text.Document,%20java.lang.String,%20int,%20int))([Document](http://docs.google.com/javax/swing/text/Document.html) doc, [String](http://docs.google.com/java/lang/String.html) text, int rows, int columns)            Constructs a new JTextArea with the specified number of rows and columns, and the given model. |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea(int,%20int))(int rows, int columns)            Constructs a new empty TextArea with the specified number of rows and columns. |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Constructs a new TextArea with the specified text displayed. |
| [**JTextArea**](http://docs.google.com/javax/swing/JTextArea.html#JTextArea(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) text, int rows, int columns)            Constructs a new TextArea with the specified text and number of rows and columns. |

| **Method Summary** | |
| --- | --- |
| void | [**append**](http://docs.google.com/javax/swing/JTextArea.html#append(java.lang.String))([String](http://docs.google.com/java/lang/String.html) str)            Appends the given text to the end of the document. |
| protected  [Document](http://docs.google.com/javax/swing/text/Document.html) | [**createDefaultModel**](http://docs.google.com/javax/swing/JTextArea.html#createDefaultModel())()            Creates the default implementation of the model to be used at construction if one isn't explicitly given. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JTextArea.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JTextArea. |
| int | [**getColumns**](http://docs.google.com/javax/swing/JTextArea.html#getColumns())()            Returns the number of columns in the TextArea. |
| protected  int | [**getColumnWidth**](http://docs.google.com/javax/swing/JTextArea.html#getColumnWidth())()            Gets column width. |
| int | [**getLineCount**](http://docs.google.com/javax/swing/JTextArea.html#getLineCount())()            Determines the number of lines contained in the area. |
| int | [**getLineEndOffset**](http://docs.google.com/javax/swing/JTextArea.html#getLineEndOffset(int))(int line)            Determines the offset of the end of the given line. |
| int | [**getLineOfOffset**](http://docs.google.com/javax/swing/JTextArea.html#getLineOfOffset(int))(int offset)            Translates an offset into the components text to a line number. |
| int | [**getLineStartOffset**](http://docs.google.com/javax/swing/JTextArea.html#getLineStartOffset(int))(int line)            Determines the offset of the start of the given line. |
| boolean | [**getLineWrap**](http://docs.google.com/javax/swing/JTextArea.html#getLineWrap())()            Gets the line-wrapping policy of the text area. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredScrollableViewportSize**](http://docs.google.com/javax/swing/JTextArea.html#getPreferredScrollableViewportSize())()            Returns the preferred size of the viewport if this component is embedded in a JScrollPane. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/JTextArea.html#getPreferredSize())()            Returns the preferred size of the TextArea. |
| protected  int | [**getRowHeight**](http://docs.google.com/javax/swing/JTextArea.html#getRowHeight())()            Defines the meaning of the height of a row. |
| int | [**getRows**](http://docs.google.com/javax/swing/JTextArea.html#getRows())()            Returns the number of rows in the TextArea. |
| boolean | [**getScrollableTracksViewportWidth**](http://docs.google.com/javax/swing/JTextArea.html#getScrollableTracksViewportWidth())()            Returns true if a viewport should always force the width of this Scrollable to match the width of the viewport. |
| int | [**getScrollableUnitIncrement**](http://docs.google.com/javax/swing/JTextArea.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect, int orientation, int direction)            Components that display logical rows or columns should compute the scroll increment that will completely expose one new row or column, depending on the value of orientation. |
| int | [**getTabSize**](http://docs.google.com/javax/swing/JTextArea.html#getTabSize())()            Gets the number of characters used to expand tabs. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JTextArea.html#getUIClassID())()            Returns the class ID for the UI. |
| boolean | [**getWrapStyleWord**](http://docs.google.com/javax/swing/JTextArea.html#getWrapStyleWord())()            Gets the style of wrapping used if the text area is wrapping lines. |
| void | [**insert**](http://docs.google.com/javax/swing/JTextArea.html#insert(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) str, int pos)            Inserts the specified text at the specified position. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JTextArea.html#paramString())()            Returns a string representation of this JTextArea. |
| void | [**replaceRange**](http://docs.google.com/javax/swing/JTextArea.html#replaceRange(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) str, int start, int end)            Replaces text from the indicated start to end position with the new text specified. |
| void | [**setColumns**](http://docs.google.com/javax/swing/JTextArea.html#setColumns(int))(int columns)            Sets the number of columns for this TextArea. |
| void | [**setFont**](http://docs.google.com/javax/swing/JTextArea.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Sets the current font. |
| void | [**setLineWrap**](http://docs.google.com/javax/swing/JTextArea.html#setLineWrap(boolean))(boolean wrap)            Sets the line-wrapping policy of the text area. |
| void | [**setRows**](http://docs.google.com/javax/swing/JTextArea.html#setRows(int))(int rows)            Sets the number of rows for this TextArea. |
| void | [**setTabSize**](http://docs.google.com/javax/swing/JTextArea.html#setTabSize(int))(int size)            Sets the number of characters to expand tabs to. |
| void | [**setWrapStyleWord**](http://docs.google.com/javax/swing/JTextArea.html#setWrapStyleWord(boolean))(boolean word)            Sets the style of wrapping used if the text area is wrapping lines. |

| **Methods inherited from class javax.swing.text.**[**JTextComponent**](http://docs.google.com/javax/swing/text/JTextComponent.html) |
| --- |
| [addCaretListener](http://docs.google.com/javax/swing/text/JTextComponent.html#addCaretListener(javax.swing.event.CaretListener)), [addInputMethodListener](http://docs.google.com/javax/swing/text/JTextComponent.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#addKeymap(java.lang.String,%20javax.swing.text.Keymap)), [copy](http://docs.google.com/javax/swing/text/JTextComponent.html#copy()), [cut](http://docs.google.com/javax/swing/text/JTextComponent.html#cut()), [fireCaretUpdate](http://docs.google.com/javax/swing/text/JTextComponent.html#fireCaretUpdate(javax.swing.event.CaretEvent)), [getActions](http://docs.google.com/javax/swing/text/JTextComponent.html#getActions()), [getCaret](http://docs.google.com/javax/swing/text/JTextComponent.html#getCaret()), [getCaretColor](http://docs.google.com/javax/swing/text/JTextComponent.html#getCaretColor()), [getCaretListeners](http://docs.google.com/javax/swing/text/JTextComponent.html#getCaretListeners()), [getCaretPosition](http://docs.google.com/javax/swing/text/JTextComponent.html#getCaretPosition()), [getDisabledTextColor](http://docs.google.com/javax/swing/text/JTextComponent.html#getDisabledTextColor()), [getDocument](http://docs.google.com/javax/swing/text/JTextComponent.html#getDocument()), [getDragEnabled](http://docs.google.com/javax/swing/text/JTextComponent.html#getDragEnabled()), [getDropLocation](http://docs.google.com/javax/swing/text/JTextComponent.html#getDropLocation()), [getDropMode](http://docs.google.com/javax/swing/text/JTextComponent.html#getDropMode()), [getFocusAccelerator](http://docs.google.com/javax/swing/text/JTextComponent.html#getFocusAccelerator()), [getHighlighter](http://docs.google.com/javax/swing/text/JTextComponent.html#getHighlighter()), [getInputMethodRequests](http://docs.google.com/javax/swing/text/JTextComponent.html#getInputMethodRequests()), [getKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#getKeymap()), [getKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#getKeymap(java.lang.String)), [getMargin](http://docs.google.com/javax/swing/text/JTextComponent.html#getMargin()), [getNavigationFilter](http://docs.google.com/javax/swing/text/JTextComponent.html#getNavigationFilter()), [getPrintable](http://docs.google.com/javax/swing/text/JTextComponent.html#getPrintable(java.text.MessageFormat,%20java.text.MessageFormat)), [getScrollableBlockIncrement](http://docs.google.com/javax/swing/text/JTextComponent.html#getScrollableBlockIncrement(java.awt.Rectangle,%20int,%20int)), [getScrollableTracksViewportHeight](http://docs.google.com/javax/swing/text/JTextComponent.html#getScrollableTracksViewportHeight()), [getSelectedText](http://docs.google.com/javax/swing/text/JTextComponent.html#getSelectedText()), [getSelectedTextColor](http://docs.google.com/javax/swing/text/JTextComponent.html#getSelectedTextColor()), [getSelectionColor](http://docs.google.com/javax/swing/text/JTextComponent.html#getSelectionColor()), [getSelectionEnd](http://docs.google.com/javax/swing/text/JTextComponent.html#getSelectionEnd()), [getSelectionStart](http://docs.google.com/javax/swing/text/JTextComponent.html#getSelectionStart()), [getText](http://docs.google.com/javax/swing/text/JTextComponent.html#getText()), [getText](http://docs.google.com/javax/swing/text/JTextComponent.html#getText(int,%20int)), [getToolTipText](http://docs.google.com/javax/swing/text/JTextComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getUI](http://docs.google.com/javax/swing/text/JTextComponent.html#getUI()), [isEditable](http://docs.google.com/javax/swing/text/JTextComponent.html#isEditable()), [loadKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#loadKeymap(javax.swing.text.Keymap,%20javax.swing.text.JTextComponent.KeyBinding%5B%5D,%20javax.swing.Action%5B%5D)), [modelToView](http://docs.google.com/javax/swing/text/JTextComponent.html#modelToView(int)), [moveCaretPosition](http://docs.google.com/javax/swing/text/JTextComponent.html#moveCaretPosition(int)), [paste](http://docs.google.com/javax/swing/text/JTextComponent.html#paste()), [print](http://docs.google.com/javax/swing/text/JTextComponent.html#print()), [print](http://docs.google.com/javax/swing/text/JTextComponent.html#print(java.text.MessageFormat,%20java.text.MessageFormat)), [print](http://docs.google.com/javax/swing/text/JTextComponent.html#print(java.text.MessageFormat,%20java.text.MessageFormat,%20boolean,%20javax.print.PrintService,%20javax.print.attribute.PrintRequestAttributeSet,%20boolean)), [processInputMethodEvent](http://docs.google.com/javax/swing/text/JTextComponent.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [read](http://docs.google.com/javax/swing/text/JTextComponent.html#read(java.io.Reader,%20java.lang.Object)), [removeCaretListener](http://docs.google.com/javax/swing/text/JTextComponent.html#removeCaretListener(javax.swing.event.CaretListener)), [removeKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#removeKeymap(java.lang.String)), [removeNotify](http://docs.google.com/javax/swing/text/JTextComponent.html#removeNotify()), [replaceSelection](http://docs.google.com/javax/swing/text/JTextComponent.html#replaceSelection(java.lang.String)), [select](http://docs.google.com/javax/swing/text/JTextComponent.html#select(int,%20int)), [selectAll](http://docs.google.com/javax/swing/text/JTextComponent.html#selectAll()), [setCaret](http://docs.google.com/javax/swing/text/JTextComponent.html#setCaret(javax.swing.text.Caret)), [setCaretColor](http://docs.google.com/javax/swing/text/JTextComponent.html#setCaretColor(java.awt.Color)), [setCaretPosition](http://docs.google.com/javax/swing/text/JTextComponent.html#setCaretPosition(int)), [setComponentOrientation](http://docs.google.com/javax/swing/text/JTextComponent.html#setComponentOrientation(java.awt.ComponentOrientation)), [setDisabledTextColor](http://docs.google.com/javax/swing/text/JTextComponent.html#setDisabledTextColor(java.awt.Color)), [setDocument](http://docs.google.com/javax/swing/text/JTextComponent.html#setDocument(javax.swing.text.Document)), [setDragEnabled](http://docs.google.com/javax/swing/text/JTextComponent.html#setDragEnabled(boolean)), [setDropMode](http://docs.google.com/javax/swing/text/JTextComponent.html#setDropMode(javax.swing.DropMode)), [setEditable](http://docs.google.com/javax/swing/text/JTextComponent.html#setEditable(boolean)), [setFocusAccelerator](http://docs.google.com/javax/swing/text/JTextComponent.html#setFocusAccelerator(char)), [setHighlighter](http://docs.google.com/javax/swing/text/JTextComponent.html#setHighlighter(javax.swing.text.Highlighter)), [setKeymap](http://docs.google.com/javax/swing/text/JTextComponent.html#setKeymap(javax.swing.text.Keymap)), [setMargin](http://docs.google.com/javax/swing/text/JTextComponent.html#setMargin(java.awt.Insets)), [setNavigationFilter](http://docs.google.com/javax/swing/text/JTextComponent.html#setNavigationFilter(javax.swing.text.NavigationFilter)), [setSelectedTextColor](http://docs.google.com/javax/swing/text/JTextComponent.html#setSelectedTextColor(java.awt.Color)), [setSelectionColor](http://docs.google.com/javax/swing/text/JTextComponent.html#setSelectionColor(java.awt.Color)), [setSelectionEnd](http://docs.google.com/javax/swing/text/JTextComponent.html#setSelectionEnd(int)), [setSelectionStart](http://docs.google.com/javax/swing/text/JTextComponent.html#setSelectionStart(int)), [setText](http://docs.google.com/javax/swing/text/JTextComponent.html#setText(java.lang.String)), [setUI](http://docs.google.com/javax/swing/text/JTextComponent.html#setUI(javax.swing.plaf.TextUI)), [updateUI](http://docs.google.com/javax/swing/text/JTextComponent.html#updateUI()), [viewToModel](http://docs.google.com/javax/swing/text/JTextComponent.html#viewToModel(java.awt.Point)), [write](http://docs.google.com/javax/swing/text/JTextComponent.html#write(java.io.Writer)) |

| **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)), [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()), [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)), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

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

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [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()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [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)), [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)) |

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

### JTextArea

public **JTextArea**()

Constructs a new TextArea. A default model is set, the initial string is null, and rows/columns are set to 0.

### JTextArea

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

Constructs a new TextArea with the specified text displayed. A default model is created and rows/columns are set to 0.

**Parameters:**text - the text to be displayed, or null

### JTextArea

public **JTextArea**(int rows,  
 int columns)

Constructs a new empty TextArea with the specified number of rows and columns. A default model is created, and the initial string is null.

**Parameters:**rows - the number of rows >= 0columns - the number of columns >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the rows or columns arguments are negative.

### JTextArea

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

Constructs a new TextArea with the specified text and number of rows and columns. A default model is created.

**Parameters:**text - the text to be displayed, or nullrows - the number of rows >= 0columns - the number of columns >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the rows or columns arguments are negative.

### JTextArea

public **JTextArea**([Document](http://docs.google.com/javax/swing/text/Document.html) doc)

Constructs a new JTextArea with the given document model, and defaults for all of the other arguments (null, 0, 0).

**Parameters:**doc - the model to use

### JTextArea

public **JTextArea**([Document](http://docs.google.com/javax/swing/text/Document.html) doc,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 int rows,  
 int columns)

Constructs a new JTextArea with the specified number of rows and columns, and the given model. All of the constructors feed through this constructor.

**Parameters:**doc - the model to use, or create a default one if nulltext - the text to be displayed, null if nonerows - the number of rows >= 0columns - the number of columns >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the rows or columns arguments are negative.

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

### getUIClassID

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

Returns the class ID for the UI.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the ID ("TextAreaUI")**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))

### createDefaultModel

protected [Document](http://docs.google.com/javax/swing/text/Document.html) **createDefaultModel**()

Creates the default implementation of the model to be used at construction if one isn't explicitly given. A new instance of PlainDocument is returned.

**Returns:**the default document model

### setTabSize

public void **setTabSize**(int size)

Sets the number of characters to expand tabs to. This will be multiplied by the maximum advance for variable width fonts. A PropertyChange event ("tabSize") is fired when the tab size changes.

**Parameters:**size - number of characters to expand to**See Also:**[getTabSize()](http://docs.google.com/javax/swing/JTextArea.html#getTabSize())

### getTabSize

public int **getTabSize**()

Gets the number of characters used to expand tabs. If the document is null or doesn't have a tab setting, return a default of 8.

**Returns:**the number of characters

### setLineWrap

public void **setLineWrap**(boolean wrap)

Sets the line-wrapping policy of the text area. If set to true the lines will be wrapped if they are too long to fit within the allocated width. If set to false, the lines will always be unwrapped. A PropertyChange event ("lineWrap") is fired when the policy is changed. By default this property is false.

**Parameters:**wrap - indicates if lines should be wrapped**See Also:**[getLineWrap()](http://docs.google.com/javax/swing/JTextArea.html#getLineWrap())

### getLineWrap

public boolean **getLineWrap**()

Gets the line-wrapping policy of the text area. If set to true the lines will be wrapped if they are too long to fit within the allocated width. If set to false, the lines will always be unwrapped.

**Returns:**if lines will be wrapped

### setWrapStyleWord

public void **setWrapStyleWord**(boolean word)

Sets the style of wrapping used if the text area is wrapping lines. If set to true the lines will be wrapped at word boundaries (whitespace) if they are too long to fit within the allocated width. If set to false, the lines will be wrapped at character boundaries. By default this property is false.

**Parameters:**word - indicates if word boundaries should be used for line wrapping**See Also:**[getWrapStyleWord()](http://docs.google.com/javax/swing/JTextArea.html#getWrapStyleWord())

### getWrapStyleWord

public boolean **getWrapStyleWord**()

Gets the style of wrapping used if the text area is wrapping lines. If set to true the lines will be wrapped at word boundaries (ie whitespace) if they are too long to fit within the allocated width. If set to false, the lines will be wrapped at character boundaries.

**Returns:**if the wrap style should be word boundaries instead of character boundaries**See Also:**[setWrapStyleWord(boolean)](http://docs.google.com/javax/swing/JTextArea.html#setWrapStyleWord(boolean))

### getLineOfOffset

public int **getLineOfOffset**(int offset)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Translates an offset into the components text to a line number.

**Parameters:**offset - the offset >= 0 **Returns:**the line number >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - thrown if the offset is less than zero or greater than the document length.

### getLineCount

public int **getLineCount**()

Determines the number of lines contained in the area.

**Returns:**the number of lines > 0

### getLineStartOffset

public int **getLineStartOffset**(int line)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Determines the offset of the start of the given line.

**Parameters:**line - the line number to translate >= 0 **Returns:**the offset >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - thrown if the line is less than zero or greater or equal to the number of lines contained in the document (as reported by getLineCount).

### getLineEndOffset

public int **getLineEndOffset**(int line)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

Determines the offset of the end of the given line.

**Parameters:**line - the line >= 0 **Returns:**the offset >= 0 **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - Thrown if the line is less than zero or greater or equal to the number of lines contained in the document (as reported by getLineCount).

### insert

public void **insert**([String](http://docs.google.com/java/lang/String.html) str,  
 int pos)

Inserts the specified text at the specified position. Does nothing if the model is null or if the text is null or empty.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Parameters:**str - the text to insertpos - the position at which to insert >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if pos is an invalid position in the model**See Also:**[TextComponent.setText(java.lang.String)](http://docs.google.com/java/awt/TextComponent.html#setText(java.lang.String)), [replaceRange(java.lang.String, int, int)](http://docs.google.com/javax/swing/JTextArea.html#replaceRange(java.lang.String,%20int,%20int))

### append

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

Appends the given text to the end of the document. Does nothing if the model is null or the string is null or empty.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Parameters:**str - the text to insert**See Also:**[insert(java.lang.String, int)](http://docs.google.com/javax/swing/JTextArea.html#insert(java.lang.String,%20int))

### replaceRange

public void **replaceRange**([String](http://docs.google.com/java/lang/String.html) str,  
 int start,  
 int end)

Replaces text from the indicated start to end position with the new text specified. Does nothing if the model is null. Simply does a delete if the new string is null or empty.

This method is thread safe, although most Swing methods are not. Please see [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html) for more information.

**Parameters:**str - the text to use as the replacementstart - the start position >= 0end - the end position >= start **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if part of the range is an invalid position in the model**See Also:**[insert(java.lang.String, int)](http://docs.google.com/javax/swing/JTextArea.html#insert(java.lang.String,%20int)), [replaceRange(java.lang.String, int, int)](http://docs.google.com/javax/swing/JTextArea.html#replaceRange(java.lang.String,%20int,%20int))

### getRows

public int **getRows**()

Returns the number of rows in the TextArea.

**Returns:**the number of rows >= 0

### setRows

public void **setRows**(int rows)

Sets the number of rows for this TextArea. Calls invalidate() after setting the new value.

**Parameters:**rows - the number of rows >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if rows is less than 0**See Also:**[getRows()](http://docs.google.com/javax/swing/JTextArea.html#getRows())

### getRowHeight

protected int **getRowHeight**()

Defines the meaning of the height of a row. This defaults to the height of the font.

**Returns:**the height >= 1

### getColumns

public int **getColumns**()

Returns the number of columns in the TextArea.

**Returns:**number of columns >= 0

### setColumns

public void **setColumns**(int columns)

Sets the number of columns for this TextArea. Does an invalidate() after setting the new value.

**Parameters:**columns - the number of columns >= 0 **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if columns is less than 0**See Also:**[getColumns()](http://docs.google.com/javax/swing/JTextArea.html#getColumns())

### getColumnWidth

protected int **getColumnWidth**()

Gets column width. The meaning of what a column is can be considered a fairly weak notion for some fonts. This method is used to define the width of a column. By default this is defined to be the width of the character *m* for the font used. This method can be redefined to be some alternative amount.

**Returns:**the column width >= 1

### getPreferredSize

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

Returns the preferred size of the TextArea. This is the maximum of the size needed to display the text and the size requested for the viewport.

**Overrides:**[getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the size**See Also:**[JComponent.setPreferredSize(java.awt.Dimension)](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)

### setFont

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

Sets the current font. This removes cached row height and column width so the new font will be reflected, and calls revalidate().

**Overrides:**[setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**f - the font to use as the current font**See Also:**[Component.getFont()](http://docs.google.com/java/awt/Component.html#getFont())

### paramString

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

Returns a string representation of this JTextArea. 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/text/JTextComponent.html#paramString()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**a string representation of this JTextArea.

### getScrollableTracksViewportWidth

public boolean **getScrollableTracksViewportWidth**()

Returns true if a viewport should always force the width of this Scrollable to match the width of the viewport. This is implemented to return true if the line wrapping policy is true, and false if lines are not being wrapped.

**Specified by:**[getScrollableTracksViewportWidth](http://docs.google.com/javax/swing/Scrollable.html#getScrollableTracksViewportWidth()) in interface [Scrollable](http://docs.google.com/javax/swing/Scrollable.html)**Overrides:**[getScrollableTracksViewportWidth](http://docs.google.com/javax/swing/text/JTextComponent.html#getScrollableTracksViewportWidth()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**true if a viewport should force the Scrollables width to match its own.

### getPreferredScrollableViewportSize

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

Returns the preferred size of the viewport if this component is embedded in a JScrollPane. This uses the desired column and row settings if they have been set, otherwise the superclass behavior is used.

**Specified by:**[getPreferredScrollableViewportSize](http://docs.google.com/javax/swing/Scrollable.html#getPreferredScrollableViewportSize()) in interface [Scrollable](http://docs.google.com/javax/swing/Scrollable.html)**Overrides:**[getPreferredScrollableViewportSize](http://docs.google.com/javax/swing/text/JTextComponent.html#getPreferredScrollableViewportSize()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**The preferredSize of a JViewport whose view is this Scrollable.**See Also:**[JComponent.getPreferredSize()](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize())

### getScrollableUnitIncrement

public int **getScrollableUnitIncrement**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) visibleRect,  
 int orientation,  
 int direction)

Components that display logical rows or columns should compute the scroll increment that will completely expose one new row or column, depending on the value of orientation. This is implemented to use the values returned by the getRowHeight and getColumnWidth methods.

Scrolling containers, like JScrollPane, will use this method each time the user requests a unit scroll.

**Specified by:**[getScrollableUnitIncrement](http://docs.google.com/javax/swing/Scrollable.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int)) in interface [Scrollable](http://docs.google.com/javax/swing/Scrollable.html)**Overrides:**[getScrollableUnitIncrement](http://docs.google.com/javax/swing/text/JTextComponent.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int)) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Parameters:**visibleRect - the view area visible within the viewportorientation - Either SwingConstants.VERTICAL or SwingConstants.HORIZONTAL.direction - Less than zero to scroll up/left, greater than zero for down/right. **Returns:**The "unit" increment for scrolling in the specified direction **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - for an invalid orientation**See Also:**[JScrollBar.setUnitIncrement(int)](http://docs.google.com/javax/swing/JScrollBar.html#setUnitIncrement(int)), [getRowHeight()](http://docs.google.com/javax/swing/JTextArea.html#getRowHeight()), [getColumnWidth()](http://docs.google.com/javax/swing/JTextArea.html#getColumnWidth())

### getAccessibleContext

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

Gets the AccessibleContext associated with this JTextArea. For JTextAreas, the AccessibleContext takes the form of an AccessibleJTextArea. A new AccessibleJTextArea instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/javax/swing/text/JTextComponent.html#getAccessibleContext()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**an AccessibleJTextArea that serves as the AccessibleContext of this JTextArea

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTextArea.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/JTable.PrintMode.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTextArea.AccessibleJTextArea.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTextArea.html)    [**NO FRAMES**](http://docs.google.com/JTextArea.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: FIELD | [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).