| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TextArea.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/SystemTray.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextArea.AccessibleAWTTextArea.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextArea.html)    [**NO FRAMES**](http://docs.google.com/TextArea.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#1y810tw) | [METHOD](#qsh70q) |

## **java.awt**

Class TextArea

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

**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)

public class **TextArea**extends [TextComponent](http://docs.google.com/java/awt/TextComponent.html)

A TextArea object is a multi-line region that displays text. It can be set to allow editing or to be read-only.

The following image shows the appearance of a text area:



This text area could be created by the following line of code:

new TextArea("Hello", 5, 40);

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

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

| **Nested classes/interfaces inherited from class java.awt.**[**TextComponent**](http://docs.google.com/java/awt/TextComponent.html) |
| --- |
| [TextComponent.AccessibleAWTTextComponent](http://docs.google.com/java/awt/TextComponent.AccessibleAWTTextComponent.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** | |
| --- | --- |
| static int | [**SCROLLBARS\_BOTH**](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH)            Create and display both vertical and horizontal scrollbars. |
| static int | [**SCROLLBARS\_HORIZONTAL\_ONLY**](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_HORIZONTAL_ONLY)            Create and display horizontal scrollbar only. |
| static int | [**SCROLLBARS\_NONE**](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_NONE)            Do not create or display any scrollbars for the text area. |
| static int | [**SCROLLBARS\_VERTICAL\_ONLY**](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_VERTICAL_ONLY)            Create and display vertical scrollbar only. |

| **Fields inherited from class java.awt.**[**TextComponent**](http://docs.google.com/java/awt/TextComponent.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) |

| **Constructor Summary** | |
| --- | --- |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html#TextArea())()            Constructs a new text area with the empty string as text. |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html#TextArea(int,%20int))(int rows, int columns)            Constructs a new text area with the specified number of rows and columns and the empty string as text. |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html#TextArea(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Constructs a new text area with the specified text. |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html#TextArea(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) text, int rows, int columns)            Constructs a new text area with the specified text, and with the specified number of rows and columns. |
| [**TextArea**](http://docs.google.com/java/awt/TextArea.html#TextArea(java.lang.String,%20int,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) text, int rows, int columns, int scrollbars)            Constructs a new text area with the specified text, and with the rows, columns, and scroll bar visibility as specified. |

| **Method Summary** | |
| --- | --- |
| void | [**addNotify**](http://docs.google.com/java/awt/TextArea.html#addNotify())()            Creates the TextArea's peer. |
| void | [**append**](http://docs.google.com/java/awt/TextArea.html#append(java.lang.String))([String](http://docs.google.com/java/lang/String.html) str)            Appends the given text to the text area's current text. |
| void | [**appendText**](http://docs.google.com/java/awt/TextArea.html#appendText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) str)  **Deprecated.** *As of JDK version 1.1, replaced by append(String).* |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/TextArea.html#getAccessibleContext())()            Returns the AccessibleContext associated with this TextArea. |
| int | [**getColumns**](http://docs.google.com/java/awt/TextArea.html#getColumns())()            Returns the number of columns in this text area. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/TextArea.html#getMinimumSize())()            Determines the minimum size of this text area. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/TextArea.html#getMinimumSize(int,%20int))(int rows, int columns)            Determines the minimum size of a text area with the specified number of rows and columns. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/TextArea.html#getPreferredSize())()            Determines the preferred size of this text area. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/TextArea.html#getPreferredSize(int,%20int))(int rows, int columns)            Determines the preferred size of a text area with the specified number of rows and columns. |
| int | [**getRows**](http://docs.google.com/java/awt/TextArea.html#getRows())()            Returns the number of rows in the text area. |
| int | [**getScrollbarVisibility**](http://docs.google.com/java/awt/TextArea.html#getScrollbarVisibility())()            Returns an enumerated value that indicates which scroll bars the text area uses. |
| void | [**insert**](http://docs.google.com/java/awt/TextArea.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 in this text area. |
| void | [**insertText**](http://docs.google.com/java/awt/TextArea.html#insertText(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) str, int pos)  **Deprecated.** *As of JDK version 1.1, replaced by insert(String, int).* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/java/awt/TextArea.html#minimumSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize().* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/java/awt/TextArea.html#minimumSize(int,%20int))(int rows, int columns)  **Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize(int, int).* |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/TextArea.html#paramString())()            Returns a string representing the state of this TextArea. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/java/awt/TextArea.html#preferredSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize().* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/java/awt/TextArea.html#preferredSize(int,%20int))(int rows, int columns)  **Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize(int, int).* |
| void | [**replaceRange**](http://docs.google.com/java/awt/TextArea.html#replaceRange(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) str, int start, int end)            Replaces text between the indicated start and end positions with the specified replacement text. |
| void | [**replaceText**](http://docs.google.com/java/awt/TextArea.html#replaceText(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) str, int start, int end)  **Deprecated.** *As of JDK version 1.1, replaced by replaceRange(String, int, int).* |
| void | [**setColumns**](http://docs.google.com/java/awt/TextArea.html#setColumns(int))(int columns)            Sets the number of columns for this text area. |
| void | [**setRows**](http://docs.google.com/java/awt/TextArea.html#setRows(int))(int rows)            Sets the number of rows for this text area. |

| **Methods inherited from class java.awt.**[**TextComponent**](http://docs.google.com/java/awt/TextComponent.html) |
| --- |
| [addTextListener](http://docs.google.com/java/awt/TextComponent.html#addTextListener(java.awt.event.TextListener)), [enableInputMethods](http://docs.google.com/java/awt/TextComponent.html#enableInputMethods(boolean)), [getBackground](http://docs.google.com/java/awt/TextComponent.html#getBackground()), [getCaretPosition](http://docs.google.com/java/awt/TextComponent.html#getCaretPosition()), [getInputMethodRequests](http://docs.google.com/java/awt/TextComponent.html#getInputMethodRequests()), [getListeners](http://docs.google.com/java/awt/TextComponent.html#getListeners(java.lang.Class)), [getSelectedText](http://docs.google.com/java/awt/TextComponent.html#getSelectedText()), [getSelectionEnd](http://docs.google.com/java/awt/TextComponent.html#getSelectionEnd()), [getSelectionStart](http://docs.google.com/java/awt/TextComponent.html#getSelectionStart()), [getText](http://docs.google.com/java/awt/TextComponent.html#getText()), [getTextListeners](http://docs.google.com/java/awt/TextComponent.html#getTextListeners()), [isEditable](http://docs.google.com/java/awt/TextComponent.html#isEditable()), [processEvent](http://docs.google.com/java/awt/TextComponent.html#processEvent(java.awt.AWTEvent)), [processTextEvent](http://docs.google.com/java/awt/TextComponent.html#processTextEvent(java.awt.event.TextEvent)), [removeNotify](http://docs.google.com/java/awt/TextComponent.html#removeNotify()), [removeTextListener](http://docs.google.com/java/awt/TextComponent.html#removeTextListener(java.awt.event.TextListener)), [select](http://docs.google.com/java/awt/TextComponent.html#select(int,%20int)), [selectAll](http://docs.google.com/java/awt/TextComponent.html#selectAll()), [setBackground](http://docs.google.com/java/awt/TextComponent.html#setBackground(java.awt.Color)), [setCaretPosition](http://docs.google.com/java/awt/TextComponent.html#setCaretPosition(int)), [setEditable](http://docs.google.com/java/awt/TextComponent.html#setEditable(boolean)), [setSelectionEnd](http://docs.google.com/java/awt/TextComponent.html#setSelectionEnd(int)), [setSelectionStart](http://docs.google.com/java/awt/TextComponent.html#setSelectionStart(int)), [setText](http://docs.google.com/java/awt/TextComponent.html#setText(java.lang.String)) |

| **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()), [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()), [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)), [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)), [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** |
| --- |

### SCROLLBARS\_BOTH

public static final int **SCROLLBARS\_BOTH**

Create and display both vertical and horizontal scrollbars.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.TextArea.SCROLLBARS_BOTH)

### SCROLLBARS\_VERTICAL\_ONLY

public static final int **SCROLLBARS\_VERTICAL\_ONLY**

Create and display vertical scrollbar only.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.TextArea.SCROLLBARS_VERTICAL_ONLY)

### SCROLLBARS\_HORIZONTAL\_ONLY

public static final int **SCROLLBARS\_HORIZONTAL\_ONLY**

Create and display horizontal scrollbar only.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.TextArea.SCROLLBARS_HORIZONTAL_ONLY)

### SCROLLBARS\_NONE

public static final int **SCROLLBARS\_NONE**

Do not create or display any scrollbars for the text area.

**Since:** JDK1.1 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.TextArea.SCROLLBARS_NONE)

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

### TextArea

public **TextArea**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new text area with the empty string as text. This text area is created with scrollbar visibility equal to [SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH), so both vertical and horizontal scrollbars will be visible for this text area.

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### TextArea

public **TextArea**([String](http://docs.google.com/java/lang/String.html) text)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new text area with the specified text. This text area is created with scrollbar visibility equal to [SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH), so both vertical and horizontal scrollbars will be visible for this text area.

**Parameters:**text - the text to be displayed; if text is null, the empty string "" will be displayed **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### TextArea

public **TextArea**(int rows,  
 int columns)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new text area with the specified number of rows and columns and the empty string as text. A column is an approximate average character width that is platform-dependent. The text area is created with scrollbar visibility equal to [SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH), so both vertical and horizontal scrollbars will be visible for this text area.

**Parameters:**rows - the number of rowscolumns - the number of columns **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### TextArea

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

Constructs a new text area with the specified text, and with the specified number of rows and columns. A column is an approximate average character width that is platform-dependent. The text area is created with scrollbar visibility equal to [SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH), so both vertical and horizontal scrollbars will be visible for this text area.

**Parameters:**text - the text to be displayed; if text is null, the empty string "" will be displayedrows - the number of rowscolumns - the number of columns **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### TextArea

public **TextArea**([String](http://docs.google.com/java/lang/String.html) text,  
 int rows,  
 int columns,  
 int scrollbars)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new text area with the specified text, and with the rows, columns, and scroll bar visibility as specified. All TextArea constructors defer to this one.

The TextArea class defines several constants that can be supplied as values for the scrollbars argument:

* SCROLLBARS\_BOTH,
* SCROLLBARS\_VERTICAL\_ONLY,
* SCROLLBARS\_HORIZONTAL\_ONLY,
* SCROLLBARS\_NONE.

Any other value for the scrollbars argument is invalid and will result in this text area being created with scrollbar visibility equal to the default value of [SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH).

**Parameters:**text - the text to be displayed; if text is null, the empty string "" will be displayedrows - the number of rows; if rows is less than 0, rows is set to 0columns - the number of columns; if columns is less than 0, columns is set to 0scrollbars - a constant that determines what scrollbars are created to view the text area **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**Since:** JDK1.1 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

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

### addNotify

public void **addNotify**()

Creates the TextArea's peer. The peer allows us to modify the appearance of the TextArea without changing any of its functionality.

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

### insert

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

Inserts the specified text at the specified position in this text area.

Note that passing null or inconsistent parameters is invalid and will result in unspecified behavior.

**Parameters:**str - the non-null text to insertpos - the position at which to insert**Since:** JDK1.1 **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/java/awt/TextArea.html#replaceRange(java.lang.String,%20int,%20int)), [append(java.lang.String)](http://docs.google.com/java/awt/TextArea.html#append(java.lang.String))

### insertText

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

**Deprecated.** *As of JDK version 1.1, replaced by insert(String, int).*

### append

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

Appends the given text to the text area's current text.

Note that passing null or inconsistent parameters is invalid and will result in unspecified behavior.

**Parameters:**str - the non-null text to append**Since:** JDK1.1 **See Also:**[insert(java.lang.String, int)](http://docs.google.com/java/awt/TextArea.html#insert(java.lang.String,%20int))

### appendText

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

**Deprecated.** *As of JDK version 1.1, replaced by append(String).*

### replaceRange

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

Replaces text between the indicated start and end positions with the specified replacement text. The text at the end position will not be replaced. The text at the start position will be replaced (unless the start position is the same as the end position). The text position is zero-based. The inserted substring may be of a different length than the text it replaces.

Note that passing null or inconsistent parameters is invalid and will result in unspecified behavior.

**Parameters:**str - the non-null text to use as the replacementstart - the start positionend - the end position**Since:** JDK1.1 **See Also:**[insert(java.lang.String, int)](http://docs.google.com/java/awt/TextArea.html#insert(java.lang.String,%20int))

### replaceText

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

**Deprecated.** *As of JDK version 1.1, replaced by replaceRange(String, int, int).*

### getRows

public int **getRows**()

Returns the number of rows in the text area.

**Returns:**the number of rows in the text area**Since:** JDK1 **See Also:**[setRows(int)](http://docs.google.com/java/awt/TextArea.html#setRows(int)), [getColumns()](http://docs.google.com/java/awt/TextArea.html#getColumns())

### setRows

public void **setRows**(int rows)

Sets the number of rows for this text area.

**Parameters:**rows - the number of rows **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the value supplied for rows is less than 0**Since:** JDK1.1 **See Also:**[getRows()](http://docs.google.com/java/awt/TextArea.html#getRows()), [setColumns(int)](http://docs.google.com/java/awt/TextArea.html#setColumns(int))

### getColumns

public int **getColumns**()

Returns the number of columns in this text area.

**Returns:**the number of columns in the text area**See Also:**[setColumns(int)](http://docs.google.com/java/awt/TextArea.html#setColumns(int)), [getRows()](http://docs.google.com/java/awt/TextArea.html#getRows())

### setColumns

public void **setColumns**(int columns)

Sets the number of columns for this text area.

**Parameters:**columns - the number of columns **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the value supplied for columns is less than 0**Since:** JDK1.1 **See Also:**[getColumns()](http://docs.google.com/java/awt/TextArea.html#getColumns()), [setRows(int)](http://docs.google.com/java/awt/TextArea.html#setRows(int))

### getScrollbarVisibility

public int **getScrollbarVisibility**()

Returns an enumerated value that indicates which scroll bars the text area uses.

The TextArea class defines four integer constants that are used to specify which scroll bars are available. TextArea has one constructor that gives the application discretion over scroll bars.

**Returns:**an integer that indicates which scroll bars are used**Since:** JDK1.1 **See Also:**[SCROLLBARS\_BOTH](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_BOTH), [SCROLLBARS\_VERTICAL\_ONLY](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_VERTICAL_ONLY), [SCROLLBARS\_HORIZONTAL\_ONLY](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_HORIZONTAL_ONLY), [SCROLLBARS\_NONE](http://docs.google.com/java/awt/TextArea.html#SCROLLBARS_NONE), [TextArea(java.lang.String, int, int, int)](http://docs.google.com/java/awt/TextArea.html#TextArea(java.lang.String,%20int,%20int,%20int))

### getPreferredSize

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

Determines the preferred size of a text area with the specified number of rows and columns.

**Parameters:**rows - the number of rowscolumns - the number of columns **Returns:**the preferred dimensions required to display the text area with the specified number of rows and columns**Since:** JDK1.1 **See Also:**[Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize())

### preferredSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **preferredSize**(int rows,  
 int columns)

**Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize(int, int).*

### getPreferredSize

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

Determines the preferred size of this text area.

**Overrides:**[getPreferredSize](http://docs.google.com/java/awt/Component.html#getPreferredSize()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**the preferred dimensions needed for this text area**Since:** JDK1.1 **See Also:**[Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize())

### preferredSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **preferredSize**()

**Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize().*

**Overrides:**[preferredSize](http://docs.google.com/java/awt/Component.html#preferredSize()) in class [Component](http://docs.google.com/java/awt/Component.html)

### getMinimumSize

public [Dimension](http://docs.google.com/java/awt/Dimension.html) **getMinimumSize**(int rows,  
 int columns)

Determines the minimum size of a text area with the specified number of rows and columns.

**Parameters:**rows - the number of rowscolumns - the number of columns **Returns:**the minimum dimensions required to display the text area with the specified number of rows and columns**Since:** JDK1.1 **See Also:**[Component.getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize())

### minimumSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumSize**(int rows,  
 int columns)

**Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize(int, int).*

### getMinimumSize

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

Determines the minimum size of this text area.

**Overrides:**[getMinimumSize](http://docs.google.com/java/awt/Component.html#getMinimumSize()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**the preferred dimensions needed for this text area**Since:** JDK1.1 **See Also:**[Component.getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize())

### minimumSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumSize**()

**Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize().*

**Overrides:**[minimumSize](http://docs.google.com/java/awt/Component.html#minimumSize()) in class [Component](http://docs.google.com/java/awt/Component.html)

### paramString

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

Returns a string representing the state of this TextArea. 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/TextComponent.html#paramString()) in class [TextComponent](http://docs.google.com/java/awt/TextComponent.html) **Returns:**the parameter string of this text area

### getAccessibleContext

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

Returns the AccessibleContext associated with this TextArea. For text areas, the AccessibleContext takes the form of an AccessibleAWTTextArea. A new AccessibleAWTTextArea 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/TextComponent.html#getAccessibleContext()) in class [TextComponent](http://docs.google.com/java/awt/TextComponent.html) **Returns:**an AccessibleAWTTextArea that serves as the AccessibleContext of this TextArea**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/TextArea.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/SystemTray.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextArea.AccessibleAWTTextArea.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextArea.html)    [**NO FRAMES**](http://docs.google.com/TextArea.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#3dy6vkm) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#1y810tw) | [METHOD](#qsh70q) |

[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).