| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TextField.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/TextComponent.AccessibleAWTTextComponent.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextField.AccessibleAWTTextField.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextField.html)    [**NO FRAMES**](http://docs.google.com/TextField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: FIELD | [CONSTR](#1ksv4uv) | [METHOD](#1y810tw) |

## **java.awt**

Class TextField

[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.TextField**

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

A TextField object is a text component that allows for the editing of a single line of text.

For example, the following image depicts a frame with four text fields of varying widths. Two of these text fields display the predefined text "Hello".



Here is the code that produces these four text fields:

TextField tf1, tf2, tf3, tf4;  
 // a blank text field  
 tf1 = new TextField();  
 // blank field of 20 columns  
 tf2 = new TextField("", 20);  
 // predefined text displayed  
 tf3 = new TextField("Hello!");  
 // predefined text in 30 columns  
 tf4 = new TextField("Hello", 30);

Every time the user types a key in the text field, one or more key events are sent to the text field. A KeyEvent may be one of three types: keyPressed, keyReleased, or keyTyped. The properties of a key event indicate which of these types it is, as well as additional information about the event, such as what modifiers are applied to the key event and the time at which the event occurred.

The key event is passed to every KeyListener or KeyAdapter object which registered to receive such events using the component's addKeyListener method. (KeyAdapter objects implement the KeyListener interface.)

It is also possible to fire an ActionEvent. If action events are enabled for the text field, they may be fired by pressing the Return key.

The TextField class's processEvent method examines the action event and passes it along to processActionEvent. The latter method redirects the event to any ActionListener objects that have registered to receive action events generated by this text field.

**Since:** JDK1.0 **See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyAdapter](http://docs.google.com/java/awt/event/KeyAdapter.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html), [Component.addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [processEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/TextField.html#processEvent(java.awt.AWTEvent)), [processActionEvent(java.awt.event.ActionEvent)](http://docs.google.com/java/awt/TextField.html#processActionEvent(java.awt.event.ActionEvent)), [addActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#addActionListener(java.awt.event.ActionListener)), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.TextField)

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

| **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** | |
| --- | --- |
| [**TextField**](http://docs.google.com/java/awt/TextField.html#TextField())()            Constructs a new text field. |
| [**TextField**](http://docs.google.com/java/awt/TextField.html#TextField(int))(int columns)            Constructs a new empty text field with the specified number of columns. |
| [**TextField**](http://docs.google.com/java/awt/TextField.html#TextField(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Constructs a new text field initialized with the specified text. |
| [**TextField**](http://docs.google.com/java/awt/TextField.html#TextField(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) text, int columns)            Constructs a new text field initialized with the specified text to be displayed, and wide enough to hold the specified number of columns. |

| **Method Summary** | |
| --- | --- |
| void | [**addActionListener**](http://docs.google.com/java/awt/TextField.html#addActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Adds the specified action listener to receive action events from this text field. |
| void | [**addNotify**](http://docs.google.com/java/awt/TextField.html#addNotify())()            Creates the TextField's peer. |
| boolean | [**echoCharIsSet**](http://docs.google.com/java/awt/TextField.html#echoCharIsSet())()            Indicates whether or not this text field has a character set for echoing. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/TextField.html#getAccessibleContext())()            Gets the AccessibleContext associated with this TextField. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/java/awt/TextField.html#getActionListeners())()            Returns an array of all the action listeners registered on this textfield. |
| int | [**getColumns**](http://docs.google.com/java/awt/TextField.html#getColumns())()            Gets the number of columns in this text field. |
| char | [**getEchoChar**](http://docs.google.com/java/awt/TextField.html#getEchoChar())()            Gets the character that is to be used for echoing. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/TextField.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this TextField. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/TextField.html#getMinimumSize())()            Gets the minumum dimensions for this text field. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/TextField.html#getMinimumSize(int))(int columns)            Gets the minumum dimensions for a text field with the specified number of columns. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/TextField.html#getPreferredSize())()            Gets the preferred size of this text field. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/TextField.html#getPreferredSize(int))(int columns)            Gets the preferred size of this text field with the specified number of columns. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/java/awt/TextField.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/TextField.html#minimumSize(int))(int columns)  **Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize(int).* |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/TextField.html#paramString())()            Returns a string representing the state of this TextField. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/java/awt/TextField.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/TextField.html#preferredSize(int))(int columns)  **Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize(int).* |
| protected  void | [**processActionEvent**](http://docs.google.com/java/awt/TextField.html#processActionEvent(java.awt.event.ActionEvent))([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)            Processes action events occurring on this text field by dispatching them to any registered ActionListener objects. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/TextField.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events on this text field. |
| void | [**removeActionListener**](http://docs.google.com/java/awt/TextField.html#removeActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Removes the specified action listener so that it no longer receives action events from this text field. |
| void | [**setColumns**](http://docs.google.com/java/awt/TextField.html#setColumns(int))(int columns)            Sets the number of columns in this text field. |
| void | [**setEchoChar**](http://docs.google.com/java/awt/TextField.html#setEchoChar(char))(char c)            Sets the echo character for this text field. |
| void | [**setEchoCharacter**](http://docs.google.com/java/awt/TextField.html#setEchoCharacter(char))(char c)  **Deprecated.** *As of JDK version 1.1, replaced by setEchoChar(char).* |
| void | [**setText**](http://docs.google.com/java/awt/TextField.html#setText(java.lang.String))([String](http://docs.google.com/java/lang/String.html) t)            Sets the text that is presented by this text component to be the specified text. |

| **Methods inherited from class java.awt.**[**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()), [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()), [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)) |

| **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)) |

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

### TextField

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

Constructs a new text field.

**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())

### TextField

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

Constructs a new text field initialized with the specified text.

**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())

### TextField

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

Constructs a new empty text field with the specified number of columns. A column is an approximate average character width that is platform-dependent.

**Parameters:**columns - the number of columns. If columns is less than 0, columns is set to 0. **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())

### TextField

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

Constructs a new text field initialized with the specified text to be displayed, and wide enough to hold the specified number of columns. A column is an approximate average character width that is platform-dependent.

**Parameters:**text - the text to be displayed. If text is null, the empty string "" will be displayed.columns - the number of columns. If columns is less than 0, columns is set to 0. **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())

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

### addNotify

public void **addNotify**()

Creates the TextField's peer. The peer allows us to modify the appearance of the TextField without changing 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())

### getEchoChar

public char **getEchoChar**()

Gets the character that is to be used for echoing.

An echo character is useful for text fields where user input should not be echoed to the screen, as in the case of a text field for entering a password. If echoChar = 0, user input is echoed to the screen unchanged.

A Java platform implementation may support only a limited, non-empty set of echo characters. This function returns the echo character originally requested via setEchoChar(). The echo character actually used by the TextField implementation might be different.

**Returns:**the echo character for this text field.**See Also:**[echoCharIsSet()](http://docs.google.com/java/awt/TextField.html#echoCharIsSet()), [setEchoChar(char)](http://docs.google.com/java/awt/TextField.html#setEchoChar(char))

### setEchoChar

public void **setEchoChar**(char c)

Sets the echo character for this text field.

An echo character is useful for text fields where user input should not be echoed to the screen, as in the case of a text field for entering a password. Setting echoChar = 0 allows user input to be echoed to the screen again.

A Java platform implementation may support only a limited, non-empty set of echo characters. Attempts to set an unsupported echo character will cause the default echo character to be used instead. Subsequent calls to getEchoChar() will return the echo character originally requested. This might or might not be identical to the echo character actually used by the TextField implementation.

**Parameters:**c - the echo character for this text field.**Since:** JDK1.1 **See Also:**[echoCharIsSet()](http://docs.google.com/java/awt/TextField.html#echoCharIsSet()), [getEchoChar()](http://docs.google.com/java/awt/TextField.html#getEchoChar())

### setEchoCharacter

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **setEchoCharacter**(char c)

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

### setText

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

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

**Overrides:**[setText](http://docs.google.com/java/awt/TextComponent.html#setText(java.lang.String)) in class [TextComponent](http://docs.google.com/java/awt/TextComponent.html) **Parameters:**t - the new text.**See Also:**[TextComponent.getText()](http://docs.google.com/java/awt/TextComponent.html#getText())

### echoCharIsSet

public boolean **echoCharIsSet**()

Indicates whether or not this text field has a character set for echoing.

An echo character is useful for text fields where user input should not be echoed to the screen, as in the case of a text field for entering a password.

**Returns:**true if this text field has a character set for echoing; false otherwise.**See Also:**[setEchoChar(char)](http://docs.google.com/java/awt/TextField.html#setEchoChar(char)), [getEchoChar()](http://docs.google.com/java/awt/TextField.html#getEchoChar())

### getColumns

public int **getColumns**()

Gets the number of columns in this text field. A column is an approximate average character width that is platform-dependent.

**Returns:**the number of columns.**Since:** JDK1.1 **See Also:**[setColumns(int)](http://docs.google.com/java/awt/TextField.html#setColumns(int))

### setColumns

public void **setColumns**(int columns)

Sets the number of columns in this text field. A column is an approximate average character width that is platform-dependent.

**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/TextField.html#getColumns())

### getPreferredSize

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

Gets the preferred size of this text field with the specified number of columns.

**Parameters:**columns - the number of columns in this text field. **Returns:**the preferred dimensions for displaying this text field.**Since:** JDK1.1

### preferredSize

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

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

### getPreferredSize

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

Gets the preferred size of this text field.

**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 for displaying this text field.**Since:** JDK1.1 **See Also:**[Component.getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### 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 columns)

Gets the minumum dimensions for a text field with the specified number of columns.

**Parameters:**columns - the number of columns in this text field.**Since:** JDK1.1

### minimumSize

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

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

### getMinimumSize

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

Gets the minumum dimensions for this text field.

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

### 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)

### addActionListener

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

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

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

**Parameters:**l - the action listener.**Since:** JDK1.1 **See Also:**[removeActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#removeActionListener(java.awt.event.ActionListener)), [getActionListeners()](http://docs.google.com/java/awt/TextField.html#getActionListeners()), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)

### removeActionListener

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

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

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

**Parameters:**l - the action listener.**Since:** JDK1.1 **See Also:**[addActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#addActionListener(java.awt.event.ActionListener)), [getActionListeners()](http://docs.google.com/java/awt/TextField.html#getActionListeners()), [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)

### getActionListeners

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

Returns an array of all the action listeners registered on this textfield.

**Returns:**all of this textfield's ActionListeners or an empty array if no action listeners are currently registered**Since:** 1.4 **See Also:**[addActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#addActionListener(java.awt.event.ActionListener)), [removeActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#removeActionListener(java.awt.event.ActionListener)), [java.awt.event](http://docs.google.com/java/awt/event/package-summary.html)

### getListeners

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

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

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

ActionListener[] als = (ActionListener[])(t.getListeners(ActionListener.class));

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

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

### processEvent

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

Processes events on this text field. If the event is an instance of ActionEvent, it invokes the processActionEvent method. Otherwise, it invokes processEvent on the superclass.

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

**Overrides:**[processEvent](http://docs.google.com/java/awt/TextComponent.html#processEvent(java.awt.AWTEvent)) in class [TextComponent](http://docs.google.com/java/awt/TextComponent.html) **Parameters:**e - the event**Since:** JDK1.1 **See Also:**[ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html), [processActionEvent(java.awt.event.ActionEvent)](http://docs.google.com/java/awt/TextField.html#processActionEvent(java.awt.event.ActionEvent))

### processActionEvent

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

Processes action events occurring on this text field by dispatching them to any registered ActionListener objects.

This method is not called unless action events are enabled for this component. Action events are enabled when one of the following occurs:

* An ActionListener object is registered via addActionListener.
* Action events are enabled via enableEvents.

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

**Parameters:**e - the action event**Since:** JDK1.1 **See Also:**[ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [addActionListener(java.awt.event.ActionListener)](http://docs.google.com/java/awt/TextField.html#addActionListener(java.awt.event.ActionListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### paramString

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

Returns a string representing the state of this TextField. 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 field

### getAccessibleContext

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

Gets the AccessibleContext associated with this TextField. For text fields, the AccessibleContext takes the form of an AccessibleAWTTextField. A new AccessibleAWTTextField 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 AccessibleAWTTextField that serves as the AccessibleContext of this TextField**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/TextField.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/TextComponent.AccessibleAWTTextComponent.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/TextField.AccessibleAWTTextField.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/TextField.html)    [**NO FRAMES**](http://docs.google.com/TextField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: FIELD | [CONSTR](#1ksv4uv) | [METHOD](#1y810tw) |

[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).