| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTextField.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JTextArea.AccessibleJTextArea.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTextField.AccessibleJTextField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTextField.html)    [**NO FRAMES**](http://docs.google.com/JTextField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#35nkun2) | [METHOD](#1ksv4uv) | DETAIL: [FIELD](#4i7ojhp) | [CONSTR](#1ci93xb) | [METHOD](#49x2ik5) |

## **javax.swing**

Class JTextField

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.Component](http://docs.google.com/java/awt/Component.html)  
 [java.awt.Container](http://docs.google.com/java/awt/Container.html)  
 [javax.swing.JComponent](http://docs.google.com/javax/swing/JComponent.html)  
 [javax.swing.text.JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html)  
 **javax.swing.JTextField**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [Scrollable](http://docs.google.com/javax/swing/Scrollable.html), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html) **Direct Known Subclasses:** [DefaultTreeCellEditor.DefaultTextField](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.DefaultTextField.html), [JFormattedTextField](http://docs.google.com/javax/swing/JFormattedTextField.html), [JPasswordField](http://docs.google.com/javax/swing/JPasswordField.html)

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

JTextField is a lightweight component that allows the editing of a single line of text. For information on and examples of using text fields, see [How to Use Text Fields](http://java.sun.com/docs/books/tutorial/uiswing/components/textfield.html) in *The Java Tutorial.*

JTextField is intended to be source-compatible with java.awt.TextField where it is reasonable to do so. This component has capabilities not found in the java.awt.TextField class. The superclass should be consulted for additional capabilities.

JTextField has a method to establish the string used as the command string for the action event that gets fired. The java.awt.TextField used the text of the field as the command string for the ActionEvent. JTextField will use the command string set with the setActionCommand method if not null, otherwise it will use the text of the field as a compatibility with java.awt.TextField.

The method setEchoChar and getEchoChar are not provided directly to avoid a new implementation of a pluggable look-and-feel inadvertently exposing password characters. To provide password-like services a separate class JPasswordField extends JTextField to provide this service with an independently pluggable look-and-feel.

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

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

The horizontal alignment of JTextField can be set to be left justified, leading justified, centered, right justified or trailing justified. Right/trailing justification is useful if the required size of the field text is smaller than the size allocated to it. This is determined by the setHorizontalAlignment and getHorizontalAlignment methods. The default is to be leading justified.

How the text field consumes VK\_ENTER events depends on whether the text field has any action listeners. If so, then VK\_ENTER results in the listeners getting an ActionEvent, and the VK\_ENTER event is consumed. This is compatible with how AWT text fields handle VK\_ENTER events. If the text field has no action listeners, then as of v 1.3 the VK\_ENTER event is not consumed. Instead, the bindings of ancestor components are processed, which enables the default button feature of JFC/Swing to work.

Customized fields can easily be created by extending the model and changing the default model provided. For example, the following piece of code will create a field that holds only upper case characters. It will work even if text is pasted into from the clipboard or it is altered via programmatic changes.

 public class UpperCaseField extends JTextField {  
   
  public UpperCaseField(int cols) {  
  super(cols);  
  }  
   
  protected Document createDefaultModel() {  
  return new UpperCaseDocument();  
  }  
   
  static class UpperCaseDocument extends PlainDocument {  
   
  public void insertString(int offs, String str, AttributeSet a)   
  throws BadLocationException {  
   
  if (str == null) {  
  return;  
  }  
  char[] upper = str.toCharArray();  
  for (int i = 0; i < upper.length; i++) {  
  upper[i] = Character.toUpperCase(upper[i]);  
  }  
  super.insertString(offs, new String(upper), a);  
  }  
  }  
 }

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[setActionCommand(java.lang.String)](http://docs.google.com/javax/swing/JTextField.html#setActionCommand(java.lang.String)), [JPasswordField](http://docs.google.com/javax/swing/JPasswordField.html), [addActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/JTextField.html#addActionListener(java.awt.event.ActionListener))

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

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

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

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

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**notifyAction**](http://docs.google.com/javax/swing/JTextField.html#notifyAction)            Name of the action to send notification that the contents of the field have been accepted. |

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

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

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

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| [**JTextField**](http://docs.google.com/javax/swing/JTextField.html#JTextField())()            Constructs a new TextField. |
| [**JTextField**](http://docs.google.com/javax/swing/JTextField.html#JTextField(javax.swing.text.Document,%20java.lang.String,%20int))([Document](http://docs.google.com/javax/swing/text/Document.html) doc, [String](http://docs.google.com/java/lang/String.html) text, int columns)            Constructs a new JTextField that uses the given text storage model and the given number of columns. |
| [**JTextField**](http://docs.google.com/javax/swing/JTextField.html#JTextField(int))(int columns)            Constructs a new empty TextField with the specified number of columns. |
| [**JTextField**](http://docs.google.com/javax/swing/JTextField.html#JTextField(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Constructs a new TextField initialized with the specified text. |
| [**JTextField**](http://docs.google.com/javax/swing/JTextField.html#JTextField(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) text, int columns)            Constructs a new TextField initialized with the specified text and columns. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**actionPropertyChanged**](http://docs.google.com/javax/swing/JTextField.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String))([Action](http://docs.google.com/javax/swing/Action.html) action, [String](http://docs.google.com/java/lang/String.html) propertyName)            Updates the textfield's state in response to property changes in associated action. |
| void | [**addActionListener**](http://docs.google.com/javax/swing/JTextField.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 textfield. |
| protected  void | [**configurePropertiesFromAction**](http://docs.google.com/javax/swing/JTextField.html#configurePropertiesFromAction(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Sets the properties on this textfield to match those in the specified Action. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createActionPropertyChangeListener**](http://docs.google.com/javax/swing/JTextField.html#createActionPropertyChangeListener(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Creates and returns a PropertyChangeListener that is responsible for listening for changes from the specified Action and updating the appropriate properties. |
| protected  [Document](http://docs.google.com/javax/swing/text/Document.html) | [**createDefaultModel**](http://docs.google.com/javax/swing/JTextField.html#createDefaultModel())()            Creates the default implementation of the model to be used at construction if one isn't explicitly given. |
| protected  void | [**fireActionPerformed**](http://docs.google.com/javax/swing/JTextField.html#fireActionPerformed())()            Notifies all listeners that have registered interest for notification on this event type. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JTextField.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JTextField. |
| [Action](http://docs.google.com/javax/swing/Action.html) | [**getAction**](http://docs.google.com/javax/swing/JTextField.html#getAction())()            Returns the currently set Action for this ActionEvent source, or null if no Action is set. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/javax/swing/JTextField.html#getActionListeners())()            Returns an array of all the ActionListeners added to this JTextField with addActionListener(). |
| [Action](http://docs.google.com/javax/swing/Action.html)[] | [**getActions**](http://docs.google.com/javax/swing/JTextField.html#getActions())()            Fetches the command list for the editor. |
| int | [**getColumns**](http://docs.google.com/javax/swing/JTextField.html#getColumns())()            Returns the number of columns in this TextField. |
| protected  int | [**getColumnWidth**](http://docs.google.com/javax/swing/JTextField.html#getColumnWidth())()            Returns the column width. |
| int | [**getHorizontalAlignment**](http://docs.google.com/javax/swing/JTextField.html#getHorizontalAlignment())()            Returns the horizontal alignment of the text. |
| [BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html) | [**getHorizontalVisibility**](http://docs.google.com/javax/swing/JTextField.html#getHorizontalVisibility())()            Gets the visibility of the text field. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/javax/swing/JTextField.html#getPreferredSize())()            Returns the preferred size Dimensions needed for this TextField. |
| int | [**getScrollOffset**](http://docs.google.com/javax/swing/JTextField.html#getScrollOffset())()            Gets the scroll offset, in pixels. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JTextField.html#getUIClassID())()            Gets the class ID for a UI. |
| boolean | [**isValidateRoot**](http://docs.google.com/javax/swing/JTextField.html#isValidateRoot())()            Calls to revalidate that come from within the textfield itself will be handled by validating the textfield, unless the textfield is contained within a JViewport, in which case this returns false. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JTextField.html#paramString())()            Returns a string representation of this JTextField. |
| void | [**postActionEvent**](http://docs.google.com/javax/swing/JTextField.html#postActionEvent())()            Processes action events occurring on this textfield by dispatching them to any registered ActionListener objects. |
| void | [**removeActionListener**](http://docs.google.com/javax/swing/JTextField.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 textfield. |
| void | [**scrollRectToVisible**](http://docs.google.com/javax/swing/JTextField.html#scrollRectToVisible(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Scrolls the field left or right. |
| void | [**setAction**](http://docs.google.com/javax/swing/JTextField.html#setAction(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Sets the Action for the ActionEvent source. |
| void | [**setActionCommand**](http://docs.google.com/javax/swing/JTextField.html#setActionCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) command)            Sets the command string used for action events. |
| void | [**setColumns**](http://docs.google.com/javax/swing/JTextField.html#setColumns(int))(int columns)            Sets the number of columns in this TextField, and then invalidate the layout. |
| void | [**setDocument**](http://docs.google.com/javax/swing/JTextField.html#setDocument(javax.swing.text.Document))([Document](http://docs.google.com/javax/swing/text/Document.html) doc)            Associates the editor with a text document. |
| void | [**setFont**](http://docs.google.com/javax/swing/JTextField.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Sets the current font. |
| void | [**setHorizontalAlignment**](http://docs.google.com/javax/swing/JTextField.html#setHorizontalAlignment(int))(int alignment)            Sets the horizontal alignment of the text. |
| void | [**setScrollOffset**](http://docs.google.com/javax/swing/JTextField.html#setScrollOffset(int))(int scrollOffset)            Sets the scroll offset, in pixels. |

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

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

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

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [show](http://docs.google.com/java/awt/Component.html#show()), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

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

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

### notifyAction

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

Name of the action to send notification that the contents of the field have been accepted. Typically this is bound to a carriage-return.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JTextField.notifyAction)

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

### JTextField

public **JTextField**()

Constructs a new TextField. A default model is created, the initial string is null, and the number of columns is set to 0.

### JTextField

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

Constructs a new TextField initialized with the specified text. A default model is created and the number of columns is 0.

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

### JTextField

public **JTextField**(int columns)

Constructs a new empty TextField with the specified number of columns. A default model is created and the initial string is set to null.

**Parameters:**columns - the number of columns to use to calculate the preferred width; if columns is set to zero, the preferred width will be whatever naturally results from the component implementation

### JTextField

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

Constructs a new TextField initialized with the specified text and columns. A default model is created.

**Parameters:**text - the text to be displayed, or nullcolumns - the number of columns to use to calculate the preferred width; if columns is set to zero, the preferred width will be whatever naturally results from the component implementation

### JTextField

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

Constructs a new JTextField that uses the given text storage model and the given number of columns. This is the constructor through which the other constructors feed. If the document is null, a default model is created.

**Parameters:**doc - the text storage to use; if this is null, a default will be provided by calling the createDefaultModel methodtext - the initial string to display, or nullcolumns - the number of columns to use to calculate the preferred width >= 0; if columns is set to zero, the preferred width will be whatever naturally results from the component implementation **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if columns < 0

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

### getUIClassID

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

Gets the class ID for a UI.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "TextFieldUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### setDocument

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

Associates the editor with a text document. The currently registered factory is used to build a view for the document, which gets displayed by the editor after revalidation. A PropertyChange event ("document") is propagated to each listener.

**Overrides:**[setDocument](http://docs.google.com/javax/swing/text/JTextComponent.html#setDocument(javax.swing.text.Document)) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Parameters:**doc - the document to display/edit**See Also:**[JTextComponent.getDocument()](http://docs.google.com/javax/swing/text/JTextComponent.html#getDocument())

### isValidateRoot

public boolean **isValidateRoot**()

Calls to revalidate that come from within the textfield itself will be handled by validating the textfield, unless the textfield is contained within a JViewport, in which case this returns false.

**Overrides:**[isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**if the parent of this textfield is a JViewPort return false, otherwise return true**See Also:**[JComponent.revalidate()](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [JComponent.isValidateRoot()](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot())

### getHorizontalAlignment

public int **getHorizontalAlignment**()

Returns the horizontal alignment of the text. Valid keys are:

* JTextField.LEFT
* JTextField.CENTER
* JTextField.RIGHT
* JTextField.LEADING
* JTextField.TRAILING

**Returns:**the horizontal alignment

### setHorizontalAlignment

public void **setHorizontalAlignment**(int alignment)

Sets the horizontal alignment of the text. Valid keys are:

* JTextField.LEFT
* JTextField.CENTER
* JTextField.RIGHT
* JTextField.LEADING
* JTextField.TRAILING

invalidate and repaint are called when the alignment is set, and a PropertyChange event ("horizontalAlignment") is fired.

**Parameters:**alignment - the alignment **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if alignment is not a valid key

### createDefaultModel

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

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

**Returns:**the default model implementation

### getColumns

public int **getColumns**()

Returns the number of columns in this TextField.

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

### setColumns

public void **setColumns**(int columns)

Sets the number of columns in this TextField, and then invalidate the layout.

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

### getColumnWidth

protected int **getColumnWidth**()

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

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

### getPreferredSize

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

Returns the preferred size Dimensions needed for this TextField. If a non-zero number of columns has been set, the width is set to the columns multiplied by the column width.

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

### setFont

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

Sets the current font. This removes cached row height and column width so the new font will be reflected. revalidate is called after setting the font.

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

### 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 textfield.

**Parameters:**l - the action listener to be added

### 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 textfield.

**Parameters:**l - the action listener to be removed

### getActionListeners

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

Returns an array of all the ActionListeners added to this JTextField with addActionListener().

**Returns:**all of the ActionListeners added or an empty array if no listeners have been added**Since:** 1.4

### fireActionPerformed

protected void **fireActionPerformed**()

Notifies all listeners that have registered interest for notification on this event type. The event instance is lazily created. The listener list is processed in last to first order.

**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### setActionCommand

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

Sets the command string used for action events.

**Parameters:**command - the command string

### setAction

public void **setAction**([Action](http://docs.google.com/javax/swing/Action.html) a)

Sets the Action for the ActionEvent source. The new Action replaces any previously set Action but does not affect ActionListeners independently added with addActionListener. If the Action is already a registered ActionListener for the ActionEvent source, it is not re-registered.

Setting the Action results in immediately changing all the properties described in  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions). Subsequently, the textfield's properties are automatically updated as the Action's properties change.

This method uses three other methods to set and help track the Action's property values. It uses the configurePropertiesFromAction method to immediately change the textfield's properties. To track changes in the Action's property values, this method registers the PropertyChangeListener returned by createActionPropertyChangeListener. The default PropertyChangeListener invokes the actionPropertyChanged method when a property in the Action changes.

**Parameters:**a - the Action for the JTextField, or null**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [getAction()](http://docs.google.com/javax/swing/JTextField.html#getAction()), [configurePropertiesFromAction(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#configurePropertiesFromAction(javax.swing.Action)), [createActionPropertyChangeListener(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#createActionPropertyChangeListener(javax.swing.Action)), [actionPropertyChanged(javax.swing.Action, java.lang.String)](http://docs.google.com/javax/swing/JTextField.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String))

### getAction

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

Returns the currently set Action for this ActionEvent source, or null if no Action is set.

**Returns:**the Action for this ActionEvent source, or null**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#setAction(javax.swing.Action))

### configurePropertiesFromAction

protected void **configurePropertiesFromAction**([Action](http://docs.google.com/javax/swing/Action.html) a)

Sets the properties on this textfield to match those in the specified Action. Refer to  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions) for more details as to which properties this sets.

**Parameters:**a - the Action from which to get the properties, or null**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#setAction(javax.swing.Action))

### actionPropertyChanged

protected void **actionPropertyChanged**([Action](http://docs.google.com/javax/swing/Action.html) action,  
 [String](http://docs.google.com/java/lang/String.html) propertyName)

Updates the textfield's state in response to property changes in associated action. This method is invoked from the PropertyChangeListener returned from createActionPropertyChangeListener. Subclasses do not normally need to invoke this. Subclasses that support additional Action properties should override this and configurePropertiesFromAction.

Refer to the table at  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions) for a list of the properties this method sets.

**Parameters:**action - the Action associated with this textfieldpropertyName - the name of the property that changed**Since:** 1.6 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [configurePropertiesFromAction(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#configurePropertiesFromAction(javax.swing.Action))

### createActionPropertyChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createActionPropertyChangeListener**([Action](http://docs.google.com/javax/swing/Action.html) a)

Creates and returns a PropertyChangeListener that is responsible for listening for changes from the specified Action and updating the appropriate properties.

**Warning:** If you subclass this do not create an anonymous inner class. If you do the lifetime of the textfield will be tied to that of the Action.

**Parameters:**a - the textfield's action**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [setAction(javax.swing.Action)](http://docs.google.com/javax/swing/JTextField.html#setAction(javax.swing.Action))

### getActions

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

Fetches the command list for the editor. This is the list of commands supported by the plugged-in UI augmented by the collection of commands that the editor itself supports. These are useful for binding to events, such as in a keymap.

**Overrides:**[getActions](http://docs.google.com/javax/swing/text/JTextComponent.html#getActions()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**the command list

### postActionEvent

public void **postActionEvent**()

Processes action events occurring on this textfield by dispatching them to any registered ActionListener objects. This is normally called by the controller registered with textfield.

### getHorizontalVisibility

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

Gets the visibility of the text field. This can be adjusted to change the location of the visible area if the size of the field is greater than the area that was allocated to the field.

The fields look-and-feel implementation manages the values of the minimum, maximum, and extent properties on the BoundedRangeModel.

**Returns:**the visibility**See Also:**[BoundedRangeModel](http://docs.google.com/javax/swing/BoundedRangeModel.html)

### getScrollOffset

public int **getScrollOffset**()

Gets the scroll offset, in pixels.

**Returns:**the offset >= 0

### setScrollOffset

public void **setScrollOffset**(int scrollOffset)

Sets the scroll offset, in pixels.

**Parameters:**scrollOffset - the offset >= 0

### scrollRectToVisible

public void **scrollRectToVisible**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Scrolls the field left or right.

**Overrides:**[scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**r - the region to scroll**See Also:**[JViewport](http://docs.google.com/javax/swing/JViewport.html)

### paramString

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

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

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

### getAccessibleContext

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

Gets the AccessibleContext associated with this JTextField. For JTextFields, the AccessibleContext takes the form of an AccessibleJTextField. A new AccessibleJTextField instance is created if necessary.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JTextField.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JTextArea.AccessibleJTextArea.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JTextField.AccessibleJTextField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JTextField.html)    [**NO FRAMES**](http://docs.google.com/JTextField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#35nkun2) | [METHOD](#1ksv4uv) | DETAIL: [FIELD](#4i7ojhp) | [CONSTR](#1ci93xb) | [METHOD](#49x2ik5) |

[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).