| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JPasswordField.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/JPanel.AccessibleJPanel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JPasswordField.AccessibleJPasswordField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JPasswordField.html)    [**NO FRAMES**](http://docs.google.com/JPasswordField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#17dp8vu) | [CONSTR](#44sinio) | [METHOD](#2jxsxqh) | DETAIL: FIELD | [CONSTR](#3whwml4) | [METHOD](#2p2csry) |

## **javax.swing**

Class JPasswordField

[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](http://docs.google.com/javax/swing/JTextField.html)  
 **javax.swing.JPasswordField**

**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)

public class **JPasswordField**extends [JTextField](http://docs.google.com/javax/swing/JTextField.html)

JPasswordField is a lightweight component that allows the editing of a single line of text where the view indicates something was typed, but does not show the original characters. You can find further information and examples in [How to Use Text Fields](http://java.sun.com/docs/books/tutorial/uiswing/components/textfield.html), a section in *The Java Tutorial.*

JPasswordField is intended to be source-compatible with java.awt.TextField used with echoChar set. It is provided separately to make it easier to safely change the UI for the JTextField without affecting password entries.

**NOTE:** By default, JPasswordField disables input methods; otherwise, input characters could be visible while they were composed using input methods. If an application needs the input methods support, please use the inherited method, enableInputMethods(true).

**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).

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

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

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

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

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

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

| **Field Summary** | |
| --- | --- |

| **Fields inherited from class javax.swing.**[**JTextField**](http://docs.google.com/javax/swing/JTextField.html) |
| --- |
| [notifyAction](http://docs.google.com/javax/swing/JTextField.html#notifyAction) |

| **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** | |
| --- | --- |
| [**JPasswordField**](http://docs.google.com/javax/swing/JPasswordField.html#JPasswordField())()            Constructs a new JPasswordField, with a default document, null starting text string, and 0 column width. |
| [**JPasswordField**](http://docs.google.com/javax/swing/JPasswordField.html#JPasswordField(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) txt, int columns)            Constructs a new JPasswordField that uses the given text storage model and the given number of columns. |
| [**JPasswordField**](http://docs.google.com/javax/swing/JPasswordField.html#JPasswordField(int))(int columns)            Constructs a new empty JPasswordField with the specified number of columns. |
| [**JPasswordField**](http://docs.google.com/javax/swing/JPasswordField.html#JPasswordField(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Constructs a new JPasswordField initialized with the specified text. |
| [**JPasswordField**](http://docs.google.com/javax/swing/JPasswordField.html#JPasswordField(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) text, int columns)            Constructs a new JPasswordField initialized with the specified text and columns. |

| **Method Summary** | |
| --- | --- |
| void | [**copy**](http://docs.google.com/javax/swing/JPasswordField.html#copy())()            Invokes provideErrorFeedback on the current look and feel, which typically initiates an error beep. |
| void | [**cut**](http://docs.google.com/javax/swing/JPasswordField.html#cut())()            Invokes provideErrorFeedback on the current look and feel, which typically initiates an error beep. |
| boolean | [**echoCharIsSet**](http://docs.google.com/javax/swing/JPasswordField.html#echoCharIsSet())()            Returns true if this JPasswordField has a character set for echoing. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JPasswordField.html#getAccessibleContext())()            Returns the AccessibleContext associated with this JPasswordField. |
| char | [**getEchoChar**](http://docs.google.com/javax/swing/JPasswordField.html#getEchoChar())()            Returns the character to be used for echoing. |
| char[] | [**getPassword**](http://docs.google.com/javax/swing/JPasswordField.html#getPassword())()            Returns the text contained in this TextComponent. |
| [String](http://docs.google.com/java/lang/String.html) | [**getText**](http://docs.google.com/javax/swing/JPasswordField.html#getText())()  **Deprecated.** *As of Java 2 platform v1.2, replaced by getPassword.* |
| [String](http://docs.google.com/java/lang/String.html) | [**getText**](http://docs.google.com/javax/swing/JPasswordField.html#getText(int,%20int))(int offs, int len)  **Deprecated.** *As of Java 2 platform v1.2, replaced by getPassword.* |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JPasswordField.html#getUIClassID())()            Returns the name of the L&F class that renders this component. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JPasswordField.html#paramString())()            Returns a string representation of this JPasswordField. |
| void | [**setEchoChar**](http://docs.google.com/javax/swing/JPasswordField.html#setEchoChar(char))(char c)            Sets the echo character for this JPasswordField. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JPasswordField.html#updateUI())()            Reloads the pluggable UI. |

| **Methods inherited from class javax.swing.**[**JTextField**](http://docs.google.com/javax/swing/JTextField.html) |
| --- |
| [actionPropertyChanged](http://docs.google.com/javax/swing/JTextField.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String)), [addActionListener](http://docs.google.com/javax/swing/JTextField.html#addActionListener(java.awt.event.ActionListener)), [configurePropertiesFromAction](http://docs.google.com/javax/swing/JTextField.html#configurePropertiesFromAction(javax.swing.Action)), [createActionPropertyChangeListener](http://docs.google.com/javax/swing/JTextField.html#createActionPropertyChangeListener(javax.swing.Action)), [createDefaultModel](http://docs.google.com/javax/swing/JTextField.html#createDefaultModel()), [fireActionPerformed](http://docs.google.com/javax/swing/JTextField.html#fireActionPerformed()), [getAction](http://docs.google.com/javax/swing/JTextField.html#getAction()), [getActionListeners](http://docs.google.com/javax/swing/JTextField.html#getActionListeners()), [getActions](http://docs.google.com/javax/swing/JTextField.html#getActions()), [getColumns](http://docs.google.com/javax/swing/JTextField.html#getColumns()), [getColumnWidth](http://docs.google.com/javax/swing/JTextField.html#getColumnWidth()), [getHorizontalAlignment](http://docs.google.com/javax/swing/JTextField.html#getHorizontalAlignment()), [getHorizontalVisibility](http://docs.google.com/javax/swing/JTextField.html#getHorizontalVisibility()), [getPreferredSize](http://docs.google.com/javax/swing/JTextField.html#getPreferredSize()), [getScrollOffset](http://docs.google.com/javax/swing/JTextField.html#getScrollOffset()), [isValidateRoot](http://docs.google.com/javax/swing/JTextField.html#isValidateRoot()), [postActionEvent](http://docs.google.com/javax/swing/JTextField.html#postActionEvent()), [removeActionListener](http://docs.google.com/javax/swing/JTextField.html#removeActionListener(java.awt.event.ActionListener)), [scrollRectToVisible](http://docs.google.com/javax/swing/JTextField.html#scrollRectToVisible(java.awt.Rectangle)), [setAction](http://docs.google.com/javax/swing/JTextField.html#setAction(javax.swing.Action)), [setActionCommand](http://docs.google.com/javax/swing/JTextField.html#setActionCommand(java.lang.String)), [setColumns](http://docs.google.com/javax/swing/JTextField.html#setColumns(int)), [setDocument](http://docs.google.com/javax/swing/JTextField.html#setDocument(javax.swing.text.Document)), [setFont](http://docs.google.com/javax/swing/JTextField.html#setFont(java.awt.Font)), [setHorizontalAlignment](http://docs.google.com/javax/swing/JTextField.html#setHorizontalAlignment(int)), [setScrollOffset](http://docs.google.com/javax/swing/JTextField.html#setScrollOffset(int)) |

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

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

### JPasswordField

public **JPasswordField**()

Constructs a new JPasswordField, with a default document, null starting text string, and 0 column width.

### JPasswordField

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

Constructs a new JPasswordField initialized with the specified text. The document model is set to the default, and the number of columns to 0.

**Parameters:**text - the text to be displayed, null if none

### JPasswordField

public **JPasswordField**(int columns)

Constructs a new empty JPasswordField 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 >= 0

### JPasswordField

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

Constructs a new JPasswordField initialized with the specified text and columns. The document model is set to the default.

**Parameters:**text - the text to be displayed, null if nonecolumns - the number of columns >= 0

### JPasswordField

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

Constructs a new JPasswordField that uses the given text storage model and the given number of columns. This is the constructor through which the other constructors feed. The echo character is set to '\*', but may be changed by the current Look and Feel. If the document model is null, a default one will be created.

**Parameters:**doc - the text storage to usetxt - the text to be displayed, null if nonecolumns - 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

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

### getUIClassID

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

Returns the name of the L&F class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JTextField.html#getUIClassID()) in class [JTextField](http://docs.google.com/javax/swing/JTextField.html) **Returns:**the string "PasswordFieldUI"**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))

### updateUI

public void **updateUI**()

Reloads the pluggable UI. The key used to fetch the new interface is getUIClassID(). The type of the UI is TextUI. invalidate is called after setting the UI.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/text/JTextComponent.html#updateUI()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Since:** 1.6 **See Also:**[JComponent.setUI(javax.swing.plaf.ComponentUI)](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [UIManager.getLookAndFeel()](http://docs.google.com/javax/swing/UIManager.html#getLookAndFeel()), [UIManager.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIManager.html#getUI(javax.swing.JComponent))

### getEchoChar

public char **getEchoChar**()

Returns the character to be used for echoing. The default is '\*'. The default may be different depending on the currently running Look and Feel. For example, Metal/Ocean's default is a bullet character.

**Returns:**the echo character, 0 if unset**See Also:**[setEchoChar(char)](http://docs.google.com/javax/swing/JPasswordField.html#setEchoChar(char)), [echoCharIsSet()](http://docs.google.com/javax/swing/JPasswordField.html#echoCharIsSet())

### setEchoChar

public void **setEchoChar**(char c)

Sets the echo character for this JPasswordField. Note that this is largely a suggestion, since the view that gets installed can use whatever graphic techniques it desires to represent the field. Setting a value of 0 indicates that you wish to see the text as it is typed, similar to the behavior of a standard JTextField.

**Parameters:**c - the echo character to display**See Also:**[echoCharIsSet()](http://docs.google.com/javax/swing/JPasswordField.html#echoCharIsSet()), [getEchoChar()](http://docs.google.com/javax/swing/JPasswordField.html#getEchoChar())

### echoCharIsSet

public boolean **echoCharIsSet**()

Returns true if this JPasswordField has a character set for echoing. A character is considered to be set if the echo character is not 0.

**Returns:**true if a character is set for echoing**See Also:**[setEchoChar(char)](http://docs.google.com/javax/swing/JPasswordField.html#setEchoChar(char)), [getEchoChar()](http://docs.google.com/javax/swing/JPasswordField.html#getEchoChar())

### cut

public void **cut**()

Invokes provideErrorFeedback on the current look and feel, which typically initiates an error beep. The normal behavior of transferring the currently selected range in the associated text model to the system clipboard, and removing the contents from the model, is not acceptable for a password field.

**Overrides:**[cut](http://docs.google.com/javax/swing/text/JTextComponent.html#cut()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **See Also:**[Toolkit.getSystemClipboard()](http://docs.google.com/java/awt/Toolkit.html#getSystemClipboard()), [Clipboard](http://docs.google.com/java/awt/datatransfer/Clipboard.html)

### copy

public void **copy**()

Invokes provideErrorFeedback on the current look and feel, which typically initiates an error beep. The normal behavior of transferring the currently selected range in the associated text model to the system clipboard, and leaving the contents from the model, is not acceptable for a password field.

**Overrides:**[copy](http://docs.google.com/javax/swing/text/JTextComponent.html#copy()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **See Also:**[Toolkit.getSystemClipboard()](http://docs.google.com/java/awt/Toolkit.html#getSystemClipboard()), [Clipboard](http://docs.google.com/java/awt/datatransfer/Clipboard.html)

### getText

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

**Deprecated.** *As of Java 2 platform v1.2, replaced by getPassword.*

Returns the text contained in this TextComponent. If the underlying document is null, will give a NullPointerException.

For security reasons, this method is deprecated. Use the \* getPassword method instead.

**Overrides:**[getText](http://docs.google.com/javax/swing/text/JTextComponent.html#getText()) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Returns:**the text**See Also:**[JTextComponent.setText(java.lang.String)](http://docs.google.com/javax/swing/text/JTextComponent.html#setText(java.lang.String))

### getText

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [String](http://docs.google.com/java/lang/String.html) **getText**(int offs,  
 int len)  
 throws [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html)

**Deprecated.** *As of Java 2 platform v1.2, replaced by getPassword.*

Fetches a portion of the text represented by the component. Returns an empty string if length is 0.

For security reasons, this method is deprecated. Use the getPassword method instead.

**Overrides:**[getText](http://docs.google.com/javax/swing/text/JTextComponent.html#getText(int,%20int)) in class [JTextComponent](http://docs.google.com/javax/swing/text/JTextComponent.html) **Parameters:**offs - the offset >= 0len - the length >= 0 **Returns:**the text **Throws:** [BadLocationException](http://docs.google.com/javax/swing/text/BadLocationException.html) - if the offset or length are invalid

### getPassword

public char[] **getPassword**()

Returns the text contained in this TextComponent. If the underlying document is null, will give a NullPointerException. For stronger security, it is recommended that the returned character array be cleared after use by setting each character to zero.

**Returns:**the text

### paramString

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

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

### getAccessibleContext

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

Returns the AccessibleContext associated with this JPasswordField. For password fields, the AccessibleContext takes the form of an AccessibleJPasswordField. A new AccessibleJPasswordField 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/JTextField.html#getAccessibleContext()) in class [JTextField](http://docs.google.com/javax/swing/JTextField.html) **Returns:**an AccessibleJPasswordField that serves as the AccessibleContext of this JPasswordField

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JPasswordField.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/JPanel.AccessibleJPanel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JPasswordField.AccessibleJPasswordField.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JPasswordField.html)    [**NO FRAMES**](http://docs.google.com/JPasswordField.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#17dp8vu) | [CONSTR](#44sinio) | [METHOD](#2jxsxqh) | DETAIL: FIELD | [CONSTR](#3whwml4) | [METHOD](#2p2csry) |

[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).