| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JMenuItem.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/JMenuBar.AccessibleJMenuBar.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JMenuItem.AccessibleJMenuItem.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JMenuItem.html)    [**NO FRAMES**](http://docs.google.com/JMenuItem.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#1ksv4uv) | DETAIL: FIELD | [CONSTR](#4i7ojhp) | [METHOD](#1pxezwc) |

## **javax.swing**

Class JMenuItem

[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.AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html)  
 **javax.swing.JMenuItem**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.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), [MenuElement](http://docs.google.com/javax/swing/MenuElement.html), [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html) **Direct Known Subclasses:** [JCheckBoxMenuItem](http://docs.google.com/javax/swing/JCheckBoxMenuItem.html), [JMenu](http://docs.google.com/javax/swing/JMenu.html), [JRadioButtonMenuItem](http://docs.google.com/javax/swing/JRadioButtonMenuItem.html)

public class **JMenuItem**extends [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)

An implementation of an item in a menu. A menu item is essentially a button sitting in a list. When the user selects the "button", the action associated with the menu item is performed. A JMenuItem contained in a JPopupMenu performs exactly that function.

Menu items can be configured, and to some degree controlled, by [Action](http://docs.google.com/Action.html)s. Using an Action with a menu item has many benefits beyond directly configuring a menu item. Refer to  [Swing Components Supporting Action](http://docs.google.com/Action.html#buttonActions) for more details, and you can find more information in [How to Use Actions](http://java.sun.com/docs/books/tutorial/uiswing/misc/action.html), a section in *The Java Tutorial*.

For further documentation and for examples, see [How to Use Menus](http://java.sun.com/docs/books/tutorial/uiswing/components/menu.html) in *The Java Tutorial.*

**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:**[JPopupMenu](http://docs.google.com/javax/swing/JPopupMenu.html), [JMenu](http://docs.google.com/javax/swing/JMenu.html), [JCheckBoxMenuItem](http://docs.google.com/javax/swing/JCheckBoxMenuItem.html), [JRadioButtonMenuItem](http://docs.google.com/javax/swing/JRadioButtonMenuItem.html)

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

| **Nested classes/interfaces inherited from class javax.swing.**[**AbstractButton**](http://docs.google.com/javax/swing/AbstractButton.html) |
| --- |
| [AbstractButton.AccessibleAbstractButton](http://docs.google.com/javax/swing/AbstractButton.AccessibleAbstractButton.html), [AbstractButton.ButtonChangeListener](http://docs.google.com/javax/swing/AbstractButton.ButtonChangeListener.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.**[**AbstractButton**](http://docs.google.com/javax/swing/AbstractButton.html) |
| --- |
| [actionListener](http://docs.google.com/javax/swing/AbstractButton.html#actionListener), [BORDER\_PAINTED\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#BORDER_PAINTED_CHANGED_PROPERTY), [changeEvent](http://docs.google.com/javax/swing/AbstractButton.html#changeEvent), [changeListener](http://docs.google.com/javax/swing/AbstractButton.html#changeListener), [CONTENT\_AREA\_FILLED\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#CONTENT_AREA_FILLED_CHANGED_PROPERTY), [DISABLED\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#DISABLED_ICON_CHANGED_PROPERTY), [DISABLED\_SELECTED\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#DISABLED_SELECTED_ICON_CHANGED_PROPERTY), [FOCUS\_PAINTED\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#FOCUS_PAINTED_CHANGED_PROPERTY), [HORIZONTAL\_ALIGNMENT\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#HORIZONTAL_ALIGNMENT_CHANGED_PROPERTY), [HORIZONTAL\_TEXT\_POSITION\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#HORIZONTAL_TEXT_POSITION_CHANGED_PROPERTY), [ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#ICON_CHANGED_PROPERTY), [itemListener](http://docs.google.com/javax/swing/AbstractButton.html#itemListener), [MARGIN\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#MARGIN_CHANGED_PROPERTY), [MNEMONIC\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#MNEMONIC_CHANGED_PROPERTY), [model](http://docs.google.com/javax/swing/AbstractButton.html#model), [MODEL\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#MODEL_CHANGED_PROPERTY), [PRESSED\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#PRESSED_ICON_CHANGED_PROPERTY), [ROLLOVER\_ENABLED\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#ROLLOVER_ENABLED_CHANGED_PROPERTY), [ROLLOVER\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#ROLLOVER_ICON_CHANGED_PROPERTY), [ROLLOVER\_SELECTED\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#ROLLOVER_SELECTED_ICON_CHANGED_PROPERTY), [SELECTED\_ICON\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#SELECTED_ICON_CHANGED_PROPERTY), [TEXT\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#TEXT_CHANGED_PROPERTY), [VERTICAL\_ALIGNMENT\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#VERTICAL_ALIGNMENT_CHANGED_PROPERTY), [VERTICAL\_TEXT\_POSITION\_CHANGED\_PROPERTY](http://docs.google.com/javax/swing/AbstractButton.html#VERTICAL_TEXT_POSITION_CHANGED_PROPERTY) |

| **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** | |
| --- | --- |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem())()            Creates a JMenuItem with no set text or icon. |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Creates a menu item whose properties are taken from the specified Action. |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Creates a JMenuItem with the specified icon. |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem(java.lang.String))([String](http://docs.google.com/java/lang/String.html) text)            Creates a JMenuItem with the specified text. |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem(java.lang.String,%20javax.swing.Icon))([String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Creates a JMenuItem with the specified text and icon. |
| [**JMenuItem**](http://docs.google.com/javax/swing/JMenuItem.html#JMenuItem(java.lang.String,%20int))([String](http://docs.google.com/java/lang/String.html) text, int mnemonic)            Creates a JMenuItem with the specified text and keyboard mnemonic. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**actionPropertyChanged**](http://docs.google.com/javax/swing/JMenuItem.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 button's state in response to property changes in the associated action. |
| void | [**addMenuDragMouseListener**](http://docs.google.com/javax/swing/JMenuItem.html#addMenuDragMouseListener(javax.swing.event.MenuDragMouseListener))([MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html) l)            Adds a MenuDragMouseListener to the menu item. |
| void | [**addMenuKeyListener**](http://docs.google.com/javax/swing/JMenuItem.html#addMenuKeyListener(javax.swing.event.MenuKeyListener))([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)            Adds a MenuKeyListener to the menu item. |
| protected  void | [**configurePropertiesFromAction**](http://docs.google.com/javax/swing/JMenuItem.html#configurePropertiesFromAction(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Sets the properties on this button to match those in the specified Action. |
| protected  void | [**fireMenuDragMouseDragged**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuDragMouseDragged(javax.swing.event.MenuDragMouseEvent))([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuDragMouseEntered**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuDragMouseEntered(javax.swing.event.MenuDragMouseEvent))([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuDragMouseExited**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuDragMouseExited(javax.swing.event.MenuDragMouseEvent))([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuDragMouseReleased**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuDragMouseReleased(javax.swing.event.MenuDragMouseEvent))([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuKeyPressed**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuKeyPressed(javax.swing.event.MenuKeyEvent))([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuKeyReleased**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuKeyReleased(javax.swing.event.MenuKeyEvent))([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireMenuKeyTyped**](http://docs.google.com/javax/swing/JMenuItem.html#fireMenuKeyTyped(javax.swing.event.MenuKeyEvent))([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)            Notifies all listeners that have registered interest for notification on this event type. |
| [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) | [**getAccelerator**](http://docs.google.com/javax/swing/JMenuItem.html#getAccelerator())()            Returns the KeyStroke which serves as an accelerator for the menu item. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JMenuItem.html#getAccessibleContext())()            Returns the AccessibleContext associated with this JMenuItem. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/javax/swing/JMenuItem.html#getComponent())()            Returns the java.awt.Component used to paint this object. |
| [MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html)[] | [**getMenuDragMouseListeners**](http://docs.google.com/javax/swing/JMenuItem.html#getMenuDragMouseListeners())()            Returns an array of all the MenuDragMouseListeners added to this JMenuItem with addMenuDragMouseListener(). |
| [MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html)[] | [**getMenuKeyListeners**](http://docs.google.com/javax/swing/JMenuItem.html#getMenuKeyListeners())()            Returns an array of all the MenuKeyListeners added to this JMenuItem with addMenuKeyListener(). |
| [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] | [**getSubElements**](http://docs.google.com/javax/swing/JMenuItem.html#getSubElements())()            This method returns an array containing the sub-menu components for this menu component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JMenuItem.html#getUIClassID())()            Returns the suffix used to construct the name of the L&F class used to render this component. |
| protected  void | [**init**](http://docs.google.com/javax/swing/JMenuItem.html#init(java.lang.String,%20javax.swing.Icon))([String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Initializes the menu item with the specified text and icon. |
| boolean | [**isArmed**](http://docs.google.com/javax/swing/JMenuItem.html#isArmed())()            Returns whether the menu item is "armed". |
| void | [**menuSelectionChanged**](http://docs.google.com/javax/swing/JMenuItem.html#menuSelectionChanged(boolean))(boolean isIncluded)            Called by the MenuSelectionManager when the MenuElement is selected or unselected. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JMenuItem.html#paramString())()            Returns a string representation of this JMenuItem. |
| void | [**processKeyEvent**](http://docs.google.com/javax/swing/JMenuItem.html#processKeyEvent(java.awt.event.KeyEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.MenuSelectionManager))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e, [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] path, [MenuSelectionManager](http://docs.google.com/javax/swing/MenuSelectionManager.html) manager)            Processes a key event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using MenuSelectionManager's API. |
| void | [**processMenuDragMouseEvent**](http://docs.google.com/javax/swing/JMenuItem.html#processMenuDragMouseEvent(javax.swing.event.MenuDragMouseEvent))([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) e)            Handles mouse drag in a menu. |
| void | [**processMenuKeyEvent**](http://docs.google.com/javax/swing/JMenuItem.html#processMenuKeyEvent(javax.swing.event.MenuKeyEvent))([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) e)            Handles a keystroke in a menu. |
| void | [**processMouseEvent**](http://docs.google.com/javax/swing/JMenuItem.html#processMouseEvent(java.awt.event.MouseEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.MenuSelectionManager))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e, [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] path, [MenuSelectionManager](http://docs.google.com/javax/swing/MenuSelectionManager.html) manager)            Processes a mouse event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using the MenuSelectionManager's API. |
| void | [**removeMenuDragMouseListener**](http://docs.google.com/javax/swing/JMenuItem.html#removeMenuDragMouseListener(javax.swing.event.MenuDragMouseListener))([MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html) l)            Removes a MenuDragMouseListener from the menu item. |
| void | [**removeMenuKeyListener**](http://docs.google.com/javax/swing/JMenuItem.html#removeMenuKeyListener(javax.swing.event.MenuKeyListener))([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)            Removes a MenuKeyListener from the menu item. |
| void | [**setAccelerator**](http://docs.google.com/javax/swing/JMenuItem.html#setAccelerator(javax.swing.KeyStroke))([KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) keyStroke)            Sets the key combination which invokes the menu item's action listeners without navigating the menu hierarchy. |
| void | [**setArmed**](http://docs.google.com/javax/swing/JMenuItem.html#setArmed(boolean))(boolean b)            Identifies the menu item as "armed". |
| void | [**setEnabled**](http://docs.google.com/javax/swing/JMenuItem.html#setEnabled(boolean))(boolean b)            Enables or disables the menu item. |
| void | [**setModel**](http://docs.google.com/javax/swing/JMenuItem.html#setModel(javax.swing.ButtonModel))([ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) newModel)            Sets the model that this button represents. |
| void | [**setUI**](http://docs.google.com/javax/swing/JMenuItem.html#setUI(javax.swing.plaf.MenuItemUI))([MenuItemUI](http://docs.google.com/javax/swing/plaf/MenuItemUI.html) ui)            Sets the look and feel object that renders this component. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JMenuItem.html#updateUI())()            Resets the UI property with a value from the current look and feel. |

| **Methods inherited from class javax.swing.**[**AbstractButton**](http://docs.google.com/javax/swing/AbstractButton.html) |
| --- |
| [addActionListener](http://docs.google.com/javax/swing/AbstractButton.html#addActionListener(java.awt.event.ActionListener)), [addChangeListener](http://docs.google.com/javax/swing/AbstractButton.html#addChangeListener(javax.swing.event.ChangeListener)), [addImpl](http://docs.google.com/javax/swing/AbstractButton.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [addItemListener](http://docs.google.com/javax/swing/AbstractButton.html#addItemListener(java.awt.event.ItemListener)), [checkHorizontalKey](http://docs.google.com/javax/swing/AbstractButton.html#checkHorizontalKey(int,%20java.lang.String)), [checkVerticalKey](http://docs.google.com/javax/swing/AbstractButton.html#checkVerticalKey(int,%20java.lang.String)), [createActionListener](http://docs.google.com/javax/swing/AbstractButton.html#createActionListener()), [createActionPropertyChangeListener](http://docs.google.com/javax/swing/AbstractButton.html#createActionPropertyChangeListener(javax.swing.Action)), [createChangeListener](http://docs.google.com/javax/swing/AbstractButton.html#createChangeListener()), [createItemListener](http://docs.google.com/javax/swing/AbstractButton.html#createItemListener()), [doClick](http://docs.google.com/javax/swing/AbstractButton.html#doClick()), [doClick](http://docs.google.com/javax/swing/AbstractButton.html#doClick(int)), [fireActionPerformed](http://docs.google.com/javax/swing/AbstractButton.html#fireActionPerformed(java.awt.event.ActionEvent)), [fireItemStateChanged](http://docs.google.com/javax/swing/AbstractButton.html#fireItemStateChanged(java.awt.event.ItemEvent)), [fireStateChanged](http://docs.google.com/javax/swing/AbstractButton.html#fireStateChanged()), [getAction](http://docs.google.com/javax/swing/AbstractButton.html#getAction()), [getActionCommand](http://docs.google.com/javax/swing/AbstractButton.html#getActionCommand()), [getActionListeners](http://docs.google.com/javax/swing/AbstractButton.html#getActionListeners()), [getChangeListeners](http://docs.google.com/javax/swing/AbstractButton.html#getChangeListeners()), [getDisabledIcon](http://docs.google.com/javax/swing/AbstractButton.html#getDisabledIcon()), [getDisabledSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#getDisabledSelectedIcon()), [getDisplayedMnemonicIndex](http://docs.google.com/javax/swing/AbstractButton.html#getDisplayedMnemonicIndex()), [getHideActionText](http://docs.google.com/javax/swing/AbstractButton.html#getHideActionText()), [getHorizontalAlignment](http://docs.google.com/javax/swing/AbstractButton.html#getHorizontalAlignment()), [getHorizontalTextPosition](http://docs.google.com/javax/swing/AbstractButton.html#getHorizontalTextPosition()), [getIcon](http://docs.google.com/javax/swing/AbstractButton.html#getIcon()), [getIconTextGap](http://docs.google.com/javax/swing/AbstractButton.html#getIconTextGap()), [getItemListeners](http://docs.google.com/javax/swing/AbstractButton.html#getItemListeners()), [getLabel](http://docs.google.com/javax/swing/AbstractButton.html#getLabel()), [getMargin](http://docs.google.com/javax/swing/AbstractButton.html#getMargin()), [getMnemonic](http://docs.google.com/javax/swing/AbstractButton.html#getMnemonic()), [getModel](http://docs.google.com/javax/swing/AbstractButton.html#getModel()), [getMultiClickThreshhold](http://docs.google.com/javax/swing/AbstractButton.html#getMultiClickThreshhold()), [getPressedIcon](http://docs.google.com/javax/swing/AbstractButton.html#getPressedIcon()), [getRolloverIcon](http://docs.google.com/javax/swing/AbstractButton.html#getRolloverIcon()), [getRolloverSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#getRolloverSelectedIcon()), [getSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#getSelectedIcon()), [getSelectedObjects](http://docs.google.com/javax/swing/AbstractButton.html#getSelectedObjects()), [getText](http://docs.google.com/javax/swing/AbstractButton.html#getText()), [getUI](http://docs.google.com/javax/swing/AbstractButton.html#getUI()), [getVerticalAlignment](http://docs.google.com/javax/swing/AbstractButton.html#getVerticalAlignment()), [getVerticalTextPosition](http://docs.google.com/javax/swing/AbstractButton.html#getVerticalTextPosition()), [imageUpdate](http://docs.google.com/javax/swing/AbstractButton.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [isBorderPainted](http://docs.google.com/javax/swing/AbstractButton.html#isBorderPainted()), [isContentAreaFilled](http://docs.google.com/javax/swing/AbstractButton.html#isContentAreaFilled()), [isFocusPainted](http://docs.google.com/javax/swing/AbstractButton.html#isFocusPainted()), [isRolloverEnabled](http://docs.google.com/javax/swing/AbstractButton.html#isRolloverEnabled()), [isSelected](http://docs.google.com/javax/swing/AbstractButton.html#isSelected()), [paintBorder](http://docs.google.com/javax/swing/AbstractButton.html#paintBorder(java.awt.Graphics)), [removeActionListener](http://docs.google.com/javax/swing/AbstractButton.html#removeActionListener(java.awt.event.ActionListener)), [removeChangeListener](http://docs.google.com/javax/swing/AbstractButton.html#removeChangeListener(javax.swing.event.ChangeListener)), [removeItemListener](http://docs.google.com/javax/swing/AbstractButton.html#removeItemListener(java.awt.event.ItemListener)), [removeNotify](http://docs.google.com/javax/swing/AbstractButton.html#removeNotify()), [setAction](http://docs.google.com/javax/swing/AbstractButton.html#setAction(javax.swing.Action)), [setActionCommand](http://docs.google.com/javax/swing/AbstractButton.html#setActionCommand(java.lang.String)), [setBorderPainted](http://docs.google.com/javax/swing/AbstractButton.html#setBorderPainted(boolean)), [setContentAreaFilled](http://docs.google.com/javax/swing/AbstractButton.html#setContentAreaFilled(boolean)), [setDisabledIcon](http://docs.google.com/javax/swing/AbstractButton.html#setDisabledIcon(javax.swing.Icon)), [setDisabledSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#setDisabledSelectedIcon(javax.swing.Icon)), [setDisplayedMnemonicIndex](http://docs.google.com/javax/swing/AbstractButton.html#setDisplayedMnemonicIndex(int)), [setFocusPainted](http://docs.google.com/javax/swing/AbstractButton.html#setFocusPainted(boolean)), [setHideActionText](http://docs.google.com/javax/swing/AbstractButton.html#setHideActionText(boolean)), [setHorizontalAlignment](http://docs.google.com/javax/swing/AbstractButton.html#setHorizontalAlignment(int)), [setHorizontalTextPosition](http://docs.google.com/javax/swing/AbstractButton.html#setHorizontalTextPosition(int)), [setIcon](http://docs.google.com/javax/swing/AbstractButton.html#setIcon(javax.swing.Icon)), [setIconTextGap](http://docs.google.com/javax/swing/AbstractButton.html#setIconTextGap(int)), [setLabel](http://docs.google.com/javax/swing/AbstractButton.html#setLabel(java.lang.String)), [setLayout](http://docs.google.com/javax/swing/AbstractButton.html#setLayout(java.awt.LayoutManager)), [setMargin](http://docs.google.com/javax/swing/AbstractButton.html#setMargin(java.awt.Insets)), [setMnemonic](http://docs.google.com/javax/swing/AbstractButton.html#setMnemonic(char)), [setMnemonic](http://docs.google.com/javax/swing/AbstractButton.html#setMnemonic(int)), [setMultiClickThreshhold](http://docs.google.com/javax/swing/AbstractButton.html#setMultiClickThreshhold(long)), [setPressedIcon](http://docs.google.com/javax/swing/AbstractButton.html#setPressedIcon(javax.swing.Icon)), [setRolloverEnabled](http://docs.google.com/javax/swing/AbstractButton.html#setRolloverEnabled(boolean)), [setRolloverIcon](http://docs.google.com/javax/swing/AbstractButton.html#setRolloverIcon(javax.swing.Icon)), [setRolloverSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#setRolloverSelectedIcon(javax.swing.Icon)), [setSelected](http://docs.google.com/javax/swing/AbstractButton.html#setSelected(boolean)), [setSelectedIcon](http://docs.google.com/javax/swing/AbstractButton.html#setSelectedIcon(javax.swing.Icon)), [setText](http://docs.google.com/javax/swing/AbstractButton.html#setText(java.lang.String)), [setUI](http://docs.google.com/javax/swing/AbstractButton.html#setUI(javax.swing.plaf.ButtonUI)), [setVerticalAlignment](http://docs.google.com/javax/swing/AbstractButton.html#setVerticalAlignment(int)), [setVerticalTextPosition](http://docs.google.com/javax/swing/AbstractButton.html#setVerticalTextPosition(int)) |

| **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)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [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()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [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)), [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)), [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)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [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()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [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()), [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)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [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)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [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** |
| --- |

### JMenuItem

public **JMenuItem**()

Creates a JMenuItem with no set text or icon.

### JMenuItem

public **JMenuItem**([Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Creates a JMenuItem with the specified icon.

**Parameters:**icon - the icon of the JMenuItem

### JMenuItem

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

Creates a JMenuItem with the specified text.

**Parameters:**text - the text of the JMenuItem

### JMenuItem

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

Creates a menu item whose properties are taken from the specified Action.

**Parameters:**a - the action of the JMenuItem**Since:** 1.3

### JMenuItem

public **JMenuItem**([String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Creates a JMenuItem with the specified text and icon.

**Parameters:**text - the text of the JMenuItemicon - the icon of the JMenuItem

### JMenuItem

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

Creates a JMenuItem with the specified text and keyboard mnemonic.

**Parameters:**text - the text of the JMenuItemmnemonic - the keyboard mnemonic for the JMenuItem

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

### setModel

public void **setModel**([ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) newModel)

Sets the model that this button represents.

**Overrides:**[setModel](http://docs.google.com/javax/swing/AbstractButton.html#setModel(javax.swing.ButtonModel)) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **Parameters:**newModel - the new ButtonModel**See Also:**[AbstractButton.getModel()](http://docs.google.com/javax/swing/AbstractButton.html#getModel())

### init

protected void **init**([String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Initializes the menu item with the specified text and icon.

**Overrides:**[init](http://docs.google.com/javax/swing/AbstractButton.html#init(java.lang.String,%20javax.swing.Icon)) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **Parameters:**text - the text of the JMenuItemicon - the icon of the JMenuItem

### setUI

public void **setUI**([MenuItemUI](http://docs.google.com/javax/swing/plaf/MenuItemUI.html) ui)

Sets the look and feel object that renders this component.

**Parameters:**ui - the JMenuItemUI L&F object**See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### updateUI

public void **updateUI**()

Resets the UI property with a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/AbstractButton.html#updateUI()) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

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

Returns the suffix used to construct the name of the L&F class used to render this component.

**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 "MenuItemUI"**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))

### setArmed

public void **setArmed**(boolean b)

Identifies the menu item as "armed". If the mouse button is released while it is over this item, the menu's action event will fire. If the mouse button is released elsewhere, the event will not fire and the menu item will be disarmed.

**Parameters:**b - true to arm the menu item so it can be selected

### isArmed

public boolean **isArmed**()

Returns whether the menu item is "armed".

**Returns:**true if the menu item is armed, and it can be selected**See Also:**[setArmed(boolean)](http://docs.google.com/javax/swing/JMenuItem.html#setArmed(boolean))

### setEnabled

public void **setEnabled**(boolean b)

Enables or disables the menu item.

**Overrides:**[setEnabled](http://docs.google.com/javax/swing/AbstractButton.html#setEnabled(boolean)) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **Parameters:**b - true to enable the item**See Also:**[Component.isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [Component.isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### setAccelerator

public void **setAccelerator**([KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) keyStroke)

Sets the key combination which invokes the menu item's action listeners without navigating the menu hierarchy. It is the UI's responsibility to install the correct action. Note that when the keyboard accelerator is typed, it will work whether or not the menu is currently displayed.

**Parameters:**keyStroke - the KeyStroke which will serve as an accelerator

### getAccelerator

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

Returns the KeyStroke which serves as an accelerator for the menu item.

**Returns:**a KeyStroke object identifying the accelerator key

### configurePropertiesFromAction

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

Sets the properties on this button 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.

**Overrides:**[configurePropertiesFromAction](http://docs.google.com/javax/swing/AbstractButton.html#configurePropertiesFromAction(javax.swing.Action)) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **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), [AbstractButton.setAction(javax.swing.Action)](http://docs.google.com/javax/swing/AbstractButton.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 button's state in response to property changes in the 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.

**Overrides:**[actionPropertyChanged](http://docs.google.com/javax/swing/AbstractButton.html#actionPropertyChanged(javax.swing.Action,%20java.lang.String)) in class [AbstractButton](http://docs.google.com/javax/swing/AbstractButton.html) **Parameters:**action - the Action associated with this buttonpropertyName - the name of the property that changed**Since:** 1.6 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html), [AbstractButton.configurePropertiesFromAction(javax.swing.Action)](http://docs.google.com/javax/swing/AbstractButton.html#configurePropertiesFromAction(javax.swing.Action))

### processMouseEvent

public void **processMouseEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e,  
 [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] path,  
 [MenuSelectionManager](http://docs.google.com/javax/swing/MenuSelectionManager.html) manager)

Processes a mouse event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using the MenuSelectionManager's API.

Note: you do not have to forward the event to sub-components. This is done automatically by the MenuSelectionManager.

**Specified by:**[processMouseEvent](http://docs.google.com/javax/swing/MenuElement.html#processMouseEvent(java.awt.event.MouseEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.MenuSelectionManager)) in interface [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Parameters:**e - a MouseEventpath - the MenuElement path arraymanager - the MenuSelectionManager

### processKeyEvent

public void **processKeyEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e,  
 [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] path,  
 [MenuSelectionManager](http://docs.google.com/javax/swing/MenuSelectionManager.html) manager)

Processes a key event forwarded from the MenuSelectionManager and changes the menu selection, if necessary, by using MenuSelectionManager's API.

Note: you do not have to forward the event to sub-components. This is done automatically by the MenuSelectionManager.

**Specified by:**[processKeyEvent](http://docs.google.com/javax/swing/MenuElement.html#processKeyEvent(java.awt.event.KeyEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.MenuSelectionManager)) in interface [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Parameters:**e - a KeyEventpath - the MenuElement path arraymanager - the MenuSelectionManager

### processMenuDragMouseEvent

public void **processMenuDragMouseEvent**([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) e)

Handles mouse drag in a menu.

**Parameters:**e - a MenuDragMouseEvent object

### processMenuKeyEvent

public void **processMenuKeyEvent**([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) e)

Handles a keystroke in a menu.

**Parameters:**e - a MenuKeyEvent object

### fireMenuDragMouseEntered

protected void **fireMenuDragMouseEntered**([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuDragMouseExited

protected void **fireMenuDragMouseExited**([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuDragMouseDragged

protected void **fireMenuDragMouseDragged**([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuDragMouseReleased

protected void **fireMenuDragMouseReleased**([MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuKeyPressed

protected void **fireMenuKeyPressed**([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuKeyReleased

protected void **fireMenuKeyReleased**([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### fireMenuKeyTyped

protected void **fireMenuKeyTyped**([MenuKeyEvent](http://docs.google.com/javax/swing/event/MenuKeyEvent.html) event)

Notifies all listeners that have registered interest for notification on this event type.

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

### menuSelectionChanged

public void **menuSelectionChanged**(boolean isIncluded)

Called by the MenuSelectionManager when the MenuElement is selected or unselected.

**Specified by:**[menuSelectionChanged](http://docs.google.com/javax/swing/MenuElement.html#menuSelectionChanged(boolean)) in interface [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Parameters:**isIncluded - true if this menu item is on the part of the menu path that changed, false if this menu is part of the a menu path that changed, but this particular part of that path is still the same**See Also:**[MenuSelectionManager.setSelectedPath(MenuElement[])](http://docs.google.com/javax/swing/MenuSelectionManager.html#setSelectedPath(javax.swing.MenuElement%5B%5D))

### getSubElements

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

This method returns an array containing the sub-menu components for this menu component.

**Specified by:**[getSubElements](http://docs.google.com/javax/swing/MenuElement.html#getSubElements()) in interface [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Returns:**an array of MenuElements

### getComponent

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

Returns the java.awt.Component used to paint this object. The returned component will be used to convert events and detect if an event is inside a menu component.

**Specified by:**[getComponent](http://docs.google.com/javax/swing/MenuElement.html#getComponent()) in interface [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Returns:**the Component that paints this menu item

### addMenuDragMouseListener

public void **addMenuDragMouseListener**([MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html) l)

Adds a MenuDragMouseListener to the menu item.

**Parameters:**l - the MenuDragMouseListener to be added

### removeMenuDragMouseListener

public void **removeMenuDragMouseListener**([MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html) l)

Removes a MenuDragMouseListener from the menu item.

**Parameters:**l - the MenuDragMouseListener to be removed

### getMenuDragMouseListeners

public [MenuDragMouseListener](http://docs.google.com/javax/swing/event/MenuDragMouseListener.html)[] **getMenuDragMouseListeners**()

Returns an array of all the MenuDragMouseListeners added to this JMenuItem with addMenuDragMouseListener().

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

### addMenuKeyListener

public void **addMenuKeyListener**([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)

Adds a MenuKeyListener to the menu item.

**Parameters:**l - the MenuKeyListener to be added

### removeMenuKeyListener

public void **removeMenuKeyListener**([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)

Removes a MenuKeyListener from the menu item.

**Parameters:**l - the MenuKeyListener to be removed

### getMenuKeyListeners

public [MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html)[] **getMenuKeyListeners**()

Returns an array of all the MenuKeyListeners added to this JMenuItem with addMenuKeyListener().

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

### paramString

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

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

### getAccessibleContext

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

Returns the AccessibleContext associated with this JMenuItem. For JMenuItems, the AccessibleContext takes the form of an AccessibleJMenuItem. A new AccessibleJMenuItme 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/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJMenuItem that serves as the AccessibleContext of this JMenuItem

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JMenuItem.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/JMenuBar.AccessibleJMenuBar.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JMenuItem.AccessibleJMenuItem.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JMenuItem.html)    [**NO FRAMES**](http://docs.google.com/JMenuItem.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2s8eyo1) | [CONSTR](#35nkun2) | [METHOD](#1ksv4uv) | DETAIL: FIELD | [CONSTR](#4i7ojhp) | [METHOD](#1pxezwc) |

[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).