| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JPopupMenu.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/JPasswordField.AccessibleJPasswordField.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JPopupMenu.AccessibleJPopupMenu.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JPopupMenu.html)    [**NO FRAMES**](http://docs.google.com/JPopupMenu.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: FIELD | [CONSTR](#2jxsxqh) | [METHOD](#1y810tw) |

## **javax.swing**

Class JPopupMenu

[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.JPopupMenu**

**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), [MenuElement](http://docs.google.com/javax/swing/MenuElement.html) **Direct Known Subclasses:** [BasicComboPopup](http://docs.google.com/javax/swing/plaf/basic/BasicComboPopup.html)

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

An implementation of a popup menu -- a small window that pops up and displays a series of choices. A JPopupMenu is used for the menu that appears when the user selects an item on the menu bar. It is also used for "pull-right" menu that appears when the selects a menu item that activates it. Finally, a JPopupMenu can also be used anywhere else you want a menu to appear. For example, when the user right-clicks in a specified area.

For information and examples of using popup menus, 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).

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JPopupMenu.AccessibleJPopupMenu**](http://docs.google.com/javax/swing/JPopupMenu.AccessibleJPopupMenu.html)            This class implements accessibility support for the JPopupMenu class. |
| static class | [**JPopupMenu.Separator**](http://docs.google.com/javax/swing/JPopupMenu.Separator.html)            A popup menu-specific separator. |

| **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.**[**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 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** | |
| --- | --- |
| [**JPopupMenu**](http://docs.google.com/javax/swing/JPopupMenu.html#JPopupMenu())()            Constructs a JPopupMenu without an "invoker". |
| [**JPopupMenu**](http://docs.google.com/javax/swing/JPopupMenu.html#JPopupMenu(java.lang.String))([String](http://docs.google.com/java/lang/String.html) label)            Constructs a JPopupMenu with the specified title. |

| **Method Summary** | |
| --- | --- |
| [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) | [**add**](http://docs.google.com/javax/swing/JPopupMenu.html#add(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Appends a new menu item to the end of the menu which dispatches the specified Action object. |
| [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) | [**add**](http://docs.google.com/javax/swing/JPopupMenu.html#add(javax.swing.JMenuItem))([JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) menuItem)            Appends the specified menu item to the end of this menu. |
| [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) | [**add**](http://docs.google.com/javax/swing/JPopupMenu.html#add(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Creates a new menu item with the specified text and appends it to the end of this menu. |
| void | [**addMenuKeyListener**](http://docs.google.com/javax/swing/JPopupMenu.html#addMenuKeyListener(javax.swing.event.MenuKeyListener))([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)            Adds a MenuKeyListener to the popup menu. |
| void | [**addPopupMenuListener**](http://docs.google.com/javax/swing/JPopupMenu.html#addPopupMenuListener(javax.swing.event.PopupMenuListener))([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)            Adds a PopupMenu listener. |
| void | [**addSeparator**](http://docs.google.com/javax/swing/JPopupMenu.html#addSeparator())()            Appends a new separator at the end of the menu. |
| protected  [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) | [**createActionChangeListener**](http://docs.google.com/javax/swing/JPopupMenu.html#createActionChangeListener(javax.swing.JMenuItem))([JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) b)            Returns a properly configured PropertyChangeListener which updates the control as changes to the Action occur. |
| protected  [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) | [**createActionComponent**](http://docs.google.com/javax/swing/JPopupMenu.html#createActionComponent(javax.swing.Action))([Action](http://docs.google.com/javax/swing/Action.html) a)            Factory method which creates the JMenuItem for Actions added to the JPopupMenu. |
| protected  void | [**firePopupMenuCanceled**](http://docs.google.com/javax/swing/JPopupMenu.html#firePopupMenuCanceled())()            Notifies PopupMenuListeners that this popup menu is cancelled. |
| protected  void | [**firePopupMenuWillBecomeInvisible**](http://docs.google.com/javax/swing/JPopupMenu.html#firePopupMenuWillBecomeInvisible())()            Notifies PopupMenuListeners that this popup menu will become invisible. |
| protected  void | [**firePopupMenuWillBecomeVisible**](http://docs.google.com/javax/swing/JPopupMenu.html#firePopupMenuWillBecomeVisible())()            Notifies PopupMenuListeners that this popup menu will become visible. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JPopupMenu.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JPopupMenu. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/javax/swing/JPopupMenu.html#getComponent())()            Returns this JPopupMenu component. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAtIndex**](http://docs.google.com/javax/swing/JPopupMenu.html#getComponentAtIndex(int))(int i)  **Deprecated.** *replaced by* [*Container.getComponent(int)*](http://docs.google.com/java/awt/Container.html#getComponent(int)) |
| int | [**getComponentIndex**](http://docs.google.com/javax/swing/JPopupMenu.html#getComponentIndex(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the index of the specified component. |
| static boolean | [**getDefaultLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JPopupMenu.html#getDefaultLightWeightPopupEnabled())()            Gets the defaultLightWeightPopupEnabled property, which by default is true. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getInvoker**](http://docs.google.com/javax/swing/JPopupMenu.html#getInvoker())()            Returns the component which is the 'invoker' of this popup menu. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLabel**](http://docs.google.com/javax/swing/JPopupMenu.html#getLabel())()            Returns the popup menu's label |
| [Insets](http://docs.google.com/java/awt/Insets.html) | [**getMargin**](http://docs.google.com/javax/swing/JPopupMenu.html#getMargin())()            Returns the margin, in pixels, between the popup menu's border and its containees. |
| [MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html)[] | [**getMenuKeyListeners**](http://docs.google.com/javax/swing/JPopupMenu.html#getMenuKeyListeners())()            Returns an array of all the MenuKeyListeners added to this JPopupMenu with addMenuKeyListener(). |
| [PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html)[] | [**getPopupMenuListeners**](http://docs.google.com/javax/swing/JPopupMenu.html#getPopupMenuListeners())()            Returns an array of all the PopupMenuListeners added to this JMenuItem with addPopupMenuListener(). |
| [SingleSelectionModel](http://docs.google.com/javax/swing/SingleSelectionModel.html) | [**getSelectionModel**](http://docs.google.com/javax/swing/JPopupMenu.html#getSelectionModel())()            Returns the model object that handles single selections. |
| [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] | [**getSubElements**](http://docs.google.com/javax/swing/JPopupMenu.html#getSubElements())()            Returns an array of MenuElements containing the submenu for this menu component. |
| [PopupMenuUI](http://docs.google.com/javax/swing/plaf/PopupMenuUI.html) | [**getUI**](http://docs.google.com/javax/swing/JPopupMenu.html#getUI())()            Returns the look and feel (L&F) object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JPopupMenu.html#getUIClassID())()            Returns the name of the L&F class that renders this component. |
| void | [**insert**](http://docs.google.com/javax/swing/JPopupMenu.html#insert(javax.swing.Action,%20int))([Action](http://docs.google.com/javax/swing/Action.html) a, int index)            Inserts a menu item for the specified Action object at a given position. |
| void | [**insert**](http://docs.google.com/javax/swing/JPopupMenu.html#insert(java.awt.Component,%20int))([Component](http://docs.google.com/java/awt/Component.html) component, int index)            Inserts the specified component into the menu at a given position. |
| boolean | [**isBorderPainted**](http://docs.google.com/javax/swing/JPopupMenu.html#isBorderPainted())()            Checks whether the border should be painted. |
| boolean | [**isLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JPopupMenu.html#isLightWeightPopupEnabled())()            Gets the lightWeightPopupEnabled property. |
| boolean | [**isPopupTrigger**](http://docs.google.com/javax/swing/JPopupMenu.html#isPopupTrigger(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)            Returns true if the MouseEvent is considered a popup trigger by the JPopupMenu's currently installed UI. |
| boolean | [**isVisible**](http://docs.google.com/javax/swing/JPopupMenu.html#isVisible())()            Returns true if the popup menu is visible (currently being displayed). |
| void | [**menuSelectionChanged**](http://docs.google.com/javax/swing/JPopupMenu.html#menuSelectionChanged(boolean))(boolean isIncluded)            Messaged when the menubar selection changes to activate or deactivate this menu. |
| void | [**pack**](http://docs.google.com/javax/swing/JPopupMenu.html#pack())()            Lays out the container so that it uses the minimum space needed to display its contents. |
| protected  void | [**paintBorder**](http://docs.google.com/javax/swing/JPopupMenu.html#paintBorder(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints the popup menu's border if the borderPainted property is true. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JPopupMenu.html#paramString())()            Returns a string representation of this JPopupMenu. |
| protected  void | [**processFocusEvent**](http://docs.google.com/javax/swing/JPopupMenu.html#processFocusEvent(java.awt.event.FocusEvent))([FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html) evt)            Processes focus events occurring on this component by dispatching them to any registered FocusListener objects. |
| protected  void | [**processKeyEvent**](http://docs.google.com/javax/swing/JPopupMenu.html#processKeyEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) evt)            Processes key stroke events such as mnemonics and accelerators. |
| void | [**processKeyEvent**](http://docs.google.com/javax/swing/JPopupMenu.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 | [**processMouseEvent**](http://docs.google.com/javax/swing/JPopupMenu.html#processMouseEvent(java.awt.event.MouseEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.MenuSelectionManager))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) event, [MenuElement](http://docs.google.com/javax/swing/MenuElement.html)[] path, [MenuSelectionManager](http://docs.google.com/javax/swing/MenuSelectionManager.html) manager)            This method is required to conform to the MenuElement interface, but it not implemented. |
| void | [**remove**](http://docs.google.com/javax/swing/JPopupMenu.html#remove(int))(int pos)            Removes the component at the specified index from this popup menu. |
| void | [**removeMenuKeyListener**](http://docs.google.com/javax/swing/JPopupMenu.html#removeMenuKeyListener(javax.swing.event.MenuKeyListener))([MenuKeyListener](http://docs.google.com/javax/swing/event/MenuKeyListener.html) l)            Removes a MenuKeyListener from the popup menu. |
| void | [**removePopupMenuListener**](http://docs.google.com/javax/swing/JPopupMenu.html#removePopupMenuListener(javax.swing.event.PopupMenuListener))([PopupMenuListener](http://docs.google.com/javax/swing/event/PopupMenuListener.html) l)            Removes a PopupMenu listener. |
| void | [**setBorderPainted**](http://docs.google.com/javax/swing/JPopupMenu.html#setBorderPainted(boolean))(boolean b)            Sets whether the border should be painted. |
| static void | [**setDefaultLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JPopupMenu.html#setDefaultLightWeightPopupEnabled(boolean))(boolean aFlag)            Sets the default value of the lightWeightPopupEnabled property. |
| void | [**setInvoker**](http://docs.google.com/javax/swing/JPopupMenu.html#setInvoker(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) invoker)            Sets the invoker of this popup menu -- the component in which the popup menu menu is to be displayed. |
| void | [**setLabel**](http://docs.google.com/javax/swing/JPopupMenu.html#setLabel(java.lang.String))([String](http://docs.google.com/java/lang/String.html) label)            Sets the popup menu's label. |
| void | [**setLightWeightPopupEnabled**](http://docs.google.com/javax/swing/JPopupMenu.html#setLightWeightPopupEnabled(boolean))(boolean aFlag)            Sets the value of the lightWeightPopupEnabled property, which by default is true. |
| void | [**setLocation**](http://docs.google.com/javax/swing/JPopupMenu.html#setLocation(int,%20int))(int x, int y)            Sets the location of the upper left corner of the popup menu using x, y coordinates. |
| void | [**setPopupSize**](http://docs.google.com/javax/swing/JPopupMenu.html#setPopupSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) d)            Sets the size of the Popup window using a Dimension object. |
| void | [**setPopupSize**](http://docs.google.com/javax/swing/JPopupMenu.html#setPopupSize(int,%20int))(int width, int height)            Sets the size of the Popup window to the specified width and height. |
| void | [**setSelected**](http://docs.google.com/javax/swing/JPopupMenu.html#setSelected(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) sel)            Sets the currently selected component, This will result in a change to the selection model. |
| void | [**setSelectionModel**](http://docs.google.com/javax/swing/JPopupMenu.html#setSelectionModel(javax.swing.SingleSelectionModel))([SingleSelectionModel](http://docs.google.com/javax/swing/SingleSelectionModel.html) model)            Sets the model object to handle single selections. |
| void | [**setUI**](http://docs.google.com/javax/swing/JPopupMenu.html#setUI(javax.swing.plaf.PopupMenuUI))([PopupMenuUI](http://docs.google.com/javax/swing/plaf/PopupMenuUI.html) ui)            Sets the L&F object that renders this component. |
| void | [**setVisible**](http://docs.google.com/javax/swing/JPopupMenu.html#setVisible(boolean))(boolean b)            Sets the visibility of the popup menu. |
| void | [**show**](http://docs.google.com/javax/swing/JPopupMenu.html#show(java.awt.Component,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) invoker, int x, int y)            Displays the popup menu at the position x,y in the coordinate space of the component invoker. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JPopupMenu.html#updateUI())()            Resets the UI property to a value from the current look and feel. |

| **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)), [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)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [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)), [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)), [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)), [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)), [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)), [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()), [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()), [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)), [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(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** |
| --- |

### JPopupMenu

public **JPopupMenu**()

Constructs a JPopupMenu without an "invoker".

### JPopupMenu

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

Constructs a JPopupMenu with the specified title.

**Parameters:**label - the string that a UI may use to display as a title for the popup menu.

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

### setDefaultLightWeightPopupEnabled

public static void **setDefaultLightWeightPopupEnabled**(boolean aFlag)

Sets the default value of the lightWeightPopupEnabled property.

**Parameters:**aFlag - true if popups can be lightweight, otherwise false**See Also:**[getDefaultLightWeightPopupEnabled()](http://docs.google.com/javax/swing/JPopupMenu.html#getDefaultLightWeightPopupEnabled()), [setLightWeightPopupEnabled(boolean)](http://docs.google.com/javax/swing/JPopupMenu.html#setLightWeightPopupEnabled(boolean))

### getDefaultLightWeightPopupEnabled

public static boolean **getDefaultLightWeightPopupEnabled**()

Gets the defaultLightWeightPopupEnabled property, which by default is true.

**Returns:**the value of the defaultLightWeightPopupEnabled property**See Also:**[setDefaultLightWeightPopupEnabled(boolean)](http://docs.google.com/javax/swing/JPopupMenu.html#setDefaultLightWeightPopupEnabled(boolean))

### getUI

public [PopupMenuUI](http://docs.google.com/javax/swing/plaf/PopupMenuUI.html) **getUI**()

Returns the look and feel (L&F) object that renders this component.

**Returns:**the PopupMenuUI object that renders this component

### setUI

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

Sets the L&F object that renders this component.

**Parameters:**ui - the new PopupMenuUI 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 to a value from the current look and feel.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.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 name of the L&F class that renders 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 "PopupMenuUI"**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))

### processFocusEvent

protected void **processFocusEvent**([FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html) evt)

**Description copied from class:** [**Component**](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)) Processes focus events occurring on this component by dispatching them to any registered FocusListener objects.

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

* A FocusListener object is registered via addFocusListener.
* Focus events are enabled via enableEvents.

If focus events are enabled for a Component, the current KeyboardFocusManager determines whether or not a focus event should be dispatched to registered FocusListener objects. If the events are to be dispatched, the KeyboardFocusManager calls the Component's dispatchEvent method, which results in a call to the Component's processFocusEvent method.

If focus events are enabled for a Component, calling the Component's dispatchEvent method with a FocusEvent as the argument will result in a call to the Component's processFocusEvent method regardless of the current KeyboardFocusManager.

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

**Overrides:**[processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**evt - the focus event**See Also:**[FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html), [Component.addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [Component.dispatchEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent))

### processKeyEvent

protected void **processKeyEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) evt)

Processes key stroke events such as mnemonics and accelerators.

**Overrides:**[processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**evt - the key event to be processed**See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html), [DefaultKeyboardFocusManager](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html), [Component.processEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)), [Component.dispatchEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [Component.addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [Component.isShowing()](http://docs.google.com/java/awt/Component.html#isShowing())

### getSelectionModel

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

Returns the model object that handles single selections.

**Returns:**the selectionModel property**See Also:**[SingleSelectionModel](http://docs.google.com/javax/swing/SingleSelectionModel.html)

### setSelectionModel

public void **setSelectionModel**([SingleSelectionModel](http://docs.google.com/javax/swing/SingleSelectionModel.html) model)

Sets the model object to handle single selections.

**Parameters:**model - the new SingleSelectionModel**See Also:**[SingleSelectionModel](http://docs.google.com/javax/swing/SingleSelectionModel.html)

### add

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

Appends the specified menu item to the end of this menu.

**Parameters:**menuItem - the JMenuItem to add **Returns:**the JMenuItem added

### add

public [JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) **add**([String](http://docs.google.com/java/lang/String.html) s)

Creates a new menu item with the specified text and appends it to the end of this menu.

**Parameters:**s - the string for the menu item to be added

### add

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

Appends a new menu item to the end of the menu which dispatches the specified Action object.

**Parameters:**a - the Action to add to the menu **Returns:**the new menu item**See Also:**[Action](http://docs.google.com/javax/swing/Action.html)

### createActionComponent

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

Factory method which creates the JMenuItem for Actions added to the JPopupMenu.

**Parameters:**a - the Action for the menu item to be added **Returns:**the new menu item**Since:** 1.3 **See Also:**[Action](http://docs.google.com/javax/swing/Action.html)

### createActionChangeListener

protected [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) **createActionChangeListener**([JMenuItem](http://docs.google.com/javax/swing/JMenuItem.html) b)

Returns a properly configured PropertyChangeListener which updates the control as changes to the Action occur.

### remove

public void **remove**(int pos)

Removes the component at the specified index from this popup menu.

**Overrides:**[remove](http://docs.google.com/java/awt/Container.html#remove(int)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**pos - the position of the item to be removed **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the value of pos < 0, or if the value of pos is greater than the number of items**See Also:**[Container.add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [Container.validate()](http://docs.google.com/java/awt/Container.html#validate()), [Container.getComponentCount()](http://docs.google.com/java/awt/Container.html#getComponentCount())

### setLightWeightPopupEnabled

public void **setLightWeightPopupEnabled**(boolean aFlag)

Sets the value of the lightWeightPopupEnabled property, which by default is true. By default, when a look and feel displays a popup, it can choose to use a lightweight (all-Java) popup. Lightweight popup windows are more efficient than heavyweight (native peer) windows, but lightweight and heavyweight components do not mix well in a GUI. If your application mixes lightweight and heavyweight components, you should disable lightweight popups. Some look and feels might always use heavyweight popups, no matter what the value of this property.

**Parameters:**aFlag - false to disable lightweight popups**See Also:**[isLightWeightPopupEnabled()](http://docs.google.com/javax/swing/JPopupMenu.html#isLightWeightPopupEnabled())

### isLightWeightPopupEnabled

public boolean **isLightWeightPopupEnabled**()

Gets the lightWeightPopupEnabled property.

**Returns:**the value of the lightWeightPopupEnabled property**See Also:**[setLightWeightPopupEnabled(boolean)](http://docs.google.com/javax/swing/JPopupMenu.html#setLightWeightPopupEnabled(boolean))

### getLabel

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

Returns the popup menu's label

**Returns:**a string containing the popup menu's label**See Also:**[setLabel(java.lang.String)](http://docs.google.com/javax/swing/JPopupMenu.html#setLabel(java.lang.String))

### setLabel

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

Sets the popup menu's label. Different look and feels may choose to display or not display this.

**Parameters:**label - a string specifying the label for the popup menu**See Also:**[setLabel(java.lang.String)](http://docs.google.com/javax/swing/JPopupMenu.html#setLabel(java.lang.String))

### addSeparator

public void **addSeparator**()

Appends a new separator at the end of the menu.

### insert

public void **insert**([Action](http://docs.google.com/javax/swing/Action.html) a,  
 int index)

Inserts a menu item for the specified Action object at a given position.

**Parameters:**a - the Action object to insertindex - specifies the position at which to insert the Action, where 0 is the first **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index < 0**See Also:**[Action](http://docs.google.com/javax/swing/Action.html)

### insert

public void **insert**([Component](http://docs.google.com/java/awt/Component.html) component,  
 int index)

Inserts the specified component into the menu at a given position.

**Parameters:**component - the Component to insertindex - specifies the position at which to insert the component, where 0 is the first **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index < 0

### addPopupMenuListener

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

Adds a PopupMenu listener.

**Parameters:**l - the PopupMenuListener to add

### removePopupMenuListener

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

Removes a PopupMenu listener.

**Parameters:**l - the PopupMenuListener to remove

### getPopupMenuListeners

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

Returns an array of all the PopupMenuListeners added to this JMenuItem with addPopupMenuListener().

**Returns:**all of the PopupMenuListeners 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 popup menu.

**Parameters:**l - the MenuKeyListener to be added**Since:** 1.5

### removeMenuKeyListener

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

Removes a MenuKeyListener from the popup menu.

**Parameters:**l - the MenuKeyListener to be removed**Since:** 1.5

### getMenuKeyListeners

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

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

**Returns:**all of the MenuKeyListeners added or an empty array if no listeners have been added**Since:** 1.5

### firePopupMenuWillBecomeVisible

protected void **firePopupMenuWillBecomeVisible**()

Notifies PopupMenuListeners that this popup menu will become visible.

### firePopupMenuWillBecomeInvisible

protected void **firePopupMenuWillBecomeInvisible**()

Notifies PopupMenuListeners that this popup menu will become invisible.

### firePopupMenuCanceled

protected void **firePopupMenuCanceled**()

Notifies PopupMenuListeners that this popup menu is cancelled.

### pack

public void **pack**()

Lays out the container so that it uses the minimum space needed to display its contents.

### setVisible

public void **setVisible**(boolean b)

Sets the visibility of the popup menu.

**Overrides:**[setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**b - true to make the popup visible, or false to hide it**See Also:**[Component.isVisible()](http://docs.google.com/java/awt/Component.html#isVisible())

### isVisible

public boolean **isVisible**()

Returns true if the popup menu is visible (currently being displayed).

**Overrides:**[isVisible](http://docs.google.com/java/awt/Component.html#isVisible()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**true if the component is visible, false otherwise**See Also:**[Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### setLocation

public void **setLocation**(int x,  
 int y)

Sets the location of the upper left corner of the popup menu using x, y coordinates.

**Overrides:**[setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**x - the x coordinate of the popup's new position in the screen's coordinate spacey - the y coordinate of the popup's new position in the screen's coordinate space**See Also:**[Component.getLocation()](http://docs.google.com/java/awt/Component.html#getLocation()), [Component.setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### getInvoker

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

Returns the component which is the 'invoker' of this popup menu.

**Returns:**the Component in which the popup menu is displayed

### setInvoker

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

Sets the invoker of this popup menu -- the component in which the popup menu menu is to be displayed.

**Parameters:**invoker - the Component in which the popup menu is displayed

### show

public void **show**([Component](http://docs.google.com/java/awt/Component.html) invoker,  
 int x,  
 int y)

Displays the popup menu at the position x,y in the coordinate space of the component invoker.

**Parameters:**invoker - the component in whose space the popup menu is to appearx - the x coordinate in invoker's coordinate space at which the popup menu is to be displayedy - the y coordinate in invoker's coordinate space at which the popup menu is to be displayed

### getComponentAtIndex

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Component](http://docs.google.com/java/awt/Component.html) **getComponentAtIndex**(int i)

**Deprecated.** *replaced by* [*Container.getComponent(int)*](http://docs.google.com/java/awt/Container.html#getComponent(int))

Returns the component at the specified index.

**Parameters:**i - the index of the component, where 0 is the first **Returns:**the Component at that index

### getComponentIndex

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

Returns the index of the specified component.

**Parameters:**c - the Component to find **Returns:**the index of the component, where 0 is the first; or -1 if the component is not found

### setPopupSize

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

Sets the size of the Popup window using a Dimension object. This is equivalent to setPreferredSize(d).

**Parameters:**d - the Dimension specifying the new size of this component.

### setPopupSize

public void **setPopupSize**(int width,  
 int height)

Sets the size of the Popup window to the specified width and height. This is equivalent to setPreferredSize(new Dimension(width, height)).

**Parameters:**width - the new width of the Popup in pixelsheight - the new height of the Popup in pixels

### setSelected

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

Sets the currently selected component, This will result in a change to the selection model.

**Parameters:**sel - the Component to select

### isBorderPainted

public boolean **isBorderPainted**()

Checks whether the border should be painted.

**Returns:**true if the border is painted, false otherwise**See Also:**[setBorderPainted(boolean)](http://docs.google.com/javax/swing/JPopupMenu.html#setBorderPainted(boolean))

### setBorderPainted

public void **setBorderPainted**(boolean b)

Sets whether the border should be painted.

**Parameters:**b - if true, the border is painted.**See Also:**[isBorderPainted()](http://docs.google.com/javax/swing/JPopupMenu.html#isBorderPainted())

### paintBorder

protected void **paintBorder**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Paints the popup menu's border if the borderPainted property is true.

**Overrides:**[paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**g - the Graphics object**See Also:**[JComponent.paint(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [JComponent.setBorder(javax.swing.border.Border)](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border))

### getMargin

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

Returns the margin, in pixels, between the popup menu's border and its containees.

**Returns:**an Insets object containing the margin values.

### paramString

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

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

### getAccessibleContext

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

Gets the AccessibleContext associated with this JPopupMenu. For JPopupMenus, the AccessibleContext takes the form of an AccessibleJPopupMenu. A new AccessibleJPopupMenu 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 AccessibleJPopupMenu that serves as the AccessibleContext of this JPopupMenu

### processMouseEvent

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

This method is required to conform to the MenuElement interface, but it not implemented.

**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) **See Also:**[MenuElement.processMouseEvent(MouseEvent, MenuElement[], MenuSelectionManager)](http://docs.google.com/javax/swing/MenuElement.html#processMouseEvent(java.awt.event.MouseEvent,%20javax.swing.MenuElement%5B%5D,%20javax.swing.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

### menuSelectionChanged

public void **menuSelectionChanged**(boolean isIncluded)

Messaged when the menubar selection changes to activate or deactivate this menu. This implements the javax.swing.MenuElement interface. Overrides MenuElement.menuSelectionChanged.

**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 is active, false if it is not**See Also:**[MenuElement.menuSelectionChanged(boolean)](http://docs.google.com/javax/swing/MenuElement.html#menuSelectionChanged(boolean))

### getSubElements

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

Returns an array of MenuElements containing the submenu for this menu component. It will only return items conforming to the JMenuElement interface. If popup menu is null returns an empty array. This method is required to conform to the MenuElement interface.

**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 MenuElement objects**See Also:**[MenuElement.getSubElements()](http://docs.google.com/javax/swing/MenuElement.html#getSubElements())

### getComponent

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

Returns this JPopupMenu 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:**this JPopupMenu object**See Also:**[MenuElement.getComponent()](http://docs.google.com/javax/swing/MenuElement.html#getComponent())

### isPopupTrigger

public boolean **isPopupTrigger**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)

Returns true if the MouseEvent is considered a popup trigger by the JPopupMenu's currently installed UI.

**Returns:**true if the mouse event is a popup trigger**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JPopupMenu.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/JPasswordField.AccessibleJPasswordField.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JPopupMenu.AccessibleJPopupMenu.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JPopupMenu.html)    [**NO FRAMES**](http://docs.google.com/JPopupMenu.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#3rdcrjn) | [METHOD](#26in1rg) | DETAIL: FIELD | [CONSTR](#2jxsxqh) | [METHOD](#1y810tw) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).