| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Action.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/AbstractSpinnerModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ActionMap.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/Action.html)    [**NO FRAMES**](http://docs.google.com/Action.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | CONSTR | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | CONSTR | [METHOD](#3j2qqm3) |

## **javax.swing**

Interface Action

**All Superinterfaces:** [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html), [EventListener](http://docs.google.com/java/util/EventListener.html) **All Known Implementing Classes:** [AbstractAction](http://docs.google.com/javax/swing/AbstractAction.html), [BasicDesktopPaneUI.CloseAction](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.CloseAction.html), [BasicDesktopPaneUI.MaximizeAction](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.MaximizeAction.html), [BasicDesktopPaneUI.MinimizeAction](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.MinimizeAction.html), [BasicDesktopPaneUI.NavigateAction](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.NavigateAction.html), [BasicDesktopPaneUI.OpenAction](http://docs.google.com/javax/swing/plaf/basic/BasicDesktopPaneUI.OpenAction.html), [BasicFileChooserUI.ApproveSelectionAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.ApproveSelectionAction.html), [BasicFileChooserUI.CancelSelectionAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.CancelSelectionAction.html), [BasicFileChooserUI.ChangeToParentDirectoryAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.ChangeToParentDirectoryAction.html), [BasicFileChooserUI.GoHomeAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.GoHomeAction.html), [BasicFileChooserUI.NewFolderAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.NewFolderAction.html), [BasicFileChooserUI.UpdateAction](http://docs.google.com/javax/swing/plaf/basic/BasicFileChooserUI.UpdateAction.html), [BasicInternalFrameTitlePane.CloseAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.CloseAction.html), [BasicInternalFrameTitlePane.IconifyAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.IconifyAction.html), [BasicInternalFrameTitlePane.MaximizeAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.MaximizeAction.html), [BasicInternalFrameTitlePane.MoveAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.MoveAction.html), [BasicInternalFrameTitlePane.RestoreAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.RestoreAction.html), [BasicInternalFrameTitlePane.SizeAction](http://docs.google.com/javax/swing/plaf/basic/BasicInternalFrameTitlePane.SizeAction.html), [BasicSliderUI.ActionScroller](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ActionScroller.html), [BasicTreeUI.TreeCancelEditingAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeCancelEditingAction.html), [BasicTreeUI.TreeHomeAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeHomeAction.html), [BasicTreeUI.TreeIncrementAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeIncrementAction.html), [BasicTreeUI.TreePageAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreePageAction.html), [BasicTreeUI.TreeToggleAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeToggleAction.html), [BasicTreeUI.TreeTraverseAction](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.TreeTraverseAction.html), [DefaultEditorKit.BeepAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.BeepAction.html), [DefaultEditorKit.CopyAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.CopyAction.html), [DefaultEditorKit.CutAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.CutAction.html), [DefaultEditorKit.DefaultKeyTypedAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.DefaultKeyTypedAction.html), [DefaultEditorKit.InsertBreakAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.InsertBreakAction.html), [DefaultEditorKit.InsertContentAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.InsertContentAction.html), [DefaultEditorKit.InsertTabAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.InsertTabAction.html), [DefaultEditorKit.PasteAction](http://docs.google.com/javax/swing/text/DefaultEditorKit.PasteAction.html), [HTMLEditorKit.HTMLTextAction](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.HTMLTextAction.html), [HTMLEditorKit.InsertHTMLTextAction](http://docs.google.com/javax/swing/text/html/HTMLEditorKit.InsertHTMLTextAction.html), [MetalFileChooserUI.DirectoryComboBoxAction](http://docs.google.com/javax/swing/plaf/metal/MetalFileChooserUI.DirectoryComboBoxAction.html), [StyledEditorKit.AlignmentAction](http://docs.google.com/javax/swing/text/StyledEditorKit.AlignmentAction.html), [StyledEditorKit.BoldAction](http://docs.google.com/javax/swing/text/StyledEditorKit.BoldAction.html), [StyledEditorKit.FontFamilyAction](http://docs.google.com/javax/swing/text/StyledEditorKit.FontFamilyAction.html), [StyledEditorKit.FontSizeAction](http://docs.google.com/javax/swing/text/StyledEditorKit.FontSizeAction.html), [StyledEditorKit.ForegroundAction](http://docs.google.com/javax/swing/text/StyledEditorKit.ForegroundAction.html), [StyledEditorKit.ItalicAction](http://docs.google.com/javax/swing/text/StyledEditorKit.ItalicAction.html), [StyledEditorKit.StyledTextAction](http://docs.google.com/javax/swing/text/StyledEditorKit.StyledTextAction.html), [StyledEditorKit.UnderlineAction](http://docs.google.com/javax/swing/text/StyledEditorKit.UnderlineAction.html), [TextAction](http://docs.google.com/javax/swing/text/TextAction.html)

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

The Action interface provides a useful extension to the ActionListener interface in cases where the same functionality may be accessed by several controls.

In addition to the actionPerformed method defined by the ActionListener interface, this interface allows the application to define, in a single place:

* One or more text strings that describe the function. These strings can be used, for example, to display the flyover text for a button or to set the text in a menu item.
* One or more icons that depict the function. These icons can be used for the images in a menu control, or for composite entries in a more sophisticated user interface.
* The enabled/disabled state of the functionality. Instead of having to separately disable the menu item and the toolbar button, the application can disable the function that implements this interface. All components which are registered as listeners for the state change then know to disable event generation for that item and to modify the display accordingly.

This interface can be added to an existing class or used to create an adapter (typically, by subclassing AbstractAction). The Action object can then be added to multiple Action-aware containers and connected to Action-capable components. The GUI controls can then be activated or deactivated all at once by invoking the Action object's setEnabled method.

Note that Action implementations tend to be more expensive in terms of storage than a typical ActionListener, which does not offer the benefits of centralized control of functionality and broadcast of property changes. For this reason, you should take care to only use Actions where their benefits are desired, and use simple ActionListeners elsewhere.

#### Swing Components Supporting Action

Many of Swing's components have an Action property. When an Action is set on a component, the following things happen:

* The Action is added as an ActionListener to the component.
* The component configures some of its properties to match the Action.
* The component installs a PropertyChangeListener on the Action so that the component can change its properties to reflect changes in the Action's properties.

The following table describes the properties used by Swing components that support Actions. In the table, *button* refers to any AbstractButton subclass, which includes not only JButton but also classes such as JMenuItem. Unless otherwise stated, a null property value in an Action (or a Action that is null) results in the button's corresponding property being set to null.

| Component Property | Components | Action Key | Notes |
| --- | --- | --- | --- |
| **enabled** | All | The isEnabled method |  |
| **toolTipText** | All | SHORT\_DESCRIPTION |  |
| **actionCommand** | All | ACTION\_COMMAND\_KEY |  |
| **mnemonic** | All buttons | MNEMONIC\_KEY | A null value or Action results in the button's mnemonic property being set to '\0'. |
| **text** | All buttons | NAME | If you do not want the text of the button to mirror that of the Action, set the property hideActionText to true. If hideActionText is true, setting the Action changes the text of the button to null and any changes to NAME are ignored. hideActionText is useful for tool bar buttons that typically only show an Icon. JToolBar.add(Action) sets the property to true if the Action has a non-null value for LARGE\_ICON\_KEY or SMALL\_ICON. |
| **displayedMnemonicIndex** | All buttons | DISPLAYED\_MNEMONIC\_INDEX\_KEY | If the value of DISPLAYED\_MNEMONIC\_INDEX\_KEY is beyond the bounds of the text, it is ignored. When setAction is called, if the value from the Action is null, the displayed mnemonic index is not updated. In any subsequent changes to DISPLAYED\_MNEMONIC\_INDEX\_KEY, null is treated as -1. |
| **icon** | All buttons except of JCheckBox, JToggleButton and JRadioButton. | either LARGE\_ICON\_KEY or SMALL\_ICON | The JMenuItem subclasses only use SMALL\_ICON. All other buttons will use LARGE\_ICON\_KEY; if the value is null they use SMALL\_ICON. |
| **accelerator** | All JMenuItem subclasses, with the exception of JMenu. | ACCELERATOR\_KEY |  |
| **selected** | JToggleButton, JCheckBox, JRadioButton, JCheckBoxMenuItem and JRadioButtonMenuItem | SELECTED\_KEY | Components that honor this property only use the value if it is non-null. For example, if you set an Action that has a null value for SELECTED\_KEY on a JToggleButton, the JToggleButton will not update it's selected state in any way. Similarly, any time the JToggleButton's selected state changes it will only set the value back on the Action if the Action has a non-null value for SELECTED\_KEY.  Components that honor this property keep their selected state in sync with this property. When the same Action is used with multiple components, all the components keep their selected state in sync with this property. Mutually exclusive buttons, such as JToggleButtons in a ButtonGroup, force only one of the buttons to be selected. As such, do not use the same Action that defines a value for the SELECTED\_KEY property with multiple mutually exclusive buttons. |

JPopupMenu, JToolBar and JMenu all provide convenience methods for creating a component and setting the Action on the corresponding component. Refer to each of these classes for more information.

Action uses PropertyChangeListener to inform listeners the Action has changed. The beans specification indicates that a null property name can be used to indicate multiple values have changed. By default Swing components that take an Action do not handle such a change. To indicate that Swing should treat null according to the beans specification set the system property swing.actions.reconfigureOnNull to the String value true.

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

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**ACCELERATOR\_KEY**](http://docs.google.com/javax/swing/Action.html#ACCELERATOR_KEY)            The key used for storing a KeyStroke to be used as the accelerator for the action. |
| static [String](http://docs.google.com/java/lang/String.html) | [**ACTION\_COMMAND\_KEY**](http://docs.google.com/javax/swing/Action.html#ACTION_COMMAND_KEY)            The key used to determine the command String for the ActionEvent that will be created when an Action is going to be notified as the result of residing in a Keymap associated with a JComponent. |
| static [String](http://docs.google.com/java/lang/String.html) | [**DEFAULT**](http://docs.google.com/javax/swing/Action.html#DEFAULT)            Not currently used. |
| static [String](http://docs.google.com/java/lang/String.html) | [**DISPLAYED\_MNEMONIC\_INDEX\_KEY**](http://docs.google.com/javax/swing/Action.html#DISPLAYED_MNEMONIC_INDEX_KEY)            The key used for storing an Integer that corresponds to the index in the text (identified by the NAME property) that the decoration for a mnemonic should be rendered at. |
| static [String](http://docs.google.com/java/lang/String.html) | [**LARGE\_ICON\_KEY**](http://docs.google.com/javax/swing/Action.html#LARGE_ICON_KEY)            The key used for storing an Icon. |
| static [String](http://docs.google.com/java/lang/String.html) | [**LONG\_DESCRIPTION**](http://docs.google.com/javax/swing/Action.html#LONG_DESCRIPTION)            The key used for storing a longer String description for the action, could be used for context-sensitive help. |
| static [String](http://docs.google.com/java/lang/String.html) | [**MNEMONIC\_KEY**](http://docs.google.com/javax/swing/Action.html#MNEMONIC_KEY)            The key used for storing an Integer that corresponds to one of the KeyEvent key codes. |
| static [String](http://docs.google.com/java/lang/String.html) | [**NAME**](http://docs.google.com/javax/swing/Action.html#NAME)            The key used for storing the String name for the action, used for a menu or button. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SELECTED\_KEY**](http://docs.google.com/javax/swing/Action.html#SELECTED_KEY)            The key used for storing a Boolean that corresponds to the selected state. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SHORT\_DESCRIPTION**](http://docs.google.com/javax/swing/Action.html#SHORT_DESCRIPTION)            The key used for storing a short String description for the action, used for tooltip text. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SMALL\_ICON**](http://docs.google.com/javax/swing/Action.html#SMALL_ICON)            The key used for storing a small Icon, such as ImageIcon. |

| **Method Summary** | |
| --- | --- |
| void | [**addPropertyChangeListener**](http://docs.google.com/javax/swing/Action.html#addPropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChange listener. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getValue**](http://docs.google.com/javax/swing/Action.html#getValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) key)            Gets one of this object's properties using the associated key. |
| boolean | [**isEnabled**](http://docs.google.com/javax/swing/Action.html#isEnabled())()            Returns the enabled state of the Action. |
| void | [**putValue**](http://docs.google.com/javax/swing/Action.html#putValue(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) key, [Object](http://docs.google.com/java/lang/Object.html) value)            Sets one of this object's properties using the associated key. |
| void | [**removePropertyChangeListener**](http://docs.google.com/javax/swing/Action.html#removePropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Removes a PropertyChange listener. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/Action.html#setEnabled(boolean))(boolean b)            Sets the enabled state of the Action. |

| **Methods inherited from interface java.awt.event.**[**ActionListener**](http://docs.google.com/java/awt/event/ActionListener.html) |
| --- |
| [actionPerformed](http://docs.google.com/java/awt/event/ActionListener.html#actionPerformed(java.awt.event.ActionEvent)) |

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

### DEFAULT

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

Not currently used.

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

### NAME

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

The key used for storing the String name for the action, used for a menu or button.

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

### SHORT\_DESCRIPTION

static final [String](http://docs.google.com/java/lang/String.html) **SHORT\_DESCRIPTION**

The key used for storing a short String description for the action, used for tooltip text.

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

### LONG\_DESCRIPTION

static final [String](http://docs.google.com/java/lang/String.html) **LONG\_DESCRIPTION**

The key used for storing a longer String description for the action, could be used for context-sensitive help.

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

### SMALL\_ICON

static final [String](http://docs.google.com/java/lang/String.html) **SMALL\_ICON**

The key used for storing a small Icon, such as ImageIcon. This is typically used with menus such as JMenuItem.

If the same Action is used with menus and buttons you'll typically specify both a SMALL\_ICON and a LARGE\_ICON\_KEY. The menu will use the SMALL\_ICON and the button will use the LARGE\_ICON\_KEY.

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

### ACTION\_COMMAND\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **ACTION\_COMMAND\_KEY**

The key used to determine the command String for the ActionEvent that will be created when an Action is going to be notified as the result of residing in a Keymap associated with a JComponent.

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

### ACCELERATOR\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **ACCELERATOR\_KEY**

The key used for storing a KeyStroke to be used as the accelerator for the action.

**Since:** 1.3 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.Action.ACCELERATOR_KEY)

### MNEMONIC\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **MNEMONIC\_KEY**

The key used for storing an Integer that corresponds to one of the KeyEvent key codes. The value is commonly used to specify a mnemonic. For example: myAction.putValue(Action.MNEMONIC\_KEY, KeyEvent.VK\_A) sets the mnemonic of myAction to 'a'.

**Since:** 1.3 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.Action.MNEMONIC_KEY)

### SELECTED\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **SELECTED\_KEY**

The key used for storing a Boolean that corresponds to the selected state. This is typically used only for components that have a meaningful selection state. For example, JRadioButton and JCheckBox make use of this but instances of JMenu don't.

This property differs from the others in that it is both read by the component and set by the component. For example, if an Action is attached to a JCheckBox the selected state of the JCheckBox will be set from that of the Action. If the user clicks on the JCheckBox the selected state of the JCheckBox **and** the Action will **both** be updated.

Note: the value of this field is prefixed with 'Swing' to avoid possible collisions with existing Actions.

**Since:** 1.6 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.Action.SELECTED_KEY)

### DISPLAYED\_MNEMONIC\_INDEX\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **DISPLAYED\_MNEMONIC\_INDEX\_KEY**

The key used for storing an Integer that corresponds to the index in the text (identified by the NAME property) that the decoration for a mnemonic should be rendered at. If the value of this property is greater than or equal to the length of the text, it will treated as -1.

Note: the value of this field is prefixed with 'Swing' to avoid possible collisions with existing Actions.

**Since:** 1.6 **See Also:**[AbstractButton.setDisplayedMnemonicIndex(int)](http://docs.google.com/javax/swing/AbstractButton.html#setDisplayedMnemonicIndex(int)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.Action.DISPLAYED_MNEMONIC_INDEX_KEY)

### LARGE\_ICON\_KEY

static final [String](http://docs.google.com/java/lang/String.html) **LARGE\_ICON\_KEY**

The key used for storing an Icon. This is typically used by buttons, such as JButton and JToggleButton.

If the same Action is used with menus and buttons you'll typically specify both a SMALL\_ICON and a LARGE\_ICON\_KEY. The menu will use the SMALL\_ICON and the button the LARGE\_ICON\_KEY.

Note: the value of this field is prefixed with 'Swing' to avoid possible collisions with existing Actions.

**Since:** 1.6 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.Action.LARGE_ICON_KEY)

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

### getValue

[Object](http://docs.google.com/java/lang/Object.html) **getValue**([String](http://docs.google.com/java/lang/String.html) key)

Gets one of this object's properties using the associated key.

**See Also:**[putValue(java.lang.String, java.lang.Object)](http://docs.google.com/javax/swing/Action.html#putValue(java.lang.String,%20java.lang.Object))

### putValue

void **putValue**([String](http://docs.google.com/java/lang/String.html) key,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

Sets one of this object's properties using the associated key. If the value has changed, a PropertyChangeEvent is sent to listeners.

**Parameters:**key - a String containing the keyvalue - an Object value

### setEnabled

void **setEnabled**(boolean b)

Sets the enabled state of the Action. When enabled, any component associated with this object is active and able to fire this object's actionPerformed method. If the value has changed, a PropertyChangeEvent is sent to listeners.

**Parameters:**b - true to enable this Action, false to disable it

### isEnabled

boolean **isEnabled**()

Returns the enabled state of the Action. When enabled, any component associated with this object is active and able to fire this object's actionPerformed method.

**Returns:**true if this Action is enabled

### addPropertyChangeListener

void **addPropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChange listener. Containers and attached components use these methods to register interest in this Action object. When its enabled state or other property changes, the registered listeners are informed of the change.

**Parameters:**listener - a PropertyChangeListener object

### removePropertyChangeListener

void **removePropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Removes a PropertyChange listener.

**Parameters:**listener - a PropertyChangeListener object**See Also:**[addPropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/javax/swing/Action.html#addPropertyChangeListener(java.beans.PropertyChangeListener))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Action.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/AbstractSpinnerModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/ActionMap.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/Action.html)    [**NO FRAMES**](http://docs.google.com/Action.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | CONSTR | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | CONSTR | [METHOD](#3j2qqm3) |

[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).