| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ButtonModel.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/ButtonGroup.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/CellEditor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ButtonModel.html)    [**NO FRAMES**](http://docs.google.com/ButtonModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

## **javax.swing**

Interface ButtonModel

**All Superinterfaces:** [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **All Known Implementing Classes:** [DefaultButtonModel](http://docs.google.com/javax/swing/DefaultButtonModel.html), [JToggleButton.ToggleButtonModel](http://docs.google.com/javax/swing/JToggleButton.ToggleButtonModel.html)

public interface **ButtonModel**extends [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html)

State model for buttons.

This model is used for regular buttons, as well as check boxes and radio buttons, which are special kinds of buttons. In practice, a button's UI takes the responsibility of calling methods on its model to manage the state, as detailed below:

In simple terms, pressing and releasing the mouse over a regular button triggers the button and causes and ActionEvent to be fired. The same behavior can be produced via a keyboard key defined by the look and feel of the button (typically the SPACE BAR). Pressing and releasing this key while the button has focus will give the same results. For check boxes and radio buttons, the mouse or keyboard equivalent sequence just described causes the button to become selected.

In details, the state model for buttons works as follows when used with the mouse:

Pressing the mouse on top of a button makes the model both armed and pressed. As long as the mouse remains down, the model remains pressed, even if the mouse moves outside the button. On the contrary, the model is only armed while the mouse remains pressed within the bounds of the button (it can move in or out of the button, but the model is only armed during the portion of time spent within the button). A button is triggered, and an ActionEvent is fired, when the mouse is released while the model is armed - meaning when it is released over top of the button after the mouse has previously been pressed on that button (and not already released). Upon mouse release, the model becomes unarmed and unpressed.

In details, the state model for buttons works as follows when used with the keyboard:

Pressing the look and feel defined keyboard key while the button has focus makes the model both armed and pressed. As long as this key remains down, the model remains in this state. Releasing the key sets the model to unarmed and unpressed, triggers the button, and causes an ActionEvent to be fired.

| **Method Summary** | |
| --- | --- |
| void | [**addActionListener**](http://docs.google.com/javax/swing/ButtonModel.html#addActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Adds an ActionListener to the model. |
| void | [**addChangeListener**](http://docs.google.com/javax/swing/ButtonModel.html#addChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Adds a ChangeListener to the model. |
| void | [**addItemListener**](http://docs.google.com/javax/swing/ButtonModel.html#addItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)            Adds an ItemListener to the model. |
| [String](http://docs.google.com/java/lang/String.html) | [**getActionCommand**](http://docs.google.com/javax/swing/ButtonModel.html#getActionCommand())()            Returns the action command string for the button. |
| int | [**getMnemonic**](http://docs.google.com/javax/swing/ButtonModel.html#getMnemonic())()            Gets the keyboard mnemonic for the button. |
| boolean | [**isArmed**](http://docs.google.com/javax/swing/ButtonModel.html#isArmed())()            Indicates partial commitment towards triggering the button. |
| boolean | [**isEnabled**](http://docs.google.com/javax/swing/ButtonModel.html#isEnabled())()            Indicates if the button can be selected or triggered by an input device, such as a mouse pointer. |
| boolean | [**isPressed**](http://docs.google.com/javax/swing/ButtonModel.html#isPressed())()            Indicates if the button is pressed. |
| boolean | [**isRollover**](http://docs.google.com/javax/swing/ButtonModel.html#isRollover())()            Indicates that the mouse is over the button. |
| boolean | [**isSelected**](http://docs.google.com/javax/swing/ButtonModel.html#isSelected())()            Indicates if the button has been selected. |
| void | [**removeActionListener**](http://docs.google.com/javax/swing/ButtonModel.html#removeActionListener(java.awt.event.ActionListener))([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)            Removes an ActionListener from the model. |
| void | [**removeChangeListener**](http://docs.google.com/javax/swing/ButtonModel.html#removeChangeListener(javax.swing.event.ChangeListener))([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)            Removes a ChangeListener from the model. |
| void | [**removeItemListener**](http://docs.google.com/javax/swing/ButtonModel.html#removeItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)            Removes an ItemListener from the model. |
| void | [**setActionCommand**](http://docs.google.com/javax/swing/ButtonModel.html#setActionCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Sets the action command string that gets sent as part of the ActionEvent when the button is triggered. |
| void | [**setArmed**](http://docs.google.com/javax/swing/ButtonModel.html#setArmed(boolean))(boolean b)            Marks the button as armed or unarmed. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/ButtonModel.html#setEnabled(boolean))(boolean b)            Enables or disables the button. |
| void | [**setGroup**](http://docs.google.com/javax/swing/ButtonModel.html#setGroup(javax.swing.ButtonGroup))([ButtonGroup](http://docs.google.com/javax/swing/ButtonGroup.html) group)            Identifies the group the button belongs to -- needed for radio buttons, which are mutually exclusive within their group. |
| void | [**setMnemonic**](http://docs.google.com/javax/swing/ButtonModel.html#setMnemonic(int))(int key)            Sets the keyboard mnemonic (shortcut key or accelerator key) for the button. |
| void | [**setPressed**](http://docs.google.com/javax/swing/ButtonModel.html#setPressed(boolean))(boolean b)            Sets the button to pressed or unpressed. |
| void | [**setRollover**](http://docs.google.com/javax/swing/ButtonModel.html#setRollover(boolean))(boolean b)            Sets or clears the button's rollover state |
| void | [**setSelected**](http://docs.google.com/javax/swing/ButtonModel.html#setSelected(boolean))(boolean b)            Selects or deselects the button. |

| **Methods inherited from interface java.awt.**[**ItemSelectable**](http://docs.google.com/java/awt/ItemSelectable.html) |
| --- |
| [getSelectedObjects](http://docs.google.com/java/awt/ItemSelectable.html#getSelectedObjects()) |

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

### isArmed

boolean **isArmed**()

Indicates partial commitment towards triggering the button.

**Returns:**true if the button is armed, and ready to be triggered**See Also:**[setArmed(boolean)](http://docs.google.com/javax/swing/ButtonModel.html#setArmed(boolean))

### isSelected

boolean **isSelected**()

Indicates if the button has been selected. Only needed for certain types of buttons - such as radio buttons and check boxes.

**Returns:**true if the button is selected

### isEnabled

boolean **isEnabled**()

Indicates if the button can be selected or triggered by an input device, such as a mouse pointer.

**Returns:**true if the button is enabled

### isPressed

boolean **isPressed**()

Indicates if the button is pressed.

**Returns:**true if the button is pressed

### isRollover

boolean **isRollover**()

Indicates that the mouse is over the button.

**Returns:**true if the mouse is over the button

### setArmed

void **setArmed**(boolean b)

Marks the button as armed or unarmed.

**Parameters:**b - whether or not the button should be armed

### setSelected

void **setSelected**(boolean b)

Selects or deselects the button.

**Parameters:**b - true selects the button, false deselects the button

### setEnabled

void **setEnabled**(boolean b)

Enables or disables the button.

**Parameters:**b - whether or not the button should be enabled**See Also:**[isEnabled()](http://docs.google.com/javax/swing/ButtonModel.html#isEnabled())

### setPressed

void **setPressed**(boolean b)

Sets the button to pressed or unpressed.

**Parameters:**b - whether or not the button should be pressed**See Also:**[isPressed()](http://docs.google.com/javax/swing/ButtonModel.html#isPressed())

### setRollover

void **setRollover**(boolean b)

Sets or clears the button's rollover state

**Parameters:**b - whether or not the button is in the rollover state**See Also:**[isRollover()](http://docs.google.com/javax/swing/ButtonModel.html#isRollover())

### setMnemonic

void **setMnemonic**(int key)

Sets the keyboard mnemonic (shortcut key or accelerator key) for the button.

**Parameters:**key - an int specifying the accelerator key

### getMnemonic

int **getMnemonic**()

Gets the keyboard mnemonic for the button.

**Returns:**an int specifying the accelerator key**See Also:**[setMnemonic(int)](http://docs.google.com/javax/swing/ButtonModel.html#setMnemonic(int))

### setActionCommand

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

Sets the action command string that gets sent as part of the ActionEvent when the button is triggered.

**Parameters:**s - the String that identifies the generated event**See Also:**[getActionCommand()](http://docs.google.com/javax/swing/ButtonModel.html#getActionCommand()), [ActionEvent.getActionCommand()](http://docs.google.com/java/awt/event/ActionEvent.html#getActionCommand())

### getActionCommand

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

Returns the action command string for the button.

**Returns:**the String that identifies the generated event**See Also:**[setActionCommand(java.lang.String)](http://docs.google.com/javax/swing/ButtonModel.html#setActionCommand(java.lang.String))

### setGroup

void **setGroup**([ButtonGroup](http://docs.google.com/javax/swing/ButtonGroup.html) group)

Identifies the group the button belongs to -- needed for radio buttons, which are mutually exclusive within their group.

**Parameters:**group - the ButtonGroup the button belongs to

### addActionListener

void **addActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)

Adds an ActionListener to the model.

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

### removeActionListener

void **removeActionListener**([ActionListener](http://docs.google.com/java/awt/event/ActionListener.html) l)

Removes an ActionListener from the model.

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

### addItemListener

void **addItemListener**([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)

Adds an ItemListener to the model.

**Specified by:**[addItemListener](http://docs.google.com/java/awt/ItemSelectable.html#addItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**l - the listener to add**See Also:**[ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html)

### removeItemListener

void **removeItemListener**([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)

Removes an ItemListener from the model.

**Specified by:**[removeItemListener](http://docs.google.com/java/awt/ItemSelectable.html#removeItemListener(java.awt.event.ItemListener)) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html) **Parameters:**l - the listener to remove**See Also:**[ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html)

### addChangeListener

void **addChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Adds a ChangeListener to the model.

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

### removeChangeListener

void **removeChangeListener**([ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html) l)

Removes a ChangeListener from the model.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ButtonModel.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/ButtonGroup.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/CellEditor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/ButtonModel.html)    [**NO FRAMES**](http://docs.google.com/ButtonModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#tyjcwt) |

[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).