| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DefaultButtonModel.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/DefaultBoundedRangeModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultCellEditor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DefaultButtonModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultButtonModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3j2qqm3) | [METHOD](#4i7ojhp) |

## **javax.swing**

Class DefaultButtonModel

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.DefaultButtonModel**

**All Implemented Interfaces:** [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Direct Known Subclasses:** [JToggleButton.ToggleButtonModel](http://docs.google.com/javax/swing/JToggleButton.ToggleButtonModel.html)

public class **DefaultButtonModel**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

The default implementation of a Button component's data model.

**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).

| **Field Summary** | |
| --- | --- |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**actionCommand**](http://docs.google.com/javax/swing/DefaultButtonModel.html#actionCommand)            The action command string fired by the button. |
| static int | [**ARMED**](http://docs.google.com/javax/swing/DefaultButtonModel.html#ARMED)            Identifies the "armed" bit in the bitmask, which indicates partial commitment towards choosing/triggering the button. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | [**changeEvent**](http://docs.google.com/javax/swing/DefaultButtonModel.html#changeEvent)            Only one ChangeEvent is needed per button model instance since the event's only state is the source property. |
| static int | [**ENABLED**](http://docs.google.com/javax/swing/DefaultButtonModel.html#ENABLED)            Identifies the "enabled" bit in the bitmask, which indicates that the button can be selected by an input device (such as a mouse pointer). |
| protected  [ButtonGroup](http://docs.google.com/javax/swing/ButtonGroup.html) | [**group**](http://docs.google.com/javax/swing/DefaultButtonModel.html#group)            The button group that the button belongs to. |
| protected  [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) | [**listenerList**](http://docs.google.com/javax/swing/DefaultButtonModel.html#listenerList)            Stores the listeners on this model. |
| protected  int | [**mnemonic**](http://docs.google.com/javax/swing/DefaultButtonModel.html#mnemonic)            The button's mnemonic. |
| static int | [**PRESSED**](http://docs.google.com/javax/swing/DefaultButtonModel.html#PRESSED)            Identifies the "pressed" bit in the bitmask, which indicates that the button is pressed. |
| static int | [**ROLLOVER**](http://docs.google.com/javax/swing/DefaultButtonModel.html#ROLLOVER)            Identifies the "rollover" bit in the bitmask, which indicates that the mouse is over the button. |
| static int | [**SELECTED**](http://docs.google.com/javax/swing/DefaultButtonModel.html#SELECTED)            Identifies the "selected" bit in the bitmask, which indicates that the button has been selected. |
| protected  int | [**stateMask**](http://docs.google.com/javax/swing/DefaultButtonModel.html#stateMask)            The bitmask used to store the state of the button. |

| **Constructor Summary** | |
| --- | --- |
| [**DefaultButtonModel**](http://docs.google.com/javax/swing/DefaultButtonModel.html#DefaultButtonModel())()            Constructs a DefaultButtonModel. |

| **Method Summary** | |
| --- | --- |
| void | [**addActionListener**](http://docs.google.com/javax/swing/DefaultButtonModel.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/DefaultButtonModel.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/DefaultButtonModel.html#addItemListener(java.awt.event.ItemListener))([ItemListener](http://docs.google.com/java/awt/event/ItemListener.html) l)            Adds an ItemListener to the model. |
| protected  void | [**fireActionPerformed**](http://docs.google.com/javax/swing/DefaultButtonModel.html#fireActionPerformed(java.awt.event.ActionEvent))([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireItemStateChanged**](http://docs.google.com/javax/swing/DefaultButtonModel.html#fireItemStateChanged(java.awt.event.ItemEvent))([ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html) e)            Notifies all listeners that have registered interest for notification on this event type. |
| protected  void | [**fireStateChanged**](http://docs.google.com/javax/swing/DefaultButtonModel.html#fireStateChanged())()            Notifies all listeners that have registered interest for notification on this event type. |
| [String](http://docs.google.com/java/lang/String.html) | [**getActionCommand**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getActionCommand())()            Returns the action command string for the button. |
| [ActionListener](http://docs.google.com/java/awt/event/ActionListener.html)[] | [**getActionListeners**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getActionListeners())()            Returns an array of all the action listeners registered on this DefaultButtonModel. |
| [ChangeListener](http://docs.google.com/javax/swing/event/ChangeListener.html)[] | [**getChangeListeners**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getChangeListeners())()            Returns an array of all the change listeners registered on this DefaultButtonModel. |
| [ButtonGroup](http://docs.google.com/javax/swing/ButtonGroup.html) | [**getGroup**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getGroup())()            Returns the group that the button belongs to. |
| [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)[] | [**getItemListeners**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getItemListeners())()            Returns an array of all the item listeners registered on this DefaultButtonModel. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this model. |
| int | [**getMnemonic**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getMnemonic())()            Gets the keyboard mnemonic for the button. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getSelectedObjects**](http://docs.google.com/javax/swing/DefaultButtonModel.html#getSelectedObjects())()            Overridden to return null. |
| boolean | [**isArmed**](http://docs.google.com/javax/swing/DefaultButtonModel.html#isArmed())()            Indicates partial commitment towards triggering the button. |
| boolean | [**isEnabled**](http://docs.google.com/javax/swing/DefaultButtonModel.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/DefaultButtonModel.html#isPressed())()            Indicates if the button is pressed. |
| boolean | [**isRollover**](http://docs.google.com/javax/swing/DefaultButtonModel.html#isRollover())()            Indicates that the mouse is over the button. |
| boolean | [**isSelected**](http://docs.google.com/javax/swing/DefaultButtonModel.html#isSelected())()            Indicates if the button has been selected. |
| void | [**removeActionListener**](http://docs.google.com/javax/swing/DefaultButtonModel.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/DefaultButtonModel.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/DefaultButtonModel.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/DefaultButtonModel.html#setActionCommand(java.lang.String))([String](http://docs.google.com/java/lang/String.html) actionCommand)            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/DefaultButtonModel.html#setArmed(boolean))(boolean b)            Marks the button as armed or unarmed. |
| void | [**setEnabled**](http://docs.google.com/javax/swing/DefaultButtonModel.html#setEnabled(boolean))(boolean b)            Enables or disables the button. |
| void | [**setGroup**](http://docs.google.com/javax/swing/DefaultButtonModel.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/DefaultButtonModel.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/DefaultButtonModel.html#setPressed(boolean))(boolean b)            Sets the button to pressed or unpressed. |
| void | [**setRollover**](http://docs.google.com/javax/swing/DefaultButtonModel.html#setRollover(boolean))(boolean b)            Sets or clears the button's rollover state |
| void | [**setSelected**](http://docs.google.com/javax/swing/DefaultButtonModel.html#setSelected(boolean))(boolean b)            Selects or deselects the button. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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)) |

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

### stateMask

protected int **stateMask**

The bitmask used to store the state of the button.

### actionCommand

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

The action command string fired by the button.

### group

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

The button group that the button belongs to.

### mnemonic

protected int **mnemonic**

The button's mnemonic.

### changeEvent

protected transient [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) **changeEvent**

Only one ChangeEvent is needed per button model instance since the event's only state is the source property. The source of events generated is always "this".

### listenerList

protected [EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html) **listenerList**

Stores the listeners on this model.

### ARMED

public static final int **ARMED**

Identifies the "armed" bit in the bitmask, which indicates partial commitment towards choosing/triggering the button.

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

### SELECTED

public static final int **SELECTED**

Identifies the "selected" bit in the bitmask, which indicates that the button has been selected. Only needed for certain types of buttons - such as radio button or check box.

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

### PRESSED

public static final int **PRESSED**

Identifies the "pressed" bit in the bitmask, which indicates that the button is pressed.

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

### ENABLED

public static final int **ENABLED**

Identifies the "enabled" bit in the bitmask, which indicates that the button can be selected by an input device (such as a mouse pointer).

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

### ROLLOVER

public static final int **ROLLOVER**

Identifies the "rollover" bit in the bitmask, which indicates that the mouse is over the button.

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

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

### DefaultButtonModel

public **DefaultButtonModel**()

Constructs a DefaultButtonModel.

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

### setActionCommand

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

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

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

### getActionCommand

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

Returns the action command string for the button.

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

### isArmed

public boolean **isArmed**()

Indicates partial commitment towards triggering the button.

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

### isSelected

public boolean **isSelected**()

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

**Specified by:**[isSelected](http://docs.google.com/javax/swing/ButtonModel.html#isSelected()) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Returns:**true if the button is selected

### isEnabled

public boolean **isEnabled**()

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

**Specified by:**[isEnabled](http://docs.google.com/javax/swing/ButtonModel.html#isEnabled()) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Returns:**true if the button is enabled

### isPressed

public boolean **isPressed**()

Indicates if the button is pressed.

**Specified by:**[isPressed](http://docs.google.com/javax/swing/ButtonModel.html#isPressed()) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Returns:**true if the button is pressed

### isRollover

public boolean **isRollover**()

Indicates that the mouse is over the button.

**Specified by:**[isRollover](http://docs.google.com/javax/swing/ButtonModel.html#isRollover()) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Returns:**true if the mouse is over the button

### setArmed

public void **setArmed**(boolean b)

Marks the button as armed or unarmed.

**Specified by:**[setArmed](http://docs.google.com/javax/swing/ButtonModel.html#setArmed(boolean)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**b - whether or not the button should be armed

### setEnabled

public void **setEnabled**(boolean b)

Enables or disables the button.

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

### setSelected

public void **setSelected**(boolean b)

Selects or deselects the button.

**Specified by:**[setSelected](http://docs.google.com/javax/swing/ButtonModel.html#setSelected(boolean)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**b - true selects the button, false deselects the button

### setPressed

public void **setPressed**(boolean b)

Sets the button to pressed or unpressed.

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

### setRollover

public void **setRollover**(boolean b)

Sets or clears the button's rollover state

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

### setMnemonic

public void **setMnemonic**(int key)

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

**Specified by:**[setMnemonic](http://docs.google.com/javax/swing/ButtonModel.html#setMnemonic(int)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**key - an int specifying the accelerator key

### getMnemonic

public int **getMnemonic**()

Gets the keyboard mnemonic for the button.

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

### addChangeListener

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

Adds a ChangeListener to the model.

**Specified by:**[addChangeListener](http://docs.google.com/javax/swing/ButtonModel.html#addChangeListener(javax.swing.event.ChangeListener)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**l - the listener to add

### removeChangeListener

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

Removes a ChangeListener from the model.

**Specified by:**[removeChangeListener](http://docs.google.com/javax/swing/ButtonModel.html#removeChangeListener(javax.swing.event.ChangeListener)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**l - the listener to remove

### getChangeListeners

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

Returns an array of all the change listeners registered on this DefaultButtonModel.

**Returns:**all of this model's ChangeListeners or an empty array if no change listeners are currently registered**Since:** 1.4 **See Also:**[addChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#addChangeListener(javax.swing.event.ChangeListener)), [removeChangeListener(javax.swing.event.ChangeListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#removeChangeListener(javax.swing.event.ChangeListener))

### fireStateChanged

protected void **fireStateChanged**()

Notifies all listeners that have registered interest for notification on this event type. The event instance is created lazily.

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

### addActionListener

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

Adds an ActionListener to the model.

**Specified by:**[addActionListener](http://docs.google.com/javax/swing/ButtonModel.html#addActionListener(java.awt.event.ActionListener)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**l - the listener to add

### removeActionListener

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

Removes an ActionListener from the model.

**Specified by:**[removeActionListener](http://docs.google.com/javax/swing/ButtonModel.html#removeActionListener(java.awt.event.ActionListener)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**l - the listener to remove

### getActionListeners

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

Returns an array of all the action listeners registered on this DefaultButtonModel.

**Returns:**all of this model's ActionListeners or an empty array if no action listeners are currently registered**Since:** 1.4 **See Also:**[addActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#addActionListener(java.awt.event.ActionListener)), [removeActionListener(java.awt.event.ActionListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#removeActionListener(java.awt.event.ActionListener))

### fireActionPerformed

protected void **fireActionPerformed**([ActionEvent](http://docs.google.com/java/awt/event/ActionEvent.html) e)

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

**Parameters:**e - the ActionEvent to deliver to listeners**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### addItemListener

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

### removeItemListener

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

### getItemListeners

public [ItemListener](http://docs.google.com/java/awt/event/ItemListener.html)[] **getItemListeners**()

Returns an array of all the item listeners registered on this DefaultButtonModel.

**Returns:**all of this model's ItemListeners or an empty array if no item listeners are currently registered**Since:** 1.4 **See Also:**[addItemListener(java.awt.event.ItemListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#addItemListener(java.awt.event.ItemListener)), [removeItemListener(java.awt.event.ItemListener)](http://docs.google.com/javax/swing/DefaultButtonModel.html#removeItemListener(java.awt.event.ItemListener))

### fireItemStateChanged

protected void **fireItemStateChanged**([ItemEvent](http://docs.google.com/java/awt/event/ItemEvent.html) e)

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

**Parameters:**e - the ItemEvent to deliver to listeners**See Also:**[EventListenerList](http://docs.google.com/javax/swing/event/EventListenerList.html)

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this model. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a DefaultButtonModel instance m for its action listeners with the following code:

ActionListener[] als = (ActionListener[])(m.getListeners(ActionListener.class));

If no such listeners exist, this method returns an empty array.

**Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this model, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getActionListeners()](http://docs.google.com/javax/swing/DefaultButtonModel.html#getActionListeners()), [getChangeListeners()](http://docs.google.com/javax/swing/DefaultButtonModel.html#getChangeListeners()), [getItemListeners()](http://docs.google.com/javax/swing/DefaultButtonModel.html#getItemListeners())

### getSelectedObjects

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

Overridden to return null.

**Specified by:**[getSelectedObjects](http://docs.google.com/java/awt/ItemSelectable.html#getSelectedObjects()) in interface [ItemSelectable](http://docs.google.com/java/awt/ItemSelectable.html)

### setGroup

public 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.

**Specified by:**[setGroup](http://docs.google.com/javax/swing/ButtonModel.html#setGroup(javax.swing.ButtonGroup)) in interface [ButtonModel](http://docs.google.com/javax/swing/ButtonModel.html) **Parameters:**group - the ButtonGroup the button belongs to

### getGroup

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

Returns the group that the button belongs to. Normally used with radio buttons, which are mutually exclusive within their group.

**Returns:**the ButtonGroup that the button belongs to**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/DefaultButtonModel.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/DefaultBoundedRangeModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/DefaultCellEditor.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/DefaultButtonModel.html)    [**NO FRAMES**](http://docs.google.com/DefaultButtonModel.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3j2qqm3) | [METHOD](#4i7ojhp) |

[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).