| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?javax/swing/event//class-useChangeEvent.html)    [**NO FRAMES**](http://docs.google.com/ChangeEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

**Uses of Class**

**javax.swing.event.ChangeEvent**

| Packages that use [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| [**javax.swing**](#3znysh7) | Provides a set of "lightweight" (all-Java language) components that, to the maximum degree possible, work the same on all platforms. |
| [**javax.swing.colorchooser**](#2et92p0) | Contains classes and interfaces used by the JColorChooser component. |
| [**javax.swing.event**](#tyjcwt) | Provides for events fired by Swing components. |
| [**javax.swing.plaf.basic**](#3dy6vkm) | Provides user interface objects built according to the Basic look and feel. |
| [**javax.swing.table**](#1t3h5sf) | Provides classes and interfaces for dealing with javax.swing.JTable. |
| [**javax.swing.text**](#4d34og8) | Provides classes and interfaces that deal with editable and noneditable text components. |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) | |
| --- | --- |

| Fields in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) declared as [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **JProgressBar.**[**changeEvent**](http://docs.google.com/javax/swing/JProgressBar.html#changeEvent)            Only one ChangeEvent is needed per instance since the event's only interesting property is the immutable source, which is the progress bar. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **JSlider.**[**changeEvent**](http://docs.google.com/javax/swing/JSlider.html#changeEvent)            Only one ChangeEvent is needed per slider instance since the event's only (read-only) state is the source property. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **JTabbedPane.**[**changeEvent**](http://docs.google.com/javax/swing/JTabbedPane.html#changeEvent)            Only one ChangeEvent is needed per TabPane instance since the event's only (read-only) state is the source property. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultBoundedRangeModel.**[**changeEvent**](http://docs.google.com/javax/swing/DefaultBoundedRangeModel.html#changeEvent)            Only one ChangeEvent is needed per model instance since the event's only (read-only) state is the source property. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **AbstractButton.**[**changeEvent**](http://docs.google.com/javax/swing/AbstractButton.html#changeEvent)            Only one ChangeEvent is needed per button instance since the event's only state is the source property. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultSingleSelectionModel.**[**changeEvent**](http://docs.google.com/javax/swing/DefaultSingleSelectionModel.html#changeEvent) |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultButtonModel.**[**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. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **AbstractCellEditor.**[**changeEvent**](http://docs.google.com/javax/swing/AbstractCellEditor.html#changeEvent) |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **MenuSelectionManager.**[**changeEvent**](http://docs.google.com/javax/swing/MenuSelectionManager.html#changeEvent)            Only one ChangeEvent is needed per button model instance since the event's only state is the source property. |

| Methods in [javax.swing](http://docs.google.com/javax/swing/package-summary.html) with parameters of type [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| void | **JTable.**[**columnMarginChanged**](http://docs.google.com/javax/swing/JTable.html#columnMarginChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when a column is moved due to a margin change. |
| void | **JTable.AccessibleJTable.**[**columnMarginChanged**](http://docs.google.com/javax/swing/JTable.AccessibleJTable.html#columnMarginChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Track changes of a column moving due to margin changes. |
| void | **JTable.**[**editingCanceled**](http://docs.google.com/javax/swing/JTable.html#editingCanceled(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when editing is canceled. |
| void | **JTable.AccessibleJTable.**[**editingCanceled**](http://docs.google.com/javax/swing/JTable.AccessibleJTable.html#editingCanceled(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when editing is canceled. |
| void | **JTable.**[**editingStopped**](http://docs.google.com/javax/swing/JTable.html#editingStopped(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when editing is finished. |
| void | **JTable.AccessibleJTable.**[**editingStopped**](http://docs.google.com/javax/swing/JTable.AccessibleJTable.html#editingStopped(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Track changes to a cell's contents. |
| void | **ProgressMonitor.AccessibleProgressMonitor.**[**stateChanged**](http://docs.google.com/javax/swing/ProgressMonitor.AccessibleProgressMonitor.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when the target of the listener has changed its state. |
| void | **JSpinner.DefaultEditor.**[**stateChanged**](http://docs.google.com/javax/swing/JSpinner.DefaultEditor.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            This method is called when the spinner's model's state changes. |
| void | **JSpinner.AccessibleJSpinner.**[**stateChanged**](http://docs.google.com/javax/swing/JSpinner.AccessibleJSpinner.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when the target of the listener has changed its state. |
| void | **JTabbedPane.ModelListener.**[**stateChanged**](http://docs.google.com/javax/swing/JTabbedPane.ModelListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **JTabbedPane.AccessibleJTabbedPane.**[**stateChanged**](http://docs.google.com/javax/swing/JTabbedPane.AccessibleJTabbedPane.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **JScrollPane.AccessibleJScrollPane.**[**stateChanged**](http://docs.google.com/javax/swing/JScrollPane.AccessibleJScrollPane.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when the target of the listener has changed its state. |
| void | **JMenuItem.AccessibleJMenuItem.**[**stateChanged**](http://docs.google.com/javax/swing/JMenuItem.AccessibleJMenuItem.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Supports the change listener interface and fires property changes. |
| void | **AbstractButton.ButtonChangeListener.**[**stateChanged**](http://docs.google.com/javax/swing/AbstractButton.ButtonChangeListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing.colorchooser](http://docs.google.com/javax/swing/colorchooser/package-summary.html) | |
| --- | --- |

| Fields in [javax.swing.colorchooser](http://docs.google.com/javax/swing/colorchooser/package-summary.html) declared as [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultColorSelectionModel.**[**changeEvent**](http://docs.google.com/javax/swing/colorchooser/DefaultColorSelectionModel.html#changeEvent)            Only one ChangeEvent is needed per model instance since the event's only (read-only) state is the source property. |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing.event](http://docs.google.com/javax/swing/event/package-summary.html) | |
| --- | --- |

| Methods in [javax.swing.event](http://docs.google.com/javax/swing/event/package-summary.html) with parameters of type [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| void | **TableColumnModelListener.**[**columnMarginChanged**](http://docs.google.com/javax/swing/event/TableColumnModelListener.html#columnMarginChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Tells listeners that a column was moved due to a margin change. |
| void | **CellEditorListener.**[**editingCanceled**](http://docs.google.com/javax/swing/event/CellEditorListener.html#editingCanceled(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            This tells the listeners the editor has canceled editing |
| void | **CellEditorListener.**[**editingStopped**](http://docs.google.com/javax/swing/event/CellEditorListener.html#editingStopped(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            This tells the listeners the editor has ended editing |
| void | **ChangeListener.**[**stateChanged**](http://docs.google.com/javax/swing/event/ChangeListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when the target of the listener has changed its state. |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing.plaf.basic](http://docs.google.com/javax/swing/plaf/basic/package-summary.html) | |
| --- | --- |

| Methods in [javax.swing.plaf.basic](http://docs.google.com/javax/swing/plaf/basic/package-summary.html) with parameters of type [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| void | **BasicTreeUI.CellEditorHandler.**[**editingCanceled**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.CellEditorHandler.html#editingCanceled(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Messaged when editing has been canceled in the tree. |
| void | **BasicTreeUI.CellEditorHandler.**[**editingStopped**](http://docs.google.com/javax/swing/plaf/basic/BasicTreeUI.CellEditorHandler.html#editingStopped(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Messaged when editing has stopped in the tree. |
| void | **BasicTabbedPaneUI.TabSelectionHandler.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicTabbedPaneUI.TabSelectionHandler.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicSliderUI.ChangeHandler.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicSliderUI.ChangeHandler.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicScrollPaneUI.ViewportChangeHandler.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollPaneUI.ViewportChangeHandler.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicScrollPaneUI.HSBChangeListener.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollPaneUI.HSBChangeListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicScrollPaneUI.VSBChangeListener.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollPaneUI.VSBChangeListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicScrollBarUI.ModelListener.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicScrollBarUI.ModelListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicProgressBarUI.ChangeHandler.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicProgressBarUI.ChangeHandler.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicMenuUI.ChangeHandler.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicMenuUI.ChangeHandler.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |
| void | **BasicButtonListener.**[**stateChanged**](http://docs.google.com/javax/swing/plaf/basic/BasicButtonListener.html#stateChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e) |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing.table](http://docs.google.com/javax/swing/table/package-summary.html) | |
| --- | --- |

| Fields in [javax.swing.table](http://docs.google.com/javax/swing/table/package-summary.html) declared as [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultTableColumnModel.**[**changeEvent**](http://docs.google.com/javax/swing/table/DefaultTableColumnModel.html#changeEvent)            Change event (only one needed) |

| Methods in [javax.swing.table](http://docs.google.com/javax/swing/table/package-summary.html) with parameters of type [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| void | **JTableHeader.**[**columnMarginChanged**](http://docs.google.com/javax/swing/table/JTableHeader.html#columnMarginChanged(javax.swing.event.ChangeEvent))([ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) e)            Invoked when a column is moved due to a margin change. |

| Uses of [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) | |
| --- | --- |

| Fields in [javax.swing.text](http://docs.google.com/javax/swing/text/package-summary.html) declared as [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | |
| --- | --- |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **DefaultCaret.**[**changeEvent**](http://docs.google.com/javax/swing/text/DefaultCaret.html#changeEvent)            The change event for the model. |
| protected  [ChangeEvent](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **StyleContext.NamedStyle.**[**changeEvent**](http://docs.google.com/javax/swing/text/StyleContext.NamedStyle.html#changeEvent)            Only one ChangeEvent is needed per model instance since the event's only (read-only) state is the source property. |

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | [**Class**](http://docs.google.com/javax/swing/event/ChangeEvent.html) | **Use** | [**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   NEXT | [**FRAMES**](http://docs.google.com/index.html?javax/swing/event//class-useChangeEvent.html)    [**NO FRAMES**](http://docs.google.com/ChangeEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).