| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CellEditor.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/ButtonModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/CellRendererPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/CellEditor.html)    [**NO FRAMES**](http://docs.google.com/CellEditor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

## **javax.swing**

Interface CellEditor

**All Known Subinterfaces:** [TableCellEditor](http://docs.google.com/javax/swing/table/TableCellEditor.html), [TreeCellEditor](http://docs.google.com/javax/swing/tree/TreeCellEditor.html) **All Known Implementing Classes:** [AbstractCellEditor](http://docs.google.com/javax/swing/AbstractCellEditor.html), [DefaultCellEditor](http://docs.google.com/javax/swing/DefaultCellEditor.html), [DefaultTreeCellEditor](http://docs.google.com/javax/swing/tree/DefaultTreeCellEditor.html)

public interface **CellEditor**

This interface defines the methods any general editor should be able to implement.

Having this interface enables complex components (the client of the editor) such as JTree and JTable to allow any generic editor to edit values in a table cell, or tree cell, etc. Without this generic editor interface, JTable would have to know about specific editors, such as JTextField, JCheckBox, JComboBox, etc. In addition, without this interface, clients of editors such as JTable would not be able to work with any editors developed in the future by the user or a 3rd party ISV.

To use this interface, a developer creating a new editor can have the new component implement the interface. Or the developer can choose a wrapper based approach and provide a companion object which implements the CellEditor interface (See JCellEditor for example). The wrapper approach is particularly useful if the user want to use a 3rd party ISV editor with JTable, but the ISV didn't implement the CellEditor interface. The user can simply create an object that contains an instance of the 3rd party editor object and "translate" the CellEditor API into the 3rd party editor's API.

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

| **Method Summary** | |
| --- | --- |
| void | [**addCellEditorListener**](http://docs.google.com/javax/swing/CellEditor.html#addCellEditorListener(javax.swing.event.CellEditorListener))([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)            Adds a listener to the list that's notified when the editor stops, or cancels editing. |
| void | [**cancelCellEditing**](http://docs.google.com/javax/swing/CellEditor.html#cancelCellEditing())()            Tells the editor to cancel editing and not accept any partially edited value. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getCellEditorValue**](http://docs.google.com/javax/swing/CellEditor.html#getCellEditorValue())()            Returns the value contained in the editor. |
| boolean | [**isCellEditable**](http://docs.google.com/javax/swing/CellEditor.html#isCellEditable(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) anEvent)            Asks the editor if it can start editing using anEvent. |
| void | [**removeCellEditorListener**](http://docs.google.com/javax/swing/CellEditor.html#removeCellEditorListener(javax.swing.event.CellEditorListener))([CellEditorListener](http://docs.google.com/javax/swing/event/CellEditorListener.html) l)            Removes a listener from the list that's notified |
| boolean | [**shouldSelectCell**](http://docs.google.com/javax/swing/CellEditor.html#shouldSelectCell(java.util.EventObject))([EventObject](http://docs.google.com/java/util/EventObject.html) anEvent)            Returns true if the editing cell should be selected, false otherwise. |
| boolean | [**stopCellEditing**](http://docs.google.com/javax/swing/CellEditor.html#stopCellEditing())()            Tells the editor to stop editing and accept any partially edited value as the value of the editor. |

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

### getCellEditorValue

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

Returns the value contained in the editor.

**Returns:**the value contained in the editor

### isCellEditable

boolean **isCellEditable**([EventObject](http://docs.google.com/java/util/EventObject.html) anEvent)

Asks the editor if it can start editing using anEvent. anEvent is in the invoking component coordinate system. The editor can not assume the Component returned by getCellEditorComponent is installed. This method is intended for the use of client to avoid the cost of setting up and installing the editor component if editing is not possible. If editing can be started this method returns true.

**Parameters:**anEvent - the event the editor should use to consider whether to begin editing or not **Returns:**true if editing can be started**See Also:**[shouldSelectCell(java.util.EventObject)](http://docs.google.com/javax/swing/CellEditor.html#shouldSelectCell(java.util.EventObject))

### shouldSelectCell

boolean **shouldSelectCell**([EventObject](http://docs.google.com/java/util/EventObject.html) anEvent)

Returns true if the editing cell should be selected, false otherwise. Typically, the return value is true, because is most cases the editing cell should be selected. However, it is useful to return false to keep the selection from changing for some types of edits. eg. A table that contains a column of check boxes, the user might want to be able to change those checkboxes without altering the selection. (See Netscape Communicator for just such an example) Of course, it is up to the client of the editor to use the return value, but it doesn't need to if it doesn't want to.

**Parameters:**anEvent - the event the editor should use to start editing **Returns:**true if the editor would like the editing cell to be selected; otherwise returns false**See Also:**[isCellEditable(java.util.EventObject)](http://docs.google.com/javax/swing/CellEditor.html#isCellEditable(java.util.EventObject))

### stopCellEditing

boolean **stopCellEditing**()

Tells the editor to stop editing and accept any partially edited value as the value of the editor. The editor returns false if editing was not stopped; this is useful for editors that validate and can not accept invalid entries.

**Returns:**true if editing was stopped; false otherwise

### cancelCellEditing

void **cancelCellEditing**()

Tells the editor to cancel editing and not accept any partially edited value.

### addCellEditorListener

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

Adds a listener to the list that's notified when the editor stops, or cancels editing.

**Parameters:**l - the CellEditorListener

### removeCellEditorListener

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

Removes a listener from the list that's notified

**Parameters:**l - the CellEditorListener

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CellEditor.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/ButtonModel.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/CellRendererPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/CellEditor.html)    [**NO FRAMES**](http://docs.google.com/CellEditor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#3znysh7) | DETAIL: FIELD | CONSTR | [METHOD](#2et92p0) |

[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).