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

## **javax.swing.undo**

Interface UndoableEdit

**All Known Implementing Classes:** [AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html), [AbstractDocument.ElementEdit](http://docs.google.com/javax/swing/text/AbstractDocument.ElementEdit.html), [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html), [CompoundEdit](http://docs.google.com/javax/swing/undo/CompoundEdit.html), [DefaultStyledDocument.AttributeUndoableEdit](http://docs.google.com/javax/swing/text/DefaultStyledDocument.AttributeUndoableEdit.html), [StateEdit](http://docs.google.com/javax/swing/undo/StateEdit.html), [UndoManager](http://docs.google.com/javax/swing/undo/UndoManager.html)

public interface **UndoableEdit**

An UndoableEdit represents an edit. The edit may be undone, or if already undone the edit may be redone.

UndoableEdit is designed to be used with the UndoManager. As UndoableEdits are generated by an UndoableEditListener they are typically added to the UndoManager. When an UndoableEdit is added to an UndoManager the following occurs (assuming end has not been called on the UndoManager):

1. If the UndoManager contains edits it will call addEdit on the current edit passing in the new edit as the argument. If addEdit returns true the new edit is assumed to have been incorporated into the current edit and the new edit will not be added to the list of current edits. Edits can use addEdit as a way for smaller edits to be incorporated into a larger edit and treated as a single edit.
2. If addEdit returns false replaceEdit is called on the new edit with the current edit passed in as the argument. This is the inverse of addEdit — if the new edit returns true from replaceEdit, the new edit replaces the current edit.

The UndoManager makes use of isSignificant to determine how many edits should be undone or redone. The UndoManager will undo or redo all insignificant edits (isSignificant returns false) between the current edit and the last or next significant edit. addEdit and replaceEdit can be used to treat multiple edits as a single edit, returning false from isSignificant allows for treating can be used to have many smaller edits undone or redone at once. Similar functionality can also be done using the addEdit method.

| **Method Summary** | |
| --- | --- |
| boolean | [**addEdit**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#addEdit(javax.swing.undo.UndoableEdit))([UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) anEdit)            Adds an UndoableEdit to this UndoableEdit. |
| boolean | [**canRedo**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#canRedo())()            Returns true if this edit may be redone. |
| boolean | [**canUndo**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#canUndo())()            Returns true if this edit may be undone. |
| void | [**die**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#die())()            Informs the edit that it should no longer be used. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPresentationName**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getPresentationName())()            Returns a localized, human-readable description of this edit, suitable for use in a change log, for example. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRedoPresentationName**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getRedoPresentationName())()            Returns a localized, human-readable description of the redoable form of this edit, suitable for use as a Redo menu item, for example. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUndoPresentationName**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getUndoPresentationName())()            Returns a localized, human-readable description of the undoable form of this edit, suitable for use as an Undo menu item, for example. |
| boolean | [**isSignificant**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#isSignificant())()            Returns true if this edit is considered significant. |
| void | [**redo**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#redo())()            Re-applies the edit. |
| boolean | [**replaceEdit**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#replaceEdit(javax.swing.undo.UndoableEdit))([UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) anEdit)            Returns true if this UndoableEdit should replace anEdit. |
| void | [**undo**](http://docs.google.com/javax/swing/undo/UndoableEdit.html#undo())()            Undo the edit. |

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

### undo

void **undo**()  
 throws [CannotUndoException](http://docs.google.com/javax/swing/undo/CannotUndoException.html)

Undo the edit.

**Throws:** [CannotUndoException](http://docs.google.com/javax/swing/undo/CannotUndoException.html) - if this edit can not be undone

### canUndo

boolean **canUndo**()

Returns true if this edit may be undone.

**Returns:**true if this edit may be undone

### redo

void **redo**()  
 throws [CannotRedoException](http://docs.google.com/javax/swing/undo/CannotRedoException.html)

Re-applies the edit.

**Throws:** [CannotRedoException](http://docs.google.com/javax/swing/undo/CannotRedoException.html) - if this edit can not be redone

### canRedo

boolean **canRedo**()

Returns true if this edit may be redone.

**Returns:**true if this edit may be redone

### die

void **die**()

Informs the edit that it should no longer be used. Once an UndoableEdit has been marked as dead it can no longer be undone or redone.

This is a useful hook for cleaning up state no longer needed once undoing or redoing is impossible--for example, deleting file resources used by objects that can no longer be undeleted. UndoManager calls this before it dequeues edits.

Note that this is a one-way operation. There is no "un-die" method.

**See Also:**[CompoundEdit.die()](http://docs.google.com/javax/swing/undo/CompoundEdit.html#die())

### addEdit

boolean **addEdit**([UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) anEdit)

Adds an UndoableEdit to this UndoableEdit. This method can be used to coalesce smaller edits into a larger compound edit. For example, text editors typically allow undo operations to apply to words or sentences. The text editor may choose to generate edits on each key event, but allow those edits to be coalesced into a more user-friendly unit, such as a word. In this case, the UndoableEdit would override addEdit to return true when the edits may be coalesced.

A return value of true indicates anEdit was incorporated into this edit. A return value of false indicates anEdit may not be incorporated into this edit.

Typically the receiver is already in the queue of a UndoManager (or other UndoableEditListener), and is being given a chance to incorporate anEdit rather than letting it be added to the queue in turn.

If true is returned, from now on anEdit must return false from canUndo and canRedo, and must throw the appropriate exception on undo or redo.

**Parameters:**anEdit - the edit to be added **Returns:**true if anEdit may be incorporated into this edit

### replaceEdit

boolean **replaceEdit**([UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) anEdit)

Returns true if this UndoableEdit should replace anEdit. This method is used by CompoundEdit and the UndoManager; it is called if anEdit could not be added to the current edit (addEdit returns false).

This method provides a way for an edit to replace an existing edit.

This message is the opposite of addEdit--anEdit has typically already been queued in an UndoManager (or other UndoableEditListener), and the receiver is being given a chance to take its place.

If true is returned, from now on anEdit must return false from canUndo() and canRedo(), and must throw the appropriate exception on undo() or redo().

**Parameters:**anEdit - the edit that replaces the current edit **Returns:**true if this edit should replace anEdit

### isSignificant

boolean **isSignificant**()

Returns true if this edit is considered significant. A significant edit is typically an edit that should be presented to the user, perhaps on a menu item or tooltip. The UndoManager will undo, or redo, all insignificant edits to the next significant edit.

**Returns:**true if this edit is significant

### getPresentationName

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

Returns a localized, human-readable description of this edit, suitable for use in a change log, for example.

**Returns:**description of this edit

### getUndoPresentationName

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

Returns a localized, human-readable description of the undoable form of this edit, suitable for use as an Undo menu item, for example. This is typically derived from getPresentationName.

**Returns:**a description of the undoable form of this edit

### getRedoPresentationName

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

Returns a localized, human-readable description of the redoable form of this edit, suitable for use as a Redo menu item, for example. This is typically derived from getPresentationName.

**Returns:**a description of the redoable form of this edit

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/UndoableEdit.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/undo/StateEditable.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/undo/UndoableEditSupport.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/undo/UndoableEdit.html)    [**NO FRAMES**](http://docs.google.com/UndoableEdit.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).