| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CompoundEdit.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/CannotUndoException.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/undo/StateEdit.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/undo/CompoundEdit.html)    [**NO FRAMES**](http://docs.google.com/CompoundEdit.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

## **javax.swing.undo**

Class CompoundEdit

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) **Direct Known Subclasses:** [AbstractDocument.DefaultDocumentEvent](http://docs.google.com/javax/swing/text/AbstractDocument.DefaultDocumentEvent.html), [UndoManager](http://docs.google.com/javax/swing/undo/UndoManager.html)

public class **CompoundEdit**extends [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html)

A concrete subclass of AbstractUndoableEdit, used to assemble little UndoableEdits into great big ones.

| **Field Summary** | |
| --- | --- |
| protected  [Vector](http://docs.google.com/java/util/Vector.html)<[UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)> | [**edits**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#edits)            The collection of UndoableEdits undone/redone en masse by this CompoundEdit. |

| **Fields inherited from class javax.swing.undo.**[**AbstractUndoableEdit**](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) |
| --- |
| [RedoName](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#RedoName), [UndoName](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#UndoName) |

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

| **Method Summary** | |
| --- | --- |
| boolean | [**addEdit**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#addEdit(javax.swing.undo.UndoableEdit))([UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) anEdit)            If this edit is inProgress, accepts anEdit and returns true. |
| boolean | [**canRedo**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#canRedo())()            Returns false if isInProgress or if super returns false. |
| boolean | [**canUndo**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#canUndo())()            Returns false if isInProgress or if super returns false. |
| void | [**die**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#die())()            Sends die to each subedit, in the reverse of the order that they were added. |
| void | [**end**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#end())()            Sets inProgress to false. |
| [String](http://docs.google.com/java/lang/String.html) | [**getPresentationName**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#getPresentationName())()            Returns getPresentationName from the last UndoableEdit added to edits. |
| [String](http://docs.google.com/java/lang/String.html) | [**getRedoPresentationName**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#getRedoPresentationName())()            Returns getRedoPresentationName from the last UndoableEdit added to edits. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUndoPresentationName**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#getUndoPresentationName())()            Returns getUndoPresentationName from the last UndoableEdit added to edits. |
| boolean | [**isInProgress**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#isInProgress())()            Returns true if this edit is in progress--that is, it has not received end. |
| boolean | [**isSignificant**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#isSignificant())()            Returns true if any of the UndoableEdits in edits do. |
| protected  [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) | [**lastEdit**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#lastEdit())()            Returns the last UndoableEdit in edits, or null if edits is empty. |
| void | [**redo**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#redo())()            Sends redo to all contained UndoableEdits in the order in which they were added. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#toString())()            Returns a string that displays and identifies this object's properties. |
| void | [**undo**](http://docs.google.com/javax/swing/undo/CompoundEdit.html#undo())()            Sends undo to all contained UndoableEdits in the reverse of the order in which they were added. |

| **Methods inherited from class javax.swing.undo.**[**AbstractUndoableEdit**](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) |
| --- |
| [replaceEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#replaceEdit(javax.swing.undo.UndoableEdit)) |

| **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()), [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** |
| --- |

### edits

protected [Vector](http://docs.google.com/java/util/Vector.html)<[UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)> **edits**

The collection of UndoableEdits undone/redone en masse by this CompoundEdit.

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

### CompoundEdit

public **CompoundEdit**()

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

### undo

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

Sends undo to all contained UndoableEdits in the reverse of the order in which they were added.

**Specified by:**[undo](http://docs.google.com/javax/swing/undo/UndoableEdit.html#undo()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[undo](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#undo()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Throws:** [CannotUndoException](http://docs.google.com/javax/swing/undo/CannotUndoException.html) - if canUndo returns false**See Also:**[AbstractUndoableEdit.canUndo()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#canUndo())

### redo

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

Sends redo to all contained UndoableEdits in the order in which they were added.

**Specified by:**[redo](http://docs.google.com/javax/swing/undo/UndoableEdit.html#redo()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[redo](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#redo()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Throws:** [CannotRedoException](http://docs.google.com/javax/swing/undo/CannotRedoException.html) - if canRedo returns false**See Also:**[AbstractUndoableEdit.canRedo()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#canRedo())

### lastEdit

protected [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html) **lastEdit**()

Returns the last UndoableEdit in edits, or null if edits is empty.

### die

public void **die**()

Sends die to each subedit, in the reverse of the order that they were added.

**Specified by:**[die](http://docs.google.com/javax/swing/undo/UndoableEdit.html#die()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[die](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#die()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **See Also:**[die()](http://docs.google.com/javax/swing/undo/CompoundEdit.html#die())

### addEdit

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

If this edit is inProgress, accepts anEdit and returns true.

The last edit added to this CompoundEdit is given a chance to addEdit(anEdit). If it refuses (returns false), anEdit is given a chance to replaceEdit the last edit. If anEdit returns false here, it is added to edits.

**Specified by:**[addEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html#addEdit(javax.swing.undo.UndoableEdit)) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[addEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#addEdit(javax.swing.undo.UndoableEdit)) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Parameters:**anEdit - the edit to be added **Returns:**true if the edit is inProgress; otherwise returns false**See Also:**[UndoableEdit.addEdit(javax.swing.undo.UndoableEdit)](http://docs.google.com/javax/swing/undo/UndoableEdit.html#addEdit(javax.swing.undo.UndoableEdit))

### end

public void **end**()

Sets inProgress to false.

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

### canUndo

public boolean **canUndo**()

Returns false if isInProgress or if super returns false.

**Specified by:**[canUndo](http://docs.google.com/javax/swing/undo/UndoableEdit.html#canUndo()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[canUndo](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#canUndo()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**true if this edit is alive and hasBeenDone is true**See Also:**[isInProgress()](http://docs.google.com/javax/swing/undo/CompoundEdit.html#isInProgress())

### canRedo

public boolean **canRedo**()

Returns false if isInProgress or if super returns false.

**Specified by:**[canRedo](http://docs.google.com/javax/swing/undo/UndoableEdit.html#canRedo()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[canRedo](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#canRedo()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**true if this edit is alive and hasBeenDone is false**See Also:**[isInProgress()](http://docs.google.com/javax/swing/undo/CompoundEdit.html#isInProgress())

### isInProgress

public boolean **isInProgress**()

Returns true if this edit is in progress--that is, it has not received end. This generally means that edits are still being added to it.

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

### isSignificant

public boolean **isSignificant**()

Returns true if any of the UndoableEdits in edits do. Returns false if they all return false.

**Specified by:**[isSignificant](http://docs.google.com/javax/swing/undo/UndoableEdit.html#isSignificant()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[isSignificant](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#isSignificant()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**true**See Also:**[UndoableEdit.isSignificant()](http://docs.google.com/javax/swing/undo/UndoableEdit.html#isSignificant())

### getPresentationName

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

Returns getPresentationName from the last UndoableEdit added to edits. If edits is empty, calls super.

**Specified by:**[getPresentationName](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getPresentationName()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[getPresentationName](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getPresentationName()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**the empty string ""**See Also:**[AbstractUndoableEdit.getUndoPresentationName()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getUndoPresentationName()), [AbstractUndoableEdit.getRedoPresentationName()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getRedoPresentationName())

### getUndoPresentationName

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

Returns getUndoPresentationName from the last UndoableEdit added to edits. If edits is empty, calls super.

**Specified by:**[getUndoPresentationName](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getUndoPresentationName()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[getUndoPresentationName](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getUndoPresentationName()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**the value from the defaults table with key AbstractUndoableEdit.undoText, followed by a space, followed by getPresentationName unless getPresentationName is "" in which case, the defaults value is returned alone.**See Also:**[AbstractUndoableEdit.getPresentationName()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getPresentationName())

### getRedoPresentationName

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

Returns getRedoPresentationName from the last UndoableEdit added to edits. If edits is empty, calls super.

**Specified by:**[getRedoPresentationName](http://docs.google.com/javax/swing/undo/UndoableEdit.html#getRedoPresentationName()) in interface [UndoableEdit](http://docs.google.com/javax/swing/undo/UndoableEdit.html)**Overrides:**[getRedoPresentationName](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getRedoPresentationName()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**the value from the defaults table with key AbstractUndoableEdit.redoText, followed by a space, followed by getPresentationName unless getPresentationName is "" in which case, the defaults value is returned alone.**See Also:**[AbstractUndoableEdit.getPresentationName()](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#getPresentationName())

### toString

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

Returns a string that displays and identifies this object's properties.

**Overrides:**[toString](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html#toString()) in class [AbstractUndoableEdit](http://docs.google.com/javax/swing/undo/AbstractUndoableEdit.html) **Returns:**a String representation of this object

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/CompoundEdit.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/CannotUndoException.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/undo/StateEdit.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/undo/CompoundEdit.html)    [**NO FRAMES**](http://docs.google.com/CompoundEdit.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#3rdcrjn) | [METHOD](#lnxbz9) |

[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).