| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TreeModelEvent.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/event/TreeExpansionListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/event/TreeModelListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/event/TreeModelEvent.html)    [**NO FRAMES**](http://docs.google.com/TreeModelEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#z337ya) |

## **javax.swing.event**

Class TreeModelEvent

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.EventObject](http://docs.google.com/java/util/EventObject.html)  
 **javax.swing.event.TreeModelEvent**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

public class **TreeModelEvent**extends [EventObject](http://docs.google.com/java/util/EventObject.html)

Encapsulates information describing changes to a tree model, and used to notify tree model listeners of the change. For more information and examples see [How to Write a Tree Model Listener](http://java.sun.com/docs/books/tutorial/uiswing/events/treemodellistener.html), a section in *The Java Tutorial.*

**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  int[] | [**childIndices**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#childIndices)            Indices identifying the position of where the children were. |
| protected  [Object](http://docs.google.com/java/lang/Object.html)[] | [**children**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#children)            Children that have been removed. |
| protected  [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**path**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#path)            Path to the parent of the nodes that have changed. |

| **Fields inherited from class java.util.**[**EventObject**](http://docs.google.com/java/util/EventObject.html) |
| --- |
| [source](http://docs.google.com/java/util/EventObject.html#source) |

| **Constructor Summary** | |
| --- | --- |
| [**TreeModelEvent**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) source, [Object](http://docs.google.com/java/lang/Object.html)[] path)            Used to create an event when the node structure has changed in some way, identifying the path to the root of a modified subtree as an array of Objects. |
| [**TreeModelEvent**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20java.lang.Object%5B%5D,%20int%5B%5D,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) source, [Object](http://docs.google.com/java/lang/Object.html)[] path, int[] childIndices, [Object](http://docs.google.com/java/lang/Object.html)[] children)            Used to create an event when nodes have been changed, inserted, or removed, identifying the path to the parent of the modified items as an array of Objects. |
| [**TreeModelEvent**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20javax.swing.tree.TreePath))([Object](http://docs.google.com/java/lang/Object.html) source, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)            Used to create an event when the node structure has changed in some way, identifying the path to the root of the modified subtree as a TreePath object. |
| [**TreeModelEvent**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20javax.swing.tree.TreePath,%20int%5B%5D,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) source, [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path, int[] childIndices, [Object](http://docs.google.com/java/lang/Object.html)[] children)            Used to create an event when nodes have been changed, inserted, or removed, identifying the path to the parent of the modified items as a TreePath object. |

| **Method Summary** | |
| --- | --- |
| int[] | [**getChildIndices**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getChildIndices())()            Returns the values of the child indexes. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getChildren**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getChildren())()            Returns the objects that are children of the node identified by getPath at the locations specified by getChildIndices. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getPath**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getPath())()            Convenience method to get the array of objects from the TreePath instance that this event wraps. |
| [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) | [**getTreePath**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getTreePath())()            For all events, except treeStructureChanged, returns the parent of the changed nodes. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/swing/event/TreeModelEvent.html#toString())()            Returns a string that displays and identifies this object's properties. |

| **Methods inherited from class java.util.**[**EventObject**](http://docs.google.com/java/util/EventObject.html) |
| --- |
| [getSource](http://docs.google.com/java/util/EventObject.html#getSource()) |

| **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** |
| --- |

### path

protected [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **path**

Path to the parent of the nodes that have changed.

### childIndices

protected int[] **childIndices**

Indices identifying the position of where the children were.

### children

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

Children that have been removed.

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

### TreeModelEvent

public **TreeModelEvent**([Object](http://docs.google.com/java/lang/Object.html) source,  
 [Object](http://docs.google.com/java/lang/Object.html)[] path,  
 int[] childIndices,  
 [Object](http://docs.google.com/java/lang/Object.html)[] children)

Used to create an event when nodes have been changed, inserted, or removed, identifying the path to the parent of the modified items as an array of Objects. All of the modified objects are siblings which are direct descendents (not grandchildren) of the specified parent. The positions at which the inserts, deletes, or changes occurred are specified by an array of int. The indexes in that array must be in order, from lowest to highest.

For changes, the indexes in the model correspond exactly to the indexes of items currently displayed in the UI. As a result, it is not really critical if the indexes are not in their exact order. But after multiple inserts or deletes, the items currently in the UI no longer correspond to the items in the model. It is therefore critical to specify the indexes properly for inserts and deletes.

For inserts, the indexes represent the *final* state of the tree, after the inserts have occurred. Since the indexes must be specified in order, the most natural processing methodology is to do the inserts starting at the lowest index and working towards the highest. Accumulate a Vector of Integer objects that specify the insert-locations as you go, then convert the Vector to an array of int to create the event. When the postition-index equals zero, the node is inserted at the beginning of the list. When the position index equals the size of the list, the node is "inserted" at (appended to) the end of the list.

For deletes, the indexes represent the *initial* state of the tree, before the deletes have occurred. Since the indexes must be specified in order, the most natural processing methodology is to use a delete-counter. Start by initializing the counter to zero and start work through the list from lowest to higest. Every time you do a delete, add the current value of the delete-counter to the index-position where the delete occurred, and append the result to a Vector of delete-locations, using addElement(). Then increment the delete-counter. The index positions stored in the Vector therefore reflect the effects of all previous deletes, so they represent each object's position in the initial tree. (You could also start at the highest index and working back towards the lowest, accumulating a Vector of delete-locations as you go using the insertElementAt(Integer, 0).) However you produce the Vector of initial-positions, you then need to convert the Vector of Integer objects to an array of int to create the event.

**Notes:**

* Like the insertNodeInto method in the DefaultTreeModel class, insertElementAt appends to the Vector when the index matches the size of the vector. So you can use insertElementAt(Integer, 0) even when the vector is empty.

To create a node changed event for the root node, specify the parent and the child indices as null.**Parameters:**source - the Object responsible for generating the event (typically the creator of the event object passes this for its value)path - an array of Object identifying the path to the parent of the modified item(s), where the first element of the array is the Object stored at the root node and the last element is the Object stored at the parent nodechildIndices - an array of int that specifies the index values of the removed items. The indices must be in sorted order, from lowest to highestchildren - an array of Object containing the inserted, removed, or changed objects**See Also:**[TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)

### TreeModelEvent

public **TreeModelEvent**([Object](http://docs.google.com/java/lang/Object.html) source,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path,  
 int[] childIndices,  
 [Object](http://docs.google.com/java/lang/Object.html)[] children)

Used to create an event when nodes have been changed, inserted, or removed, identifying the path to the parent of the modified items as a TreePath object. For more information on how to specify the indexes and objects, see TreeModelEvent(Object,Object[],int[],Object[]).

**Parameters:**source - the Object responsible for generating the event (typically the creator of the event object passes this for its value)path - a TreePath object that identifies the path to the parent of the modified item(s)childIndices - an array of int that specifies the index values of the modified itemschildren - an array of Object containing the inserted, removed, or changed objects**See Also:**[TreeModelEvent(Object,Object[],int[],Object[])](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20java.lang.Object%5B%5D,%20int%5B%5D,%20java.lang.Object%5B%5D))

### TreeModelEvent

public **TreeModelEvent**([Object](http://docs.google.com/java/lang/Object.html) source,  
 [Object](http://docs.google.com/java/lang/Object.html)[] path)

Used to create an event when the node structure has changed in some way, identifying the path to the root of a modified subtree as an array of Objects. A structure change event might involve nodes swapping position, for example, or it might encapsulate multiple inserts and deletes in the subtree stemming from the node, where the changes may have taken place at different levels of the subtree.**Note:**

JTree collapses all nodes under the specified node, so that only its immediate children are visible.

**Parameters:**source - the Object responsible for generating the event (typically the creator of the event object passes this for its value)path - an array of Object identifying the path to the root of the modified subtree, where the first element of the array is the object stored at the root node and the last element is the object stored at the changed node**See Also:**[TreePath](http://docs.google.com/javax/swing/tree/TreePath.html)

### TreeModelEvent

public **TreeModelEvent**([Object](http://docs.google.com/java/lang/Object.html) source,  
 [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) path)

Used to create an event when the node structure has changed in some way, identifying the path to the root of the modified subtree as a TreePath object. For more information on this event specification, see TreeModelEvent(Object,Object[]).

**Parameters:**source - the Object responsible for generating the event (typically the creator of the event object passes this for its value)path - a TreePath object that identifies the path to the change. In the DefaultTreeModel, this object contains an array of user-data objects, but a subclass of TreePath could use some totally different mechanism -- for example, a node ID number**See Also:**[TreeModelEvent(Object,Object[])](http://docs.google.com/javax/swing/event/TreeModelEvent.html#TreeModelEvent(java.lang.Object,%20java.lang.Object%5B%5D))

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

### getTreePath

public [TreePath](http://docs.google.com/javax/swing/tree/TreePath.html) **getTreePath**()

For all events, except treeStructureChanged, returns the parent of the changed nodes. For treeStructureChanged events, returns the ancestor of the structure that has changed. This and getChildIndices are used to get a list of the effected nodes.

The one exception to this is a treeNodesChanged event that is to identify the root, in which case this will return the root and getChildIndices will return null.

**Returns:**the TreePath used in identifying the changed nodes.**See Also:**[TreePath.getLastPathComponent()](http://docs.google.com/javax/swing/tree/TreePath.html#getLastPathComponent())

### getPath

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

Convenience method to get the array of objects from the TreePath instance that this event wraps.

**Returns:**an array of Objects, where the first Object is the one stored at the root and the last object is the one stored at the node identified by the path

### getChildren

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

Returns the objects that are children of the node identified by getPath at the locations specified by getChildIndices. If this is a removal event the returned objects are no longer children of the parent node.

**Returns:**an array of Object containing the children specified by the event**See Also:**[getPath()](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getPath()), [getChildIndices()](http://docs.google.com/javax/swing/event/TreeModelEvent.html#getChildIndices())

### getChildIndices

public int[] **getChildIndices**()

Returns the values of the child indexes. If this is a removal event the indexes point to locations in the initial list where items were removed. If it is an insert, the indices point to locations in the final list where the items were added. For node changes, the indices point to the locations of the modified nodes.

**Returns:**an array of int containing index locations for the children specified by the event

### 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/java/util/EventObject.html#toString()) in class [EventObject](http://docs.google.com/java/util/EventObject.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/TreeModelEvent.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/event/TreeExpansionListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/event/TreeModelListener.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/event/TreeModelEvent.html)    [**NO FRAMES**](http://docs.google.com/TreeModelEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#lnxbz9) | [METHOD](#z337ya) |

[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).