| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NamingEvent.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/naming/event/NamespaceChangeListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/event/NamingExceptionEvent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/event/NamingEvent.html)    [**NO FRAMES**](http://docs.google.com/NamingEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

## **javax.naming.event**

Class NamingEvent

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

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

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

This class represents an event fired by a naming/directory service.

The NamingEvent's state consists of

* The event source: the EventContext which fired this event.
* The event type.
* The new binding: information about the object after the change.
* The old binding: information about the object before the change.
* Change information: information about the change that triggered this event; usually service provider-specific or server-specific information.

Note that the event source is always the same EventContext *instance* that the listener has registered with. Furthermore, the names of the bindings in the NamingEvent are always relative to that instance. For example, suppose a listener makes the following registration:

NamespaceChangeListener listener = ...;  
 src.addNamingListener("x", SUBTREE\_SCOPE, listener);

When an object named "x/y" is subsequently deleted, the corresponding NamingEvent (evt) must contain:

evt.getEventContext() == src  
 evt.getOldBinding().getName().equals("x/y")

Care must be taken when multiple threads are accessing the same EventContext concurrently. See the [package description](http://docs.google.com/package-summary.html#THREADING) for more information on threading issues.

**Since:** 1.3 **See Also:**[NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html), [EventContext](http://docs.google.com/javax/naming/event/EventContext.html), [Serialized Form](http://docs.google.com/serialized-form.html#javax.naming.event.NamingEvent)

| **Field Summary** | |
| --- | --- |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**changeInfo**](http://docs.google.com/javax/naming/event/NamingEvent.html#changeInfo)            Contains information about the change that generated this event. |
| protected  [Binding](http://docs.google.com/javax/naming/Binding.html) | [**newBinding**](http://docs.google.com/javax/naming/event/NamingEvent.html#newBinding)            Contains information about the object after the change. |
| static int | [**OBJECT\_ADDED**](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_ADDED)            Naming event type for indicating that a new object has been added. |
| static int | [**OBJECT\_CHANGED**](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_CHANGED)            Naming event type for indicating that an object has been changed. |
| static int | [**OBJECT\_REMOVED**](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_REMOVED)            Naming event type for indicating that an object has been removed. |
| static int | [**OBJECT\_RENAMED**](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_RENAMED)            Naming event type for indicating that an object has been renamed. |
| protected  [Binding](http://docs.google.com/javax/naming/Binding.html) | [**oldBinding**](http://docs.google.com/javax/naming/event/NamingEvent.html#oldBinding)            Contains information about the object before the change. |
| protected  int | [**type**](http://docs.google.com/javax/naming/event/NamingEvent.html#type)            Contains the type of this event. |

| **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** | |
| --- | --- |
| [**NamingEvent**](http://docs.google.com/javax/naming/event/NamingEvent.html#NamingEvent(javax.naming.event.EventContext,%20int,%20javax.naming.Binding,%20javax.naming.Binding,%20java.lang.Object))([EventContext](http://docs.google.com/javax/naming/event/EventContext.html) source, int type, [Binding](http://docs.google.com/javax/naming/Binding.html) newBd, [Binding](http://docs.google.com/javax/naming/Binding.html) oldBd, [Object](http://docs.google.com/java/lang/Object.html) changeInfo)            Constructs an instance of NamingEvent. |

| **Method Summary** | |
| --- | --- |
| void | [**dispatch**](http://docs.google.com/javax/naming/event/NamingEvent.html#dispatch(javax.naming.event.NamingListener))([NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) listener)            Invokes the appropriate listener method on this event. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getChangeInfo**](http://docs.google.com/javax/naming/event/NamingEvent.html#getChangeInfo())()            Retrieves the change information for this event. |
| [EventContext](http://docs.google.com/javax/naming/event/EventContext.html) | [**getEventContext**](http://docs.google.com/javax/naming/event/NamingEvent.html#getEventContext())()            Retrieves the event source that fired this event. |
| [Binding](http://docs.google.com/javax/naming/Binding.html) | [**getNewBinding**](http://docs.google.com/javax/naming/event/NamingEvent.html#getNewBinding())()            Retrieves the binding of the object after the change. |
| [Binding](http://docs.google.com/javax/naming/Binding.html) | [**getOldBinding**](http://docs.google.com/javax/naming/event/NamingEvent.html#getOldBinding())()            Retrieves the binding of the object before the change. |
| int | [**getType**](http://docs.google.com/javax/naming/event/NamingEvent.html#getType())()            Returns the type of this event. |

| **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()), [toString](http://docs.google.com/java/util/EventObject.html#toString()) |

| **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** |
| --- |

### OBJECT\_ADDED

public static final int **OBJECT\_ADDED**

Naming event type for indicating that a new object has been added. The value of this constant is 0.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.event.NamingEvent.OBJECT_ADDED)

### OBJECT\_REMOVED

public static final int **OBJECT\_REMOVED**

Naming event type for indicating that an object has been removed. The value of this constant is 1.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.event.NamingEvent.OBJECT_REMOVED)

### OBJECT\_RENAMED

public static final int **OBJECT\_RENAMED**

Naming event type for indicating that an object has been renamed. Note that some services might fire multiple events for a single logical rename operation. For example, the rename operation might be implemented by adding a binding with the new name and removing the old binding.

The old/new binding in NamingEvent may be null if the old name or new name is outside of the scope for which the listener has registered.

When an interior node in the namespace tree has been renamed, the topmost node which is part of the listener's scope should used to generate a rename event. The extent to which this can be supported is provider-specific. For example, a service might generate rename notifications for all descendants of the changed interior node and the corresponding provider might not be able to prevent those notifications from being propagated to the listeners.

The value of this constant is 2.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.event.NamingEvent.OBJECT_RENAMED)

### OBJECT\_CHANGED

public static final int **OBJECT\_CHANGED**

Naming event type for indicating that an object has been changed. The changes might include the object's attributes, or the object itself. Note that some services might fire multiple events for a single modification. For example, the modification might be implemented by first removing the old binding and adding a new binding containing the same name but a different object.

The value of this constant is 3.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.event.NamingEvent.OBJECT_CHANGED)

### changeInfo

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

Contains information about the change that generated this event.

### type

protected int **type**

Contains the type of this event.

**See Also:**[OBJECT\_ADDED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_ADDED), [OBJECT\_REMOVED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_REMOVED), [OBJECT\_RENAMED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_RENAMED), [OBJECT\_CHANGED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_CHANGED)

### oldBinding

protected [Binding](http://docs.google.com/javax/naming/Binding.html) **oldBinding**

Contains information about the object before the change.

### newBinding

protected [Binding](http://docs.google.com/javax/naming/Binding.html) **newBinding**

Contains information about the object after the change.

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

### NamingEvent

public **NamingEvent**([EventContext](http://docs.google.com/javax/naming/event/EventContext.html) source,  
 int type,  
 [Binding](http://docs.google.com/javax/naming/Binding.html) newBd,  
 [Binding](http://docs.google.com/javax/naming/Binding.html) oldBd,  
 [Object](http://docs.google.com/java/lang/Object.html) changeInfo)

Constructs an instance of NamingEvent.

The names in newBd and oldBd are to be resolved relative to the event source source. For an OBJECT\_ADDED event type, newBd must not be null. For an OBJECT\_REMOVED event type, oldBd must not be null. For an OBJECT\_CHANGED event type, newBd and oldBd must not be null. For an OBJECT\_RENAMED event type, one of newBd or oldBd may be null if the new or old binding is outside of the scope for which the listener has registered.

**Parameters:**source - The non-null context that fired this event.type - The type of the event.newBd - A possibly null binding before the change. See method description.oldBd - A possibly null binding after the change. See method description.changeInfo - A possibly null object containing information about the change.**See Also:**[OBJECT\_ADDED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_ADDED), [OBJECT\_REMOVED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_REMOVED), [OBJECT\_RENAMED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_RENAMED), [OBJECT\_CHANGED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_CHANGED)

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

### getType

public int **getType**()

Returns the type of this event.

**Returns:**The type of this event.**See Also:**[OBJECT\_ADDED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_ADDED), [OBJECT\_REMOVED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_REMOVED), [OBJECT\_RENAMED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_RENAMED), [OBJECT\_CHANGED](http://docs.google.com/javax/naming/event/NamingEvent.html#OBJECT_CHANGED)

### getEventContext

public [EventContext](http://docs.google.com/javax/naming/event/EventContext.html) **getEventContext**()

Retrieves the event source that fired this event. This returns the same object as EventObject.getSource().

If the result of this method is used to access the event source, for example, to look up the object or get its attributes, then it needs to be locked because implementations of Context are not guaranteed to be thread-safe (and EventContext is a subinterface of Context). See the [package description](http://docs.google.com/package-summary.html#THREADING) for more information on threading issues.

**Returns:**The non-null context that fired this event.

### getOldBinding

public [Binding](http://docs.google.com/javax/naming/Binding.html) **getOldBinding**()

Retrieves the binding of the object before the change.

The binding must be nonnull if the object existed before the change relative to the source context (getEventContext()). That is, it must be nonnull for OBJECT\_REMOVED and OBJECT\_CHANGED. For OBJECT\_RENAMED, it is null if the object before the rename is outside of the scope for which the listener has registered interest; it is nonnull if the object is inside the scope before the rename.

The name in the binding is to be resolved relative to the event source getEventContext(). The object returned by Binding.getObject() may be null if such information is unavailable.

**Returns:**The possibly null binding of the object before the change.

### getNewBinding

public [Binding](http://docs.google.com/javax/naming/Binding.html) **getNewBinding**()

Retrieves the binding of the object after the change.

The binding must be nonnull if the object existed after the change relative to the source context (getEventContext()). That is, it must be nonnull for OBJECT\_ADDED and OBJECT\_CHANGED. For OBJECT\_RENAMED, it is null if the object after the rename is outside the scope for which the listener registered interest; it is nonnull if the object is inside the scope after the rename.

The name in the binding is to be resolved relative to the event source getEventContext(). The object returned by Binding.getObject() may be null if such information is unavailable.

**Returns:**The possibly null binding of the object after the change.

### getChangeInfo

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

Retrieves the change information for this event. The value of the change information is service-specific. For example, it could be an ID that identifies the change in a change log on the server.

**Returns:**The possibly null change information of this event.

### dispatch

public void **dispatch**([NamingListener](http://docs.google.com/javax/naming/event/NamingListener.html) listener)

Invokes the appropriate listener method on this event. The default implementation of this method handles the following event types: OBJECT\_ADDED, OBJECT\_REMOVED, OBJECT\_RENAMED, OBJECT\_CHANGED.

The listener method is executed in the same thread as this method. See the [package description](http://docs.google.com/package-summary.html#THREADING) for more information on threading issues.

**Parameters:**listener - The nonnull listener.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NamingEvent.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/naming/event/NamespaceChangeListener.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/event/NamingExceptionEvent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/event/NamingEvent.html)    [**NO FRAMES**](http://docs.google.com/NamingEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#z337ya) | [METHOD](#1y810tw) |

[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).