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

## **javax.management**

Interface NotificationBroadcaster

**All Known Subinterfaces:** [ModelMBean](http://docs.google.com/javax/management/modelmbean/ModelMBean.html), [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html), [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) **All Known Implementing Classes:** [CounterMonitor](http://docs.google.com/javax/management/monitor/CounterMonitor.html), [GaugeMonitor](http://docs.google.com/javax/management/monitor/GaugeMonitor.html), [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html), [MBeanServerDelegate](http://docs.google.com/javax/management/MBeanServerDelegate.html), [Monitor](http://docs.google.com/javax/management/monitor/Monitor.html), [NotificationBroadcasterSupport](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html), [RelationService](http://docs.google.com/javax/management/relation/RelationService.html), [RequiredModelMBean](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html), [RMIConnectorServer](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html), [StandardEmitterMBean](http://docs.google.com/javax/management/StandardEmitterMBean.html), [StringMonitor](http://docs.google.com/javax/management/monitor/StringMonitor.html), [Timer](http://docs.google.com/javax/management/timer/Timer.html)

public interface **NotificationBroadcaster**

Interface implemented by an MBean that emits Notifications. It allows a listener to be registered with the MBean as a notification listener.

### Notification dispatch

When an MBean emits a notification, it considers each listener that has been added with [addNotificationListener](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) and not subsequently removed with [removeNotificationListener](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener)). If a filter was provided with that listener, and if the filter's [isNotificationEnabled](http://docs.google.com/javax/management/NotificationFilter.html#isNotificationEnabled(javax.management.Notification)) method returns false, the listener is ignored. Otherwise, the listener's [handleNotification](http://docs.google.com/javax/management/NotificationListener.html#handleNotification(javax.management.Notification,%20java.lang.Object)) method is called with the notification, as well as the handback object that was provided to addNotificationListener.

If the same listener is added more than once, it is considered as many times as it was added. It is often useful to add the same listener with different filters or handback objects.

Implementations of this interface can differ regarding the thread in which the methods of filters and listeners are called.

If the method call of a filter or listener throws an [Exception](http://docs.google.com/java/lang/Exception.html), then that exception should not prevent other listeners from being invoked. However, if the method call throws an [Error](http://docs.google.com/java/lang/Error.html), then it is recommended that processing of the notification stop at that point, and if it is possible to propagate the Error to the sender of the notification, this should be done.

New code should use the [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) interface instead.

Implementations of this interface and of NotificationEmitter should be careful about synchronization. In particular, it is not a good idea for an implementation to hold any locks while it is calling a listener. To deal with the possibility that the list of listeners might change while a notification is being dispatched, a good strategy is to use a [CopyOnWriteArrayList](http://docs.google.com/java/util/concurrent/CopyOnWriteArrayList.html) for this list.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener, [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) filter, [Object](http://docs.google.com/java/lang/Object.html) handback)            Adds a listener to this MBean. |
| [MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)[] | [**getNotificationInfo**](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo())()            Returns an array indicating, for each notification this MBean may send, the name of the Java class of the notification and the notification type. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)            Removes a listener from this MBean. |

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

### addNotificationListener

void **addNotificationListener**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener,  
 [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) filter,  
 [Object](http://docs.google.com/java/lang/Object.html) handback)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Adds a listener to this MBean.

**Parameters:**listener - The listener object which will handle the notifications emitted by the broadcaster.filter - The filter object. If filter is null, no filtering will be performed before handling notifications.handback - An opaque object to be sent back to the listener when a notification is emitted. This object cannot be used by the Notification broadcaster object. It should be resent unchanged with the notification to the listener. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - Listener parameter is null.**See Also:**[removeNotificationListener(javax.management.NotificationListener)](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener))

### removeNotificationListener

void **removeNotificationListener**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)  
 throws [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html)

Removes a listener from this MBean. If the listener has been registered with different handback objects or notification filters, all entries corresponding to the listener will be removed.

**Parameters:**listener - A listener that was previously added to this MBean. **Throws:** [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered with the MBean.**See Also:**[addNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [NotificationEmitter.removeNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/NotificationEmitter.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### getNotificationInfo

[MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)[] **getNotificationInfo**()

Returns an array indicating, for each notification this MBean may send, the name of the Java class of the notification and the notification type.

It is not illegal for the MBean to send notifications not described in this array. However, some clients of the MBean server may depend on the array being complete for their correct functioning.

**Returns:**the array of possible notifications.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NotificationBroadcaster.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/management/Notification.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/NotificationBroadcaster.html)    [**NO FRAMES**](http://docs.google.com/NotificationBroadcaster.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).