| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NotificationBroadcasterSupport.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/NotificationBroadcaster.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/NotificationEmitter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/NotificationBroadcasterSupport.html)    [**NO FRAMES**](http://docs.google.com/NotificationBroadcasterSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

## **javax.management**

Class NotificationBroadcasterSupport

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.management.NotificationBroadcasterSupport**

**All Implemented Interfaces:** [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html), [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) **Direct Known Subclasses:** [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html), [Monitor](http://docs.google.com/javax/management/monitor/Monitor.html), [RelationService](http://docs.google.com/javax/management/relation/RelationService.html), [Timer](http://docs.google.com/javax/management/timer/Timer.html)

public class **NotificationBroadcasterSupport**extends [Object](http://docs.google.com/java/lang/Object.html)implements [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html)

Provides an implementation of [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) interface. This can be used as the super class of an MBean that sends notifications.

By default, the notification dispatch model is synchronous. That is, when a thread calls sendNotification, the NotificationListener.handleNotification method of each listener is called within that thread. You can override this default by overriding handleNotification in a subclass, or by passing an Executor to the constructor.

If the method call of a filter or listener throws an [Exception](http://docs.google.com/java/lang/Exception.html), then that exception does not prevent other listeners from being invoked. However, if the method call of a filter or of Executor.execute or of handleNotification (when no Excecutor is specified) throws an [Error](http://docs.google.com/java/lang/Error.html), then that Error is propagated to the caller of [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

Remote listeners added using the JMX Remote API (see JMXConnector) are not usually called synchronously. That is, when sendNotification returns, it is not guaranteed that any remote listeners have yet received the notification.

**Since:** 1.5

| **Constructor Summary** | |
| --- | --- |
| [**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport())()            Constructs a NotificationBroadcasterSupport where each listener is invoked by the thread sending the notification. |
| [**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(java.util.concurrent.Executor))([Executor](http://docs.google.com/java/util/concurrent/Executor.html) executor)            Constructs a NotificationBroadcasterSupport where each listener is invoked using the given [Executor](http://docs.google.com/java/util/concurrent/Executor.html). |
| [**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(java.util.concurrent.Executor,%20javax.management.MBeanNotificationInfo...))([Executor](http://docs.google.com/java/util/concurrent/Executor.html) executor, [MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)... info)            Constructs a NotificationBroadcasterSupport with information about the notifications that may be sent, and where each listener is invoked using the given [Executor](http://docs.google.com/java/util/concurrent/Executor.html). |
| [**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(javax.management.MBeanNotificationInfo...))([MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)... info)            Constructs a NotificationBroadcasterSupport with information about the notifications that may be sent. |

| **Method Summary** | |
| --- | --- |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.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. |
| [MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)[] | [**getNotificationInfo**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.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. |
| protected  void | [**handleNotification**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#handleNotification(javax.management.NotificationListener,%20javax.management.Notification,%20java.lang.Object))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener, [Notification](http://docs.google.com/javax/management/Notification.html) notif, [Object](http://docs.google.com/java/lang/Object.html) handback)            This method is called by [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)) for each listener in order to send the notification to that listener. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#removeNotificationListener(javax.management.NotificationListener))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)            Removes a listener from this MBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#removeNotificationListener(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)            Removes a listener from this MBean. |
| void | [**sendNotification**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification))([Notification](http://docs.google.com/javax/management/Notification.html) notification)            Sends a notification. |

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

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

### NotificationBroadcasterSupport

public **NotificationBroadcasterSupport**()

Constructs a NotificationBroadcasterSupport where each listener is invoked by the thread sending the notification. This constructor is equivalent to [NotificationBroadcasterSupport(null, null)](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(java.util.concurrent.Executor,%20javax.management.MBeanNotificationInfo...)).

### NotificationBroadcasterSupport

public **NotificationBroadcasterSupport**([Executor](http://docs.google.com/java/util/concurrent/Executor.html) executor)

Constructs a NotificationBroadcasterSupport where each listener is invoked using the given [Executor](http://docs.google.com/java/util/concurrent/Executor.html). When [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)) is called, a listener is selected if it was added with a null [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html), or if [isNotificationEnabled](http://docs.google.com/javax/management/NotificationFilter.html#isNotificationEnabled(javax.management.Notification)) returns true for the notification being sent. The call to NotificationFilter.isNotificationEnabled takes place in the thread that called sendNotification. Then, for each selected listener, [executor.execute](http://docs.google.com/java/util/concurrent/Executor.html#execute(java.lang.Runnable)) is called with a command that calls the handleNotification method. This constructor is equivalent to [NotificationBroadcasterSupport(executor, null)](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(java.util.concurrent.Executor,%20javax.management.MBeanNotificationInfo...)).

**Parameters:**executor - an executor used by the method sendNotification to send each notification. If it is null, the thread calling sendNotification will invoke the handleNotification method itself.**Since:** 1.6

### NotificationBroadcasterSupport

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

Constructs a NotificationBroadcasterSupport with information about the notifications that may be sent. Each listener is invoked by the thread sending the notification. This constructor is equivalent to [NotificationBroadcasterSupport(null, info)](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#NotificationBroadcasterSupport(java.util.concurrent.Executor,%20javax.management.MBeanNotificationInfo...)).

If the info array is not empty, then it is cloned by the constructor as if by info.clone(), and each call to [getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#getNotificationInfo()) returns a new clone.

**Parameters:**info - an array indicating, for each notification this MBean may send, the name of the Java class of the notification and the notification type. Can be null, which is equivalent to an empty array.**Since:** 1.6

### NotificationBroadcasterSupport

public **NotificationBroadcasterSupport**([Executor](http://docs.google.com/java/util/concurrent/Executor.html) executor,  
 [MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)... info)

Constructs a NotificationBroadcasterSupport with information about the notifications that may be sent, and where each listener is invoked using the given [Executor](http://docs.google.com/java/util/concurrent/Executor.html).

When [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)) is called, a listener is selected if it was added with a null [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html), or if [isNotificationEnabled](http://docs.google.com/javax/management/NotificationFilter.html#isNotificationEnabled(javax.management.Notification)) returns true for the notification being sent. The call to NotificationFilter.isNotificationEnabled takes place in the thread that called sendNotification. Then, for each selected listener, [executor.execute](http://docs.google.com/java/util/concurrent/Executor.html#execute(java.lang.Runnable)) is called with a command that calls the handleNotification method.

If the info array is not empty, then it is cloned by the constructor as if by info.clone(), and each call to [getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#getNotificationInfo()) returns a new clone.

**Parameters:**executor - an executor used by the method sendNotification to send each notification. If it is null, the thread calling sendNotification will invoke the handleNotification method itself.info - an array indicating, for each notification this MBean may send, the name of the Java class of the notification and the notification type. Can be null, which is equivalent to an empty array.**Since:** 1.6

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

### addNotificationListener

public 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)

Adds a listener.

**Specified by:**[addNotificationListener](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html) **Parameters:**listener - The listener to receive notifications.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) - thrown if the listener is null.**See Also:**[removeNotificationListener(javax.management.NotificationListener)](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#removeNotificationListener(javax.management.NotificationListener))

### removeNotificationListener

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

**Description copied from interface:** [**NotificationBroadcaster**](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener)) 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.

**Specified by:**[removeNotificationListener](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener)) in interface [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html) **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:**[NotificationBroadcaster.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))

### removeNotificationListener

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

**Description copied from interface:** [**NotificationEmitter**](http://docs.google.com/javax/management/NotificationEmitter.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

Removes a listener from this MBean. The MBean must have a listener that exactly matches the given listener, filter, and handback parameters. If there is more than one such listener, only one is removed.

The filter and handback parameters may be null if and only if they are null in a listener to be removed.

**Specified by:**[removeNotificationListener](http://docs.google.com/javax/management/NotificationEmitter.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) **Parameters:**listener - A listener that was previously added to this MBean.filter - The filter that was specified when the listener was added.handback - The handback that was specified when the listener was added. **Throws:** [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered with the MBean, or it is not registered with the given filter and handback.

### getNotificationInfo

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

**Description copied from interface:** [**NotificationBroadcaster**](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.

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.

**Specified by:**[getNotificationInfo](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo()) in interface [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html) **Returns:**the array of possible notifications.

### sendNotification

public void **sendNotification**([Notification](http://docs.google.com/javax/management/Notification.html) notification)

Sends a notification. If an Executor was specified in the constructor, it will be given one task per selected listener to deliver the notification to that listener.

**Parameters:**notification - The notification to send.

### handleNotification

protected void **handleNotification**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener,  
 [Notification](http://docs.google.com/javax/management/Notification.html) notif,  
 [Object](http://docs.google.com/java/lang/Object.html) handback)

This method is called by [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)) for each listener in order to send the notification to that listener. It can be overridden in subclasses to change the behavior of notification delivery, for instance to deliver the notification in a separate thread.

The default implementation of this method is equivalent to

listener.handleNotification(notif, handback);

**Parameters:**listener - the listener to which the notification is being delivered.notif - the notification being delivered to the listener.handback - the handback object that was supplied when the listener was added.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/NotificationBroadcasterSupport.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/NotificationBroadcaster.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/NotificationEmitter.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/NotificationBroadcasterSupport.html)    [**NO FRAMES**](http://docs.google.com/NotificationBroadcasterSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

[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).