| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StandardEmitterMBean.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/ServiceNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/StandardMBean.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/StandardEmitterMBean.html)    [**NO FRAMES**](http://docs.google.com/StandardEmitterMBean.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#35nkun2) |

## **javax.management**

Class StandardEmitterMBean

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

**All Implemented Interfaces:** [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html), [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html), [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html), [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html)

public class **StandardEmitterMBean**extends [StandardMBean](http://docs.google.com/javax/management/StandardMBean.html)implements [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html)

An MBean whose management interface is determined by reflection on a Java interface, and that emits notifications.

The following example shows how to use the public constructor [StandardEmitterMBean(implementation, mbeanInterface, emitter)](http://docs.google.com/javax/management/StandardEmitterMBean.html#StandardEmitterMBean(T,%20java.lang.Class,%20javax.management.NotificationEmitter)) to create an MBean emitting notifications with any implementation class name *Impl*, with a management interface defined (as for current Standard MBeans) by any interface *Intf*, and with any implementation of the interface [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html). The example uses the class [NotificationBroadcasterSupport](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html) as an implementation of the interface [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html).

MBeanServer mbs;  
 ...  
 final String[] types = new String[] {"sun.disc.space","sun.disc.alarm"};  
 final MBeanNotificationInfo info = new MBeanNotificationInfo(  
 types,  
 Notification.class.getName(),  
 "Notification about disc info.");  
 final NotificationEmitter emitter =   
 new NotificationBroadcasterSupport(info);  
  
 final Intf impl = new Impl(...);  
 final Object mbean = new StandardEmitterMBean(  
 impl, Intf.class, emitter);  
 mbs.registerMBean(mbean, objectName);

**Since:** 1.6 **See Also:**[StandardMBean](http://docs.google.com/javax/management/StandardMBean.html)

| **Constructor Summary** | |
| --- | --- |
| protected | [**StandardEmitterMBean**](http://docs.google.com/javax/management/StandardEmitterMBean.html#StandardEmitterMBean(java.lang.Class,%20boolean,%20javax.management.NotificationEmitter))([Class](http://docs.google.com/java/lang/Class.html)<?> mbeanInterface, boolean isMXBean, [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)            Make an MBean whose management interface is specified by mbeanInterface, and where notifications are handled by the given NotificationEmitter. |
| protected | [**StandardEmitterMBean**](http://docs.google.com/javax/management/StandardEmitterMBean.html#StandardEmitterMBean(java.lang.Class,%20javax.management.NotificationEmitter))([Class](http://docs.google.com/java/lang/Class.html)<?> mbeanInterface, [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)            Make an MBean whose management interface is specified by mbeanInterface, and where notifications are handled by the given NotificationEmitter. |
|  | [**StandardEmitterMBean**](http://docs.google.com/javax/management/StandardEmitterMBean.html#StandardEmitterMBean(T,%20java.lang.Class,%20boolean,%20javax.management.NotificationEmitter))(T implementation, [Class](http://docs.google.com/java/lang/Class.html)<T> mbeanInterface, boolean isMXBean, [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)            Make an MBean whose management interface is specified by mbeanInterface, with the given implementation and where notifications are handled by the given NotificationEmitter. |
|  | [**StandardEmitterMBean**](http://docs.google.com/javax/management/StandardEmitterMBean.html#StandardEmitterMBean(T,%20java.lang.Class,%20javax.management.NotificationEmitter))(T implementation, [Class](http://docs.google.com/java/lang/Class.html)<T> mbeanInterface, [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)            Make an MBean whose management interface is specified by mbeanInterface, with the given implementation and where notifications are handled by the given NotificationEmitter. |

| **Method Summary** | |
| --- | --- |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/StandardEmitterMBean.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/StandardEmitterMBean.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/StandardEmitterMBean.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/StandardEmitterMBean.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/StandardEmitterMBean.html#sendNotification(javax.management.Notification))([Notification](http://docs.google.com/javax/management/Notification.html) n)            Sends a notification. |

| **Methods inherited from class javax.management.**[**StandardMBean**](http://docs.google.com/javax/management/StandardMBean.html) |
| --- |
| [cacheMBeanInfo](http://docs.google.com/javax/management/StandardMBean.html#cacheMBeanInfo(javax.management.MBeanInfo)), [getAttribute](http://docs.google.com/javax/management/StandardMBean.html#getAttribute(java.lang.String)), [getAttributes](http://docs.google.com/javax/management/StandardMBean.html#getAttributes(java.lang.String%5B%5D)), [getCachedMBeanInfo](http://docs.google.com/javax/management/StandardMBean.html#getCachedMBeanInfo()), [getClassName](http://docs.google.com/javax/management/StandardMBean.html#getClassName(javax.management.MBeanInfo)), [getConstructors](http://docs.google.com/javax/management/StandardMBean.html#getConstructors(javax.management.MBeanConstructorInfo%5B%5D,%20java.lang.Object)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanAttributeInfo)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanConstructorInfo)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanConstructorInfo,%20javax.management.MBeanParameterInfo,%20int)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanFeatureInfo)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanInfo)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanOperationInfo)), [getDescription](http://docs.google.com/javax/management/StandardMBean.html#getDescription(javax.management.MBeanOperationInfo,%20javax.management.MBeanParameterInfo,%20int)), [getImpact](http://docs.google.com/javax/management/StandardMBean.html#getImpact(javax.management.MBeanOperationInfo)), [getImplementation](http://docs.google.com/javax/management/StandardMBean.html#getImplementation()), [getImplementationClass](http://docs.google.com/javax/management/StandardMBean.html#getImplementationClass()), [getMBeanInfo](http://docs.google.com/javax/management/StandardMBean.html#getMBeanInfo()), [getMBeanInterface](http://docs.google.com/javax/management/StandardMBean.html#getMBeanInterface()), [getParameterName](http://docs.google.com/javax/management/StandardMBean.html#getParameterName(javax.management.MBeanConstructorInfo,%20javax.management.MBeanParameterInfo,%20int)), [getParameterName](http://docs.google.com/javax/management/StandardMBean.html#getParameterName(javax.management.MBeanOperationInfo,%20javax.management.MBeanParameterInfo,%20int)), [invoke](http://docs.google.com/javax/management/StandardMBean.html#invoke(java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)), [postDeregister](http://docs.google.com/javax/management/StandardMBean.html#postDeregister()), [postRegister](http://docs.google.com/javax/management/StandardMBean.html#postRegister(java.lang.Boolean)), [preDeregister](http://docs.google.com/javax/management/StandardMBean.html#preDeregister()), [preRegister](http://docs.google.com/javax/management/StandardMBean.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)), [setAttribute](http://docs.google.com/javax/management/StandardMBean.html#setAttribute(javax.management.Attribute)), [setAttributes](http://docs.google.com/javax/management/StandardMBean.html#setAttributes(javax.management.AttributeList)), [setImplementation](http://docs.google.com/javax/management/StandardMBean.html#setImplementation(java.lang.Object)) |

| **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** |
| --- |

### StandardEmitterMBean

public **StandardEmitterMBean**(T implementation,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> mbeanInterface,  
 [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)

Make an MBean whose management interface is specified by mbeanInterface, with the given implementation and where notifications are handled by the given NotificationEmitter. The resultant MBean implements the NotificationEmitter interface by forwarding its methods to emitter. It is legal and useful for implementation and emitter to be the same object.

If emitter is an instance of NotificationBroadcasterSupport then the MBean's [sendNotification](http://docs.google.com/javax/management/StandardEmitterMBean.html#sendNotification(javax.management.Notification)) method will call emitter.[sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

The array returned by [getNotificationInfo()](http://docs.google.com/javax/management/StandardEmitterMBean.html#getNotificationInfo()) on the new MBean is a copy of the array returned by emitter.[getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo()) at the time of construction. If the array returned by emitter.getNotificationInfo() later changes, that will have no effect on this object's getNotificationInfo().

**Parameters:**implementation - the implementation of the MBean interface.mbeanInterface - a Standard MBean interface.emitter - the object that will handle notifications. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the mbeanInterface does not follow JMX design patterns for Management Interfaces, or if the given implementation does not implement the specified interface, or if emitter is null.

### StandardEmitterMBean

public **StandardEmitterMBean**(T implementation,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> mbeanInterface,  
 boolean isMXBean,  
 [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)

Make an MBean whose management interface is specified by mbeanInterface, with the given implementation and where notifications are handled by the given NotificationEmitter. This constructor can be used to make either Standard MBeans or MXBeans. The resultant MBean implements the NotificationEmitter interface by forwarding its methods to emitter. It is legal and useful for implementation and emitter to be the same object.

If emitter is an instance of NotificationBroadcasterSupport then the MBean's [sendNotification](http://docs.google.com/javax/management/StandardEmitterMBean.html#sendNotification(javax.management.Notification)) method will call emitter.[sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

The array returned by [getNotificationInfo()](http://docs.google.com/javax/management/StandardEmitterMBean.html#getNotificationInfo()) on the new MBean is a copy of the array returned by emitter.[getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo()) at the time of construction. If the array returned by emitter.getNotificationInfo() later changes, that will have no effect on this object's getNotificationInfo().

**Parameters:**implementation - the implementation of the MBean interface.mbeanInterface - a Standard MBean interface.isMXBean - If true, the mbeanInterface parameter names an MXBean interface and the resultant MBean is an MXBean.emitter - the object that will handle notifications. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the mbeanInterface does not follow JMX design patterns for Management Interfaces, or if the given implementation does not implement the specified interface, or if emitter is null.

### StandardEmitterMBean

protected **StandardEmitterMBean**([Class](http://docs.google.com/java/lang/Class.html)<?> mbeanInterface,  
 [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)

Make an MBean whose management interface is specified by mbeanInterface, and where notifications are handled by the given NotificationEmitter. The resultant MBean implements the NotificationEmitter interface by forwarding its methods to emitter.

If emitter is an instance of NotificationBroadcasterSupport then the MBean's [sendNotification](http://docs.google.com/javax/management/StandardEmitterMBean.html#sendNotification(javax.management.Notification)) method will call emitter.[sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

The array returned by [getNotificationInfo()](http://docs.google.com/javax/management/StandardEmitterMBean.html#getNotificationInfo()) on the new MBean is a copy of the array returned by emitter.[getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo()) at the time of construction. If the array returned by emitter.getNotificationInfo() later changes, that will have no effect on this object's getNotificationInfo().

This constructor must be called from a subclass that implements the given mbeanInterface.

**Parameters:**mbeanInterface - a StandardMBean interface.emitter - the object that will handle notifications. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the mbeanInterface does not follow JMX design patterns for Management Interfaces, or if this does not implement the specified interface, or if emitter is null.

### StandardEmitterMBean

protected **StandardEmitterMBean**([Class](http://docs.google.com/java/lang/Class.html)<?> mbeanInterface,  
 boolean isMXBean,  
 [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) emitter)

Make an MBean whose management interface is specified by mbeanInterface, and where notifications are handled by the given NotificationEmitter. This constructor can be used to make either Standard MBeans or MXBeans. The resultant MBean implements the NotificationEmitter interface by forwarding its methods to emitter.

If emitter is an instance of NotificationBroadcasterSupport then the MBean's [sendNotification](http://docs.google.com/javax/management/StandardEmitterMBean.html#sendNotification(javax.management.Notification)) method will call emitter.[sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

The array returned by [getNotificationInfo()](http://docs.google.com/javax/management/StandardEmitterMBean.html#getNotificationInfo()) on the new MBean is a copy of the array returned by emitter.[getNotificationInfo()](http://docs.google.com/javax/management/NotificationBroadcaster.html#getNotificationInfo()) at the time of construction. If the array returned by emitter.getNotificationInfo() later changes, that will have no effect on this object's getNotificationInfo().

This constructor must be called from a subclass that implements the given mbeanInterface.

**Parameters:**mbeanInterface - a StandardMBean interface.isMXBean - If true, the mbeanInterface parameter names an MXBean interface and the resultant MBean is an MXBean.emitter - the object that will handle notifications. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the mbeanInterface does not follow JMX design patterns for Management Interfaces, or if this does not implement the specified interface, or if emitter is null.

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

### 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.

### 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)

**Description copied from interface:** [**NotificationBroadcaster**](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) Adds a listener to this MBean.

**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 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.**See Also:**[NotificationBroadcaster.removeNotificationListener(javax.management.NotificationListener)](http://docs.google.com/javax/management/NotificationBroadcaster.html#removeNotificationListener(javax.management.NotificationListener))

### 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) n)

Sends a notification.

If the emitter parameter to the constructor was an instance of NotificationBroadcasterSupport then this method will call emitter.[sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)).

**Parameters:**n - the notification to send. **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if the emitter parameter to the constructor was not a NotificationBroadcasterSupport.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/StandardEmitterMBean.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/ServiceNotFoundException.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/StandardMBean.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/StandardEmitterMBean.html)    [**NO FRAMES**](http://docs.google.com/StandardEmitterMBean.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#35nkun2) |

[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).