| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RequiredModelMBean.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/modelmbean/ModelMBeanOperationInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/modelmbean/XMLParseException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/modelmbean/RequiredModelMBean.html)    [**NO FRAMES**](http://docs.google.com/RequiredModelMBean.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **javax.management.modelmbean**

Class RequiredModelMBean

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

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

public class **RequiredModelMBean**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ModelMBean](http://docs.google.com/javax/management/modelmbean/ModelMBean.html), [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html), [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html)

This class is the implementation of a ModelMBean. An appropriate implementation of a ModelMBean must be shipped with every JMX Agent and the class must be named RequiredModelMBean.

Java resources wishing to be manageable instantiate the RequiredModelMBean using the MBeanServer's createMBean method. The resource then sets the MBeanInfo and Descriptors for the RequiredModelMBean instance. The attributes and operations exposed via the ModelMBeanInfo for the ModelMBean are accessible from MBeans, connectors/adaptors like other MBeans. Through the Descriptors, values and methods in the managed application can be defined and mapped to attributes and operations of the ModelMBean. This mapping can be defined in an XML formatted file or dynamically and programmatically at runtime.

Every RequiredModelMBean which is instantiated in the MBeanServer becomes manageable:

its attributes and operations become remotely accessible through the connectors/adaptors connected to that MBeanServer.

A Java object cannot be registered in the MBeanServer unless it is a JMX compliant MBean. By instantiating a RequiredModelMBean, resources are guaranteed that the MBean is valid. MBeanException and RuntimeOperationsException must be thrown on every public method. This allows for wrapping exceptions from distributed communications (RMI, EJB, etc.)

**Since:** 1.5

| **Constructor Summary** | |
| --- | --- |
| [**RequiredModelMBean**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#RequiredModelMBean())()            Constructs an RequiredModelMBean with an empty ModelMBeanInfo. |
| [**RequiredModelMBean**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#RequiredModelMBean(javax.management.modelmbean.ModelMBeanInfo))([ModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanInfo.html) mbi)            Constructs a RequiredModelMBean object using ModelMBeanInfo passed in. |

| **Method Summary** | |
| --- | --- |
| void | [**addAttributeChangeNotificationListener**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#addAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String,%20java.lang.Object))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) inlistener, [String](http://docs.google.com/java/lang/String.html) inAttributeName, [Object](http://docs.google.com/java/lang/Object.html) inhandback)            Registers an object which implements the NotificationListener interface as a listener. |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.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)            Registers an object which implements the NotificationListener interface as a listener. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getAttribute**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) attrName)            Returns the value of a specific attribute defined for this ModelMBean. |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**getAttributes**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getAttributes(java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] attrNames)            Returns the values of several attributes in the ModelMBean. |
| protected  [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) | [**getClassLoaderRepository**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getClassLoaderRepository())()            Return the Class Loader Repository used to perform class loading. |
| [MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html) | [**getMBeanInfo**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getMBeanInfo())()            Returns the attributes, operations, constructors and notifications that this RequiredModelMBean exposes for management. |
| [MBeanNotificationInfo](http://docs.google.com/javax/management/MBeanNotificationInfo.html)[] | [**getNotificationInfo**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getNotificationInfo())()            Returns the array of Notifications always generated by the RequiredModelMBean. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#invoke(java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html) opName, [Object](http://docs.google.com/java/lang/Object.html)[] opArgs, [String](http://docs.google.com/java/lang/String.html)[] sig)            Invokes a method on or through a RequiredModelMBean and returns the result of the method execution. |
| void | [**load**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#load())()            Instantiates this MBean instance with the data found for the MBean in the persistent store. |
| void | [**postDeregister**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#postDeregister())()            Allows the MBean to perform any operations needed after having been unregistered in the MBean server. |
| void | [**postRegister**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#postRegister(java.lang.Boolean))([Boolean](http://docs.google.com/java/lang/Boolean.html) registrationDone)            Allows the MBean to perform any operations needed after having been registered in the MBean server or after the registration has failed. |
| void | [**preDeregister**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#preDeregister())()            Allows the MBean to perform any operations it needs before being unregistered by the MBean server. |
| [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**preRegister**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName))([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) server, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Allows the MBean to perform any operations it needs before being registered in the MBean server. |
| void | [**removeAttributeChangeNotificationListener**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#removeAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) inlistener, [String](http://docs.google.com/java/lang/String.html) inAttributeName)            Removes a listener for attributeChangeNotifications from the RequiredModelMBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#removeNotificationListener(javax.management.NotificationListener))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)            Removes a listener for Notifications from the RequiredModelMBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.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 | [**sendAttributeChangeNotification**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#sendAttributeChangeNotification(javax.management.Attribute,%20javax.management.Attribute))([Attribute](http://docs.google.com/javax/management/Attribute.html) inOldVal, [Attribute](http://docs.google.com/javax/management/Attribute.html) inNewVal)            Sends an attributeChangeNotification which contains the old value and new value for the attribute to the registered AttributeChangeNotification listeners on the ModelMBean. |
| void | [**sendAttributeChangeNotification**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#sendAttributeChangeNotification(javax.management.AttributeChangeNotification))([AttributeChangeNotification](http://docs.google.com/javax/management/AttributeChangeNotification.html) ntfyObj)            Sends an attributeChangeNotification which is passed in to the registered attributeChangeNotification listeners on the ModelMBean. |
| void | [**sendNotification**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#sendNotification(javax.management.Notification))([Notification](http://docs.google.com/javax/management/Notification.html) ntfyObj)            Sends a Notification which is passed in to the registered Notification listeners on the ModelMBean as a jmx.modelmbean.generic notification. |
| void | [**sendNotification**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#sendNotification(java.lang.String))([String](http://docs.google.com/java/lang/String.html) ntfyText)            Sends a Notification which contains the text string that is passed in to the registered Notification listeners on the ModelMBean. |
| void | [**setAttribute**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setAttribute(javax.management.Attribute))([Attribute](http://docs.google.com/javax/management/Attribute.html) attribute)            Sets the value of a specific attribute of a named ModelMBean. |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**setAttributes**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setAttributes(javax.management.AttributeList))([AttributeList](http://docs.google.com/javax/management/AttributeList.html) attributes)            Sets the values of an array of attributes of this ModelMBean. |
| void | [**setManagedResource**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setManagedResource(java.lang.Object,%20java.lang.String))([Object](http://docs.google.com/java/lang/Object.html) mr, [String](http://docs.google.com/java/lang/String.html) mr\_type)            Sets the instance handle of the object against which to execute all methods in this ModelMBean management interface (MBeanInfo and Descriptors). |
| void | [**setModelMBeanInfo**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo))([ModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanInfo.html) mbi)            Initializes a ModelMBean object using ModelMBeanInfo passed in. |
| void | [**store**](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#store())()            Captures the current state of this MBean instance and writes it out to the persistent store. |

| **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** |
| --- |

### RequiredModelMBean

public **RequiredModelMBean**()  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Constructs an RequiredModelMBean with an empty ModelMBeanInfo.

The RequiredModelMBean's MBeanInfo and Descriptors can be customized using the [setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)) method. After the RequiredModelMBean's MBeanInfo and Descriptors are customized, the RequiredModelMBean can be registered with the MBeanServer.

**Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) during the construction of the object.

### RequiredModelMBean

public **RequiredModelMBean**([ModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanInfo.html) mbi)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Constructs a RequiredModelMBean object using ModelMBeanInfo passed in. As long as the RequiredModelMBean is not registered with the MBeanServer yet, the RequiredModelMBean's MBeanInfo and Descriptors can be customized using the [setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)) method. After the RequiredModelMBean's MBeanInfo and Descriptors are customized, the RequiredModelMBean can be registered with the MBeanServer.

**Parameters:**mbi - The ModelMBeanInfo object to be used by the RequiredModelMBean. The given ModelMBeanInfo is cloned and modified as specified by [setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)) **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an {link java.lang.IllegalArgumentException}: The MBeanInfo passed in parameter is null.

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

### setModelMBeanInfo

public void **setModelMBeanInfo**([ModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanInfo.html) mbi)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Initializes a ModelMBean object using ModelMBeanInfo passed in. This method makes it possible to set a customized ModelMBeanInfo on the ModelMBean as long as it is not registered with the MBeanServer.

Once the ModelMBean's ModelMBeanInfo (with Descriptors) are customized and set on the ModelMBean, the ModelMBean be registered with the MBeanServer.

If the ModelMBean is currently registered, this method throws a [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) wrapping an [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html)

If the given inModelMBeanInfo does not contain any [ModelMBeanNotificationInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationInfo.html) for the GENERIC or ATTRIBUTE\_CHANGE notifications, then the RequiredModelMBean will supply its own default [ModelMBeanNotificationInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationInfo.html)s for those missing notifications.

**Specified by:**[setModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBean.html#setModelMBeanInfo(javax.management.modelmbean.ModelMBeanInfo)) in interface [ModelMBean](http://docs.google.com/javax/management/modelmbean/ModelMBean.html) **Parameters:**mbi - The ModelMBeanInfo object to be used by the ModelMBean. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) -

* Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) if the MBeanInfo passed in parameter is null.
* Wraps an [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) if the ModelMBean is currently registered in the MBeanServer.

### setManagedResource

public void **setManagedResource**([Object](http://docs.google.com/java/lang/Object.html) mr,  
 [String](http://docs.google.com/java/lang/String.html) mr\_type)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [InvalidTargetObjectTypeException](http://docs.google.com/javax/management/modelmbean/InvalidTargetObjectTypeException.html)

Sets the instance handle of the object against which to execute all methods in this ModelMBean management interface (MBeanInfo and Descriptors).

**Specified by:**[setManagedResource](http://docs.google.com/javax/management/modelmbean/ModelMBean.html#setManagedResource(java.lang.Object,%20java.lang.String)) in interface [ModelMBean](http://docs.google.com/javax/management/modelmbean/ModelMBean.html) **Parameters:**mr - Object that is the managed resourcemr\_type - The type of reference for the managed resource.

Can be: "ObjectReference", "Handle", "IOR", "EJBHandle", or "RMIReference".

In this implementation only "ObjectReference" is supported. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The initializer of the object has thrown an exception. [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The managed resource object could not be found [InvalidTargetObjectTypeException](http://docs.google.com/javax/management/modelmbean/InvalidTargetObjectTypeException.html) - The managed resource type should be "ObjectReference". [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) when setting the resource.

### load

public void **load**()  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Instantiates this MBean instance with the data found for the MBean in the persistent store. The data loaded could include attribute and operation values.

This method should be called during construction or initialization of this instance, and before the MBean is registered with the MBeanServer.

If the implementation of this class does not support persistence, an [MBeanException](http://docs.google.com/javax/management/MBeanException.html) wrapping a [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) is thrown.

**Specified by:**[load](http://docs.google.com/javax/management/PersistentMBean.html#load()) in interface [PersistentMBean](http://docs.google.com/javax/management/PersistentMBean.html) **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps another exception, or persistence is not supported [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps exceptions from the persistence mechanism [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - Could not find or load this MBean from persistent storage

### store

public void **store**()  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Captures the current state of this MBean instance and writes it out to the persistent store. The state stored could include attribute and operation values.

If the implementation of this class does not support persistence, an [MBeanException](http://docs.google.com/javax/management/MBeanException.html) wrapping a [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) is thrown.

Persistence policy from the MBean and attribute descriptor is used to guide execution of this method. The MBean should be stored if 'persistPolicy' field is:

!= "never"  
 = "always"  
 = "onTimer" and now > 'lastPersistTime' + 'persistPeriod'  
 = "NoMoreOftenThan" and now > 'lastPersistTime' + 'persistPeriod'  
 = "onUnregister"

Do not store the MBean if 'persistPolicy' field is:

= "never"  
 = "onUpdate"  
 = "onTimer" && now < 'lastPersistTime' + 'persistPeriod'

**Specified by:**[store](http://docs.google.com/javax/management/PersistentMBean.html#store()) in interface [PersistentMBean](http://docs.google.com/javax/management/PersistentMBean.html) **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps another exception, or persistence is not supported [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps exceptions from the persistence mechanism [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - Could not find/access the persistent store

### getMBeanInfo

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

Returns the attributes, operations, constructors and notifications that this RequiredModelMBean exposes for management.

**Specified by:**[getMBeanInfo](http://docs.google.com/javax/management/DynamicMBean.html#getMBeanInfo()) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Returns:**An instance of ModelMBeanInfo allowing retrieval all attributes, operations, and Notifications of this MBean.

### invoke

public [Object](http://docs.google.com/java/lang/Object.html) **invoke**([String](http://docs.google.com/java/lang/String.html) opName,  
 [Object](http://docs.google.com/java/lang/Object.html)[] opArgs,  
 [String](http://docs.google.com/java/lang/String.html)[] sig)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

Invokes a method on or through a RequiredModelMBean and returns the result of the method execution.

If the given method to be invoked, together with the provided signature, matches one of RequiredModelMbean accessible methods, this one will be call. Otherwise the call to the given method will be tried on the managed resource.

The last value returned by an operation may be cached in the operation's descriptor which is in the ModelMBeanOperationInfo's descriptor. The valid value will be in the 'value' field if there is one. If the 'currencyTimeLimit' field in the descriptor is:

* **<0** Then the value is not cached and is never valid. The operation method is invoked. The 'value' and 'lastUpdatedTimeStamp' fields are cleared.
* **=0** Then the value is always cached and always valid. The 'value' field is returned. If there is no 'value' field then the operation method is invoked for the attribute. The 'lastUpdatedTimeStamp' field and `value' fields are set to the operation's return value and the current time stamp.
* **>0** Represents the number of seconds that the 'value' field is valid. The 'value' field is no longer valid when 'lastUpdatedTimeStamp' + 'currencyTimeLimit' > Now.
  + When 'value' is valid, 'value' is returned.
  + When 'value' is no longer valid then the operation method is invoked. The 'lastUpdatedTimeStamp' field and `value' fields are updated.

**Note:** because of inconsistencies in previous versions of this specification, it is recommended not to use negative or zero values for currencyTimeLimit. To indicate that a cached value is never valid, omit the currencyTimeLimit field. To indicate that it is always valid, use a very large number for this field.

**Specified by:**[invoke](http://docs.google.com/javax/management/DynamicMBean.html#invoke(java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Parameters:**opName - The name of the method to be invoked. The name can be the fully qualified method name including the classname, or just the method name if the classname is defined in the 'class' field of the operation descriptor.opArgs - An array containing the parameters to be set when the operation is invokedsig - An array containing the signature of the operation. The class objects will be loaded using the same class loader as the one used for loading the MBean on which the operation was invoked. **Returns:**The object returned by the method, which represents the result of invoking the method on the specified managed resource. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps one of the following Exceptions:

* An Exception thrown by the managed object's invoked method.
* [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html): No ModelMBeanOperationInfo or no descriptor defined for the specified operation or the managed resource is null.
* [InvalidTargetObjectTypeException](http://docs.google.com/javax/management/modelmbean/InvalidTargetObjectTypeException.html): The 'targetType' field value is not 'objectReference'.

[ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps an [Exception](http://docs.google.com/java/lang/Exception.html) thrown while trying to invoke the method. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) Method name is null.

### getAttribute

public [Object](http://docs.google.com/java/lang/Object.html) **getAttribute**([String](http://docs.google.com/java/lang/String.html) attrName)  
 throws [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

Returns the value of a specific attribute defined for this ModelMBean. The last value returned by an attribute may be cached in the attribute's descriptor. The valid value will be in the 'value' field if there is one. If the 'currencyTimeLimit' field in the descriptor is:

* **<0** Then the value is not cached and is never valid. The getter method is invoked for the attribute. The 'value' and 'lastUpdatedTimeStamp' fields are cleared.
* **=0** Then the value is always cached and always valid. The 'value' field is returned. If there is no'value' field then the getter method is invoked for the attribute. The 'lastUpdatedTimeStamp' field and `value' fields are set to the attribute's value and the current time stamp.
* **>0** Represents the number of seconds that the 'value' field is valid. The 'value' field is no longer valid when 'lastUpdatedTimeStamp' + 'currencyTimeLimit' > Now.
  + When 'value' is valid, 'value' is returned.
  + When 'value' is no longer valid then the getter method is invoked for the attribute. The 'lastUpdatedTimeStamp' field and `value' fields are updated.

**Note:** because of inconsistencies in previous versions of this specification, it is recommended not to use negative or zero values for currencyTimeLimit. To indicate that a cached value is never valid, omit the currencyTimeLimit field. To indicate that it is always valid, use a very large number for this field.

If the 'getMethod' field contains the name of a valid operation descriptor, then the method described by the operation descriptor is executed. The response from the method is returned as the value of the attribute. If the operation fails or the returned value is not compatible with the declared type of the attribute, an exception will be thrown.

If no 'getMethod' field is defined then the default value of the attribute is returned. If the returned value is not compatible with the declared type of the attribute, an exception will be thrown.

The declared type of the attribute is the String returned by [MBeanAttributeInfo.getType()](http://docs.google.com/javax/management/MBeanAttributeInfo.html#getType()). A value is compatible with this type if one of the following is true:

* the value is null;
* the declared name is a primitive type name (such as "int") and the value is an instance of the corresponding wrapper type (such as java.lang.Integer);
* the name of the value's class is identical to the declared name;
* the declared name can be loaded by the value's class loader and produces a class to which the value can be assigned.

In this implementation, in every case where the getMethod needs to be called, because the method is invoked through the standard "invoke" method and thus needs operationInfo, an operation must be specified for that getMethod so that the invocation works correctly.

**Specified by:**[getAttribute](http://docs.google.com/javax/management/DynamicMBean.html#getAttribute(java.lang.String)) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Parameters:**attrName - A String specifying the name of the attribute to be retrieved. It must match the name of a ModelMBeanAttributeInfo. **Returns:**The value of the retrieved attribute from the descriptor 'value' field or from the invocation of the operation in the 'getMethod' field of the descriptor. **Throws:** [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html) - The specified attribute is not accessible in the MBean. The following cases may result in an AttributeNotFoundException:

* No ModelMBeanInfo was found for the Model MBean.
* No ModelMBeanAttributeInfo was found for the specified attribute name.
* The ModelMBeanAttributeInfo isReadable method returns 'false'.

[MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps one of the following Exceptions:

* [InvalidAttributeValueException](http://docs.google.com/javax/management/InvalidAttributeValueException.html): A wrong value type was received from the attribute's getter method or no 'getMethod' field defined in the descriptor for the attribute and no default value exists.
* [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html): No ModelMBeanOperationInfo defined for the attribute's getter method or no descriptor associated with the ModelMBeanOperationInfo or the managed resource is null.
* [InvalidTargetObjectTypeException](http://docs.google.com/javax/management/modelmbean/InvalidTargetObjectTypeException.html) The 'targetType' field value is not 'objectReference'.
* An Exception thrown by the managed object's getter.

[ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps an [Exception](http://docs.google.com/java/lang/Exception.html) thrown while trying to invoke the getter. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html): The attribute name in parameter is null.**See Also:**[setAttribute(javax.management.Attribute)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setAttribute(javax.management.Attribute))

### getAttributes

public [AttributeList](http://docs.google.com/javax/management/AttributeList.html) **getAttributes**([String](http://docs.google.com/java/lang/String.html)[] attrNames)

Returns the values of several attributes in the ModelMBean. Executes a getAttribute for each attribute name in the attrNames array passed in.

**Specified by:**[getAttributes](http://docs.google.com/javax/management/DynamicMBean.html#getAttributes(java.lang.String%5B%5D)) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Parameters:**attrNames - A String array of names of the attributes to be retrieved. **Returns:**The array of the retrieved attributes. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html): The object name in parameter is null or attributes in parameter is null.**See Also:**[setAttributes(javax.management.AttributeList)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#setAttributes(javax.management.AttributeList))

### setAttribute

public void **setAttribute**([Attribute](http://docs.google.com/javax/management/Attribute.html) attribute)  
 throws [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html),  
 [InvalidAttributeValueException](http://docs.google.com/javax/management/InvalidAttributeValueException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

Sets the value of a specific attribute of a named ModelMBean. If the 'setMethod' field of the attribute's descriptor contains the name of a valid operation descriptor, then the method described by the operation descriptor is executed. In this implementation, the operation descriptor must be specified correctly and assigned to the modelMBeanInfo so that the 'setMethod' works correctly. The response from the method is set as the value of the attribute in the descriptor.

If currencyTimeLimit is > 0, then the new value for the attribute is cached in the attribute descriptor's 'value' field and the 'lastUpdatedTimeStamp' field is set to the current time stamp.

If the persist field of the attribute's descriptor is not null then Persistence policy from the attribute descriptor is used to guide storing the attribute in a persistent store.

Store the MBean if 'persistPolicy' field is:

* != "never"
* = "always"
* = "onUpdate"
* = "onTimer" and now > 'lastPersistTime' + 'persistPeriod'
* = "NoMoreOftenThan" and now > 'lastPersistTime' + 'persistPeriod'

Do not store the MBean if 'persistPolicy' field is:

* = "never"
* = "onTimer" && now < 'lastPersistTime' + 'persistPeriod'
* = "onUnregister"
* = "NoMoreOftenThan" and now < 'lastPersistTime' + 'persistPeriod'

The ModelMBeanInfo of the Model MBean is stored in a file.

**Specified by:**[setAttribute](http://docs.google.com/javax/management/DynamicMBean.html#setAttribute(javax.management.Attribute)) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Parameters:**attribute - The Attribute instance containing the name of the attribute to be set and the value it is to be set to. **Throws:** [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html) - The specified attribute is not accessible in the MBean.

The following cases may result in an AttributeNotFoundException:

* No ModelMBeanAttributeInfo is found for the specified attribute.
* The ModelMBeanAttributeInfo's isWritable method returns 'false'.

[InvalidAttributeValueException](http://docs.google.com/javax/management/InvalidAttributeValueException.html) - No descriptor is defined for the specified attribute. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps one of the following Exceptions:

* An Exception thrown by the managed object's setter.
* A [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) if a setMethod field is defined in the descriptor for the attribute and the managed resource is null; or if no setMethod field is defined and caching is not enabled for the attribute. Note that if there is no getMethod field either, then caching is automatically enabled.
* [InvalidTargetObjectTypeException](http://docs.google.com/javax/management/modelmbean/InvalidTargetObjectTypeException.html) The 'targetType' field value is not 'objectReference'.
* An Exception thrown by the managed object's getter.

[ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps an [Exception](http://docs.google.com/java/lang/Exception.html) thrown while trying to invoke the setter. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html): The attribute in parameter is null.**See Also:**[getAttribute(java.lang.String)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getAttribute(java.lang.String))

### setAttributes

public [AttributeList](http://docs.google.com/javax/management/AttributeList.html) **setAttributes**([AttributeList](http://docs.google.com/javax/management/AttributeList.html) attributes)

Sets the values of an array of attributes of this ModelMBean. Executes the setAttribute() method for each attribute in the list.

**Specified by:**[setAttributes](http://docs.google.com/javax/management/DynamicMBean.html#setAttributes(javax.management.AttributeList)) in interface [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) **Parameters:**attributes - A list of attributes: The identification of the attributes to be set and the values they are to be set to. **Returns:**The array of attributes that were set, with their new values in Attribute instances. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html): The object name in parameter is null or attributes in parameter is null.**See Also:**[getAttributes(java.lang.String[])](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#getAttributes(java.lang.String%5B%5D))

### 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)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Registers an object which implements the NotificationListener interface as a listener. This object's 'handleNotification()' method will be invoked when any notification is issued through or by the ModelMBean. This does not include attributeChangeNotifications. They must be registered for independently.

**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 handles notifications emitted by the registered MBean.filter - The filter object. If null, no filtering will be performed before handling notifications.handback - The context to be sent to the listener with the notification when a notification is emitted. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - The listener cannot be null.**See Also:**[removeNotificationListener(javax.management.NotificationListener)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.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)

Removes a listener for Notifications from the RequiredModelMBean.

**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 - The listener name which was handling notifications emitted by the registered MBean. This method will remove all information related to this listener. **Throws:** [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean or is null.**See Also:**[addNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/modelmbean/RequiredModelMBean.html#addNotificationListener(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.

### sendNotification

public void **sendNotification**([Notification](http://docs.google.com/javax/management/Notification.html) ntfyObj)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendNotification(javax.management.Notification)) Sends a Notification which is passed in to the registered Notification listeners on the ModelMBean as a jmx.modelmbean.generic notification.

**Specified by:**[sendNotification](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendNotification(javax.management.Notification)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**ntfyObj - The notification which is to be passed to the 'handleNotification' method of the listener object. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException: The Notification object passed in parameter is null.

### sendNotification

public void **sendNotification**([String](http://docs.google.com/java/lang/String.html) ntfyText)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendNotification(java.lang.String)) Sends a Notification which contains the text string that is passed in to the registered Notification listeners on the ModelMBean.

**Specified by:**[sendNotification](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendNotification(java.lang.String)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**ntfyText - The text which is to be passed in the Notification to the 'handleNotification' method of the listener object. the constructed Notification will be: type "jmx.modelmbean.generic" source this ModelMBean instance sequence 1 **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException: The Notification text string passed in parameter is null.

### getNotificationInfo

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

Returns the array of Notifications always generated by the RequiredModelMBean.

RequiredModelMBean may always send also two additional notifications:

* One with descriptor "name=GENERIC,descriptorType=notification,log=T,severity=6,displayName=jmx.modelmbean.generic"
* Second is a standard attribute change notification with descriptor "name=ATTRIBUTE\_CHANGE,descriptorType=notification,log=T,severity=6,displayName=jmx.attribute.change"

Thus these two notifications are always added to those specified by the application.

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

### addAttributeChangeNotificationListener

public void **addAttributeChangeNotificationListener**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) inlistener,  
 [String](http://docs.google.com/java/lang/String.html) inAttributeName,  
 [Object](http://docs.google.com/java/lang/Object.html) inhandback)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html),  
 [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#addAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String,%20java.lang.Object)) Registers an object which implements the NotificationListener interface as a listener. This object's 'handleNotification()' method will be invoked when any attributeChangeNotification is issued through or by the ModelMBean. This does not include other Notifications. They must be registered for independently. An AttributeChangeNotification will be generated for this attributeName.

**Specified by:**[addAttributeChangeNotificationListener](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#addAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String,%20java.lang.Object)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**inlistener - The listener object which will handles notifications emitted by the registered MBean.inAttributeName - The name of the ModelMBean attribute for which to receive change notifications. If null, then all attribute changes will cause an attributeChangeNotification to be issued.inhandback - The context to be sent to the listener with the notification when a notification is emitted. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException The attribute name passed in parameter does not exist. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - The listener cannot be null.**See Also:**[ModelMBeanNotificationBroadcaster.removeAttributeChangeNotificationListener(javax.management.NotificationListener, java.lang.String)](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#removeAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String))

### removeAttributeChangeNotificationListener

public void **removeAttributeChangeNotificationListener**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) inlistener,  
 [String](http://docs.google.com/java/lang/String.html) inAttributeName)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#removeAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String)) Removes a listener for attributeChangeNotifications from the RequiredModelMBean.

**Specified by:**[removeAttributeChangeNotificationListener](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#removeAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**inlistener - The listener name which was handling notifications emitted by the registered MBean. This method will remove all information related to this listener.inAttributeName - The attribute for which the listener no longer wants to receive attributeChangeNotifications. If null the listener will be removed for all attributeChangeNotifications. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException If the inAttributeName parameter does not correspond to an attribute name. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean or is null.**See Also:**[ModelMBeanNotificationBroadcaster.addAttributeChangeNotificationListener(javax.management.NotificationListener, java.lang.String, java.lang.Object)](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#addAttributeChangeNotificationListener(javax.management.NotificationListener,%20java.lang.String,%20java.lang.Object))

### sendAttributeChangeNotification

public void **sendAttributeChangeNotification**([AttributeChangeNotification](http://docs.google.com/javax/management/AttributeChangeNotification.html) ntfyObj)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendAttributeChangeNotification(javax.management.AttributeChangeNotification)) Sends an attributeChangeNotification which is passed in to the registered attributeChangeNotification listeners on the ModelMBean.

**Specified by:**[sendAttributeChangeNotification](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendAttributeChangeNotification(javax.management.AttributeChangeNotification)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**ntfyObj - The notification which is to be passed to the 'handleNotification' method of the listener object. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException: The AttributeChangeNotification object passed in parameter is null.

### sendAttributeChangeNotification

public void **sendAttributeChangeNotification**([Attribute](http://docs.google.com/javax/management/Attribute.html) inOldVal,  
 [Attribute](http://docs.google.com/javax/management/Attribute.html) inNewVal)  
 throws [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

**Description copied from interface:** [**ModelMBeanNotificationBroadcaster**](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendAttributeChangeNotification(javax.management.Attribute,%20javax.management.Attribute)) Sends an attributeChangeNotification which contains the old value and new value for the attribute to the registered AttributeChangeNotification listeners on the ModelMBean.

**Specified by:**[sendAttributeChangeNotification](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html#sendAttributeChangeNotification(javax.management.Attribute,%20javax.management.Attribute)) in interface [ModelMBeanNotificationBroadcaster](http://docs.google.com/javax/management/modelmbean/ModelMBeanNotificationBroadcaster.html) **Parameters:**inOldVal - The original value for the AttributeinNewVal - The current value for the Attribute

The constructed attributeChangeNotification will be:  
 type "jmx.attribute.change"  
 source this ModelMBean instance  
 sequence 1  
 attributeName oldValue.getName()  
 attributeType oldValue's class  
 attributeOldValue oldValue.getValue()  
 attributeNewValue newValue.getValue()

**Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps a distributed communication Exception. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps an IllegalArgumentException: An Attribute object passed in parameter is null or the names of the two Attribute objects in parameter are not the same.

### getClassLoaderRepository

protected [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) **getClassLoaderRepository**()

Return the Class Loader Repository used to perform class loading. Subclasses may wish to redefine this method in order to return the appropriate [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) that should be used in this object.

**Returns:**the Class Loader Repository.

### preRegister

public [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **preRegister**([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) server,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Allows the MBean to perform any operations it needs before being registered in the MBean server. If the name of the MBean is not specified, the MBean can provide a name for its registration. If any exception is raised, the MBean will not be registered in the MBean server.

In order to ensure proper run-time semantics of RequireModelMBean, Any subclass of RequiredModelMBean overloading or overriding this method should call super.preRegister(server, name) in its own preRegister implementation.

**Specified by:**[preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Parameters:**server - The MBean server in which the MBean will be registered.name - The object name of the MBean. This name is null if the name parameter to one of the createMBean or registerMBean methods in the [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) interface is null. In that case, this method must return a non-null ObjectName for the new MBean. **Returns:**The name under which the MBean is to be registered. This value must not be null. If the name parameter is not null, it will usually but not necessarily be the returned value. **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - This exception will be caught by the MBean server and re-thrown as an [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html).

### postRegister

public void **postRegister**([Boolean](http://docs.google.com/java/lang/Boolean.html) registrationDone)

Allows the MBean to perform any operations needed after having been registered in the MBean server or after the registration has failed.

In order to ensure proper run-time semantics of RequireModelMBean, Any subclass of RequiredModelMBean overloading or overriding this method should call super.postRegister(registrationDone) in its own postRegister implementation.

**Specified by:**[postRegister](http://docs.google.com/javax/management/MBeanRegistration.html#postRegister(java.lang.Boolean)) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Parameters:**registrationDone - Indicates whether or not the MBean has been successfully registered in the MBean server. The value false means that the registration phase has failed.

### preDeregister

public void **preDeregister**()  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Allows the MBean to perform any operations it needs before being unregistered by the MBean server.

In order to ensure proper run-time semantics of RequireModelMBean, Any subclass of RequiredModelMBean overloading or overriding this method should call super.preDeregister() in its own preDeregister implementation.

**Specified by:**[preDeregister](http://docs.google.com/javax/management/MBeanRegistration.html#preDeregister()) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - This exception will be caught by the MBean server and re-thrown as an [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html).

### postDeregister

public void **postDeregister**()

Allows the MBean to perform any operations needed after having been unregistered in the MBean server.

In order to ensure proper run-time semantics of RequireModelMBean, Any subclass of RequiredModelMBean overloading or overriding this method should call super.postDeregister() in its own postDeregister implementation.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RequiredModelMBean.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/modelmbean/ModelMBeanOperationInfo.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/modelmbean/XMLParseException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/modelmbean/RequiredModelMBean.html)    [**NO FRAMES**](http://docs.google.com/RequiredModelMBean.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).