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

## **javax.management**

Interface MBeanServer

**All Superinterfaces:** [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **All Known Subinterfaces:** [MBeanServerForwarder](http://docs.google.com/javax/management/remote/MBeanServerForwarder.html)

public interface **MBeanServer**extends [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html)

This is the interface for MBean manipulation on the agent side. It contains the methods necessary for the creation, registration, and deletion of MBeans as well as the access methods for registered MBeans. This is the core component of the JMX infrastructure.

User code does not usually implement this interface. Instead, an object that implements this interface is obtained with one of the methods in the [MBeanServerFactory](http://docs.google.com/javax/management/MBeanServerFactory.html) class.

Every MBean which is added to the MBean server becomes manageable: its attributes and operations become remotely accessible through the connectors/adaptors connected to that MBean server. A Java object cannot be registered in the MBean server unless it is a JMX compliant MBean.

When an MBean is registered or unregistered in the MBean server a [MBeanServerNotification](http://docs.google.com/javax/management/MBeanServerNotification.html) Notification is emitted. To register an object as listener to MBeanServerNotifications you should call the MBean server method [addNotificationListener](http://docs.google.com/javax/management/MBeanServer.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) with ObjectName the ObjectName of the [MBeanServerDelegate](http://docs.google.com/javax/management/MBeanServerDelegate.html). This ObjectName is:

JMImplementation:type=MBeanServerDelegate.

An object obtained from the [createMBeanServer](http://docs.google.com/javax/management/MBeanServerFactory.html#createMBeanServer(java.lang.String)) or [newMBeanServer](http://docs.google.com/javax/management/MBeanServerFactory.html#newMBeanServer(java.lang.String)) methods of the [MBeanServerFactory](http://docs.google.com/javax/management/MBeanServerFactory.html) class applies security checks to its methods, as follows.

First, if there is no security manager ([System.getSecurityManager()](http://docs.google.com/java/lang/System.html#getSecurityManager()) is null), then an implementation of this interface is free not to make any checks.

Assuming that there is a security manager, or that the implementation chooses to make checks anyway, the checks are made as detailed below. In what follows, className is the string returned by [MBeanInfo.getClassName()](http://docs.google.com/javax/management/MBeanInfo.html#getClassName()) for the target MBean.

If a security check fails, the method throws [SecurityException](http://docs.google.com/java/lang/SecurityException.html).

For methods that can throw [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html), this exception is thrown for a non-existent MBean, regardless of permissions. This is because a non-existent MBean has no className.

* For the [invoke](http://docs.google.com/javax/management/MBeanServer.html#invoke(javax.management.ObjectName,%20java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) method, the caller's permissions must imply [MBeanPermission(className, operationName, name, "invoke")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getAttribute](http://docs.google.com/javax/management/MBeanServer.html#getAttribute(javax.management.ObjectName,%20java.lang.String)) method, the caller's permissions must imply [MBeanPermission(className, attribute, name, "getAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getAttributes](http://docs.google.com/javax/management/MBeanServer.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "getAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)). Additionally, for each attribute *a* in the [AttributeList](http://docs.google.com/javax/management/AttributeList.html), if the caller's permissions do not imply [MBeanPermission(className, *a*, name, "getAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)), the MBean server will behave as if that attribute had not been in the supplied list.
* For the [setAttribute](http://docs.google.com/javax/management/MBeanServer.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute)) method, the caller's permissions must imply [MBeanPermission(className, attrName, name, "setAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)), where attrName is [attribute.getName()](http://docs.google.com/javax/management/Attribute.html#getName()).
* For the [setAttributes](http://docs.google.com/javax/management/MBeanServer.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "setAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)). Additionally, for each attribute *a* in the [AttributeList](http://docs.google.com/javax/management/AttributeList.html), if the caller's permissions do not imply [MBeanPermission(className, *a*, name, "setAttribute")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)), the MBean server will behave as if that attribute had not been in the supplied list.
* For the addNotificationListener methods, the caller's permissions must imply [MBeanPermission(className, null, name, "addNotificationListener")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the removeNotificationListener methods, the caller's permissions must imply [MBeanPermission(className, null, name, "removeNotificationListener")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getMBeanInfo](http://docs.google.com/javax/management/MBeanServer.html#getMBeanInfo(javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "getMBeanInfo")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getObjectInstance](http://docs.google.com/javax/management/MBeanServer.html#getObjectInstance(javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "getObjectInstance")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [isInstanceOf](http://docs.google.com/javax/management/MBeanServer.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "isInstanceOf")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [queryMBeans](http://docs.google.com/javax/management/MBeanServer.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)) method, the caller's permissions must imply [MBeanPermission(null, null, name, "queryMBeans")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)). Additionally, for each MBean that matches name, if the caller's permissions do not imply [MBeanPermission(className, null, name, "queryMBeans")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)), the MBean server will behave as if that MBean did not exist.  
  Certain query elements perform operations on the MBean server. If the caller does not have the required permissions for a given MBean, that MBean will not be included in the result of the query. The standard query elements that are affected are [Query.attr(String)](http://docs.google.com/javax/management/Query.html#attr(java.lang.String)), [Query.attr(String,String)](http://docs.google.com/javax/management/Query.html#attr(java.lang.String,%20java.lang.String)), and [Query.classattr()](http://docs.google.com/javax/management/Query.html#classattr()).
* For the [queryNames](http://docs.google.com/javax/management/MBeanServer.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)) method, the checks are the same as for queryMBeans except that "queryNames" is used instead of "queryMBeans" in the MBeanPermission objects. Note that a "queryMBeans" permission implies the corresponding "queryNames" permission.
* For the [getDomains](http://docs.google.com/javax/management/MBeanServer.html#getDomains()) method, the caller's permissions must imply [MBeanPermission(null, null, name, "getDomains")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)). Additionally, for each domain d in the returned array, if the caller's permissions do not imply [MBeanPermission(null, null, new ObjectName("d:x=x"), "getDomains")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)), the domain is eliminated from the array. Here, x=x is any key=value pair, needed to satisfy ObjectName's constructor but not otherwise relevant.
* For the [getClassLoader](http://docs.google.com/javax/management/MBeanServer.html#getClassLoader(javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, loaderName, "getClassLoader")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getClassLoaderFor](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderFor(javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, mbeanName, "getClassLoaderFor")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [getClassLoaderRepository](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) method, the caller's permissions must imply [MBeanPermission(null, null, null, "getClassLoaderRepository")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the deprecated deserialize methods, the required permissions are the same as for the methods that replace them.
* For the instantiate methods, the caller's permissions must imply [MBeanPermission(className, null, null, "instantiate")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).
* For the [registerMBean](http://docs.google.com/javax/management/MBeanServer.html#registerMBean(java.lang.Object,%20javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "registerMBean")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)). Here className is the string returned by [MBeanInfo.getClassName()](http://docs.google.com/javax/management/MBeanInfo.html#getClassName()) for an object of this class.  
  If the MBeanPermission check succeeds, the MBean's class is validated by checking that its [ProtectionDomain](http://docs.google.com/java/security/ProtectionDomain.html) implies [MBeanTrustPermission("register")](http://docs.google.com/javax/management/MBeanTrustPermission.html#MBeanTrustPermission(java.lang.String)).  
  Finally, if the name argument is null, another MBeanPermission check is made using the ObjectName returned by [MBeanRegistration.preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)).
* For the createMBean methods, the caller's permissions must imply the permissions needed by the equivalent instantiate followed by registerMBean.
* For the [unregisterMBean](http://docs.google.com/javax/management/MBeanServer.html#unregisterMBean(javax.management.ObjectName)) method, the caller's permissions must imply [MBeanPermission(className, null, name, "unregisterMBean")](http://docs.google.com/javax/management/MBeanPermission.html#MBeanPermission(java.lang.String,%20java.lang.String,%20javax.management.ObjectName,%20java.lang.String)).

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [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 a registered MBean. |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.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 a registered MBean. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/MBeanServer.html#createMBean(java.lang.String,%20javax.management.ObjectName))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Instantiates and registers an MBean in the MBean server. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/MBeanServer.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Object](http://docs.google.com/java/lang/Object.html)[] params, [String](http://docs.google.com/java/lang/String.html)[] signature)            Instantiates and registers an MBean in the MBean server. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/MBeanServer.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)            Instantiates and registers an MBean in the MBean server. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/MBeanServer.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName, [Object](http://docs.google.com/java/lang/Object.html)[] params, [String](http://docs.google.com/java/lang/String.html)[] signature)            Instantiates and registers an MBean in the MBean server. |
| [ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) | [**deserialize**](http://docs.google.com/javax/management/MBeanServer.html#deserialize(javax.management.ObjectName,%20byte%5B%5D))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, byte[] data)  **Deprecated.** *Use* [*getClassLoaderFor*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderFor(javax.management.ObjectName)) *to obtain the appropriate class loader for deserialization.* |
| [ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) | [**deserialize**](http://docs.google.com/javax/management/MBeanServer.html#deserialize(java.lang.String,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, byte[] data)  **Deprecated.** *Use* [*getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) *to obtain the class loader repository and use it to deserialize.* |
| [ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) | [**deserialize**](http://docs.google.com/javax/management/MBeanServer.html#deserialize(java.lang.String,%20javax.management.ObjectName,%20byte%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName, byte[] data)  **Deprecated.** *Use* [*getClassLoader*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoader(javax.management.ObjectName)) *to obtain the class loader for deserialization.* |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getAttribute**](http://docs.google.com/javax/management/MBeanServer.html#getAttribute(javax.management.ObjectName,%20java.lang.String))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) attribute)            Gets the value of a specific attribute of a named MBean. |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**getAttributes**](http://docs.google.com/javax/management/MBeanServer.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html)[] attributes)            Enables the values of several attributes of a named MBean. |
| [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) | [**getClassLoader**](http://docs.google.com/javax/management/MBeanServer.html#getClassLoader(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)            Return the named [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html). |
| [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) | [**getClassLoaderFor**](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderFor(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) mbeanName)            Return the [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) that was used for loading the class of the named MBean. |
| [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) | [**getClassLoaderRepository**](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository())()            Return the ClassLoaderRepository for this MBeanServer. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDefaultDomain**](http://docs.google.com/javax/management/MBeanServer.html#getDefaultDomain())()            Returns the default domain used for naming the MBean. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getDomains**](http://docs.google.com/javax/management/MBeanServer.html#getDomains())()            Returns the list of domains in which any MBean is currently registered. |
| [Integer](http://docs.google.com/java/lang/Integer.html) | [**getMBeanCount**](http://docs.google.com/javax/management/MBeanServer.html#getMBeanCount())()            Returns the number of MBeans registered in the MBean server. |
| [MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html) | [**getMBeanInfo**](http://docs.google.com/javax/management/MBeanServer.html#getMBeanInfo(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            This method discovers the attributes and operations that an MBean exposes for management. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**getObjectInstance**](http://docs.google.com/javax/management/MBeanServer.html#getObjectInstance(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Gets the ObjectInstance for a given MBean registered with the MBean server. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String))([String](http://docs.google.com/java/lang/String.html) className)            Instantiates an object using the list of all class loaders registered in the MBean server's [Class Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, [Object](http://docs.google.com/java/lang/Object.html)[] params, [String](http://docs.google.com/java/lang/String.html)[] signature)            Instantiates an object using the list of all class loaders registered in the MBean server [Class Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String,%20javax.management.ObjectName))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)            Instantiates an object using the class Loader specified by its ObjectName. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**instantiate**](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName, [Object](http://docs.google.com/java/lang/Object.html)[] params, [String](http://docs.google.com/java/lang/String.html)[] signature)            Instantiates an object. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/javax/management/MBeanServer.html#invoke(javax.management.ObjectName,%20java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) operationName, [Object](http://docs.google.com/java/lang/Object.html)[] params, [String](http://docs.google.com/java/lang/String.html)[] signature)            Invokes an operation on an MBean. |
| boolean | [**isInstanceOf**](http://docs.google.com/javax/management/MBeanServer.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) className)            Returns true if the MBean specified is an instance of the specified class, false otherwise. |
| boolean | [**isRegistered**](http://docs.google.com/javax/management/MBeanServer.html#isRegistered(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Checks whether an MBean, identified by its object name, is already registered with the MBean server. |
| [Set](http://docs.google.com/java/util/Set.html)<[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html)> | [**queryMBeans**](http://docs.google.com/javax/management/MBeanServer.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [QueryExp](http://docs.google.com/javax/management/QueryExp.html) query)            Gets MBeans controlled by the MBean server. |
| [Set](http://docs.google.com/java/util/Set.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)> | [**queryNames**](http://docs.google.com/javax/management/MBeanServer.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [QueryExp](http://docs.google.com/javax/management/QueryExp.html) query)            Gets the names of MBeans controlled by the MBean server. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**registerMBean**](http://docs.google.com/javax/management/MBeanServer.html#registerMBean(java.lang.Object,%20javax.management.ObjectName))([Object](http://docs.google.com/java/lang/Object.html) object, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Registers a pre-existing object as an MBean with the MBean server. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)            Removes a listener from a registered MBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [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 a registered MBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener)            Removes a listener from a registered MBean. |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/MBeanServer.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.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 a registered MBean. |
| void | [**setAttribute**](http://docs.google.com/javax/management/MBeanServer.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Attribute](http://docs.google.com/javax/management/Attribute.html) attribute)            Sets the value of a specific attribute of a named MBean. |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**setAttributes**](http://docs.google.com/javax/management/MBeanServer.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [AttributeList](http://docs.google.com/javax/management/AttributeList.html) attributes)            Sets the values of several attributes of a named MBean. |
| void | [**unregisterMBean**](http://docs.google.com/javax/management/MBeanServer.html#unregisterMBean(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Unregisters an MBean from the MBean server. |

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

### createMBean

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **createMBean**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName))

Instantiates and registers an MBean in the MBean server. The MBean server will use its [Default Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) to load the class of the MBean. An object name is associated to the MBean. If the object name given is null, the MBean must provide its own name by implementing the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface and returning the name from the [preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) method.

This method is equivalent to [createMBean(className, name, (Object[]) null, (String[]) null)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)).

**Specified by:**[createMBean](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**className - The class name of the MBean to be instantiated.name - The object name of the MBean. May be null. **Returns:**An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean. If the contained ObjectName is n, the contained Java class name is [getMBeanInfo(n)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or a java.lang.Exception that occurred when trying to invoke the MBean's constructor. [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) - The MBean is already under the control of the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the MBean has thrown an exception [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html) - This class is not a JMX compliant MBean

### createMBean

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **createMBean**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName))

Instantiates and registers an MBean in the MBean server. The class loader to be used is identified by its object name. An object name is associated to the MBean. If the object name of the loader is null, the ClassLoader that loaded the MBean server will be used. If the MBean's object name given is null, the MBean must provide its own name by implementing the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface and returning the name from the [preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) method.

This method is equivalent to [createMBean(className, name, loaderName, (Object[]) null, (String[]) null)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)).

**Specified by:**[createMBean](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**className - The class name of the MBean to be instantiated.name - The object name of the MBean. May be null.loaderName - The object name of the class loader to be used. **Returns:**An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean. If the contained ObjectName is n, the contained Java class name is [getMBeanInfo(n)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or a java.lang.Exception that occurred when trying to invoke the MBean's constructor. [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) - The MBean is already under the control of the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the MBean has thrown an exception [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html) - This class is not a JMX compliant MBean [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The specified class loader is not registered in the MBean server.

### createMBean

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **createMBean**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) Instantiates and registers an MBean in the MBean server. The MBean server will use its [Default Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) to load the class of the MBean. An object name is associated to the MBean. If the object name given is null, the MBean must provide its own name by implementing the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface and returning the name from the [preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) method.

**Specified by:**[createMBean](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**className - The class name of the MBean to be instantiated.name - The object name of the MBean. May be null.params - An array containing the parameters of the constructor to be invoked.signature - An array containing the signature of the constructor to be invoked. **Returns:**An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean. If the contained ObjectName is n, the contained Java class name is [getMBeanInfo(n)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or a java.lang.Exception that occurred when trying to invoke the MBean's constructor. [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) - The MBean is already under the control of the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the MBean has thrown an exception [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html) - This class is not a JMX compliant MBean

### createMBean

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **createMBean**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) Instantiates and registers an MBean in the MBean server. The class loader to be used is identified by its object name. An object name is associated to the MBean. If the object name of the loader is not specified, the ClassLoader that loaded the MBean server will be used. If the MBean object name given is null, the MBean must provide its own name by implementing the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface and returning the name from the [preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) method.

**Specified by:**[createMBean](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**className - The class name of the MBean to be instantiated.name - The object name of the MBean. May be null.loaderName - The object name of the class loader to be used.params - An array containing the parameters of the constructor to be invoked.signature - An array containing the signature of the constructor to be invoked. **Returns:**An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean. If the contained ObjectName is n, the contained Java class name is [getMBeanInfo(n)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or a java.lang.Exception that occurred when trying to invoke the MBean's constructor. [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) - The MBean is already under the control of the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the MBean has thrown an exception [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html) - This class is not a JMX compliant MBean [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The specified class loader is not registered in the MBean server.

### registerMBean

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **registerMBean**([Object](http://docs.google.com/java/lang/Object.html) object,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html)

Registers a pre-existing object as an MBean with the MBean server. If the object name given is null, the MBean must provide its own name by implementing the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface and returning the name from the [preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) method.

**Parameters:**object - The MBean to be registered as an MBean.name - The object name of the MBean. May be null. **Returns:**An ObjectInstance, containing the ObjectName and the Java class name of the newly registered MBean. If the contained ObjectName is n, the contained Java class name is [getMBeanInfo(n)](http://docs.google.com/javax/management/MBeanServer.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [InstanceAlreadyExistsException](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) - The MBean is already under the control of the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered. [NotCompliantMBeanException](http://docs.google.com/javax/management/NotCompliantMBeanException.html) - This object is not a JMX compliant MBean [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a java.lang.IllegalArgumentException: The object passed in parameter is null or no object name is specified.

### unregisterMBean

void **unregisterMBean**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#unregisterMBean(javax.management.ObjectName)) Unregisters an MBean from the MBean server. The MBean is identified by its object name. Once the method has been invoked, the MBean may no longer be accessed by its object name.

**Specified by:**[unregisterMBean](http://docs.google.com/javax/management/MBeanServerConnection.html#unregisterMBean(javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean to be unregistered. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html) - The preDeregister ((MBeanRegistration interface) method of the MBean has thrown an exception.

### getObjectInstance

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **getObjectInstance**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#getObjectInstance(javax.management.ObjectName)) Gets the ObjectInstance for a given MBean registered with the MBean server.

**Specified by:**[getObjectInstance](http://docs.google.com/javax/management/MBeanServerConnection.html#getObjectInstance(javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean. **Returns:**The ObjectInstance associated with the MBean specified by name. The contained ObjectName is name and the contained class name is [getMBeanInfo(name)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).getClassName(). **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server.

### queryMBeans

[Set](http://docs.google.com/java/util/Set.html)<[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html)> **queryMBeans**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [QueryExp](http://docs.google.com/javax/management/QueryExp.html) query)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)) Gets MBeans controlled by the MBean server. This method allows any of the following to be obtained: All MBeans, a set of MBeans specified by pattern matching on the ObjectName and/or a Query expression, a specific MBean. When the object name is null or no domain and key properties are specified, all objects are to be selected (and filtered if a query is specified). It returns the set of ObjectInstance objects (containing the ObjectName and the Java Class name) for the selected MBeans.

**Specified by:**[queryMBeans](http://docs.google.com/javax/management/MBeanServerConnection.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name pattern identifying the MBeans to be retrieved. If null or no domain and key properties are specified, all the MBeans registered will be retrieved.query - The query expression to be applied for selecting MBeans. If null no query expression will be applied for selecting MBeans. **Returns:**A set containing the ObjectInstance objects for the selected MBeans. If no MBean satisfies the query an empty list is returned.

### queryNames

[Set](http://docs.google.com/java/util/Set.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)> **queryNames**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [QueryExp](http://docs.google.com/javax/management/QueryExp.html) query)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)) Gets the names of MBeans controlled by the MBean server. This method enables any of the following to be obtained: The names of all MBeans, the names of a set of MBeans specified by pattern matching on the ObjectName and/or a Query expression, a specific MBean name (equivalent to testing whether an MBean is registered). When the object name is null or no domain and key properties are specified, all objects are selected (and filtered if a query is specified). It returns the set of ObjectNames for the MBeans selected.

**Specified by:**[queryNames](http://docs.google.com/javax/management/MBeanServerConnection.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name pattern identifying the MBean names to be retrieved. If null or no domain and key properties are specified, the name of all registered MBeans will be retrieved.query - The query expression to be applied for selecting MBeans. If null no query expression will be applied for selecting MBeans. **Returns:**A set containing the ObjectNames for the MBeans selected. If no MBean satisfies the query, an empty list is returned.

### isRegistered

boolean **isRegistered**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#isRegistered(javax.management.ObjectName)) Checks whether an MBean, identified by its object name, is already registered with the MBean server.

**Specified by:**[isRegistered](http://docs.google.com/javax/management/MBeanServerConnection.html#isRegistered(javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean to be checked. **Returns:**True if the MBean is already registered in the MBean server, false otherwise.

### getMBeanCount

[Integer](http://docs.google.com/java/lang/Integer.html) **getMBeanCount**()

Returns the number of MBeans registered in the MBean server.

**Specified by:**[getMBeanCount](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanCount()) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Returns:**the number of registered MBeans, wrapped in an Integer. If the caller's permissions are restricted, this number may be greater than the number of MBeans the caller can access.

### getAttribute

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

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String)) Gets the value of a specific attribute of a named MBean. The MBean is identified by its object name.

**Specified by:**[getAttribute](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean from which the attribute is to be retrieved.attribute - A String specifying the name of the attribute to be retrieved. **Returns:**The value of the retrieved attribute. **Throws:** [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps an exception thrown by the MBean's getter. [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html) - The attribute specified is not accessible in the MBean. [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.Exception thrown when trying to invoke the setter.**See Also:**[MBeanServerConnection.setAttribute(javax.management.ObjectName, javax.management.Attribute)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute))

### getAttributes

[AttributeList](http://docs.google.com/javax/management/AttributeList.html) **getAttributes**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [String](http://docs.google.com/java/lang/String.html)[] attributes)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D)) Enables the values of several attributes of a named MBean. The MBean is identified by its object name.

**Specified by:**[getAttributes](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean from which the attributes are retrieved.attributes - A list of the attributes to be retrieved. **Returns:**The list of the retrieved attributes. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - An exception occurred when trying to invoke the getAttributes method of a Dynamic MBean.**See Also:**[MBeanServerConnection.setAttributes(javax.management.ObjectName, javax.management.AttributeList)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList))

### setAttribute

void **setAttribute**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Attribute](http://docs.google.com/javax/management/Attribute.html) attribute)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [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)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute)) Sets the value of a specific attribute of a named MBean. The MBean is identified by its object name.

**Specified by:**[setAttribute](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean within which the attribute is to be set.attribute - The identification of the attribute to be set and the value it is to be set to. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html) - The attribute specified is not accessible in the MBean. [InvalidAttributeValueException](http://docs.google.com/javax/management/InvalidAttributeValueException.html) - The value specified for the attribute is not valid. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps an exception thrown by the MBean's setter. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.Exception thrown when trying to invoke the setter.**See Also:**[MBeanServerConnection.getAttribute(javax.management.ObjectName, java.lang.String)](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String))

### setAttributes

[AttributeList](http://docs.google.com/javax/management/AttributeList.html) **setAttributes**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [AttributeList](http://docs.google.com/javax/management/AttributeList.html) attributes)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList)) Sets the values of several attributes of a named MBean. The MBean is identified by its object name.

**Specified by:**[setAttributes](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean within which the attributes are to be set.attributes - A list of attributes: The identification of the attributes to be set and the values they are to be set to. **Returns:**The list of attributes that were set, with their new values. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - An exception occurred when trying to invoke the getAttributes method of a Dynamic MBean.**See Also:**[MBeanServerConnection.getAttributes(javax.management.ObjectName, java.lang.String[])](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D))

### invoke

[Object](http://docs.google.com/java/lang/Object.html) **invoke**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [String](http://docs.google.com/java/lang/String.html) operationName,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#invoke(javax.management.ObjectName,%20java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) Invokes an operation on an MBean.

**Specified by:**[invoke](http://docs.google.com/javax/management/MBeanServerConnection.html#invoke(javax.management.ObjectName,%20java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The object name of the MBean on which the method is to be invoked.operationName - The name of the operation to be invoked.params - An array containing the parameters to be set when the operation is invokedsignature - 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 operation, which represents the result of invoking the operation on the MBean specified. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps an exception thrown by the MBean's invoked method. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.Exception thrown while trying to invoke the method.

### getDefaultDomain

[String](http://docs.google.com/java/lang/String.html) **getDefaultDomain**()

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#getDefaultDomain()) Returns the default domain used for naming the MBean. The default domain name is used as the domain part in the ObjectName of MBeans if no domain is specified by the user.

**Specified by:**[getDefaultDomain](http://docs.google.com/javax/management/MBeanServerConnection.html#getDefaultDomain()) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Returns:**the default domain.

### getDomains

[String](http://docs.google.com/java/lang/String.html)[] **getDomains**()

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

Returns the list of domains in which any MBean is currently registered. A string is in the returned array if and only if there is at least one MBean registered with an ObjectName whose [getDomain()](http://docs.google.com/javax/management/ObjectName.html#getDomain()) is equal to that string. The order of strings within the returned array is not defined.

**Specified by:**[getDomains](http://docs.google.com/javax/management/MBeanServerConnection.html#getDomains()) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Returns:**the list of domains.

### addNotificationListener

void **addNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [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 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

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

Adds a listener to a registered MBean.

A notification emitted by an MBean will be forwarded by the MBeanServer to the listener. If the source of the notification is a reference to an MBean object, the MBean server will replace it by that MBean's ObjectName. Otherwise the source is unchanged.

**Specified by:**[addNotificationListener](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be added.listener - The listener object which will handle the notifications emitted by the registered MBean.filter - The filter object. If filter is null, no filtering will be performed before handling notifications.handback - The context to be sent to the listener when a notification is emitted. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name provided does not match any of the registered MBeans.**See Also:**[MBeanServerConnection.removeNotificationListener(ObjectName, NotificationListener)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener)), [MBeanServerConnection.removeNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### addNotificationListener

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

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

Adds a listener to a registered MBean.

A notification emitted by an MBean will be forwarded by the MBeanServer to the listener. If the source of the notification is a reference to an MBean object, the MBean server will replace it by that MBean's ObjectName. Otherwise the source is unchanged.

The listener object that receives notifications is the one that is registered with the given name at the time this method is called. Even if it is subsequently unregistered, it will continue to receive notifications.

**Specified by:**[addNotificationListener](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be added.listener - The object name of the listener which will handle the notifications emitted by the registered MBean.filter - The filter object. If filter is null, no filtering will be performed before handling notifications.handback - The context to be sent to the listener when a notification is emitted. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name of the notification listener or of the notification broadcaster does not match any of the registered MBeans.**See Also:**[MBeanServerConnection.removeNotificationListener(ObjectName, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName)), [MBeanServerConnection.removeNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeNotificationListener

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

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName)) Removes a listener from a registered MBean.

If the listener is registered more than once, perhaps with different filters or callbacks, this method will remove all those registrations.

**Specified by:**[removeNotificationListener](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be removed.listener - The object name of the listener to be removed. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name provided does not match any of the registered MBeans. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean.**See Also:**[MBeanServerConnection.addNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeNotificationListener

void **removeNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener,  
 [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) filter,  
 [Object](http://docs.google.com/java/lang/Object.html) handback)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html)

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

Removes a listener from a registered 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/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be removed.listener - The object name of the listener to be removed.filter - The filter that was specified when the listener was added.handback - The handback that was specified when the listener was added. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name provided does not match any of the registered MBeans. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean, or it is not registered with the given filter and handback.**See Also:**[MBeanServerConnection.addNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeNotificationListener

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

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener))

Removes a listener from a registered MBean.

If the listener is registered more than once, perhaps with different filters or callbacks, this method will remove all those registrations.

**Specified by:**[removeNotificationListener](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be removed.listener - The listener to be removed. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name provided does not match any of the registered MBeans. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean.**See Also:**[MBeanServerConnection.addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeNotificationListener

void **removeNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [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 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html)

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

Removes a listener from a registered 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/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean on which the listener should be removed.listener - The listener to be removed.filter - The filter that was specified when the listener was added.handback - The handback that was specified when the listener was added. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean name provided does not match any of the registered MBeans. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - The listener is not registered in the MBean, or it is not registered with the given filter and handback.**See Also:**[MBeanServerConnection.addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServerConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### getMBeanInfo

[MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html) **getMBeanInfo**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [IntrospectionException](http://docs.google.com/javax/management/IntrospectionException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)) This method discovers the attributes and operations that an MBean exposes for management.

**Specified by:**[getMBeanInfo](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The name of the MBean to analyze **Returns:**An instance of MBeanInfo allowing the retrieval of all attributes and operations of this MBean. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified was not found. [IntrospectionException](http://docs.google.com/javax/management/IntrospectionException.html) - An exception occurred during introspection. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - An exception occurred when trying to invoke the getMBeanInfo of a Dynamic MBean.

### isInstanceOf

boolean **isInstanceOf**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [String](http://docs.google.com/java/lang/String.html) className)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

**Description copied from interface:** [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String))

Returns true if the MBean specified is an instance of the specified class, false otherwise.

If name does not name an MBean, this method throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html).

Otherwise, let

X be the MBean named by name,

L be the ClassLoader of X,

N be the class name in X's [MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html).

If N equals className, the result is true.

Otherwise, if L successfully loads className and X is an instance of this class, the result is true.

Otherwise, if L successfully loads both N and className, and the second class is assignable from the first, the result is true.

Otherwise, the result is false.

**Specified by:**[isInstanceOf](http://docs.google.com/javax/management/MBeanServerConnection.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String)) in interface [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **Parameters:**name - The ObjectName of the MBean.className - The name of the class. **Returns:**true if the MBean specified is an instance of the specified class according to the rules above, false otherwise. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server.**See Also:**[Class.isInstance(java.lang.Object)](http://docs.google.com/java/lang/Class.html#isInstance(java.lang.Object))

### instantiate

[Object](http://docs.google.com/java/lang/Object.html) **instantiate**([String](http://docs.google.com/java/lang/String.html) className)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html)

Instantiates an object using the list of all class loaders registered in the MBean server's [Class Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html). The object's class should have a public constructor. This method returns a reference to the newly created object. The newly created object is not registered in the MBean server.

This method is equivalent to [instantiate(className, (Object[]) null, (String[]) null)](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)).

**Parameters:**className - The class name of the object to be instantiated. **Returns:**The newly instantiated object. **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or the java.lang.Exception that occurred when trying to invoke the object's constructor. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the object has thrown an exception [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null.

### instantiate

[Object](http://docs.google.com/java/lang/Object.html) **instantiate**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Instantiates an object using the class Loader specified by its ObjectName. If the loader name is null, the ClassLoader that loaded the MBean Server will be used. The object's class should have a public constructor. This method returns a reference to the newly created object. The newly created object is not registered in the MBean server.

This method is equivalent to [instantiate(className, loaderName, (Object[]) null, (String[]) null)](http://docs.google.com/javax/management/MBeanServer.html#instantiate(java.lang.String,%20javax.management.ObjectName,%20java.lang.Object%5B%5D,%20java.lang.String%5B%5D)).

**Parameters:**className - The class name of the MBean to be instantiated.loaderName - The object name of the class loader to be used. **Returns:**The newly instantiated object. **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or the java.lang.Exception that occurred when trying to invoke the object's constructor. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the object has thrown an exception. [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The specified class loader is not registered in the MBeanServer. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null.

### instantiate

[Object](http://docs.google.com/java/lang/Object.html) **instantiate**([String](http://docs.google.com/java/lang/String.html) className,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html)

Instantiates an object using the list of all class loaders registered in the MBean server [Class Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html). The object's class should have a public constructor. The call returns a reference to the newly created object. The newly created object is not registered in the MBean server.

**Parameters:**className - The class name of the object to be instantiated.params - An array containing the parameters of the constructor to be invoked.signature - An array containing the signature of the constructor to be invoked. **Returns:**The newly instantiated object. **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or the java.lang.Exception that occurred when trying to invoke the object's constructor. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the object has thrown an exception [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null.

### instantiate

[Object](http://docs.google.com/java/lang/Object.html) **instantiate**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName,  
 [Object](http://docs.google.com/java/lang/Object.html)[] params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature)  
 throws [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [MBeanException](http://docs.google.com/javax/management/MBeanException.html),  
 [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Instantiates an object. The class loader to be used is identified by its object name. If the object name of the loader is null, the ClassLoader that loaded the MBean server will be used. The object's class should have a public constructor. The call returns a reference to the newly created object. The newly created object is not registered in the MBean server.

**Parameters:**className - The class name of the object to be instantiated.params - An array containing the parameters of the constructor to be invoked.signature - An array containing the signature of the constructor to be invoked.loaderName - The object name of the class loader to be used. **Returns:**The newly instantiated object. **Throws:** [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - Wraps a java.lang.ClassNotFoundException or the java.lang.Exception that occurred when trying to invoke the object's constructor. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - The constructor of the object has thrown an exception [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The specified class loader is not registered in the MBean server. [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null.

### deserialize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
[ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) **deserialize**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 byte[] data)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [OperationsException](http://docs.google.com/javax/management/OperationsException.html)

**Deprecated.** *Use* [*getClassLoaderFor*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderFor(javax.management.ObjectName)) *to obtain the appropriate class loader for deserialization.*

De-serializes a byte array in the context of the class loader of an MBean.

**Parameters:**name - The name of the MBean whose class loader should be used for the de-serialization.data - The byte array to be de-sererialized. **Returns:**The de-serialized object stream. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not found. [OperationsException](http://docs.google.com/javax/management/OperationsException.html) - Any of the usual Input/Output related exceptions.

### deserialize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
[ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) **deserialize**([String](http://docs.google.com/java/lang/String.html) className,  
 byte[] data)  
 throws [OperationsException](http://docs.google.com/javax/management/OperationsException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Deprecated.** *Use* [*getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) *to obtain the class loader repository and use it to deserialize.*

De-serializes a byte array in the context of a given MBean class loader. The class loader is found by loading the class className through the [Class Loader Repository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html). The resultant class's class loader is the one to use.

**Parameters:**className - The name of the class whose class loader should be used for the de-serialization.data - The byte array to be de-sererialized. **Returns:**The de-serialized object stream. **Throws:** [OperationsException](http://docs.google.com/javax/management/OperationsException.html) - Any of the usual Input/Output related exceptions. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - The specified class could not be loaded by the class loader repository

### deserialize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
[ObjectInputStream](http://docs.google.com/java/io/ObjectInputStream.html) **deserialize**([String](http://docs.google.com/java/lang/String.html) className,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName,  
 byte[] data)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [OperationsException](http://docs.google.com/javax/management/OperationsException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html)

**Deprecated.** *Use* [*getClassLoader*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoader(javax.management.ObjectName)) *to obtain the class loader for deserialization.*

De-serializes a byte array in the context of a given MBean class loader. The class loader is the one that loaded the class with name "className". The name of the class loader to be used for loading the specified class is specified. If null, the MBean Server's class loader will be used.

**Parameters:**className - The name of the class whose class loader should be used for the de-serialization.data - The byte array to be de-sererialized.loaderName - The name of the class loader to be used for loading the specified class. If null, the MBean Server's class loader will be used. **Returns:**The de-serialized object stream. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The specified class loader MBean is not found. [OperationsException](http://docs.google.com/javax/management/OperationsException.html) - Any of the usual Input/Output related exceptions. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - The specified class could not be loaded by the specified class loader.

### getClassLoaderFor

[ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) **getClassLoaderFor**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) mbeanName)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Return the [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) that was used for loading the class of the named MBean.

**Parameters:**mbeanName - The ObjectName of the MBean. **Returns:**The ClassLoader used for that MBean. If l is the MBean's actual ClassLoader, and r is the returned value, then either:

* r is identical to l; or
* the result of r[.loadClass(s)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) is the same as l[.loadClass(s)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) for any string s.

What this means is that the ClassLoader may be wrapped in another ClassLoader for security or other reasons. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - if the named MBean is not found.

### getClassLoader

[ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) **getClassLoader**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) loaderName)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html)

Return the named [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html).

**Parameters:**loaderName - The ObjectName of the ClassLoader. May be null, in which case the MBean server's own ClassLoader is returned. **Returns:**The named ClassLoader. If l is the actual ClassLoader with that name, and r is the returned value, then either:

* r is identical to l; or
* the result of r[.loadClass(s)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) is the same as l[.loadClass(s)](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)) for any string s.

What this means is that the ClassLoader may be wrapped in another ClassLoader for security or other reasons. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - if the named ClassLoader is not found.

### getClassLoaderRepository

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

Return the ClassLoaderRepository for this MBeanServer.

**Returns:**The ClassLoaderRepository for this MBeanServer.

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

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).