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

## **javax.management.remote.rmi**

Interface RMIConnection

**All Superinterfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html), [Remote](http://docs.google.com/java/rmi/Remote.html) **All Known Implementing Classes:** [RMIConnectionImpl](http://docs.google.com/javax/management/remote/rmi/RMIConnectionImpl.html), [RMIConnectionImpl\_Stub](http://docs.google.com/javax/management/remote/rmi/RMIConnectionImpl_Stub.html)

public interface **RMIConnection**extends [Closeable](http://docs.google.com/java/io/Closeable.html), [Remote](http://docs.google.com/java/rmi/Remote.html)

RMI object used to forward an MBeanServer request from a client to its MBeanServer implementation on the server side. There is one Remote object implementing this interface for each remote client connected to an RMI connector.

User code does not usually refer to this interface. It is specified as part of the public API so that different implementations of that API will interoperate.

To ensure that client parameters will be deserialized at the server side with the correct classloader, client parameters such as parameters used to invoke a method are wrapped in a [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html). An implementation of this interface must first get the appropriate class loader for the operation and its target, then deserialize the marshalled parameters with this classloader. Except as noted, a parameter that is a MarshalledObject or MarshalledObject[] must not be null; the behavior is unspecified if it is.

Class loading aspects are detailed in the  [JMX Specification, version 1.4](http://docs.google.com/technotes/guides/jmx/JMX_1_4_specification.pdf) PDF document.

Most methods in this interface parallel methods in the [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) interface. Where an aspect of the behavior of a method is not specified here, it is the same as in the corresponding MBeanServerConnection method.

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| void | [**addNotificationListener**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) filter, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) handback, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [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)). |
| [Integer](http://docs.google.com/java/lang/Integer.html)[] | [**addNotificationListeners**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListeners(javax.management.ObjectName%5B%5D,%20java.rmi.MarshalledObject%5B%5D,%20javax.security.auth.Subject%5B%5D))([ObjectName](http://docs.google.com/javax/management/ObjectName.html)[] names, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)[] filters, [Subject](http://docs.google.com/javax/security/auth/Subject.html)[] delegationSubjects)            Handles the method [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)). |
| void | [**close**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#close())()            Closes this connection. |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.lang.String%5B%5D,%20javax.security.auth.Subject))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params, [String](http://docs.google.com/java/lang/String.html)[] signature, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.createMBean(String, ObjectName, Object[], String[])](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)). |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.lang.String%5B%5D,%20javax.security.auth.Subject))([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, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params, [String](http://docs.google.com/java/lang/String.html)[] signature, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.createMBean(String, ObjectName, ObjectName, Object[], String[])](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)). |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName,%20javax.security.auth.Subject))([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, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.createMBean(String, ObjectName, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName)). |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**createMBean**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.security.auth.Subject))([String](http://docs.google.com/java/lang/String.html) className, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.createMBean(String, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName)). |
| [NotificationResult](http://docs.google.com/javax/management/remote/NotificationResult.html) | [**fetchNotifications**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#fetchNotifications(long,%20int,%20long))(long clientSequenceNumber, int maxNotifications, long timeout)            Retrieves notifications from the connector server. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getAttribute**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) attribute, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getAttribute(ObjectName, String)](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String)). |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**getAttributes**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html)[] attributes, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getAttributes(ObjectName, String[])](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D)). |
| [String](http://docs.google.com/java/lang/String.html) | [**getConnectionId**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getConnectionId())()            Returns the connection ID. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDefaultDomain**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getDefaultDomain(javax.security.auth.Subject))([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getDefaultDomain()](http://docs.google.com/javax/management/MBeanServerConnection.html#getDefaultDomain()). |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getDomains**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getDomains(javax.security.auth.Subject))([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getDomains()](http://docs.google.com/javax/management/MBeanServerConnection.html#getDomains()). |
| [Integer](http://docs.google.com/java/lang/Integer.html) | [**getMBeanCount**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getMBeanCount(javax.security.auth.Subject))([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getMBeanCount()](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanCount()). |
| [MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html) | [**getMBeanInfo**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getMBeanInfo(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)). |
| [ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) | [**getObjectInstance**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getObjectInstance(javax.management.ObjectName,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.getObjectInstance(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#getObjectInstance(javax.management.ObjectName)). |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#invoke(javax.management.ObjectName,%20java.lang.String,%20java.rmi.MarshalledObject,%20java.lang.String%5B%5D,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) operationName, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params, [String](http://docs.google.com/java/lang/String.html)[] signature, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.invoke(ObjectName, String, Object[], String[])](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)). |
| boolean | [**isInstanceOf**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [String](http://docs.google.com/java/lang/String.html) className, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.isInstanceOf(ObjectName, String)](http://docs.google.com/javax/management/MBeanServerConnection.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String)). |
| boolean | [**isRegistered**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#isRegistered(javax.management.ObjectName,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.isRegistered(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#isRegistered(javax.management.ObjectName)). |
| [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/remote/rmi/RMIConnection.html#queryMBeans(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) query, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.queryMBeans(ObjectName, QueryExp)](http://docs.google.com/javax/management/MBeanServerConnection.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)). |
| [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/remote/rmi/RMIConnection.html#queryNames(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) query, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.queryNames(ObjectName, QueryExp)](http://docs.google.com/javax/management/MBeanServerConnection.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)). |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) filter, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) handback, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [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)). |
| void | [**removeNotificationListener**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.removeNotificationListener(ObjectName, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName)). |
| void | [**removeNotificationListeners**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#removeNotificationListeners(javax.management.ObjectName,%20java.lang.Integer%5B%5D,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Integer](http://docs.google.com/java/lang/Integer.html)[] listenerIDs, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the [removeNotificationListener(ObjectName, NotificationListener)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener)) and [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)) methods. |
| void | [**setAttribute**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#setAttribute(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) attribute, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.setAttribute(ObjectName, Attribute)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute)). |
| [AttributeList](http://docs.google.com/javax/management/AttributeList.html) | [**setAttributes**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#setAttributes(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) attributes, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.setAttributes(ObjectName, AttributeList)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList)). |
| void | [**unregisterMBean**](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#unregisterMBean(javax.management.ObjectName,%20javax.security.auth.Subject))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name, [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Handles the method [MBeanServerConnection.unregisterMBean(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#unregisterMBean(javax.management.ObjectName)). |

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

### getConnectionId

[String](http://docs.google.com/java/lang/String.html) **getConnectionId**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns the connection ID. This string is different for every open connection to a given RMI connector server.

**Returns:**the connection ID **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[RMIConnector.connect](http://docs.google.com/javax/management/remote/rmi/RMIConnector.html#connect())

### close

void **close**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Closes this connection. On return from this method, the RMI object implementing this interface is unexported, so further remote calls to it will fail.

**Specified by:**[close](http://docs.google.com/java/io/Closeable.html#close()) in interface [Closeable](http://docs.google.com/java/io/Closeable.html) **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the connection could not be closed, or the Remote object could not be unexported, or there was a communication failure when transmitting the remote close request.

### 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,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.createMBean(String, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName)).

**Parameters:**className - The class name of the MBean to be instantiated.name - The object name of the MBean. May be null.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject)).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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.createMBean(String, ObjectName, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#createMBean(java.lang.String,%20javax.management.ObjectName,%20javax.management.ObjectName)).

**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.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject)).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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.createMBean(String, ObjectName, Object[], String[])](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)). The Object[] parameter is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject. The encapsulated array can be null, equivalent to an empty array.signature - An array containing the signature of the constructor to be invoked. Can be null, equivalent to an empty array.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject)).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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern, or no ObjectName is specified for the MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.createMBean(String, ObjectName, ObjectName, Object[], String[])](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)). The Object[] parameter is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject. The encapsulated array can be null, equivalent to an empty array.signature - An array containing the signature of the constructor to be invoked. Can be null, equivalent to an empty array.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject)).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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern, or no ObjectName is specified for the MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### unregisterMBean

void **unregisterMBean**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.unregisterMBean(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#unregisterMBean(javax.management.ObjectName)).

**Parameters:**name - The object name of the MBean to be unregistered.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null or the MBean you are when trying to unregister is the [MBeanServerDelegate](http://docs.google.com/javax/management/MBeanServerDelegate.html) MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### getObjectInstance

[ObjectInstance](http://docs.google.com/javax/management/ObjectInstance.html) **getObjectInstance**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getObjectInstance(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#getObjectInstance(javax.management.ObjectName)).

**Parameters:**name - The object name of the MBean.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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/remote/rmi/RMIConnection.html#getMBeanInfo(javax.management.ObjectName,%20javax.security.auth.Subject)).getClassName(). **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified is not registered in the MBean server. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) query,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.queryMBeans(ObjectName, QueryExp)](http://docs.google.com/javax/management/MBeanServerConnection.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)). The QueryExp is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject. If the MarshalledObject encapsulates a null value no query expression will be applied for selecting MBeans.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**A set containing the ObjectInstance objects for the selected MBeans. If no MBean satisfies the query an empty list is returned. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) query,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.queryNames(ObjectName, QueryExp)](http://docs.google.com/javax/management/MBeanServerConnection.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)). The QueryExp is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject. If the MarshalledObject encapsulates a null value no query expression will be applied for selecting MBeans.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**A set containing the ObjectNames for the MBeans selected. If no MBean satisfies the query, an empty list is returned. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### isRegistered

boolean **isRegistered**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.isRegistered(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#isRegistered(javax.management.ObjectName)).

**Parameters:**name - The object name of the MBean to be checked.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**True if the MBean is already registered in the MBean server, false otherwise. **Throws:** RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### getMBeanCount

[Integer](http://docs.google.com/java/lang/Integer.html) **getMBeanCount**([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getMBeanCount()](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanCount()).

**Parameters:**delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**the number of MBeans registered. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### 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,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getAttribute(ObjectName, String)](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String)).

**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.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**The value of the retrieved attribute. **Throws:** [AttributeNotFoundException](http://docs.google.com/javax/management/AttributeNotFoundException.html) - The attribute specified is not accessible in the MBean. [MBeanException](http://docs.google.com/javax/management/MBeanException.html) - Wraps an exception thrown by the MBean's getter. [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 getter. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null or the attribute in parameter is null. RuntimeMBeanException - Wraps a runtime exception thrown by the MBean's getter. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[setAttribute(javax.management.ObjectName, java.rmi.MarshalledObject, javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#setAttribute(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))

### 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,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getAttributes(ObjectName, String[])](http://docs.google.com/javax/management/MBeanServerConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D)).

**Parameters:**name - The object name of the MBean from which the attributes are retrieved.attributes - A list of the attributes to be retrieved.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. RuntimeOperationsException - Wrap a java.lang.IllegalArgumentException: The object name in parameter is null or attributes in parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[setAttributes(javax.management.ObjectName, java.rmi.MarshalledObject, javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#setAttributes(javax.management.ObjectName,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))

### setAttribute

void **setAttribute**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) attribute,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.setAttribute(ObjectName, Attribute)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttribute(javax.management.ObjectName,%20javax.management.Attribute)). The Attribute parameter is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null or the attribute in parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[getAttribute(javax.management.ObjectName, java.lang.String, javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getAttribute(javax.management.ObjectName,%20java.lang.String,%20javax.security.auth.Subject))

### setAttributes

[AttributeList](http://docs.google.com/javax/management/AttributeList.html) **setAttributes**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) attributes,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.setAttributes(ObjectName, AttributeList)](http://docs.google.com/javax/management/MBeanServerConnection.html#setAttributes(javax.management.ObjectName,%20javax.management.AttributeList)). The AttributeList parameter is wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null or attributes in parameter is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[getAttributes(javax.management.ObjectName, java.lang.String[], javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#getAttributes(javax.management.ObjectName,%20java.lang.String%5B%5D,%20javax.security.auth.Subject))

### 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,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) params,  
 [String](http://docs.google.com/java/lang/String.html)[] signature,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.invoke(ObjectName, String, Object[], String[])](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)). The Object[] parameter is wrapped in a MarshalledObject.

**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 invoked, encapsulated into a MarshalledObject. The encapsulated array can be null, equivalent to an empty array.signature - 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. Can be null, equivalent to an empty array.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. RuntimeOperationsException - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) when name or operationName is null.

### getDefaultDomain

[String](http://docs.google.com/java/lang/String.html) **getDefaultDomain**([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getDefaultDomain()](http://docs.google.com/javax/management/MBeanServerConnection.html#getDefaultDomain()).

**Parameters:**delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**the default domain. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### getDomains

[String](http://docs.google.com/java/lang/String.html)[] **getDomains**([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getDomains()](http://docs.google.com/javax/management/MBeanServerConnection.html#getDomains()).

**Parameters:**delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**the list of domains. **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### getMBeanInfo

[MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html) **getMBeanInfo**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 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),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.getMBeanInfo(ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#getMBeanInfo(javax.management.ObjectName)).

**Parameters:**name - The name of the MBean to analyzedelegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Returns:**An instance of MBeanInfo allowing the retrieval of all attributes and operations of this MBean. **Throws:** [IntrospectionException](http://docs.google.com/javax/management/IntrospectionException.html) - An exception occurred during introspection. [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - The MBean specified was not found. [ReflectionException](http://docs.google.com/javax/management/ReflectionException.html) - An exception occurred when trying to invoke the getMBeanInfo of a Dynamic MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null.

### isInstanceOf

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

Handles the method [MBeanServerConnection.isInstanceOf(ObjectName, String)](http://docs.google.com/javax/management/MBeanServerConnection.html#isInstanceOf(javax.management.ObjectName,%20java.lang.String)).

**Parameters:**name - The ObjectName of the MBean.className - The name of the class.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. RuntimeOperationsException - Wraps a java.lang.IllegalArgumentException: The object name in parameter is null.

### addNotificationListener

void **addNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) filter,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) handback,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [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)). The NotificationFilter parameter is wrapped in a MarshalledObject. The Object (handback) parameter is also wrapped in a MarshalledObject.

**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, encapsulated into a MarshalledObject. If filter encapsulated in the MarshalledObject has a null value, no filtering will be performed before handling notifications.handback - The context to be sent to the listener when a notification is emitted, encapsulated into a MarshalledObject.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. RuntimeOperationsException - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html). The MBean named by listener exists but does not implement the [NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) interface, or name or listener is null. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.**See Also:**[removeNotificationListener(ObjectName, ObjectName, Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.security.auth.Subject)), [removeNotificationListener(ObjectName, ObjectName, MarshalledObject, MarshalledObject, Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))

### removeNotificationListener

void **removeNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [MBeanServerConnection.removeNotificationListener(ObjectName, ObjectName)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName)).

**Parameters:**name - The name of the MBean on which the listener should be removed.listener - The object name of the listener to be removed.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. RuntimeOperationsException - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) when name or listener is null.**See Also:**[addNotificationListener(javax.management.ObjectName, javax.management.ObjectName, java.rmi.MarshalledObject, java.rmi.MarshalledObject, javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))

### removeNotificationListener

void **removeNotificationListener**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) listener,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) filter,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html) handback,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [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)). The NotificationFilter parameter is wrapped in a MarshalledObject. The Object parameter is also wrapped in a MarshalledObject.

**Parameters:**name - The name of the MBean on which the listener should be removed.listener - A listener that was previously added to this MBean.filter - The filter that was specified when the listener was added, encapsulated into a MarshalledObject.handback - The handback that was specified when the listener was added, encapsulated into a MarshalledObject.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **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. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to perform this operation. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. RuntimeOperationsException - Wraps an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) when name or listener is null.**See Also:**[addNotificationListener(javax.management.ObjectName, javax.management.ObjectName, java.rmi.MarshalledObject, java.rmi.MarshalledObject, javax.security.auth.Subject)](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20java.rmi.MarshalledObject,%20java.rmi.MarshalledObject,%20javax.security.auth.Subject))

### addNotificationListeners

[Integer](http://docs.google.com/java/lang/Integer.html)[] **addNotificationListeners**([ObjectName](http://docs.google.com/javax/management/ObjectName.html)[] names,  
 [MarshalledObject](http://docs.google.com/java/rmi/MarshalledObject.html)[] filters,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html)[] delegationSubjects)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the method [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)).

Register for notifications from the given MBeans that match the given filters. The remote client can subsequently retrieve the notifications using the [fetchNotifications](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#fetchNotifications(long,%20int,%20long)) method.

For each listener, the original NotificationListener and handback are kept on the client side; in order for the client to be able to identify them, the server generates and returns a unique listenerID. This listenerID is forwarded with the Notifications to the remote client.

If any one of the given (name, filter) pairs cannot be registered, then the operation fails with an exception, and no names or filters are registered.

**Parameters:**names - the ObjectNames identifying the MBeans emitting the Notifications.filters - an array of marshalled representations of the NotificationFilters. Elements of this array can be null.delegationSubjects - the Subjects on behalf of which the listeners are being added. Elements of this array can be null. Also, the delegationSubjects parameter itself can be null, which is equivalent to an array of null values with the same size as the names and filters arrays. **Returns:**an array of listenerIDs identifying the local listeners. This array has the same number of elements as the parameters. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if names or filters is null, or if names contains a null element, or if the three arrays do not all have the same size. [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if one of the elements of filters unmarshalls as a non-null object that is not a NotificationFilter. [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - if one of the names does not correspond to any registered MBean. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if, for one of the MBeans, the client, or the delegated Subject if any, does not have permission to add a listener. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

### removeNotificationListeners

void **removeNotificationListeners**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name,  
 [Integer](http://docs.google.com/java/lang/Integer.html)[] listenerIDs,  
 [Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html),  
 [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Handles the [removeNotificationListener(ObjectName, NotificationListener)](http://docs.google.com/javax/management/MBeanServerConnection.html#removeNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener)) and [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)) methods.

This method removes one or more NotificationListeners from a given MBean in the MBean server.

The NotificationListeners are identified by the IDs which were returned by the [addNotificationListeners(ObjectName[], MarshalledObject[], Subject[])](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListeners(javax.management.ObjectName%5B%5D,%20java.rmi.MarshalledObject%5B%5D,%20javax.security.auth.Subject%5B%5D)) method.

**Parameters:**name - the ObjectName identifying the MBean emitting the Notifications.listenerIDs - the list of the IDs corresponding to the listeners to remove.delegationSubject - The Subject containing the delegation principals or null if the authentication principal is used instead. **Throws:** [InstanceNotFoundException](http://docs.google.com/javax/management/InstanceNotFoundException.html) - if the given name does not correspond to any registered MBean. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - if one of the listeners was not found on the server side. This exception can happen if the MBean discarded a listener for some reason other than a call to MBeanServer.removeNotificationListener. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the client, or the delegated Subject if any, does not have permission to remove the listeners. [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if ObjectName or listenerIds is null or if listenerIds contains a null element.

### fetchNotifications

[NotificationResult](http://docs.google.com/javax/management/remote/NotificationResult.html) **fetchNotifications**(long clientSequenceNumber,  
 int maxNotifications,  
 long timeout)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Retrieves notifications from the connector server. This method can block until there is at least one notification or until the specified timeout is reached. The method can also return at any time with zero notifications.

A notification can be included in the result if its sequence number is no less than clientSequenceNumber and this client has registered at least one listener for the MBean generating the notification, with a filter that accepts the notification. Each listener that is interested in the notification is identified by an Integer ID that was returned by [addNotificationListeners(ObjectName[], MarshalledObject[], Subject[])](http://docs.google.com/javax/management/remote/rmi/RMIConnection.html#addNotificationListeners(javax.management.ObjectName%5B%5D,%20java.rmi.MarshalledObject%5B%5D,%20javax.security.auth.Subject%5B%5D)).

**Parameters:**clientSequenceNumber - the first sequence number that the client is interested in. If negative, it is interpreted as meaning the sequence number that the next notification will have.maxNotifications - the maximum number of different notifications to return. The TargetedNotification array in the returned NotificationResult can have more elements than this if the same notification appears more than once. The behavior is unspecified if this parameter is negative.timeout - the maximum time in milliseconds to wait for a notification to arrive. This can be 0 to indicate that the method should not wait if there are no notifications, but should return at once. It can be Long.MAX\_VALUE to indicate that there is no timeout. The behavior is unspecified if this parameter is negative. **Returns:**A NotificationResult. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a general communication exception occurred.

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