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

## **javax.management.remote**

Interface JMXConnector

**All Superinterfaces:** [Closeable](http://docs.google.com/java/io/Closeable.html) **All Known Implementing Classes:** [RMIConnector](http://docs.google.com/javax/management/remote/rmi/RMIConnector.html)

public interface **JMXConnector**extends [Closeable](http://docs.google.com/java/io/Closeable.html)

The client end of a JMX API connector. An object of this type can be used to establish a connection to a connector server.

A newly-created object of this type is unconnected. Its [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect()) method must be called before it can be used. However, objects created by [JMXConnectorFactory.connect](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#connect(javax.management.remote.JMXServiceURL,%20java.util.Map)) are already connected.

**Since:** 1.5

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**CREDENTIALS**](http://docs.google.com/javax/management/remote/JMXConnector.html#CREDENTIALS)            Name of the attribute that specifies the credentials to send to the connector server during connection. |

| **Method Summary** | |
| --- | --- |
| void | [**addConnectionNotificationListener**](http://docs.google.com/javax/management/remote/JMXConnector.html#addConnectionNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener, [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) filter, [Object](http://docs.google.com/java/lang/Object.html) handback)            Adds a listener to be informed of changes in connection status. |
| void | [**close**](http://docs.google.com/javax/management/remote/JMXConnector.html#close())()            Closes the client connection to its server. |
| void | [**connect**](http://docs.google.com/javax/management/remote/JMXConnector.html#connect())()            Establishes the connection to the connector server. |
| void | [**connect**](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> env)            Establishes the connection to the connector server. |
| [String](http://docs.google.com/java/lang/String.html) | [**getConnectionId**](http://docs.google.com/javax/management/remote/JMXConnector.html#getConnectionId())()            Gets this connection's ID from the connector server. |
| [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) | [**getMBeanServerConnection**](http://docs.google.com/javax/management/remote/JMXConnector.html#getMBeanServerConnection())()            Returns an MBeanServerConnection object representing a remote MBean server. |
| [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) | [**getMBeanServerConnection**](http://docs.google.com/javax/management/remote/JMXConnector.html#getMBeanServerConnection(javax.security.auth.Subject))([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)            Returns an MBeanServerConnection object representing a remote MBean server on which operations are performed on behalf of the supplied delegation subject. |
| void | [**removeConnectionNotificationListener**](http://docs.google.com/javax/management/remote/JMXConnector.html#removeConnectionNotificationListener(javax.management.NotificationListener))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) listener)            Removes a listener from the list to be informed of changes in status. |
| void | [**removeConnectionNotificationListener**](http://docs.google.com/javax/management/remote/JMXConnector.html#removeConnectionNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) l, [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) f, [Object](http://docs.google.com/java/lang/Object.html) handback)            Removes a listener from the list to be informed of changes in status. |

| **Field Detail** |
| --- |

### CREDENTIALS

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

Name of the attribute that specifies the credentials to send to the connector server during connection. The value associated with this attribute, if any, is a serializable object of an appropriate type for the server's [JMXAuthenticator](http://docs.google.com/javax/management/remote/JMXAuthenticator.html).

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.management.remote.JMXConnector.CREDENTIALS)

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

### connect

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

Establishes the connection to the connector server. This method is equivalent to [connect(null)](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)).

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the connection could not be made because of a communication problem. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the connection could not be made for security reasons.

### connect

void **connect**([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> env)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Establishes the connection to the connector server.

If connect has already been called successfully on this object, calling it again has no effect. If, however, [close()](http://docs.google.com/javax/management/remote/JMXConnector.html#close()) was called after connect, the new connect will throw an IOException.

Otherwise, either connect has never been called on this object, or it has been called but produced an exception. Then calling connect will attempt to establish a connection to the connector server.

**Parameters:**env - the properties of the connection. Properties in this map override properties in the map specified when the JMXConnector was created, if any. This parameter can be null, which is equivalent to an empty map. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the connection could not be made because of a communication problem. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the connection could not be made for security reasons.

### getMBeanServerConnection

[MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **getMBeanServerConnection**()  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns an MBeanServerConnection object representing a remote MBean server. For a given JMXConnector, two successful calls to this method will usually return the same MBeanServerConnection object, though this is not required.

For each method in the returned MBeanServerConnection, calling the method causes the corresponding method to be called in the remote MBean server. The value returned by the MBean server method is the value returned to the client. If the MBean server method produces an Exception, the same Exception is seen by the client. If the MBean server method, or the attempt to call it, produces an Error, the Error is wrapped in a [JMXServerErrorException](http://docs.google.com/javax/management/remote/JMXServerErrorException.html), which is seen by the client.

Calling this method is equivalent to calling [getMBeanServerConnection(null)](http://docs.google.com/javax/management/remote/JMXConnector.html#getMBeanServerConnection(javax.security.auth.Subject)) meaning that no delegation subject is specified and that all the operations called on the MBeanServerConnection must use the authenticated subject, if any.

**Returns:**an object that implements the MBeanServerConnection interface by forwarding its methods to the remote MBean server. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a valid MBeanServerConnection cannot be created, for instance because the connection to the remote MBean server has not yet been established (with the [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)) method), or it has been closed, or it has broken.

### getMBeanServerConnection

[MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) **getMBeanServerConnection**([Subject](http://docs.google.com/javax/security/auth/Subject.html) delegationSubject)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Returns an MBeanServerConnection object representing a remote MBean server on which operations are performed on behalf of the supplied delegation subject. For a given JMXConnector and Subject, two successful calls to this method will usually return the same MBeanServerConnection object, though this is not required.

For each method in the returned MBeanServerConnection, calling the method causes the corresponding method to be called in the remote MBean server on behalf of the given delegation subject instead of the authenticated subject. The value returned by the MBean server method is the value returned to the client. If the MBean server method produces an Exception, the same Exception is seen by the client. If the MBean server method, or the attempt to call it, produces an Error, the Error is wrapped in a [JMXServerErrorException](http://docs.google.com/javax/management/remote/JMXServerErrorException.html), which is seen by the client.

**Parameters:**delegationSubject - the Subject on behalf of which requests will be performed. Can be null, in which case requests will be performed on behalf of the authenticated Subject, if any. **Returns:**an object that implements the MBeanServerConnection interface by forwarding its methods to the remote MBean server on behalf of a given delegation subject. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a valid MBeanServerConnection cannot be created, for instance because the connection to the remote MBean server has not yet been established (with the [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)) method), or it has been closed, or it has broken.

### close

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

Closes the client connection to its server. Any ongoing or new request using the MBeanServerConnection returned by [getMBeanServerConnection()](http://docs.google.com/javax/management/remote/JMXConnector.html#getMBeanServerConnection()) will get an IOException.

If close has already been called successfully on this object, calling it again has no effect. If close has never been called, or if it was called but produced an exception, an attempt will be made to close the connection. This attempt can succeed, in which case close will return normally, or it can generate an exception.

Closing a connection is a potentially slow operation. For example, if the server has crashed, the close operation might have to wait for a network protocol timeout. Callers that do not want to block in a close operation should do it in a separate thread.

**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 cannot be closed cleanly. If this exception is thrown, it is not known whether the server end of the connection has been cleanly closed.

### addConnectionNotificationListener

void **addConnectionNotificationListener**([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 be informed of changes in connection status. The listener will receive notifications of type [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). An implementation can send other types of notifications too.

Any number of listeners can be added with this method. The same listener can be added more than once with the same or different values for the filter and handback. There is no special treatment of a duplicate entry. For example, if a listener is registered twice with no filter, then its handleNotification method will be called twice for each notification.

**Parameters:**listener - a listener to receive connection status notifications.filter - a filter to select which notifications are to be delivered to the listener, or null if all notifications are to be delivered.handback - an object to be given to the listener along with each notification. Can be null. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if listener is null.**See Also:**[removeConnectionNotificationListener(javax.management.NotificationListener)](http://docs.google.com/javax/management/remote/JMXConnector.html#removeConnectionNotificationListener(javax.management.NotificationListener)), [NotificationBroadcaster.addNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/NotificationBroadcaster.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeConnectionNotificationListener

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

Removes a listener from the list to be informed of changes in status. The listener must previously have been added. If there is more than one matching listener, all are removed.

**Parameters:**listener - a listener to receive connection status notifications. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if listener is null. [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - if the listener is not registered with this JMXConnector.**See Also:**[removeConnectionNotificationListener(NotificationListener, NotificationFilter, Object)](http://docs.google.com/javax/management/remote/JMXConnector.html#removeConnectionNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [addConnectionNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/remote/JMXConnector.html#addConnectionNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [NotificationEmitter.removeNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/NotificationEmitter.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### removeConnectionNotificationListener

void **removeConnectionNotificationListener**([NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) l,  
 [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) f,  
 [Object](http://docs.google.com/java/lang/Object.html) handback)  
 throws [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html)

Removes a listener from the list to be informed of changes in status. The listener must previously have been added with the same three parameters. If there is more than one matching listener, only one is removed.

**Parameters:**l - a listener to receive connection status notifications.f - a filter to select which notifications are to be delivered to the listener. Can be null.handback - an object to be given to the listener along with each notification. Can be null. **Throws:** [ListenerNotFoundException](http://docs.google.com/javax/management/ListenerNotFoundException.html) - if the listener is not registered with this JMXConnector, or is not registered with the given filter and handback.**See Also:**[removeConnectionNotificationListener(NotificationListener)](http://docs.google.com/javax/management/remote/JMXConnector.html#removeConnectionNotificationListener(javax.management.NotificationListener)), [addConnectionNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/remote/JMXConnector.html#addConnectionNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [NotificationEmitter.removeNotificationListener(javax.management.NotificationListener, javax.management.NotificationFilter, java.lang.Object)](http://docs.google.com/javax/management/NotificationEmitter.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object))

### getConnectionId

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

Gets this connection's ID from the connector server. For a given connector server, every connection will have a unique id which does not change during the lifetime of the connection.

**Returns:**the unique ID of this connection. This is the same as the ID that the connector server includes in its [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html)s. The [package description](http://docs.google.com/javax/management/remote/package-summary.html) describes the conventions for connection IDs. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the connection ID cannot be obtained, for instance because the connection is closed or broken.

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

[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).