| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RMIConnectorServer.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/rmi/RMIConnector.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/remote/rmi/RMIIIOPServerImpl.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/remote/rmi/RMIConnectorServer.html)    [**NO FRAMES**](http://docs.google.com/RMIConnectorServer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#35nkun2) | [METHOD](#z337ya) |

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

Class RMIConnectorServer

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [javax.management.NotificationBroadcasterSupport](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html)  
 [javax.management.remote.JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html)  
 **javax.management.remote.rmi.RMIConnectorServer**

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

public class **RMIConnectorServer**extends [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html)

A JMX API connector server that creates RMI-based connections from remote clients. Usually, such connector servers are made using [JMXConnectorServerFactory](http://docs.google.com/javax/management/remote/JMXConnectorServerFactory.html). However, specialized applications can use this class directly, for example with an [RMIServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html) object.

**Since:** 1.5

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**JNDI\_REBIND\_ATTRIBUTE**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#JNDI_REBIND_ATTRIBUTE)            Name of the attribute that specifies whether the [RMIServer](http://docs.google.com/javax/management/remote/rmi/RMIServer.html) stub that represents an RMI connector server should override an existing stub at the same address. |
| static [String](http://docs.google.com/java/lang/String.html) | [**RMI\_CLIENT\_SOCKET\_FACTORY\_ATTRIBUTE**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMI_CLIENT_SOCKET_FACTORY_ATTRIBUTE)            Name of the attribute that specifies the [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) for the RMI objects created in conjunction with this connector. |
| static [String](http://docs.google.com/java/lang/String.html) | [**RMI\_SERVER\_SOCKET\_FACTORY\_ATTRIBUTE**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMI_SERVER_SOCKET_FACTORY_ATTRIBUTE)            Name of the attribute that specifies the [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) for the RMI objects created in conjunction with this connector. |

| **Fields inherited from class javax.management.remote.**[**JMXConnectorServer**](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) |
| --- |
| [AUTHENTICATOR](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#AUTHENTICATOR) |

| **Constructor Summary** | |
| --- | --- |
| [**RMIConnectorServer**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMIConnectorServer(javax.management.remote.JMXServiceURL,%20java.util.Map))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment)            Makes an RMIConnectorServer. |
| [**RMIConnectorServer**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMIConnectorServer(javax.management.remote.JMXServiceURL,%20java.util.Map,%20javax.management.MBeanServer))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment, [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbeanServer)            Makes an RMIConnectorServer for the given MBean server. |
| [**RMIConnectorServer**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMIConnectorServer(javax.management.remote.JMXServiceURL,%20java.util.Map,%20javax.management.remote.rmi.RMIServerImpl,%20javax.management.MBeanServer))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment, [RMIServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html) rmiServerImpl, [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbeanServer)            Makes an RMIConnectorServer for the given MBean server. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**connectionClosed**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#connectionClosed(java.lang.String,%20java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) connectionId, [String](http://docs.google.com/java/lang/String.html) message, [Object](http://docs.google.com/java/lang/Object.html) userData)            Called by a subclass when a client connection is closed normally. |
| protected  void | [**connectionFailed**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#connectionFailed(java.lang.String,%20java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) connectionId, [String](http://docs.google.com/java/lang/String.html) message, [Object](http://docs.google.com/java/lang/Object.html) userData)            Called by a subclass when a client connection fails. |
| protected  void | [**connectionOpened**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#connectionOpened(java.lang.String,%20java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) connectionId, [String](http://docs.google.com/java/lang/String.html) message, [Object](http://docs.google.com/java/lang/Object.html) userData)            Called by a subclass when a new client connection is opened. |
| [JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) | [**getAddress**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#getAddress())()            The address of this connector server. |
| [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> | [**getAttributes**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#getAttributes())()            The attributes for this connector server. |
| boolean | [**isActive**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#isActive())()            Determines whether the connector server is active. |
| void | [**setMBeanServerForwarder**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#setMBeanServerForwarder(javax.management.remote.MBeanServerForwarder))([MBeanServerForwarder](http://docs.google.com/javax/management/remote/MBeanServerForwarder.html) mbsf)            Adds an object that intercepts requests for the MBean server that arrive through this connector server. |
| void | [**start**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start())()            Activates the connector server, that is starts listening for client connections. |
| void | [**stop**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#stop())()            Deactivates the connector server, that is, stops listening for client connections. |
| [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) | [**toJMXConnector**](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#toJMXConnector(java.util.Map))([Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> env)            Returns a client stub for this connector server. |

| **Methods inherited from class javax.management.remote.**[**JMXConnectorServer**](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) |
| --- |
| [getConnectionIds](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getConnectionIds()), [getMBeanServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getMBeanServer()), [getNotificationInfo](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getNotificationInfo()), [postDeregister](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#postDeregister()), [postRegister](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#postRegister(java.lang.Boolean)), [preDeregister](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#preDeregister()), [preRegister](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) |

| **Methods inherited from class javax.management.**[**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html) |
| --- |
| [addNotificationListener](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#addNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [handleNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#handleNotification(javax.management.NotificationListener,%20javax.management.Notification,%20java.lang.Object)), [removeNotificationListener](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#removeNotificationListener(javax.management.NotificationListener)), [removeNotificationListener](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#removeNotificationListener(javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)), [sendNotification](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html#sendNotification(javax.management.Notification)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

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

### JNDI\_REBIND\_ATTRIBUTE

public static final [String](http://docs.google.com/java/lang/String.html) **JNDI\_REBIND\_ATTRIBUTE**

Name of the attribute that specifies whether the [RMIServer](http://docs.google.com/javax/management/remote/rmi/RMIServer.html) stub that represents an RMI connector server should override an existing stub at the same address. The value associated with this attribute, if any, should be a string that is equal, ignoring case, to "true" or "false". The default value is false.

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

### RMI\_CLIENT\_SOCKET\_FACTORY\_ATTRIBUTE

public static final [String](http://docs.google.com/java/lang/String.html) **RMI\_CLIENT\_SOCKET\_FACTORY\_ATTRIBUTE**

Name of the attribute that specifies the [RMIClientSocketFactory](http://docs.google.com/java/rmi/server/RMIClientSocketFactory.html) for the RMI objects created in conjunction with this connector. The value associated with this attribute must be of type RMIClientSocketFactory and can only be specified in the Map argument supplied when creating a connector server.

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

### RMI\_SERVER\_SOCKET\_FACTORY\_ATTRIBUTE

public static final [String](http://docs.google.com/java/lang/String.html) **RMI\_SERVER\_SOCKET\_FACTORY\_ATTRIBUTE**

Name of the attribute that specifies the [RMIServerSocketFactory](http://docs.google.com/java/rmi/server/RMIServerSocketFactory.html) for the RMI objects created in conjunction with this connector. The value associated with this attribute must be of type RMIServerSocketFactory and can only be specified in the Map argument supplied when creating a connector server.

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

| **Constructor Detail** |
| --- |

### RMIConnectorServer

public **RMIConnectorServer**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Makes an RMIConnectorServer. This is equivalent to calling [RMIConnectorServer(directoryURL,environment,null,null)](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMIConnectorServer(javax.management.remote.JMXServiceURL,%20java.util.Map,%20javax.management.remote.rmi.RMIServerImpl,%20javax.management.MBeanServer))

**Parameters:**url - the URL defining how to create the connector server. Cannot be null.environment - attributes governing the creation and storing of the RMI object. Can be null, which is equivalent to an empty Map. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if url is null. [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if url does not conform to the syntax for an RMI connector, or if its protocol is not recognized by this implementation. Only "rmi" and "iiop" are valid when this constructor is used. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector server cannot be created for some reason or if it is inevitable that its [start](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start()) method will fail.

### RMIConnectorServer

public **RMIConnectorServer**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment,  
 [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbeanServer)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Makes an RMIConnectorServer for the given MBean server. This is equivalent to calling [RMIConnectorServer(directoryURL,environment,null,mbeanServer)](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#RMIConnectorServer(javax.management.remote.JMXServiceURL,%20java.util.Map,%20javax.management.remote.rmi.RMIServerImpl,%20javax.management.MBeanServer))

**Parameters:**url - the URL defining how to create the connector server. Cannot be null.environment - attributes governing the creation and storing of the RMI object. Can be null, which is equivalent to an empty Map.mbeanServer - the MBean server to which the new connector server is attached, or null if it will be attached by being registered as an MBean in the MBean server. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if url is null. [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if url does not conform to the syntax for an RMI connector, or if its protocol is not recognized by this implementation. Only "rmi" and "iiop" are valid when this constructor is used. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector server cannot be created for some reason or if it is inevitable that its [start](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start()) method will fail.

### RMIConnectorServer

public **RMIConnectorServer**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) url,  
 [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment,  
 [RMIServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html) rmiServerImpl,  
 [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbeanServer)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Makes an RMIConnectorServer for the given MBean server.

**Parameters:**url - the URL defining how to create the connector server. Cannot be null.environment - attributes governing the creation and storing of the RMI object. Can be null, which is equivalent to an empty Map.rmiServerImpl - An implementation of the RMIServer interface, consistent with the protocol type specified in url. If this parameter is non null, the protocol type specified by url is not constrained, and is assumed to be valid. Otherwise, only "rmi" and "iiop" will be recognized.mbeanServer - the MBean server to which the new connector server is attached, or null if it will be attached by being registered as an MBean in the MBean server. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if url is null. [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if url does not conform to the syntax for an RMI connector, or if its protocol is not recognized by this implementation. Only "rmi" and "iiop" are recognized when rmiServerImpl is null. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector server cannot be created for some reason or if it is inevitable that its [start](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start()) method will fail.**See Also:**[start()](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start())

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

### toJMXConnector

public [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) **toJMXConnector**([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)

Returns a client stub for this connector server. A client stub is a serializable object whose [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)) method can be used to make one new connection to this connector server.

**Specified by:**[toJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html#toJMXConnector(java.util.Map)) in interface [JMXConnectorServerMBean](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html)**Overrides:**[toJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#toJMXConnector(java.util.Map)) in class [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) **Parameters:**env - client connection parameters of the same sort that could be provided to [JMXConnector.connect(Map)](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)). Can be null, which is equivalent to an empty map. **Returns:**a client stub that can be used to make a new connection to this connector server. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this connector server does not support the generation of client stubs. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the JMXConnectorServer is not started (see [isActive()](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#isActive())). [IOException](http://docs.google.com/java/io/IOException.html) - if a communications problem means that a stub cannot be created.

### start

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

Activates the connector server, that is starts listening for client connections. Calling this method when the connector server is already active has no effect. Calling this method when the connector server has been stopped will generate an IOException.

The behavior of this method when called for the first time depends on the parameters that were supplied at construction, as described below.

First, an object of a subclass of [RMIServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html) is required, to export the connector server through RMI:

* If an RMIServerImpl was supplied to the constructor, it is used.
* Otherwise, if the protocol part of the JMXServiceURL supplied to the constructor was iiop, an object of type [RMIIIOPServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIIIOPServerImpl.html) is created.
* Otherwise, if the JMXServiceURL was null, or its protocol part was rmi, an object of type [RMIJRMPServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIJRMPServerImpl.html) is created.
* Otherwise, the implementation can create an implementation-specific [RMIServerImpl](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html) or it can throw [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html).

If the given address includes a JNDI directory URL as specified in the package documentation for [javax.management.remote.rmi](http://docs.google.com/javax/management/remote/rmi/package-summary.html), then this RMIConnectorServer will bootstrap by binding the RMIServerImpl to the given address.

If the URL path part of the JMXServiceURL was empty or a single slash (/), then the RMI object will not be bound to a directory. Instead, a reference to it will be encoded in the URL path of the RMIConnectorServer address (returned by [getAddress()](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#getAddress())). The encodings for rmi and iiop are described in the package documentation for [javax.management.remote.rmi](http://docs.google.com/javax/management/remote/rmi/package-summary.html).

The behavior when the URL path is neither empty nor a JNDI directory URL, or when the protocol is neither rmi nor iiop, is implementation defined, and may include throwing [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) when the connector server is created or when it is started.

**Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if the connector server has not been attached to an MBean server. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector server cannot be started.

### stop

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

Deactivates the connector server, that is, stops listening for client connections. Calling this method will also close all client connections that were made by this server. After this method returns, whether normally or with an exception, the connector server will not create any new client connections.

Once a connector server has been stopped, it cannot be started again.

Calling this method when the connector server has already been stopped has no effect. Calling this method when the connector server has not yet been started will disable the connector server object permanently.

If closing a client connection produces an exception, that exception is not thrown from this method. A [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html) is emitted from this MBean with the connection ID of the connection that could not be closed.

Closing a connector server is a potentially slow operation. For example, if a client machine with an open connection 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.

This method calls the method [close](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html#close()) on the connector server's RMIServerImpl object.

If the RMIServerImpl was bound to a JNDI directory by the [start](http://docs.google.com/javax/management/remote/rmi/RMIConnectorServer.html#start()) method, it is unbound from the directory by this method.

**Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if the server cannot be closed cleanly, or if the RMIServerImpl cannot be unbound from the directory. When this exception is thrown, the server has already attempted to close all client connections, if appropriate; to call [RMIServerImpl.close()](http://docs.google.com/javax/management/remote/rmi/RMIServerImpl.html#close()); and to unbind the RMIServerImpl from its directory, if appropriate. All client connections are closed except possibly those that generated exceptions when the server attempted to close them.

### isActive

public boolean **isActive**()

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

Determines whether the connector server is active. A connector server starts being active when its [start](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html#start()) method returns successfully and remains active until either its [stop](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html#stop()) method is called or the connector server fails.

**Returns:**true if the connector server is active.

### getAddress

public [JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) **getAddress**()

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

The address of this connector server.

**Returns:**the address of this connector server, or null if it does not have one.

### getAttributes

public [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> **getAttributes**()

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

The attributes for this connector server.

**Returns:**a read-only map containing the attributes for this connector server. Attributes whose values are not serializable are omitted from this map. If there are no serializable attributes, the returned map is empty.

### setMBeanServerForwarder

public void **setMBeanServerForwarder**([MBeanServerForwarder](http://docs.google.com/javax/management/remote/MBeanServerForwarder.html) mbsf)

**Description copied from interface:** [**JMXConnectorServerMBean**](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html#setMBeanServerForwarder(javax.management.remote.MBeanServerForwarder))

Adds an object that intercepts requests for the MBean server that arrive through this connector server. This object will be supplied as the MBeanServer for any new connection created by this connector server. Existing connections are unaffected.

If this connector server is already associated with an MBeanServer object, then that object is given to [mbsf.setMBeanServer](http://docs.google.com/javax/management/remote/MBeanServerForwarder.html#setMBeanServer(javax.management.MBeanServer)). If doing so produces an exception, this method throws the same exception without any other effect.

If this connector is not already associated with an MBeanServer object, or if the mbsf.setMBeanServer call just mentioned succeeds, then mbsf becomes this connector server's MBeanServer.

**Specified by:**[setMBeanServerForwarder](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html#setMBeanServerForwarder(javax.management.remote.MBeanServerForwarder)) in interface [JMXConnectorServerMBean](http://docs.google.com/javax/management/remote/JMXConnectorServerMBean.html)**Overrides:**[setMBeanServerForwarder](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#setMBeanServerForwarder(javax.management.remote.MBeanServerForwarder)) in class [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) **Parameters:**mbsf - the new MBeanServerForwarder.

### connectionOpened

protected void **connectionOpened**([String](http://docs.google.com/java/lang/String.html) connectionId,  
 [String](http://docs.google.com/java/lang/String.html) message,  
 [Object](http://docs.google.com/java/lang/Object.html) userData)

**Description copied from class:** [**JMXConnectorServer**](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionOpened(java.lang.String,%20java.lang.String,%20java.lang.Object))

Called by a subclass when a new client connection is opened. Adds connectionId to the list returned by [JMXConnectorServer.getConnectionIds()](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getConnectionIds()), then emits a [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html) with type [JMXConnectionNotification.OPENED](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html#OPENED).

**Overrides:**[connectionOpened](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionOpened(java.lang.String,%20java.lang.String,%20java.lang.Object)) in class [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) **Parameters:**connectionId - the ID of the new connection. This must be different from the ID of any connection previously opened by this connector server.message - the message for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getMessage()](http://docs.google.com/javax/management/Notification.html#getMessage()).userData - the userData for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getUserData()](http://docs.google.com/javax/management/Notification.html#getUserData()).

### connectionClosed

protected void **connectionClosed**([String](http://docs.google.com/java/lang/String.html) connectionId,  
 [String](http://docs.google.com/java/lang/String.html) message,  
 [Object](http://docs.google.com/java/lang/Object.html) userData)

**Description copied from class:** [**JMXConnectorServer**](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionClosed(java.lang.String,%20java.lang.String,%20java.lang.Object))

Called by a subclass when a client connection is closed normally. Removes connectionId from the list returned by [JMXConnectorServer.getConnectionIds()](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getConnectionIds()), then emits a [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html) with type [JMXConnectionNotification.CLOSED](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html#CLOSED).

**Overrides:**[connectionClosed](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionClosed(java.lang.String,%20java.lang.String,%20java.lang.Object)) in class [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) **Parameters:**connectionId - the ID of the closed connection.message - the message for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getMessage()](http://docs.google.com/javax/management/Notification.html#getMessage()).userData - the userData for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getUserData()](http://docs.google.com/javax/management/Notification.html#getUserData()).

### connectionFailed

protected void **connectionFailed**([String](http://docs.google.com/java/lang/String.html) connectionId,  
 [String](http://docs.google.com/java/lang/String.html) message,  
 [Object](http://docs.google.com/java/lang/Object.html) userData)

**Description copied from class:** [**JMXConnectorServer**](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionFailed(java.lang.String,%20java.lang.String,%20java.lang.Object))

Called by a subclass when a client connection fails. Removes connectionId from the list returned by [JMXConnectorServer.getConnectionIds()](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#getConnectionIds()), then emits a [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html) with type [JMXConnectionNotification.FAILED](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html#FAILED).

**Overrides:**[connectionFailed](http://docs.google.com/javax/management/remote/JMXConnectorServer.html#connectionFailed(java.lang.String,%20java.lang.String,%20java.lang.Object)) in class [JMXConnectorServer](http://docs.google.com/javax/management/remote/JMXConnectorServer.html) **Parameters:**connectionId - the ID of the failed connection.message - the message for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getMessage()](http://docs.google.com/javax/management/Notification.html#getMessage()).userData - the userData for the emitted [JMXConnectionNotification](http://docs.google.com/javax/management/remote/JMXConnectionNotification.html). Can be null. See [Notification.getUserData()](http://docs.google.com/javax/management/Notification.html#getUserData()).

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RMIConnectorServer.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/rmi/RMIConnector.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/remote/rmi/RMIIIOPServerImpl.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/remote/rmi/RMIConnectorServer.html)    [**NO FRAMES**](http://docs.google.com/RMIConnectorServer.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: [FIELD](#17dp8vu) | [CONSTR](#35nkun2) | [METHOD](#z337ya) |

[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).