| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JMXConnectorFactory.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/JMXConnector.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/remote/JMXConnectorProvider.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/remote/JMXConnectorFactory.html)    [**NO FRAMES**](http://docs.google.com/JMXConnectorFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#17dp8vu) |

## **javax.management.remote**

Class JMXConnectorFactory

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

public class **JMXConnectorFactory**extends [Object](http://docs.google.com/java/lang/Object.html)

Factory to create JMX API connector clients. There are no instances of this class.

Connections are usually made using the [connect](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#connect(javax.management.remote.JMXServiceURL)) method of this class. More advanced applications can separate the creation of the connector client, using [newJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map)) and the establishment of the connection itself, using [JMXConnector.connect(Map)](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)).

Each client is created by an instance of [JMXConnectorProvider](http://docs.google.com/javax/management/remote/JMXConnectorProvider.html). This instance is found as follows. Suppose the given [JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) looks like "service:jmx:*protocol*:*remainder*". Then the factory will attempt to find the appropriate [JMXConnectorProvider](http://docs.google.com/javax/management/remote/JMXConnectorProvider.html) for *protocol*. Each occurrence of the character + or - in *protocol* is replaced by . or \_, respectively.

A *provider package list* is searched for as follows:

1. If the environment parameter to [newJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map)) contains the key jmx.remote.protocol.provider.pkgs then the associated value is the provider package list.
2. Otherwise, if the system property jmx.remote.protocol.provider.pkgs exists, then its value is the provider package list.
3. Otherwise, there is no provider package list.

The provider package list is a string that is interpreted as a list of non-empty Java package names separated by vertical bars (|). If the string is empty, then so is the provider package list. If the provider package list is not a String, or if it contains an element that is an empty string, a [JMXProviderException](http://docs.google.com/javax/management/remote/JMXProviderException.html) is thrown.

If the provider package list exists and is not empty, then for each element *pkg* of the list, the factory will attempt to load the class

*pkg*.*protocol*.ClientProvider

If the environment parameter to [newJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map)) contains the key jmx.remote.protocol.provider.class.loader then the associated value is the class loader to use to load the provider. If the associated value is not an instance of [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html), an [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) is thrown.

If the jmx.remote.protocol.provider.class.loader key is not present in the environment parameter, the calling thread's context class loader is used.

If the attempt to load this class produces a [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html), the search for a handler continues with the next element of the list.

Otherwise, a problem with the provider found is signalled by a [JMXProviderException](http://docs.google.com/javax/management/remote/JMXProviderException.html) whose [*cause*](http://docs.google.com/javax/management/remote/JMXProviderException.html#getCause()) indicates the underlying exception, as follows:

* if the attempt to load the class produces an exception other than ClassNotFoundException, that is the *cause*;
* if [Class.newInstance()](http://docs.google.com/java/lang/Class.html#newInstance()) for the class produces an exception, that is the *cause*.

If no provider is found by the above steps, including the default case where there is no provider package list, then the implementation will use its own provider for *protocol*, or it will throw a MalformedURLException if there is none. An implementation may choose to find providers by other means. For example, it may support the  [JAR conventions for service providers](http://docs.google.com/technotes/guides/jar/jar.html#Service%20Provider), where the service interface is JMXConnectorProvider.

Every implementation must support the RMI connector protocols, specified with the string rmi or iiop.

Once a provider is found, the result of the newJMXConnector method is the result of calling [newJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorProvider.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map)) on the provider.

The Map parameter passed to the JMXConnectorProvider is a new read-only Map that contains all the entries that were in the environment parameter to [JMXConnectorFactory.newJMXConnector](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map)), if there was one. Additionally, if the jmx.remote.protocol.provider.class.loader key is not present in the environment parameter, it is added to the new read-only Map. The associated value is the calling thread's context class loader.

**Since:** 1.5

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**DEFAULT\_CLASS\_LOADER**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#DEFAULT_CLASS_LOADER)            Name of the attribute that specifies the default class loader. |
| static [String](http://docs.google.com/java/lang/String.html) | [**PROTOCOL\_PROVIDER\_CLASS\_LOADER**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#PROTOCOL_PROVIDER_CLASS_LOADER)            Name of the attribute that specifies the class loader for loading protocol providers. |
| static [String](http://docs.google.com/java/lang/String.html) | [**PROTOCOL\_PROVIDER\_PACKAGES**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#PROTOCOL_PROVIDER_PACKAGES)            Name of the attribute that specifies the provider packages that are consulted when looking for the handler for a protocol. |

| **Method Summary** | |
| --- | --- |
| static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) | [**connect**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#connect(javax.management.remote.JMXServiceURL))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL)            Creates a connection to the connector server at the given address. |
| static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) | [**connect**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#connect(javax.management.remote.JMXServiceURL,%20java.util.Map))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment)            Creates a connection to the connector server at the given address. |
| static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) | [**newJMXConnector**](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#newJMXConnector(javax.management.remote.JMXServiceURL,%20java.util.Map))([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL, [Map](http://docs.google.com/java/util/Map.html)<[String](http://docs.google.com/java/lang/String.html),?> environment)            Creates a connector client for the connector server at the given address. |

| **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** |
| --- |

### DEFAULT\_CLASS\_LOADER

public static final [String](http://docs.google.com/java/lang/String.html) **DEFAULT\_CLASS\_LOADER**

Name of the attribute that specifies the default class loader. This class loader is used to deserialize return values and exceptions from remote MBeanServerConnection calls. The value associated with this attribute is an instance of [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html).

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

### PROTOCOL\_PROVIDER\_PACKAGES

public static final [String](http://docs.google.com/java/lang/String.html) **PROTOCOL\_PROVIDER\_PACKAGES**

Name of the attribute that specifies the provider packages that are consulted when looking for the handler for a protocol. The value associated with this attribute is a string with package names separated by vertical bars (|).

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

### PROTOCOL\_PROVIDER\_CLASS\_LOADER

public static final [String](http://docs.google.com/java/lang/String.html) **PROTOCOL\_PROVIDER\_CLASS\_LOADER**

Name of the attribute that specifies the class loader for loading protocol providers. The value associated with this attribute is an instance of [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html).

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

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

### connect

public static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) **connect**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL)  
 throws [IOException](http://docs.google.com/java/io/IOException.html)

Creates a connection to the connector server at the given address.

This method is equivalent to [connect(serviceURL, null)](http://docs.google.com/javax/management/remote/JMXConnectorFactory.html#connect(javax.management.remote.JMXServiceURL,%20java.util.Map)).

**Parameters:**serviceURL - the address of the connector server to connect to. **Returns:**a JMXConnector whose [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect()) method has been called. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if serviceURL is null. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector client or the connection cannot be made because of a communication problem. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the connection cannot be made for security reasons.

### connect

public static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) **connect**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL,  
 [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)

Creates a connection to the connector server at the given address.

This method is equivalent to:

JMXConnector conn = JMXConnectorFactory.newJMXConnector(serviceURL,  
 environment);  
 conn.connect(environment);

**Parameters:**serviceURL - the address of the connector server to connect to.environment - a set of attributes to determine how the connection is made. This parameter can be null. Keys in this map must be Strings. The appropriate type of each associated value depends on the attribute. The contents of environment are not changed by this call. **Returns:**a JMXConnector representing the newly-made connection. Each successful call to this method produces a different object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if serviceURL is null. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector client or the connection cannot be made because of a communication problem. [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the connection cannot be made for security reasons.

### newJMXConnector

public static [JMXConnector](http://docs.google.com/javax/management/remote/JMXConnector.html) **newJMXConnector**([JMXServiceURL](http://docs.google.com/javax/management/remote/JMXServiceURL.html) serviceURL,  
 [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)

Creates a connector client for the connector server at the given address. The resultant client is not connected until its [connect](http://docs.google.com/javax/management/remote/JMXConnector.html#connect(java.util.Map)) method is called.

**Parameters:**serviceURL - the address of the connector server to connect to.environment - a set of attributes to determine how the connection is made. This parameter can be null. Keys in this map must be Strings. The appropriate type of each associated value depends on the attribute. The contents of environment are not changed by this call. **Returns:**a JMXConnector representing the new connector client. Each successful call to this method produces a different object. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if serviceURL is null. [IOException](http://docs.google.com/java/io/IOException.html) - if the connector client cannot be made because of a communication problem. [MalformedURLException](http://docs.google.com/java/net/MalformedURLException.html) - if there is no provider for the protocol in serviceURL. [JMXProviderException](http://docs.google.com/javax/management/remote/JMXProviderException.html) - if there is a provider for the protocol in serviceURL but it cannot be used for some reason.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JMXConnectorFactory.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/JMXConnector.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/remote/JMXConnectorProvider.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/remote/JMXConnectorFactory.html)    [**NO FRAMES**](http://docs.google.com/JMXConnectorFactory.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | CONSTR | [METHOD](#2et92p0) | DETAIL: [FIELD](#3dy6vkm) | CONSTR | [METHOD](#17dp8vu) |

[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).