| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MBeanServerInvocationHandler.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/MBeanServerFactory.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/MBeanServerNotification.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/MBeanServerInvocationHandler.html)    [**NO FRAMES**](http://docs.google.com/MBeanServerInvocationHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

## **javax.management**

Class MBeanServerInvocationHandler

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

**All Implemented Interfaces:** [InvocationHandler](http://docs.google.com/java/lang/reflect/InvocationHandler.html)

public class **MBeanServerInvocationHandler**extends [Object](http://docs.google.com/java/lang/Object.html)implements [InvocationHandler](http://docs.google.com/java/lang/reflect/InvocationHandler.html)

[InvocationHandler](http://docs.google.com/java/lang/reflect/InvocationHandler.html) that forwards methods in an MBean's management interface through the MBean server to the MBean.

Given an [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html), the [ObjectName](http://docs.google.com/javax/management/ObjectName.html) of an MBean within that MBean server, and a Java interface Intf that describes the management interface of the MBean using the patterns for a Standard MBean or an MXBean, this class can be used to construct a proxy for the MBean. The proxy implements the interface Intf such that all of its methods are forwarded through the MBean server to the MBean.

If the InvocationHandler is for an MXBean, then the parameters of a method are converted from the type declared in the MXBean interface into the corresponding mapped type, and the return value is converted from the mapped type into the declared type. For example, with the method

public List<String> reverse(List<String> list);

and given that the mapped type for List<String> is String[], a call to proxy.reverse(someList) will convert someList from a List<String> to a String[], call the MBean operation reverse, then convert the returned String[] into a List<String>.

The method Object.toString(), Object.hashCode(), or Object.equals(Object), when invoked on a proxy using this invocation handler, is forwarded to the MBean server as a method on the proxied MBean only if it appears in one of the proxy's interfaces. For a proxy created with [JMX.newMBeanProxy](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class)) or [JMX.newMXBeanProxy](http://docs.google.com/javax/management/JMX.html#newMXBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class)), this means that the method must appear in the Standard MBean or MXBean interface. Otherwise these methods have the following behavior:

* toString() returns a string representation of the proxy
* hashCode() returns a hash code for the proxy such that two equal proxies have the same hash code
* equals(Object) returns true if and only if the Object argument is of the same proxy class as this proxy, with an MBeanServerInvocationHandler that has the same MBeanServerConnection and ObjectName; if one of the MBeanServerInvocationHandlers was constructed with a Class argument then the other must have been constructed with the same Class for equals to return true.

**Since:** 1.5

| **Constructor Summary** | |
| --- | --- |
| [**MBeanServerInvocationHandler**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#MBeanServerInvocationHandler(javax.management.MBeanServerConnection,%20javax.management.ObjectName))([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName)            Invocation handler that forwards methods through an MBean server to a Standard MBean. |
| [**MBeanServerInvocationHandler**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#MBeanServerInvocationHandler(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20boolean))([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName, boolean isMXBean)            Invocation handler that can forward methods through an MBean server to a Standard MBean or MXBean. |

| **Method Summary** | |
| --- | --- |
| [MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) | [**getMBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#getMBeanServerConnection())()            The MBean server connection through which the methods of a proxy using this handler are forwarded. |
| [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getObjectName**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#getObjectName())()            The name of the MBean within the MBean server to which methods are forwarded. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**invoke**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#invoke(java.lang.Object,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) proxy, [Method](http://docs.google.com/java/lang/reflect/Method.html) method, [Object](http://docs.google.com/java/lang/Object.html)[] args)            Processes a method invocation on a proxy instance and returns the result. |
| boolean | [**isMXBean**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#isMXBean())()            If true, the proxy is for an MXBean, and appropriate mappings are applied to method parameters and return values. |
| static   | <T> T | | --- | | [**newProxyInstance**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#newProxyInstance(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class,%20boolean))([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName, [Class](http://docs.google.com/java/lang/Class.html)<T> interfaceClass, boolean notificationBroadcaster)            Return a proxy that implements the given interface by forwarding its methods through the given MBean server to the named MBean. |

| **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)) |

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

### MBeanServerInvocationHandler

public **MBeanServerInvocationHandler**([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName)

Invocation handler that forwards methods through an MBean server to a Standard MBean. This constructor may be called instead of relying on [JMX.newMBeanProxy](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class)), for instance if you need to supply a different [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) to [Proxy.newProxyInstance](http://docs.google.com/java/lang/reflect/Proxy.html#newProxyInstance(java.lang.ClassLoader,%20java.lang.Class%5B%5D,%20java.lang.reflect.InvocationHandler)).

This constructor is not appropriate for an MXBean. Use [MBeanServerInvocationHandler(MBeanServerConnection, ObjectName, boolean)](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#MBeanServerInvocationHandler(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20boolean)) for that. This constructor is equivalent to new MBeanServerInvocationHandler(connection, objectName, false).

**Parameters:**connection - the MBean server connection through which all methods of a proxy using this handler will be forwarded.objectName - the name of the MBean within the MBean server to which methods will be forwarded.

### MBeanServerInvocationHandler

public **MBeanServerInvocationHandler**([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName,  
 boolean isMXBean)

Invocation handler that can forward methods through an MBean server to a Standard MBean or MXBean. This constructor may be called instead of relying on [JMX.newMXBeanProxy](http://docs.google.com/javax/management/JMX.html#newMXBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class)), for instance if you need to supply a different [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) to [Proxy.newProxyInstance](http://docs.google.com/java/lang/reflect/Proxy.html#newProxyInstance(java.lang.ClassLoader,%20java.lang.Class%5B%5D,%20java.lang.reflect.InvocationHandler)).

**Parameters:**connection - the MBean server connection through which all methods of a proxy using this handler will be forwarded.objectName - the name of the MBean within the MBean server to which methods will be forwarded.isMXBean - if true, the proxy is for an [MXBean](http://docs.google.com/javax/management/MXBean.html), and appropriate mappings will be applied to method parameters and return values.**Since:** 1.6

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

### getMBeanServerConnection

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

The MBean server connection through which the methods of a proxy using this handler are forwarded.

**Returns:**the MBean server connection.**Since:** 1.6

### getObjectName

public [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getObjectName**()

The name of the MBean within the MBean server to which methods are forwarded.

**Returns:**the object name.**Since:** 1.6

### isMXBean

public boolean **isMXBean**()

If true, the proxy is for an MXBean, and appropriate mappings are applied to method parameters and return values.

**Returns:**whether the proxy is for an MXBean.**Since:** 1.6

### newProxyInstance

public static <T> T **newProxyInstance**([MBeanServerConnection](http://docs.google.com/javax/management/MBeanServerConnection.html) connection,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) objectName,  
 [Class](http://docs.google.com/java/lang/Class.html)<T> interfaceClass,  
 boolean notificationBroadcaster)

Return a proxy that implements the given interface by forwarding its methods through the given MBean server to the named MBean. As of 1.6, the methods [JMX.newMBeanProxy(MBeanServerConnection, ObjectName, Class)](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class)) and [JMX.newMBeanProxy(MBeanServerConnection, ObjectName, Class, boolean)](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class,%20boolean)) are preferred to this method.

This method is equivalent to [Proxy.newProxyInstance](http://docs.google.com/java/lang/reflect/Proxy.html#newProxyInstance(java.lang.ClassLoader,%20java.lang.Class%5B%5D,%20java.lang.reflect.InvocationHandler))(interfaceClass.getClassLoader(), interfaces, handler). Here handler is the result of [new MBeanServerInvocationHandler(connection, objectName)](http://docs.google.com/javax/management/MBeanServerInvocationHandler.html#MBeanServerInvocationHandler(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20boolean)), and interfaces is an array that has one element if notificationBroadcaster is false and two if it is true. The first element of interfaces is interfaceClass and the second, if present, is NotificationEmitter.class.

**Type Parameters:**T - allows the compiler to know that if the interfaceClass parameter is MyMBean.class, for example, then the return type is MyMBean.**Parameters:**connection - the MBean server to forward to.objectName - the name of the MBean within connection to forward to.interfaceClass - the management interface that the MBean exports, which will also be implemented by the returned proxy.notificationBroadcaster - make the returned proxy implement [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) by forwarding its methods via connection. A call to [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)) on the proxy will result in a call to [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)), and likewise for the other methods of [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html) and [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html). **Returns:**the new proxy instance.**See Also:**[JMX.newMBeanProxy(MBeanServerConnection, ObjectName, Class)](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class))

### invoke

public [Object](http://docs.google.com/java/lang/Object.html) **invoke**([Object](http://docs.google.com/java/lang/Object.html) proxy,  
 [Method](http://docs.google.com/java/lang/reflect/Method.html) method,  
 [Object](http://docs.google.com/java/lang/Object.html)[] args)  
 throws [Throwable](http://docs.google.com/java/lang/Throwable.html)

**Description copied from interface:** [**InvocationHandler**](http://docs.google.com/java/lang/reflect/InvocationHandler.html#invoke(java.lang.Object,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D)) Processes a method invocation on a proxy instance and returns the result. This method will be invoked on an invocation handler when a method is invoked on a proxy instance that it is associated with.

**Specified by:**[invoke](http://docs.google.com/java/lang/reflect/InvocationHandler.html#invoke(java.lang.Object,%20java.lang.reflect.Method,%20java.lang.Object%5B%5D)) in interface [InvocationHandler](http://docs.google.com/java/lang/reflect/InvocationHandler.html) **Parameters:**proxy - the proxy instance that the method was invoked onmethod - the Method instance corresponding to the interface method invoked on the proxy instance. The declaring class of the Method object will be the interface that the method was declared in, which may be a superinterface of the proxy interface that the proxy class inherits the method through.args - an array of objects containing the values of the arguments passed in the method invocation on the proxy instance, or null if interface method takes no arguments. Arguments of primitive types are wrapped in instances of the appropriate primitive wrapper class, such as java.lang.Integer or java.lang.Boolean. **Returns:**the value to return from the method invocation on the proxy instance. If the declared return type of the interface method is a primitive type, then the value returned by this method must be an instance of the corresponding primitive wrapper class; otherwise, it must be a type assignable to the declared return type. If the value returned by this method is null and the interface method's return type is primitive, then a NullPointerException will be thrown by the method invocation on the proxy instance. If the value returned by this method is otherwise not compatible with the interface method's declared return type as described above, a ClassCastException will be thrown by the method invocation on the proxy instance. **Throws:** [Throwable](http://docs.google.com/java/lang/Throwable.html) - the exception to throw from the method invocation on the proxy instance. The exception's type must be assignable either to any of the exception types declared in the throws clause of the interface method or to the unchecked exception types java.lang.RuntimeException or java.lang.Error. If a checked exception is thrown by this method that is not assignable to any of the exception types declared in the throws clause of the interface method, then an [UndeclaredThrowableException](http://docs.google.com/java/lang/reflect/UndeclaredThrowableException.html) containing the exception that was thrown by this method will be thrown by the method invocation on the proxy instance.**See Also:**[UndeclaredThrowableException](http://docs.google.com/java/lang/reflect/UndeclaredThrowableException.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MBeanServerInvocationHandler.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/MBeanServerFactory.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/MBeanServerNotification.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/MBeanServerInvocationHandler.html)    [**NO FRAMES**](http://docs.google.com/MBeanServerInvocationHandler.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#2s8eyo1) |

[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).