| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/lang/model/util/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/management/loading/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package javax.management

Provides the core classes for the Java Management Extensions.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**Descriptor**](http://docs.google.com/javax/management/Descriptor.html) | Additional metadata for a JMX element. |
| [**DescriptorAccess**](http://docs.google.com/javax/management/DescriptorAccess.html) | This interface is used to gain access to descriptors of the Descriptor class which are associated with a JMX component, i.e. |
| [**DescriptorRead**](http://docs.google.com/javax/management/DescriptorRead.html) | Interface to read the Descriptor of a management interface element such as an MBeanInfo. |
| [**DynamicMBean**](http://docs.google.com/javax/management/DynamicMBean.html) | Defines the methods that should be implemented by a Dynamic MBean (MBean that exposes a dynamic management interface). |
| [**MBeanRegistration**](http://docs.google.com/javax/management/MBeanRegistration.html) | Can be implemented by an MBean in order to carry out operations before and after being registered or unregistered from the MBean server. |
| [**MBeanServer**](http://docs.google.com/javax/management/MBeanServer.html) | This is the interface for MBean manipulation on the agent side. |
| [**MBeanServerConnection**](http://docs.google.com/javax/management/MBeanServerConnection.html) | This interface represents a way to talk to an MBean server, whether local or remote. |
| [**MBeanServerDelegateMBean**](http://docs.google.com/javax/management/MBeanServerDelegateMBean.html) | Defines the management interface of an object of class MBeanServerDelegate. |
| [**NotificationBroadcaster**](http://docs.google.com/javax/management/NotificationBroadcaster.html) | Interface implemented by an MBean that emits Notifications. |
| [**NotificationEmitter**](http://docs.google.com/javax/management/NotificationEmitter.html) | Interface implemented by an MBean that emits Notifications. |
| [**NotificationFilter**](http://docs.google.com/javax/management/NotificationFilter.html) | To be implemented by a any class acting as a notification filter. |
| [**NotificationListener**](http://docs.google.com/javax/management/NotificationListener.html) | Should be implemented by an object that wants to receive notifications. |
| [**PersistentMBean**](http://docs.google.com/javax/management/PersistentMBean.html) | This class is the interface to be implemented by MBeans that are meant to be persistent. |
| [**QueryExp**](http://docs.google.com/javax/management/QueryExp.html) | Represents relational constraints that can be used in database query "where clauses". |
| [**ValueExp**](http://docs.google.com/javax/management/ValueExp.html) | Represents values that can be passed as arguments to relational expressions. |

| **Class Summary** | |
| --- | --- |
| [**Attribute**](http://docs.google.com/javax/management/Attribute.html) | Represents an MBean attribute by associating its name with its value. |
| [**AttributeChangeNotification**](http://docs.google.com/javax/management/AttributeChangeNotification.html) | Provides definitions of the attribute change notifications sent by MBeans. |
| [**AttributeChangeNotificationFilter**](http://docs.google.com/javax/management/AttributeChangeNotificationFilter.html) | This class implements of the [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) interface for the [attribute change notification](http://docs.google.com/javax/management/AttributeChangeNotification.html). |
| [**AttributeList**](http://docs.google.com/javax/management/AttributeList.html) | Represents a list of values for attributes of an MBean. |
| [**AttributeValueExp**](http://docs.google.com/javax/management/AttributeValueExp.html) | Represents attributes used as arguments to relational constraints. |
| [**DefaultLoaderRepository**](http://docs.google.com/javax/management/DefaultLoaderRepository.html) | **Deprecated.** *Use* [*MBeanServer.getClassLoaderRepository()*](http://docs.google.com/javax/management/MBeanServer.html#getClassLoaderRepository()) *instead.* |
| [**ImmutableDescriptor**](http://docs.google.com/javax/management/ImmutableDescriptor.html) | An immutable descriptor. |
| [**JMX**](http://docs.google.com/javax/management/JMX.html) | Static methods from the JMX API. |
| [**MBeanAttributeInfo**](http://docs.google.com/javax/management/MBeanAttributeInfo.html) | Describes an MBean attribute exposed for management. |
| [**MBeanConstructorInfo**](http://docs.google.com/javax/management/MBeanConstructorInfo.html) | Describes a constructor exposed by an MBean. |
| [**MBeanFeatureInfo**](http://docs.google.com/javax/management/MBeanFeatureInfo.html) | Provides general information for an MBean descriptor object. |
| [**MBeanInfo**](http://docs.google.com/javax/management/MBeanInfo.html) | Describes the management interface exposed by an MBean; that is, the set of attributes and operations which are available for management operations. |
| [**MBeanNotificationInfo**](http://docs.google.com/javax/management/MBeanNotificationInfo.html) | The MBeanNotificationInfo class is used to describe the characteristics of the different notification instances emitted by an MBean, for a given Java class of notification. |
| [**MBeanOperationInfo**](http://docs.google.com/javax/management/MBeanOperationInfo.html) | Describes a management operation exposed by an MBean. |
| [**MBeanParameterInfo**](http://docs.google.com/javax/management/MBeanParameterInfo.html) | Describes an argument of an operation exposed by an MBean. |
| [**MBeanPermission**](http://docs.google.com/javax/management/MBeanPermission.html) | Permission controlling access to MBeanServer operations. |
| [**MBeanServerBuilder**](http://docs.google.com/javax/management/MBeanServerBuilder.html) | This class represents a builder that creates a default [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) implementation. |
| [**MBeanServerDelegate**](http://docs.google.com/javax/management/MBeanServerDelegate.html) | Represents the MBean server from the management point of view. |
| [**MBeanServerFactory**](http://docs.google.com/javax/management/MBeanServerFactory.html) | Provides MBean server references. |
| [**MBeanServerInvocationHandler**](http://docs.google.com/javax/management/MBeanServerInvocationHandler.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. |
| [**MBeanServerNotification**](http://docs.google.com/javax/management/MBeanServerNotification.html) | Represents a notification emitted by the MBean server through the MBeanServerDelegate MBean. |
| [**MBeanServerPermission**](http://docs.google.com/javax/management/MBeanServerPermission.html) | A Permission to perform actions related to MBeanServers. |
| [**MBeanTrustPermission**](http://docs.google.com/javax/management/MBeanTrustPermission.html) | This permission represents "trust" in a signer or codebase. |
| [**Notification**](http://docs.google.com/javax/management/Notification.html) | The Notification class represents a notification emitted by an MBean. |
| [**NotificationBroadcasterSupport**](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html) | Provides an implementation of [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) interface. |
| [**NotificationFilterSupport**](http://docs.google.com/javax/management/NotificationFilterSupport.html) | Provides an implementation of the [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) interface. |
| [**ObjectInstance**](http://docs.google.com/javax/management/ObjectInstance.html) | Used to represent the object name of an MBean and its class name. |
| [**ObjectName**](http://docs.google.com/javax/management/ObjectName.html) | Represents the object name of an MBean, or a pattern that can match the names of several MBeans. |
| [**Query**](http://docs.google.com/javax/management/Query.html) | Constructs query object constraints. |
| [**QueryEval**](http://docs.google.com/javax/management/QueryEval.html) | Allows a query to be performed in the context of a specific MBean server. |
| [**StandardEmitterMBean**](http://docs.google.com/javax/management/StandardEmitterMBean.html) | An MBean whose management interface is determined by reflection on a Java interface, and that emits notifications. |
| [**StandardMBean**](http://docs.google.com/javax/management/StandardMBean.html) | An MBean whose management interface is determined by reflection on a Java interface. |
| [**StringValueExp**](http://docs.google.com/javax/management/StringValueExp.html) | Represents strings that are arguments to relational constraints. |

| **Exception Summary** | |
| --- | --- |
| [**AttributeNotFoundException**](http://docs.google.com/javax/management/AttributeNotFoundException.html) | The specified attribute does not exist or cannot be retrieved. |
| [**BadAttributeValueExpException**](http://docs.google.com/javax/management/BadAttributeValueExpException.html) | Thrown when an invalid MBean attribute is passed to a query constructing method. |
| [**BadBinaryOpValueExpException**](http://docs.google.com/javax/management/BadBinaryOpValueExpException.html) | Thrown when an invalid expression is passed to a method for constructing a query. |
| [**BadStringOperationException**](http://docs.google.com/javax/management/BadStringOperationException.html) | Thrown when an invalid string operation is passed to a method for constructing a query. |
| [**InstanceAlreadyExistsException**](http://docs.google.com/javax/management/InstanceAlreadyExistsException.html) | The MBean is already registered in the repository. |
| [**InstanceNotFoundException**](http://docs.google.com/javax/management/InstanceNotFoundException.html) | The specified MBean does not exist in the repository. |
| [**IntrospectionException**](http://docs.google.com/javax/management/IntrospectionException.html) | An exception occurred during the introspection of an MBean. |
| [**InvalidApplicationException**](http://docs.google.com/javax/management/InvalidApplicationException.html) | Thrown when an attempt is made to apply either of the following: A subquery expression to an MBean or a qualified attribute expression to an MBean of the wrong class. |
| [**InvalidAttributeValueException**](http://docs.google.com/javax/management/InvalidAttributeValueException.html) | The value specified is not valid for the attribute. |
| [**JMException**](http://docs.google.com/javax/management/JMException.html) | Exceptions thrown by JMX implementations. |
| [**JMRuntimeException**](http://docs.google.com/javax/management/JMRuntimeException.html) | Runtime exceptions emitted by JMX implementations. |
| [**ListenerNotFoundException**](http://docs.google.com/javax/management/ListenerNotFoundException.html) | The specified MBean listener does not exist in the repository. |
| [**MalformedObjectNameException**](http://docs.google.com/javax/management/MalformedObjectNameException.html) | The format of the string does not correspond to a valid ObjectName. |
| [**MBeanException**](http://docs.google.com/javax/management/MBeanException.html) | Represents "user defined" exceptions thrown by MBean methods in the agent. |
| [**MBeanRegistrationException**](http://docs.google.com/javax/management/MBeanRegistrationException.html) | Wraps exceptions thrown by the preRegister(), preDeregister() methods of the MBeanRegistration interface. |
| [**NotCompliantMBeanException**](http://docs.google.com/javax/management/NotCompliantMBeanException.html) | Exception which occurs when trying to register an object in the MBean server that is not a JMX compliant MBean. |
| [**OperationsException**](http://docs.google.com/javax/management/OperationsException.html) | Represents exceptions thrown in the MBean server when performing operations on MBeans. |
| [**ReflectionException**](http://docs.google.com/javax/management/ReflectionException.html) | Represents exceptions thrown in the MBean server when using the java.lang.reflect classes to invoke methods on MBeans. |
| [**RuntimeErrorException**](http://docs.google.com/javax/management/RuntimeErrorException.html) | When a java.lang.Error occurs in the agent it should be caught and re-thrown as a RuntimeErrorException. |
| [**RuntimeMBeanException**](http://docs.google.com/javax/management/RuntimeMBeanException.html) | Represents runtime exceptions thrown by MBean methods in the agent. |
| [**RuntimeOperationsException**](http://docs.google.com/javax/management/RuntimeOperationsException.html) | Represents runtime exceptions thrown in the agent when performing operations on MBeans. |
| [**ServiceNotFoundException**](http://docs.google.com/javax/management/ServiceNotFoundException.html) | Represents exceptions raised when a requested service is not supported. |

| **Annotation Types Summary** | |
| --- | --- |
| [**DescriptorKey**](http://docs.google.com/javax/management/DescriptorKey.html) | Meta-annotation that describes how an annotation element relates to a field in a [Descriptor](http://docs.google.com/javax/management/Descriptor.html). |
| [**MXBean**](http://docs.google.com/javax/management/MXBean.html) | Annotation to mark an interface explicitly as being an MXBean interface, or as not being an MXBean interface. |

## Package javax.management Description

Provides the core classes for the Java Management Extensions.

The Java Management Extensions (JMXTM) API is a standard API for management and monitoring. Typical uses include:

* consulting and changing application configuration
* accumulating statistics about application behavior and making them available
* notifying of state changes and erroneous conditions.

The JMX API can also be used as part of a solution for managing systems, networks, and so on.

The API includes remote access, so a remote management program can interact with a running application for these purposes.

## MBeans

The fundamental notion of the JMX API is the *MBean*. An MBean is a named *managed object* representing a resource. It has a *management interface* consisting of:

* named and typed attributes that can be read and/or written
* named and typed operations that can be invoked
* typed notifications that can be emitted by the MBean.

For example, an MBean representing an application's configuration could have attributes representing the different configuration items. Reading the CacheSize attribute would return the current value of that item. Writing it would update the item, potentially changing the behavior of the running application. An operation such as save could store the current configuration persistently. A notification such as ConfigurationChangedNotification could be sent every time the configuration is changed.

In the standard usage of the JMX API, MBeans are implemented as Java objects. However, as explained below, these objects are not usually referenced directly.

### Standard MBeans

To make MBean implementation simple, the JMX API includes the notion of *Standard MBeans*. A Standard MBean is one whose attributes and operations are deduced from a Java interface using certain naming patterns, similar to those used by JavaBeansTM. For example, consider an interface like this:

public interface ConfigurationMBean {  
 public int getCacheSize();  
 public void setCacheSize(int size);  
 public long getLastChangedTime();  
 public void save();  
 }

The methods getCacheSize and setCacheSize define a read-write attribute of type int called CacheSize (with an initial capital, unlike the JavaBeans convention).

The method getLastChangedTime defines an attribute of type long called LastChangedTime. This is a read-only attribute, since there is no method setLastChangedTime.

The method save defines an operation called save. It is not an attribute, since its name does not begin with get, set, or is.

The exact naming patterns for Standard MBeans are detailed in the [JMX Specification](#1t3h5sf).

There are two ways to make a Java object that is an MBean with this management interface. One is for the object to be of a class that has exactly the same name as the Java interface but without the MBean suffix. So in the example the object would be of the class Configuration, in the same Java package as ConfigurationMBean. The second way is to use the [StandardMBean](http://docs.google.com/javax/management/StandardMBean.html) class.

### MXBeans

An *MXBean* is a variant of Standard MBean where complex types are mapped to a standard set of types defined in the [javax.management.openmbean](http://docs.google.com/javax/management/openmbean/package-summary.html) package. MXBeans are appropriate if you would otherwise need to reference application-specific classes in your MBean interface. They are described in detail in the specification for [MXBean](http://docs.google.com/javax/management/MXBean.html).

### Dynamic MBeans

A *Dynamic MBean* is an MBean that defines its management interface at run-time. For example, a configuration MBean could determine the names and types of the attributes it exposes by parsing an XML file.

Any Java object of a class that implements the [DynamicMBean](http://docs.google.com/javax/management/DynamicMBean.html) interface is a Dynamic MBean.

### Open MBeans

An *Open MBean* is a kind of Dynamic MBean where the types of attributes and of operation parameters and return values are built using a small set of predefined Java classes. Open MBeans facilitate operation with remote management programs that do not necessarily have access to application-specific types, including non-Java programs. Open MBeans are defined by the package  [javax.management.openmbean](http://docs.google.com/openmbean/package-summary.html).

### Model MBeans

A *Model MBean* is a kind of Dynamic MBean that acts as a bridge between the management interface and the underlying managed resource. Both the management interface and the managed resource are specified as Java objects. The same Model MBean implementation can be reused many times with different management interfaces and managed resources, and it can provide common functionality such as persistence and caching. Model MBeans are defined by the package  [javax.management.modelmbean](http://docs.google.com/modelmbean/package-summary.html).

## MBean Server

To be useful, an MBean must be registered in an *MBean Server*. An MBean Server is a repository of MBeans. Usually the only access to the MBeans is through the MBean Server. In other words, code no longer accesses the Java object implementing the MBean directly, but instead accesses the MBean by name through the MBean Server. Each MBean has a unique name within the MBean Server, defined by the [ObjectName](http://docs.google.com/javax/management/ObjectName.html) class.

An MBean Server is an object implementing the interface [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html). The most convenient MBean Server to use is the *Platform MBean Server*. This is a single MBean Server that can be shared by different managed components running within the same Java Virtual Machine. The Platform MBean Server is accessed with the method [ManagementFactory.getPlatformMBeanServer()](http://docs.google.com/java/lang/management/ManagementFactory.html#getPlatformMBeanServer()).

Application code can also create a new MBean Server, or access already-created MBean Servers, using the [MBeanServerFactory](http://docs.google.com/javax/management/MBeanServerFactory.html) class.

### Creating MBeans in the MBean Server

There are two ways to create an MBean. One is to construct a Java object that will be the MBean, then use the [registerMBean](http://docs.google.com/javax/management/MBeanServer.html#registerMBean(java.lang.Object,%20javax.management.ObjectName)) method to register it in the MBean Server. The other is to create and register the MBean in a single operation using one of the [createMBean](http://docs.google.com/javax/management/MBeanServer.html#createMBean(java.lang.String,%20javax.management.ObjectName)) methods.

The registerMBean method is simpler for local use, but cannot be used remotely. The createMBean method can be used remotely, but sometimes requires attention to class loading issues.

An MBean can perform actions when it is registered in or unregistered from an MBean Server if it implements the [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) interface.

### Accessing MBeans in the MBean Server

Given an ObjectName name and an MBeanServer mbs, you can access attributes and operations as in this example:

int cacheSize = mbs.getAttribute(name, "CacheSize");  
 [Attribute](http://docs.google.com/javax/management/Attribute.html) newCacheSize =  
 new Attribute("CacheSize", new Integer(2000));  
 mbs.setAttribute(name, newCacheSize);  
 mbs.invoke(name, "save", new Object[0], new Class[0]);

Alternatively, if you have a Java interface that corresponds to the management interface for the MBean, you can use an *MBean proxy* like this:

ConfigurationMBean conf =  
 [JMX.newMBeanProxy](http://docs.google.com/javax/management/JMX.html#newMBeanProxy(javax.management.MBeanServerConnection,%20javax.management.ObjectName,%20java.lang.Class))(mbs, name, ConfigurationMBean.class);  
 int cacheSize = conf.getCacheSize();  
 conf.setCacheSize(2000);  
 conf.save();

Using an MBean proxy is just a convenience. The second example ends up calling the same MBeanServer operations as the first one.

An MBean Server can be queried for MBeans whose names match certain patterns and/or whose attributes meet certain constraints. Name patterns are constructed using the [ObjectName](http://docs.google.com/javax/management/ObjectName.html) class and constraints are constructed using the [Query](http://docs.google.com/javax/management/Query.html) class. The methods [queryNames](http://docs.google.com/javax/management/MBeanServer.html#queryNames(javax.management.ObjectName,%20javax.management.QueryExp)) and [queryMBeans](http://docs.google.com/javax/management/MBeanServer.html#queryMBeans(javax.management.ObjectName,%20javax.management.QueryExp)) then perform the query.

## Notifications

A *notification* is an instance of the [Notification](http://docs.google.com/javax/management/Notification.html) class or a subclass. In addition to its Java class, it has a *type* string that can distinguish it from other notifications of the same class.

An MBean that will emit notifications must implement the [NotificationBroadcaster](http://docs.google.com/javax/management/NotificationBroadcaster.html) or [NotificationEmitter](http://docs.google.com/javax/management/NotificationEmitter.html) interface. Usually, it does this by subclassing [NotificationBroadcasterSupport](http://docs.google.com/javax/management/NotificationBroadcasterSupport.html) or by delegating to an instance of that class.

Notifications can be received by a *listener*, which is an object that implements the [NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) interface. You can add a listener to an MBean with the method [MBeanServer.addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServer.html#addNotificationListener(javax.management.ObjectName,%20javax.management.NotificationListener,%20javax.management.NotificationFilter,%20java.lang.Object)). You can optionally supply a *filter* to this method, to select only notifications of interest. A filter is an object that implements the [NotificationFilter](http://docs.google.com/javax/management/NotificationFilter.html) interface.

An MBean can be a listener for notifications emitted by other MBeans in the same MBean Server. In this case, it implements [NotificationListener](http://docs.google.com/javax/management/NotificationListener.html) and the method [MBeanServer.addNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)](http://docs.google.com/javax/management/MBeanServer.html#addNotificationListener(javax.management.ObjectName,%20javax.management.ObjectName,%20javax.management.NotificationFilter,%20java.lang.Object)) is used to listen.

## Remote Access to MBeans

An MBean Server can be accessed remotely through a *connector*. A connector allows a remote Java application to access an MBean Server in essentially the same way as a local one. The package  [javax.management.remote](http://docs.google.com/remote/package-summary.html) defines connectors.

The JMX specification also defines the notion of an *adaptor*. An adaptor translates between requests in a protocol such as SNMP or HTML and accesses to an MBean Server. So for example an SNMP GET operation might result in a getAttribute on the MBean Server.

**Since:** 1.5 **See Also:** [Java SE 6 Platform documentation on JMX technology](http://docs.google.com/technotes/guides/jmx/index.html) in particular the  [JMX Specification, version 1.4(pdf).](http://docs.google.com/technotes/guides/jmx/JMX_1_4_specification.pdf)

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/javax/lang/model/util/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/javax/management/loading/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).