| | [**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/java/lang/instrument/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/lang/ref/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/management/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.lang.management

Provides the management interface for monitoring and management of the Java virtual machine as well as the operating system on which the Java virtual machine is running.

**See:**

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

| **Interface Summary** | |
| --- | --- |
| [**ClassLoadingMXBean**](http://docs.google.com/java/lang/management/ClassLoadingMXBean.html) | The management interface for the class loading system of the Java virtual machine. |
| [**CompilationMXBean**](http://docs.google.com/java/lang/management/CompilationMXBean.html) | The management interface for the compilation system of the Java virtual machine. |
| [**GarbageCollectorMXBean**](http://docs.google.com/java/lang/management/GarbageCollectorMXBean.html) | The management interface for the garbage collection of the Java virtual machine. |
| [**MemoryManagerMXBean**](http://docs.google.com/java/lang/management/MemoryManagerMXBean.html) | The management interface for a memory manager. |
| [**MemoryMXBean**](http://docs.google.com/java/lang/management/MemoryMXBean.html) | The management interface for the memory system of the Java virtual machine. |
| [**MemoryPoolMXBean**](http://docs.google.com/java/lang/management/MemoryPoolMXBean.html) | The management interface for a memory pool. |
| [**OperatingSystemMXBean**](http://docs.google.com/java/lang/management/OperatingSystemMXBean.html) | The management interface for the operating system on which the Java virtual machine is running. |
| [**RuntimeMXBean**](http://docs.google.com/java/lang/management/RuntimeMXBean.html) | The management interface for the runtime system of the Java virtual machine. |
| [**ThreadMXBean**](http://docs.google.com/java/lang/management/ThreadMXBean.html) | The management interface for the thread system of the Java virtual machine. |

| **Class Summary** | |
| --- | --- |
| [**LockInfo**](http://docs.google.com/java/lang/management/LockInfo.html) | Information about a *lock*. |
| [**ManagementFactory**](http://docs.google.com/java/lang/management/ManagementFactory.html) | The ManagementFactory class is a factory class for getting managed beans for the Java platform. |
| [**ManagementPermission**](http://docs.google.com/java/lang/management/ManagementPermission.html) | The permission which the SecurityManager will check when code that is running with a SecurityManager calls methods defined in the management interface for the Java platform. |
| [**MemoryNotificationInfo**](http://docs.google.com/java/lang/management/MemoryNotificationInfo.html) | The information about a memory notification. |
| [**MemoryUsage**](http://docs.google.com/java/lang/management/MemoryUsage.html) | A MemoryUsage object represents a snapshot of memory usage. |
| [**MonitorInfo**](http://docs.google.com/java/lang/management/MonitorInfo.html) | Information about an object monitor lock. |
| [**ThreadInfo**](http://docs.google.com/java/lang/management/ThreadInfo.html) | Thread information. |

| **Enum Summary** | |
| --- | --- |
| [**MemoryType**](http://docs.google.com/java/lang/management/MemoryType.html) | Types of [memory pools](http://docs.google.com/java/lang/management/MemoryPoolMXBean.html). |

## Package java.lang.management Description

Provides the management interface for monitoring and management of the Java virtual machine as well as the operating system on which the Java virtual machine is running. It allows both local and remote monitoring and management of the running Java virtual machine.

#### Platform MXBeans

This package defines the management interface of the following components:

| Management Interface | Description |
| --- | --- |
| [ClassLoadingMXBean](http://docs.google.com/java/lang/management/ClassLoadingMXBean.html) | Class loading system of the Java virtual machine. |
| [CompilationMXBean](http://docs.google.com/java/lang/management/CompilationMXBean.html) | Compilation system of the Java virtual machine. |
| [MemoryMXBean](http://docs.google.com/java/lang/management/MemoryMXBean.html) | Memory system of the Java virtual machine. |
| [ThreadMXBean](http://docs.google.com/java/lang/management/ThreadMXBean.html) | Threads system of the Java virtual machine. |
| [RuntimeMXBean](http://docs.google.com/java/lang/management/RuntimeMXBean.html) | Runtime system of the Java virtual machine. |
| [OperatingSystemMXBean](http://docs.google.com/java/lang/management/OperatingSystemMXBean.html) | Operating system on which the Java virtual machine is running. |
| [GarbageCollectorMXBean](http://docs.google.com/java/lang/management/GarbageCollectorMXBean.html) | Garbage collector in the Java virtual machine. |
| [MemoryManagerMXBean](http://docs.google.com/java/lang/management/MemoryManagerMXBean.html) | Memory manager in the Java virtual machine. |
| [MemoryPoolMXBean](http://docs.google.com/java/lang/management/MemoryPoolMXBean.html) | Memory pool in the Java virtual machine. |

A platform MXBean is a *managed bean* that defines the management interface for one component for the platform and is specified in the  [ManagementFactory](http://docs.google.com/ManagementFactory.html#MXBean) class.

An application can monitor the instrumentation of the Java virtual machine and manage certain characteristics in the following ways:

* *Direct access to an MXBean interface*
  1. Get the MXBean instance through the static factory method and access the MXBean interface locally of the running virtual machine.
  2. Construct an MXBean proxy instance that forwards the method calls to a given [MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) by calling [ManagementFactory.newPlatformMXBeanProxy](http://docs.google.com/java/lang/management/ManagementFactory.html#newPlatformMXBeanProxy(javax.management.MBeanServerConnection,%20java.lang.String,%20java.lang.Class)). A proxy is typically constructed to remotely access an MXBean of another running virtual machine.
* *Indirect access via* [*MBeanServer*](http://docs.google.com/javax/management/MBeanServer.html) *interface*
  1. Go through the [platform MBeanServer](http://docs.google.com/java/lang/management/ManagementFactory.html#getPlatformMBeanServer()) to access MXBeans locally or a specific MBeanServerConnection to access MXBeans remotely. The attributes and operations of an MXBean use only *JMX open types* which include basic data types, [CompositeData](http://docs.google.com/javax/management/openmbean/CompositeData.html), and [TabularData](http://docs.google.com/javax/management/openmbean/TabularData.html) defined in [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html).

Below shows a few [examples](#_2et92p0) of different ways to access MXBeans.

#### ManagementFactory

The [ManagementFactory](http://docs.google.com/java/lang/management/ManagementFactory.html) class is the management factory class for the Java platform. This class provides a set of static factory methods to obtain the MXBeans for the Java platform to allow an application to access the MXBeans directly.

A *platform MBeanServer* can be accessed with the [getPlatformMBeanServer](http://docs.google.com/java/lang/management/ManagementFactory.html#getPlatformMBeanServer()) method. On the first call to this method, it creates the platform MBeanServer and registers all platform MXBeans including platform MXBeans defined in other packages such as [LoggingMXBean](http://docs.google.com/java/util/logging/LoggingMXBean.html). Each platform MXBean is registered with a unique name defined in the [ManagementFactory](http://docs.google.com/java/lang/management/ManagementFactory.html) class for constructing [ObjectName](http://docs.google.com/javax/management/ObjectName.html). This is a single MBeanServer that can be shared by different managed components running within the same Java virtual machine.

#### Interoperability

A management application and a platform MBeanServer of a running virtual machine can interoperate without requiring classes used by the platform MXBean interfaces. The data types being transmitted between the JMX connector server and the connector client are JMX [open types](http://docs.google.com/javax/management/openmbean/OpenType.html) and this allows interoperation across versions.

A data type used by the MXBean interfaces are mapped to an open type when being accessed via MBeanServer interface. The data type mapping is specified in the [ManagementFactory](http://docs.google.com/java/lang/management/ManagementFactory.html) class.

#### Ways to Access MXBeans

There are three different ways to access the management interfaces.

1. Call the methods in the MXBean directly within the same Java virtual machine. RuntimeMXBean mxbean = ManagementFactory.getRuntimeMXBean();  
     
    // Get the standard attribute "VmVendor"  
    String vendor = mxbean.getVmVendor();
2. Go through a MBeanServerConnection connecting to the platform MBeanServer of a running virtual machine.

MBeanServerConnection mbs;  
  
 // Connect to a running JVM (or itself) and get MBeanServerConnection  
 // that has the JVM MXBeans registered in it  
 ...  
  
 try {  
 // Assuming the RuntimeMXBean has been registered in mbs  
 ObjectName oname = new ObjectName(ManagementFactory.RUNTIME\_MXBEAN\_NAME);  
   
 // Get standard attribute "VmVendor"  
 String vendor = (String) mbs.getAttribute(oname, "VmVendor");  
 } catch (....) {  
 // Catch the exceptions thrown by ObjectName constructor  
 // and MBeanServer.getAttribute method  
 ...  
 }

1. Use MXBean proxy.

MBeanServerConnection mbs;  
  
 // Connect to a running JVM (or itself) and get MBeanServerConnection  
 // that has the JVM MBeans registered in it  
 ...  
  
 // Get a MBean proxy for RuntimeMXBean interface  
 RuntimeMXBean proxy =   
 ManagementFactory.newPlatformMXBeanProxy(mbs,  
 ManagementFactory.RUNTIME\_MXBEAN\_NAME,  
 RuntimeMXBean.class);  
 // Get standard attribute "VmVendor"   
 String vendor = proxy.getVmVendor();

#### Platform Extension

A Java virtual machine implementation may add its platform extension to the management interface by defining platform-dependent interfaces that extend the standard management interfaces to include platform-specific metrics and management operations. The static factory methods in the ManagementFactory class will return the MBeans with the platform extension.

It is recommended to name the platform-specific attributes with a vendor-specific prefix such as the vendor's name to avoid collisions of the attribute name between the future extension to the standard management interface and the platform extension. If the future extension to the standard management interface defines a new attribute for a management interface and the attribute name is happened to be same as some vendor-specific attribute's name, the applications accessing that vendor-specific attribute would have to be modified to cope with versioning and compatibility issues.

Below is an example showing how to access a platform-specific attribute from Sun's implementation of the RuntimeMXBean.

1) Direct access to the Sun-specific MXBean interface

com.sun.management.RuntimeMXBean mxbean =   
 (com.sun.management.RuntimeMXBean) ManagementFactory.getRuntimeMXBean();  
  
 // Get the standard attribute "VmVendor"  
 String vendor = mxbean.getVmVendor();  
  
 // Get the platform-specific attribute "Bar"  
 BarType bar = mxbean.getBar();

2) Access the Sun-specific MXBean interface via MBeanServer

MBeanServerConnection mbs;  
  
 // Connect to a running JVM (or itself) and get MBeanServerConnection  
 // that has the JVM MXBeans registered in it  
 ...  
  
 try {  
 // Assuming the RuntimeMXBean has been registered in mbs  
 ObjectName oname = new ObjectName(ManagementFactory.RUNTIME\_MXBEAN\_NAME);  
   
 // Get standard attribute "VmVendor"  
 String vendor = (String) mbs.getAttribute(oname, "VmVendor");  
  
 // Check if this MXBean contains Sun's extension  
 if (mbs.isInstanceOf(oname, "com.sun.management.RuntimeMXBean")) {  
 // Get platform-specific attribute "Bar"  
 BarType bar = (String) mbs.getAttribute(oname, "Bar");  
 }  
 } catch (....) {  
 // Catch the exceptions thrown by ObjectName constructor  
 // and MBeanServer methods  
 ...  
 }

Unless otherwise noted, passing a null argument to a constructor or method in any class or interface in this package will cause a [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) to be thrown.

The java.lang.management API is thread-safe.

**Since:** 1.5 **See Also:** [JMX Specification.](http://docs.google.com/javax/management/package-summary.html)

| | [**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/java/lang/instrument/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/lang/ref/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/lang/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).