| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MLet.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/loading/DefaultLoaderRepository.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/loading/MLetContent.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/loading/MLet.html)    [**NO FRAMES**](http://docs.google.com/MLet.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#17dp8vu) | [METHOD](#z337ya) |

## **javax.management.loading**

Class MLet

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.lang.ClassLoader](http://docs.google.com/java/lang/ClassLoader.html)  
 [java.security.SecureClassLoader](http://docs.google.com/java/security/SecureClassLoader.html)  
 [java.net.URLClassLoader](http://docs.google.com/java/net/URLClassLoader.html)  
 **javax.management.loading.MLet**

**All Implemented Interfaces:** [Externalizable](http://docs.google.com/java/io/Externalizable.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html), [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Direct Known Subclasses:** [PrivateMLet](http://docs.google.com/javax/management/loading/PrivateMLet.html)

public class **MLet**extends [URLClassLoader](http://docs.google.com/java/net/URLClassLoader.html)implements [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html), [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html), [Externalizable](http://docs.google.com/java/io/Externalizable.html)

Allows you to instantiate and register one or several MBeans in the MBean server coming from a remote URL. M-let is a shortcut for management applet. The m-let service does this by loading an m-let text file, which specifies information on the MBeans to be obtained. The information on each MBean is specified in a single instance of a tag, called the MLET tag. The location of the m-let text file is specified by a URL.

The MLET tag has the following syntax:

<MLET

CODE = class | OBJECT = serfile

ARCHIVE = "archiveList"

[CODEBASE = codebaseURL]

[NAME = mbeanname]

[VERSION = version]

>

[arglist]

</MLET>

where:

CODE = class This attribute specifies the full Java class name, including package name, of the MBean to be obtained. The compiled .class file of the MBean must be contained in one of the .jar files specified by the ARCHIVE attribute. Either CODE or OBJECT must be present. OBJECT = serfile This attribute specifies the .ser file that contains a serialized representation of the MBean to be obtained. This file must be contained in one of the .jar files specified by the ARCHIVE attribute. If the .jar file contains a directory hierarchy, specify the path of the file within this hierarchy. Otherwise a match will not be found. Either CODE or OBJECT must be present. ARCHIVE = "archiveList" This mandatory attribute specifies one or more .jar files containing MBeans or other resources used by the MBean to be obtained. One of the .jar files must contain the file specified by the CODE or OBJECT attribute. If archivelist contains more than one file:

* Each file must be separated from the one that follows it by a comma (,).
* archivelist must be enclosed in double quote marks.

All .jar files in archivelist must be stored in the directory specified by the code base URL. CODEBASE = codebaseURL This optional attribute specifies the code base URL of the MBean to be obtained. It identifies the directory that contains the .jar files specified by the ARCHIVE attribute. Specify this attribute only if the .jar files are not in the same directory as the m-let text file. If this attribute is not specified, the base URL of the m-let text file is used. NAME = mbeanname This optional attribute specifies the object name to be assigned to the MBean instance when the m-let service registers it. If mbeanname starts with the colon character (:), the domain part of the object name is the default domain of the MBean server, as returned by [MBeanServer.getDefaultDomain()](http://docs.google.com/javax/management/MBeanServer.html#getDefaultDomain()). VERSION = version This optional attribute specifies the version number of the MBean and associated .jar files to be obtained. This version number can be used to specify that the .jar files are loaded from the server to update those stored locally in the cache the next time the m-let text file is loaded. version must be a series of non-negative decimal integers each separated by a period from the one that precedes it. arglist This optional attribute specifies a list of one or more parameters for the MBean to be instantiated. This list describes the parameters to be passed the MBean's constructor. Use the following syntax to specify each item in arglist:

<ARG TYPE=argumentType VALUE=value>

where:

* argumentType is the type of the argument that will be passed as parameter to the MBean's constructor.

The arguments' type in the argument list should be a Java primitive type or a Java basic type (java.lang.Boolean, java.lang.Byte, java.lang.Short, java.lang.Long, java.lang.Integer, java.lang.Float, java.lang.Double, java.lang.String).

When an m-let text file is loaded, an instance of each MBean specified in the file is created and registered.

The m-let service extends the java.net.URLClassLoader and can be used to load remote classes and jar files in the VM of the agent.

**Note -**  The MLet class loader uses the [MBeanServerFactory.getClassLoaderRepository(javax.management.MBeanServer)](http://docs.google.com/javax/management/MBeanServerFactory.html#getClassLoaderRepository(javax.management.MBeanServer)) to load classes that could not be found in the loaded jar files.

**Since:** 1.5 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.management.loading.MLet)

| **Constructor Summary** | |
| --- | --- |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet())()            Constructs a new MLet using the default delegation parent ClassLoader. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D))([URL](http://docs.google.com/java/net/URL.html)[] urls)            Constructs a new MLet for the specified URLs using the default delegation parent ClassLoader. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D,%20boolean))([URL](http://docs.google.com/java/net/URL.html)[] urls, boolean delegateToCLR)            Constructs a new MLet for the specified URLs using the default delegation parent ClassLoader. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D,%20java.lang.ClassLoader))([URL](http://docs.google.com/java/net/URL.html)[] urls, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent)            Constructs a new MLet for the given URLs. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D,%20java.lang.ClassLoader,%20boolean))([URL](http://docs.google.com/java/net/URL.html)[] urls, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent, boolean delegateToCLR)            Constructs a new MLet for the given URLs. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D,%20java.lang.ClassLoader,%20java.net.URLStreamHandlerFactory))([URL](http://docs.google.com/java/net/URL.html)[] urls, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent, [URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) factory)            Constructs a new MLet for the specified URLs, parent class loader, and URLStreamHandlerFactory. |
| [**MLet**](http://docs.google.com/javax/management/loading/MLet.html#MLet(java.net.URL%5B%5D,%20java.lang.ClassLoader,%20java.net.URLStreamHandlerFactory,%20boolean))([URL](http://docs.google.com/java/net/URL.html)[] urls, [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent, [URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) factory, boolean delegateToCLR)            Constructs a new MLet for the specified URLs, parent class loader, and URLStreamHandlerFactory. |

| **Method Summary** | |
| --- | --- |
| void | [**addURL**](http://docs.google.com/javax/management/loading/MLet.html#addURL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Appends the specified URL to the list of URLs to search for classes and resources. |
| void | [**addURL**](http://docs.google.com/javax/management/loading/MLet.html#addURL(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) url)            Appends the specified URL to the list of URLs to search for classes and resources. |
| protected  [URL](http://docs.google.com/java/net/URL.html) | [**check**](http://docs.google.com/javax/management/loading/MLet.html#check(java.lang.String,%20java.net.URL,%20java.lang.String,%20javax.management.loading.MLetContent))([String](http://docs.google.com/java/lang/String.html) version, [URL](http://docs.google.com/java/net/URL.html) codebase, [String](http://docs.google.com/java/lang/String.html) jarfile, [MLetContent](http://docs.google.com/javax/management/loading/MLetContent.html) mlet)            This method is to be overridden when extending this service to support caching and versioning. |
| protected  [Class](http://docs.google.com/java/lang/Class.html)<?> | [**findClass**](http://docs.google.com/javax/management/loading/MLet.html#findClass(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            This is the main method for class loaders that is being redefined. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**findLibrary**](http://docs.google.com/javax/management/loading/MLet.html#findLibrary(java.lang.String))([String](http://docs.google.com/java/lang/String.html) libname)            Returns the absolute path name of a native library. |
| [String](http://docs.google.com/java/lang/String.html) | [**getLibraryDirectory**](http://docs.google.com/javax/management/loading/MLet.html#getLibraryDirectory())()            Gets the current directory used by the library loader for storing native libraries before they are loaded into memory. |
| [Set](http://docs.google.com/java/util/Set.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**getMBeansFromURL**](http://docs.google.com/javax/management/loading/MLet.html#getMBeansFromURL(java.lang.String))([String](http://docs.google.com/java/lang/String.html) url)            Loads a text file containing MLET tags that define the MBeans to be added to the MBean server. |
| [Set](http://docs.google.com/java/util/Set.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**getMBeansFromURL**](http://docs.google.com/javax/management/loading/MLet.html#getMBeansFromURL(java.net.URL))([URL](http://docs.google.com/java/net/URL.html) url)            Loads a text file containing MLET tags that define the MBeans to be added to the MBean server. |
| [URL](http://docs.google.com/java/net/URL.html)[] | [**getURLs**](http://docs.google.com/javax/management/loading/MLet.html#getURLs())()            Returns the search path of URLs for loading classes and resources. |
| [Class](http://docs.google.com/java/lang/Class.html)<?> | [**loadClass**](http://docs.google.com/javax/management/loading/MLet.html#loadClass(java.lang.String,%20javax.management.loading.ClassLoaderRepository))([String](http://docs.google.com/java/lang/String.html) name, [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) clr)            Load a class, using the given [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) if the class is not found in this MLet's URLs. |
| void | [**postDeregister**](http://docs.google.com/javax/management/loading/MLet.html#postDeregister())()            Allows the m-let to perform any operations needed after having been unregistered in the MBean server. |
| void | [**postRegister**](http://docs.google.com/javax/management/loading/MLet.html#postRegister(java.lang.Boolean))([Boolean](http://docs.google.com/java/lang/Boolean.html) registrationDone)            Allows the m-let to perform any operations needed after having been registered in the MBean server or after the registration has failed. |
| void | [**preDeregister**](http://docs.google.com/javax/management/loading/MLet.html#preDeregister())()            Allows the m-let to perform any operations it needs before being unregistered by the MBean server. |
| [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**preRegister**](http://docs.google.com/javax/management/loading/MLet.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName))([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) server, [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Allows the m-let to perform any operations it needs before being registered in the MBean server. |
| void | [**readExternal**](http://docs.google.com/javax/management/loading/MLet.html#readExternal(java.io.ObjectInput))([ObjectInput](http://docs.google.com/java/io/ObjectInput.html) in)            Restore this MLet's contents from the given [ObjectInput](http://docs.google.com/java/io/ObjectInput.html). |
| void | [**setLibraryDirectory**](http://docs.google.com/javax/management/loading/MLet.html#setLibraryDirectory(java.lang.String))([String](http://docs.google.com/java/lang/String.html) libdir)            Sets the directory used by the library loader for storing native libraries before they are loaded into memory. |
| void | [**writeExternal**](http://docs.google.com/javax/management/loading/MLet.html#writeExternal(java.io.ObjectOutput))([ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html) out)            Save this MLet's contents to the given [ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html). |

| **Methods inherited from class java.net.**[**URLClassLoader**](http://docs.google.com/java/net/URLClassLoader.html) |
| --- |
| [definePackage](http://docs.google.com/java/net/URLClassLoader.html#definePackage(java.lang.String,%20java.util.jar.Manifest,%20java.net.URL)), [findResource](http://docs.google.com/java/net/URLClassLoader.html#findResource(java.lang.String)), [findResources](http://docs.google.com/java/net/URLClassLoader.html#findResources(java.lang.String)), [getPermissions](http://docs.google.com/java/net/URLClassLoader.html#getPermissions(java.security.CodeSource)), [newInstance](http://docs.google.com/java/net/URLClassLoader.html#newInstance(java.net.URL%5B%5D)), [newInstance](http://docs.google.com/java/net/URLClassLoader.html#newInstance(java.net.URL%5B%5D,%20java.lang.ClassLoader)) |

| **Methods inherited from class java.security.**[**SecureClassLoader**](http://docs.google.com/java/security/SecureClassLoader.html) |
| --- |
| [defineClass](http://docs.google.com/java/security/SecureClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.CodeSource)), [defineClass](http://docs.google.com/java/security/SecureClassLoader.html#defineClass(java.lang.String,%20java.nio.ByteBuffer,%20java.security.CodeSource)) |

| **Methods inherited from class java.lang.**[**ClassLoader**](http://docs.google.com/java/lang/ClassLoader.html) |
| --- |
| [clearAssertionStatus](http://docs.google.com/java/lang/ClassLoader.html#clearAssertionStatus()), [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(byte%5B%5D,%20int,%20int)), [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int)), [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20byte%5B%5D,%20int,%20int,%20java.security.ProtectionDomain)), [defineClass](http://docs.google.com/java/lang/ClassLoader.html#defineClass(java.lang.String,%20java.nio.ByteBuffer,%20java.security.ProtectionDomain)), [definePackage](http://docs.google.com/java/lang/ClassLoader.html#definePackage(java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.lang.String,%20java.net.URL)), [findLoadedClass](http://docs.google.com/java/lang/ClassLoader.html#findLoadedClass(java.lang.String)), [findSystemClass](http://docs.google.com/java/lang/ClassLoader.html#findSystemClass(java.lang.String)), [getPackage](http://docs.google.com/java/lang/ClassLoader.html#getPackage(java.lang.String)), [getPackages](http://docs.google.com/java/lang/ClassLoader.html#getPackages()), [getParent](http://docs.google.com/java/lang/ClassLoader.html#getParent()), [getResource](http://docs.google.com/java/lang/ClassLoader.html#getResource(java.lang.String)), [getResourceAsStream](http://docs.google.com/java/lang/ClassLoader.html#getResourceAsStream(java.lang.String)), [getResources](http://docs.google.com/java/lang/ClassLoader.html#getResources(java.lang.String)), [getSystemClassLoader](http://docs.google.com/java/lang/ClassLoader.html#getSystemClassLoader()), [getSystemResource](http://docs.google.com/java/lang/ClassLoader.html#getSystemResource(java.lang.String)), [getSystemResourceAsStream](http://docs.google.com/java/lang/ClassLoader.html#getSystemResourceAsStream(java.lang.String)), [getSystemResources](http://docs.google.com/java/lang/ClassLoader.html#getSystemResources(java.lang.String)), [loadClass](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String)), [loadClass](http://docs.google.com/java/lang/ClassLoader.html#loadClass(java.lang.String,%20boolean)), [resolveClass](http://docs.google.com/java/lang/ClassLoader.html#resolveClass(java.lang.Class)), [setClassAssertionStatus](http://docs.google.com/java/lang/ClassLoader.html#setClassAssertionStatus(java.lang.String,%20boolean)), [setDefaultAssertionStatus](http://docs.google.com/java/lang/ClassLoader.html#setDefaultAssertionStatus(boolean)), [setPackageAssertionStatus](http://docs.google.com/java/lang/ClassLoader.html#setPackageAssertionStatus(java.lang.String,%20boolean)), [setSigners](http://docs.google.com/java/lang/ClassLoader.html#setSigners(java.lang.Class,%20java.lang.Object%5B%5D)) |

| **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)) |

| **Methods inherited from interface javax.management.loading.**[**MLetMBean**](http://docs.google.com/javax/management/loading/MLetMBean.html) |
| --- |
| [getResource](http://docs.google.com/javax/management/loading/MLetMBean.html#getResource(java.lang.String)), [getResourceAsStream](http://docs.google.com/javax/management/loading/MLetMBean.html#getResourceAsStream(java.lang.String)), [getResources](http://docs.google.com/javax/management/loading/MLetMBean.html#getResources(java.lang.String)) |

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

### MLet

public **MLet**()

Constructs a new MLet using the default delegation parent ClassLoader.

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls)

Constructs a new MLet for the specified URLs using the default delegation parent ClassLoader. The URLs will be searched in the order specified for classes and resources after first searching in the parent class loader.

**Parameters:**urls - The URLs from which to load classes and resources.

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent)

Constructs a new MLet for the given URLs. The URLs will be searched in the order specified for classes and resources after first searching in the specified parent class loader. The parent argument will be used as the parent class loader for delegation.

**Parameters:**urls - The URLs from which to load classes and resources.parent - The parent class loader for delegation.

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent,  
 [URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) factory)

Constructs a new MLet for the specified URLs, parent class loader, and URLStreamHandlerFactory. The parent argument will be used as the parent class loader for delegation. The factory argument will be used as the stream handler factory to obtain protocol handlers when creating new URLs.

**Parameters:**urls - The URLs from which to load classes and resources.parent - The parent class loader for delegation.factory - The URLStreamHandlerFactory to use when creating URLs.

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls,  
 boolean delegateToCLR)

Constructs a new MLet for the specified URLs using the default delegation parent ClassLoader. The URLs will be searched in the order specified for classes and resources after first searching in the parent class loader.

**Parameters:**urls - The URLs from which to load classes and resources.delegateToCLR - True if, when a class is not found in either the parent ClassLoader or the URLs, the MLet should delegate to its containing MBeanServer's [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html).

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent,  
 boolean delegateToCLR)

Constructs a new MLet for the given URLs. The URLs will be searched in the order specified for classes and resources after first searching in the specified parent class loader. The parent argument will be used as the parent class loader for delegation.

**Parameters:**urls - The URLs from which to load classes and resources.parent - The parent class loader for delegation.delegateToCLR - True if, when a class is not found in either the parent ClassLoader or the URLs, the MLet should delegate to its containing MBeanServer's [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html).

### MLet

public **MLet**([URL](http://docs.google.com/java/net/URL.html)[] urls,  
 [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) parent,  
 [URLStreamHandlerFactory](http://docs.google.com/java/net/URLStreamHandlerFactory.html) factory,  
 boolean delegateToCLR)

Constructs a new MLet for the specified URLs, parent class loader, and URLStreamHandlerFactory. The parent argument will be used as the parent class loader for delegation. The factory argument will be used as the stream handler factory to obtain protocol handlers when creating new URLs.

**Parameters:**urls - The URLs from which to load classes and resources.parent - The parent class loader for delegation.factory - The URLStreamHandlerFactory to use when creating URLs.delegateToCLR - True if, when a class is not found in either the parent ClassLoader or the URLs, the MLet should delegate to its containing MBeanServer's [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html).

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

### addURL

public void **addURL**([URL](http://docs.google.com/java/net/URL.html) url)

Appends the specified URL to the list of URLs to search for classes and resources.

**Specified by:**[addURL](http://docs.google.com/javax/management/loading/MLetMBean.html#addURL(java.net.URL)) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html)**Overrides:**[addURL](http://docs.google.com/java/net/URLClassLoader.html#addURL(java.net.URL)) in class [URLClassLoader](http://docs.google.com/java/net/URLClassLoader.html) **Parameters:**url - the URL to be added to the search path of URLs

### addURL

public void **addURL**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html)

Appends the specified URL to the list of URLs to search for classes and resources.

**Specified by:**[addURL](http://docs.google.com/javax/management/loading/MLetMBean.html#addURL(java.lang.String)) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html) **Parameters:**url - the URL to add. **Throws:** [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) - The specified URL is malformed.

### getURLs

public [URL](http://docs.google.com/java/net/URL.html)[] **getURLs**()

Returns the search path of URLs for loading classes and resources. This includes the original list of URLs specified to the constructor, along with any URLs subsequently appended by the addURL() method.

**Specified by:**[getURLs](http://docs.google.com/javax/management/loading/MLetMBean.html#getURLs()) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html)**Overrides:**[getURLs](http://docs.google.com/java/net/URLClassLoader.html#getURLs()) in class [URLClassLoader](http://docs.google.com/java/net/URLClassLoader.html) **Returns:**the search path of URLs for loading classes and resources.

### getMBeansFromURL

public [Set](http://docs.google.com/java/util/Set.html)<[Object](http://docs.google.com/java/lang/Object.html)> **getMBeansFromURL**([URL](http://docs.google.com/java/net/URL.html) url)  
 throws [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html)

Loads a text file containing MLET tags that define the MBeans to be added to the MBean server. The location of the text file is specified by a URL. The MBeans specified in the MLET file will be instantiated and registered in the MBean server.

**Specified by:**[getMBeansFromURL](http://docs.google.com/javax/management/loading/MLetMBean.html#getMBeansFromURL(java.net.URL)) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html) **Parameters:**url - The URL of the text file to be loaded as URL object. **Returns:**A set containing one entry per MLET tag in the m-let text file loaded. Each entry specifies either the ObjectInstance for the created MBean, or a throwable object (that is, an error or an exception) if the MBean could not be created. **Throws:** [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) - One of the following errors has occurred: The m-let text file does not contain an MLET tag, the m-let text file is not found, a mandatory attribute of the MLET tag is not specified, the value of url is null. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - MLet MBean is not registered with an MBeanServer.

### getMBeansFromURL

public [Set](http://docs.google.com/java/util/Set.html)<[Object](http://docs.google.com/java/lang/Object.html)> **getMBeansFromURL**([String](http://docs.google.com/java/lang/String.html) url)  
 throws [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html)

Loads a text file containing MLET tags that define the MBeans to be added to the MBean server. The location of the text file is specified by a URL. The MBeans specified in the MLET file will be instantiated and registered in the MBean server.

**Specified by:**[getMBeansFromURL](http://docs.google.com/javax/management/loading/MLetMBean.html#getMBeansFromURL(java.lang.String)) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html) **Parameters:**url - The URL of the text file to be loaded as String object. **Returns:**A set containing one entry per MLET tag in the m-let text file loaded. Each entry specifies either the ObjectInstance for the created MBean, or a throwable object (that is, an error or an exception) if the MBean could not be created. **Throws:** [ServiceNotFoundException](http://docs.google.com/javax/management/ServiceNotFoundException.html) - One of the following errors has occurred: The m-let text file does not contain an MLET tag, the m-let text file is not found, a mandatory attribute of the MLET tag is not specified, the url is malformed. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - MLet MBean is not registered with an MBeanServer.

### getLibraryDirectory

public [String](http://docs.google.com/java/lang/String.html) **getLibraryDirectory**()

Gets the current directory used by the library loader for storing native libraries before they are loaded into memory.

**Specified by:**[getLibraryDirectory](http://docs.google.com/javax/management/loading/MLetMBean.html#getLibraryDirectory()) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html) **Returns:**The current directory used by the library loader. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this implementation does not support storing native libraries in this way.**See Also:**[setLibraryDirectory(java.lang.String)](http://docs.google.com/javax/management/loading/MLet.html#setLibraryDirectory(java.lang.String))

### setLibraryDirectory

public void **setLibraryDirectory**([String](http://docs.google.com/java/lang/String.html) libdir)

Sets the directory used by the library loader for storing native libraries before they are loaded into memory.

**Specified by:**[setLibraryDirectory](http://docs.google.com/javax/management/loading/MLetMBean.html#setLibraryDirectory(java.lang.String)) in interface [MLetMBean](http://docs.google.com/javax/management/loading/MLetMBean.html) **Parameters:**libdir - The directory used by the library loader. **Throws:** [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this implementation does not support storing native libraries in this way.**See Also:**[getLibraryDirectory()](http://docs.google.com/javax/management/loading/MLet.html#getLibraryDirectory())

### preRegister

public [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **preRegister**([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) server,  
 [ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Allows the m-let to perform any operations it needs before being registered in the MBean server. If the ObjectName is null, the m-let provides a default name for its registration <defaultDomain>:type=MLet

**Specified by:**[preRegister](http://docs.google.com/javax/management/MBeanRegistration.html#preRegister(javax.management.MBeanServer,%20javax.management.ObjectName)) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Parameters:**server - The MBean server in which the m-let will be registered.name - The object name of the m-let. **Returns:**The name of the m-let registered. **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - This exception should be caught by the MBean server and re-thrown as an MBeanRegistrationException.

### postRegister

public void **postRegister**([Boolean](http://docs.google.com/java/lang/Boolean.html) registrationDone)

Allows the m-let to perform any operations needed after having been registered in the MBean server or after the registration has failed.

**Specified by:**[postRegister](http://docs.google.com/javax/management/MBeanRegistration.html#postRegister(java.lang.Boolean)) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Parameters:**registrationDone - Indicates whether or not the m-let has been successfully registered in the MBean server. The value false means that either the registration phase has failed.

### preDeregister

public void **preDeregister**()  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

Allows the m-let to perform any operations it needs before being unregistered by the MBean server.

**Specified by:**[preDeregister](http://docs.google.com/javax/management/MBeanRegistration.html#preDeregister()) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html) **Throws:** java.langException - This exception should be caught by the MBean server and re-thrown as an MBeanRegistrationException. [Exception](http://docs.google.com/java/lang/Exception.html) - This exception will be caught by the MBean server and re-thrown as an [MBeanRegistrationException](http://docs.google.com/javax/management/MBeanRegistrationException.html).

### postDeregister

public void **postDeregister**()

Allows the m-let to perform any operations needed after having been unregistered in the MBean server.

**Specified by:**[postDeregister](http://docs.google.com/javax/management/MBeanRegistration.html#postDeregister()) in interface [MBeanRegistration](http://docs.google.com/javax/management/MBeanRegistration.html)

### writeExternal

public void **writeExternal**([ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html) out)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html)

Save this MLet's contents to the given [ObjectOutput](http://docs.google.com/java/io/ObjectOutput.html). Not all implementations support this method. Those that do not throw [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html). A subclass may override this method to support it or to change the format of the written data.

The format of the written data is not specified, but if an implementation supports [writeExternal(java.io.ObjectOutput)](http://docs.google.com/javax/management/loading/MLet.html#writeExternal(java.io.ObjectOutput)) it must also support [readExternal(java.io.ObjectInput)](http://docs.google.com/javax/management/loading/MLet.html#readExternal(java.io.ObjectInput)) in such a way that what is written by the former can be read by the latter.

**Specified by:**[writeExternal](http://docs.google.com/java/io/Externalizable.html#writeExternal(java.io.ObjectOutput)) in interface [Externalizable](http://docs.google.com/java/io/Externalizable.html) **Parameters:**out - The object output stream to write to. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - If a problem occurred while writing. [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - If this implementation does not support this operation.

### readExternal

public void **readExternal**([ObjectInput](http://docs.google.com/java/io/ObjectInput.html) in)  
 throws [IOException](http://docs.google.com/java/io/IOException.html),  
 [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html),  
 [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html)

Restore this MLet's contents from the given [ObjectInput](http://docs.google.com/java/io/ObjectInput.html). Not all implementations support this method. Those that do not throw [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html). A subclass may override this method to support it or to change the format of the read data.

The format of the read data is not specified, but if an implementation supports [readExternal(java.io.ObjectInput)](http://docs.google.com/javax/management/loading/MLet.html#readExternal(java.io.ObjectInput)) it must also support [writeExternal(java.io.ObjectOutput)](http://docs.google.com/javax/management/loading/MLet.html#writeExternal(java.io.ObjectOutput)) in such a way that what is written by the latter can be read by the former.

**Specified by:**[readExternal](http://docs.google.com/java/io/Externalizable.html#readExternal(java.io.ObjectInput)) in interface [Externalizable](http://docs.google.com/java/io/Externalizable.html) **Parameters:**in - The object input stream to read from. **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if a problem occurred while reading. [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - if the class for the object being restored cannot be found. [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) - if this implementation does not support this operation.

### loadClass

public [Class](http://docs.google.com/java/lang/Class.html)<?> **loadClass**([String](http://docs.google.com/java/lang/String.html) name,  
 [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) clr)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

Load a class, using the given [ClassLoaderRepository](http://docs.google.com/javax/management/loading/ClassLoaderRepository.html) if the class is not found in this MLet's URLs. The given ClassLoaderRepository can be null, in which case a [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) occurs immediately if the class is not found in this MLet's URLs.

**Parameters:**name - The name of the class we want to load.clr - The ClassLoaderRepository that will be used to search for the given class, if it is not found in this ClassLoader. May be null. **Returns:**The resulting Class object. **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - The specified class could not be found in this ClassLoader nor in the given ClassLoaderRepository.

### findClass

protected [Class](http://docs.google.com/java/lang/Class.html)<?> **findClass**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html)

This is the main method for class loaders that is being redefined.

**Overrides:**[findClass](http://docs.google.com/java/net/URLClassLoader.html#findClass(java.lang.String)) in class [URLClassLoader](http://docs.google.com/java/net/URLClassLoader.html) **Parameters:**name - The name of the class. **Returns:**The resulting Class object. **Throws:** [ClassNotFoundException](http://docs.google.com/java/lang/ClassNotFoundException.html) - The specified class could not be found.

### findLibrary

protected [String](http://docs.google.com/java/lang/String.html) **findLibrary**([String](http://docs.google.com/java/lang/String.html) libname)

Returns the absolute path name of a native library. The VM invokes this method to locate the native libraries that belong to classes loaded with this class loader. Libraries are searched in the JAR files using first just the native library name and if not found the native library name together with the architecture-specific path name (OSName/OSArch/OSVersion/lib/nativelibname), i.e.

the library stat on Solaris SPARC 5.7 will be searched in the JAR file as:

1. libstat.so
2. SunOS/sparc/5.7/lib/libstat.so

the library stat on Windows NT 4.0 will be searched in the JAR file as:

1. stat.dll
2. WindowsNT/x86/4.0/lib/stat.dll

More specifically, let *nativelibname* be the result of [System.mapLibraryName](http://docs.google.com/java/lang/System.html#mapLibraryName(java.lang.String))(libname). Then the following names are searched in the JAR files, in order:

*nativelibname*

<os.name>/<os.arch>/<os.version>/lib/*nativelibname*

where <X> means System.getProperty(X) with any spaces in the result removed, and / stands for the file separator character ([File.separator](http://docs.google.com/java/io/File.html#separator)).

If this method returns null, i.e. the libraries were not found in any of the JAR files loaded with this class loader, the VM searches the library along the path specified as the java.library.path property.

**Overrides:**[findLibrary](http://docs.google.com/java/lang/ClassLoader.html#findLibrary(java.lang.String)) in class [ClassLoader](http://docs.google.com/java/lang/ClassLoader.html) **Parameters:**libname - The library name. **Returns:**The absolute path of the native library.**See Also:**[System.loadLibrary(String)](http://docs.google.com/java/lang/System.html#loadLibrary(java.lang.String)), [System.mapLibraryName(String)](http://docs.google.com/java/lang/System.html#mapLibraryName(java.lang.String))

### check

protected [URL](http://docs.google.com/java/net/URL.html) **check**([String](http://docs.google.com/java/lang/String.html) version,  
 [URL](http://docs.google.com/java/net/URL.html) codebase,  
 [String](http://docs.google.com/java/lang/String.html) jarfile,  
 [MLetContent](http://docs.google.com/javax/management/loading/MLetContent.html) mlet)  
 throws [Exception](http://docs.google.com/java/lang/Exception.html)

This method is to be overridden when extending this service to support caching and versioning. It is called from [getMBeansFromURL](http://docs.google.com/javax/management/loading/MLet.html#getMBeansFromURL(java.net.URL)) when the version, codebase, and jarfile have been extracted from the MLet file, and can be used to verify that it is all right to load the given MBean, or to replace the given URL with a different one.

The default implementation of this method returns codebase unchanged.

**Parameters:**version - The version number of the .jar file stored locally.codebase - The base URL of the remote .jar file.jarfile - The name of the .jar file to be loaded.mlet - The MLetContent instance that represents the MLET tag. **Returns:**the codebase to use for the loaded MBean. The returned value should not be null. **Throws:** [Exception](http://docs.google.com/java/lang/Exception.html) - if the MBean is not to be loaded for some reason. The exception will be added to the set returned by [getMBeansFromURL](http://docs.google.com/javax/management/loading/MLet.html#getMBeansFromURL(java.net.URL)).

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