Java™ Management Extensions API

Reference Implementation (Early Access)

Version 2.0

901 San Antonio Road, Palo Alto, CA 94303, U.S.A.

Contents

Java Management Extensions API

Package Summary

Packages

`com.sun.management`	Provides specific classes to **Sun JMX Reference Implementation**.
`javax.management`	Provides the **core JMX classes**.
`javax.management.loading`	Provides the classes which implement **the advanced dynamic loading**.
`javax.management.monitor`	Provides the definition of the **monitor classes**.
`javax.management.timer`	Provides the definition of the **Timer MBean**.

package

com.sun.management

Description

Provides specific classes to **Sun JMX Reference Implementation**.

Class Summary

Classes

`Enumerated`	This class is used for implementing enumerated values.
`Introspector`	This class contains the methods for performing all the tests needed to verify that a class represents a JMX compliant MBean.
`ServiceName`	This class is used for storing the names of core services.
`Trace`	This class is a static class and is used to get JMX internal runtime information.
`TraceFilter`	This class is used to filter trace information.
`TraceListener`	An object of this class can be used to receive notifications sent out by the class Trace, and all notifications received will be saved to a file specified, or write to user screen.
`TraceNotification`	This class defines an object used by the class Trace to send out all internal runtime information.

com.sun.management
Enumerated

Syntax

```
public abstract class Enumerated extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--com.sun.management.Enumerated
```

All Implemented Interfaces: java.io.Serializable

Description

This class is used for implementing enumerated values. An enumeration is represented by a class derived from Enumerated. The derived class defines what are the permitted values in the enumeration. An enumerated value is represented by an instance of the derived class. It can be represented : - as an integer - as a string

Member Summary

Fields

`int`	`value`

Constructors

- `Enumerated()`
- `Enumerated(int)`
- `Enumerated(Integer)`
- `Enumerated(String)`

Methods

`boolean`	`equals(Object)`
`Hashtable`	`getIntTable()`
`Hashtable`	`getStringTable()`
`int`	`hashCode()`
`int`	`intValue()`
`String`	`toString()`
`Enumeration`	`valueIndexes()`
`Enumeration`	`valueStrings()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Fields

---

value

`protected int value`

This variable keeps the integer form of the enumerated. The string form is retreived using getIntTable().

Constructors

Enumerated()

`public  Enumerated()`

Construct an enumerated with a default value. The default value is the first available in getIntTable().

Throws: `IllegalArgumentException` - One of the arguments passed to the method is illegal or inappropriate.

Enumerated(int)

`public  Enumerated(int valueIndex)`

Construct an enumerated from its integer form.

Parameters:
`<VAR>valueIndex</VAR>` - The integer form.

Throws: `IllegalArgumentException` - One of the arguments passed to the method is illegal or inappropriate.

Enumerated(Integer)

`public  Enumerated(java.lang.Integer valueIndex)`

Construct an enumerated from its Integer form.

Parameters:
`<VAR>valueIndex</VAR>` - The Integer form.

Throws: `IllegalArgumentException` - One of the arguments passed to the method is illegal or inappropriate.

Enumerated(String)

`public  Enumerated(java.lang.String valueString)`

Construct an enumerated from its string form.

Parameters:
`<VAR>valueString</VAR>` - The string form.

Throws: `IllegalArgumentException` - One of the arguments passed to the method is illegal or inappropriate.

Methods

---

equals(Object)

`public boolean equals(java.lang.Object obj)`

Compares this enumerated to the specified enumerated. The result is true if and only if the argument is not null and is of the same class.

Overrides: java.lang.Object.equals(java.lang.Object) in class java.lang.Object

Parameters:
`<VAR>obj</VAR>` - The object to compare with.

Returns: True if this and obj are the same; false otherwise

---

getIntTable()

`protected abstract java.util.Hashtable getIntTable()`

Returns the hashtable of the integer forms. getIntTable().get(x) returns the string form associated to the integer x. This method must be implemented by the derived class.

Returns: An hashtable for read-only purpose

---

getStringTable()

`protected abstract java.util.Hashtable getStringTable()`

Returns the hashtable of the string forms. getStringTable().get(s) returns the integer form associated to the string s. This method must be implemented by the derived class.

Returns: An hashtable for read-only purpose

---

hashCode()

`public int hashCode()`

Returns the hash code for this enumerated.

Overrides: java.lang.Object.hashCode() in class java.lang.Object

Returns: A hash code value for this object.

---

intValue()

`public int intValue()`

Return the integer form of the enumerated.

Returns: The integer form

---

toString()

`public java.lang.String toString()`

Returns the string form of this enumerated.

Overrides: java.lang.Object.toString() in class java.lang.Object

Returns: The string for for this object.

---

valueIndexes()

`public java.util.Enumeration valueIndexes()`

Returns an Java enumeration of the permitted integers.

Returns: An enumeration of Integer instances

---

valueStrings()

`public java.util.Enumeration valueStrings()`

Returns an Java enumeration of the permitted strings.

Returns: An enumeration of String instances

com.sun.management
Introspector

Syntax

```
public class Introspector extends java.lang.Object
```

```
java.lang.Object
  |
  +--com.sun.management.Introspector
```

Description

This class contains the methods for performing all the tests needed to verify that a class represents a JMX compliant MBean.

Member Summary	
Constructors	
	`Introspector()`
Methods	
`Class`	`getMBeanInterface(Class)`
`MBeanInfo`	`testCompliance(Class)`

Inherited Member Summary
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Introspector()

```
public  Introspector()
```

Methods

---

getMBeanInterface(Class)

`public static java.lang.Class getMBeanInterface(java.lang.Class c)`

Basic method for testing if a given class is a dynamic MBean

Parameters:

`c` - The class to be tested

Throws: `NotCompliantMBeanException` - The specified class is not a JMX compliant MBean

---

testCompliance(Class)

`public static MBeanInfo testCompliance(java.lang.Class c)`

Basic method for testing if a given class is a JMX compliant MBean

Parameters:

`c` - The class to be tested

Throws: `NotCompliantMBeanException` - The specified class is not a JMX compliant MBean

com.sun.management
ServiceName

Syntax

```
public class ServiceName extends java.lang.Object
```

```
java.lang.Object
  |
  +--com.sun.management.ServiceName
```

Description

This class is used for storing the names of core services.

Member Summary

Fields		
	String	DELEGATE
	String	DOMAIN
	String	MLET
Constructors		
		ServiceName()

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

DELEGATE

```
public static final java.lang.String DELEGATE
```

The default key properties for the delegate object The value is `type=MBeanServerDelegate`.

DOMAIN

```
public static final java.lang.String DOMAIN
```

The default domain. The value is `DefaultDomain`.

MLET

```
public static final java.lang.String MLET
```

The class name used for registering the class loader of the MLet service. The value is `type=MLet`.

Constructors

ServiceName()

```
public  ServiceName()
```

com.sun.management
Trace

Syntax

```
public class Trace extends java.lang.Object
```

```
java.lang.Object
  |
  +--com.sun.management.Trace
```

Description

This class is a static class and is used to get JMX internal runtime information.

Two levels of information can be specified:

1) trace. It provides information to tell what JMX is doing, this information is used to help a user to develop a JMX application.

2) debug. It provides information to help a JMX developper to diagnose the JMX implementation.

A user can specify information types to select information. The following types are specified:

1) INFO_CONNECTOR_HTML. Information from HTML connector.

2) INFO_MBEANSERVER. Information from a MBean Server.

3) INFO_MLET. Information from a MLet service.

4) INFO_MONITOR. Information from a Monitor.

5) INFO_TIMER. Information from a Timer.

6) INFO_MISC. Information from all other classes except those specified types.

7) INFO_ALL. Information from all classes.

This class uses notification machanism to distribute the information. To register a listener, a TraceFilter object should be provided to do filtering. There are two ways to receive trace information:

- adding a listener with a filter in the code. It is possible to have more than one listeners but with different filters.
- specifying system properties in the command to start the Java interpreter when you run a class. In this case, the code of the class must include a call to the `parseTraceProperties()` method of this class.

To specifiy trace level, add -DLEVEL_TRACE or -DLEVEL_DEBUG to you command line. By default, the level is set to LEVEL_TRACE.

To specifiy trace types, add the selected types to the command line as: -DINFO_MLET -DINFO_TIMER to select mlet and timer information, it is possible to add more than one type. By default, the type is set to INFO_ALL.

To specify output, add -DTRACE_OUTPUT=name_of_outfile to the command line, name_of_outfile is a file name to write message. By default, the message is written to screen.

If at least of the three properties is specified in the command, all other which are not specified will be set to default value.

Member Summary

Fields

`int`	`INFO_ALL`
`int`	`INFO_CONNECTOR_HTML`
`int`	`INFO_MBEANSERVER`
`int`	`INFO_MISC`
`int`	`INFO_MLET`
`int`	`INFO_MONITOR`
`int`	`INFO_TIMER`
`int`	`LEVEL_DEBUG`
`int`	`LEVEL_TRACE`
`String`	`UNKOWNTYPE`

Constructors

	`Trace()`

Methods

`void`	`addNotificationListener(NotificationListener, NotificationFilter, Object)`
`void`	`addNotificationListener(TraceListener, Object)`
`String`	`getRIType(int)`
`boolean`	`isSelected(int, int)`
`void`	`parseTraceProperties()`
`void`	`removeAllListeners()`
`void`	`removeNotificationListener(NotificationListener)`
`boolean`	`send(int, int, String, String, String)`
`boolean`	`send(int, int, String, String, Throwable)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

INFO_ALL

```
public static int INFO_ALL
```

Information type defined to represent all types defined.

INFO_CONNECTOR_HTML

```
public static final int INFO_CONNECTOR_HTML
```

Information type defined for HTML connector information.

INFO_MBEANSERVER

`public static final int INFO_MBEANSERVER`

Information type defined for MBean Server information.

INFO_MISC

`public static final int INFO_MISC`

Information type defined for all other classes.

INFO_MLET

`public static final int INFO_MLET`

Information type defined for MLet service information.

INFO_MONITOR

`public static final int INFO_MONITOR`

Information type defined for Monitor information.

INFO_TIMER

`public static final int INFO_TIMER`

Information type defined for Timer information.

LEVEL_DEBUG

`public static final int LEVEL_DEBUG`

Information level defined for debug level. The formation will be provided to help diagnosy. Selecting this level will result to select the LEVEL_TRACE too.

LEVEL_TRACE

`public static final int LEVEL_TRACE`

Information level defined for trace level. The formation will be provided to help development of JDMK applications.

UNKOWNTYPE

`protected static final java.lang.String UNKOWNTYPE`

Constructors

Trace()

```
public  Trace()
```

Methods

addNotificationListener(NotificationListener, NotificationFilter, Object)

```
public static void addNotificationListener(NotificationListener listener,
            NotificationFilter f, java.lang.Object handback)
```

Add a listener with filtering conditions

Parameters:

`listener` - the listener to be added to receive trace notification.

`f` - the filter used to select a trace notification. This filter should be an instance of the class TraceFilter or its sub-class. If the filter is set to null all trace information with any type and any level will be send to the listener.

`handback` - the object sent back to listener.

Throws: `IllegalArgumentException` - Thrown if the specified filter is not an instance of the class TraceFilter or its sub-classes, or no listener is specified.

addNotificationListener(TraceListener, Object)

```
public static void addNotificationListener(TraceListener listener, java.lang.Object
            handback)
```

Add a listener with default filtering conditions: all types but only trace level are selected.

Parameters:

`listener` - the listener to be added to receive trace notification.

`handback` - the object will send back to listener.

Throws: `IllegalArgumentException` - Thrown if the listener is null.

getRIType(int)

```
protected static java.lang.String getRIType(int type)
```

isSelected(int, int)

```
public static boolean isSelected(int level, int type)
```

Verify whether the specified info level and the info type are selected by a listener.

It is strongly recommanded to call this method before sending an information to this Trace class.

Parameters:

`level` - the level of trace information.

`type` - the type of the trace information.

---

parseTraceProperties()

```
public static void parseTraceProperties()
```

Parses the system properties specified at the command-line to determine the specified trace conditions

---

removeAllListeners()

```
public static void removeAllListeners()
```

Remove all listeners.

---

removeNotificationListener(NotificationListener)

```
public static void removeNotificationListener(NotificationListener listener)
```

Remove a listener

Parameters:
`listener` - the listener to be removed.

---

send(int, int, String, String, String)

```
public static boolean send(int level, int type, java.lang.String className,
            java.lang.String methodName, java.lang.String info)
```

Send a new information to this Trace class

Parameters:

`level` - the level of trace information to be sent.

`type` - the type of trace information to be sent.

`className` - the name of the class from which the trace information is from.

`methodName` - the name of the method from which the trace information is from.

`info` - the trace information to be sent.

Returns: false if the level and the type are not selected.

---

send(int, int, String, String, Throwable)

```
public static boolean send(int level, int type, java.lang.String className,
            java.lang.String methodName, java.lang.Throwable exception)
```

Send an exception to this Trace class.

Parameters:

`level` - the level of trace information to be sent.

`type` - the type of trace information to be sent.

`className` - the name of the class from which the trace information is from.

`methodName` - the name of the method from which the trace information is from.

`exception` - exception sent as the trace information.

com.sun.management
TraceFilter

Syntax

```
public class TraceFilter extends java.lang.Object implements NotificationFilter

java.lang.Object
  |
  +--com.sun.management.TraceFilter
```

All Implemented Interfaces: `NotificationFilter`, java.io.Serializable

Description

This class is used to filter trace information. The class Trace needs an object of this class for each listener.

Member Summary

Fields		
	`int`	`levels`
	`int`	`types`
Constructors		
		`TraceFilter(int, int)`
Methods		
	`int`	`getLevels()`
	`int`	`getTypes()`
	`boolean`	`isNotificationEnabled(Notification)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

---

levels

```
protected int levels
```

---

types

```
protected int types
```

Constructors

TraceFilter(int, int)

```
public  TraceFilter(int levels, int types)
```

Construct a TraceFilter object with the levels and types selected. Here is an example to specify create a filter: TraceFilter myFilter = new TraceFilter(LEVEL_TRACE, INFO_MBEANSERVER | INFO_TIMER) This filter will select information from MBeanServer and Timer with the level LEVEL_TRACE.

Parameters:
`levels` - the selected levels of trace information.
`types` - the selected types of the trace information.

Throws: `IllegalArgumentException` - Thrown if the specified types or levels are illegal.

Methods

---

getLevels()

```
public int getLevels()
```

get the levels selected

Returns: the levels selected for filtering.

---

getTypes()

```
public int getTypes()
```

get types selected

Returns: the types selected for filtering.

---

isNotificationEnabled(Notification)

```
public boolean isNotificationEnabled(Notification notification)
```

Implement the interface NotificationFilter. This implementation uses the specified types and levels to do filtering.

Specified By: `isNotificationEnabled(Notification)` in interface `NotificationFilter`

Parameters:
`notification` - the notification to be sent.

com.sun.management
TraceListener

Syntax

`public class TraceListener extends java.lang.Object implements NotificationListener`

```
java.lang.Object
  |
  +--com.sun.management.TraceListener
```

All Implemented Interfaces:

java.util.EventListener, `NotificationListener`

Description

An object of this class can be used to receive notifications sent out by the class Trace, and all notifications received will be saved to a file specified, or write to user screen. This listener will print or save information with two formats:

- simple format:

(className methodName) Message. This is a default format

- completed format: all realative information is printed or saved like:

Global sequence number: 7 Sequence number: 3

Level: LEVEL_TRACE Type: INFO_CONNECTOR_RMI

Class Name: MBeanServer

Method Name: sendTraceInfo

Information: This is a example to send a trace information.

A user should call the method setFormated with the value "true" to select this format.

This class can be used as a default listener, a user can write his own listener to treat trace information the way he wants.

Member Summary

Fields		
	`boolean`	`formated`
	`boolean`	`needTobeClosed`
	`PrintStream`	`out`
Constructors		
		`TraceListener()`
		`TraceListener(PrintStream)`
		`TraceListener(String)`
Methods		
	`void`	`handleNotification(Notification, Object)`
	`void`	`setFile(String)`
	`void`	`setFormated(boolean)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

formated

`protected boolean formated`

needTobeClosed

`protected boolean needTobeClosed`

out

`protected java.io.PrintStream out`

Constructors

TraceListener()

`public  TraceListener()`

Construct a default trace listener. All information will printed on screen (System.out) with the format:

(className methodName) message.

TraceListener(PrintStream)

`public  TraceListener(java.io.PrintStream ps)`

Construct a trace listener with a specified output stream.

Parameters:

`ps` - a PrintStream object used to print trace information.

For example, it can take value as System.out or System.err.

`ps` - An PrintStream object used to print out trace information.

Throws: `IllegalArgumentException` - thrown if the parameter ps is null.

TraceListener(String)

```
public  TraceListener(java.lang.String fileName)
```

Construct a trace listener with a file specified to save all information received.

If specified file exists currently, all trace information will be appended to this file.

Parameters:
`fileName` - the file used to save information.

Throws: `IOException` - thrown if failed to open or write the file.

Methods

handleNotification(Notification, Object)

```
public void handleNotification(Notification notif, java.lang.Object handback)
```

Called by the class Trace to receive trace information.

Specified By: `handleNotification(Notification, Object)` in interface `NotificationListener`

setFile(String)

```
public void setFile(java.lang.String fileName)
```

Specify a file to save information received

Parameters:
`fileName` - the file used to save information. It will replace the old file or the PrintStream object setted before.

Throws: `IOException` - thrown if failed to open or to write the file.

setFormated(boolean)

```
public void setFormated(boolean f)
```

Choose a format to output trace information.

Parameters:
`f` - if true, the completed format will be selected.

com.sun.management
TraceNotification

Syntax

`public class TraceNotification extends Notification`

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--Notification
              |
              +--com.sun.management.TraceNotification
```

All Implemented Interfaces: java.io.Serializable

Description

This class defines an object used by the class Trace to send out all internal runtime information.

Member Summary

Fields		
	`String`	`className`
	`Throwable`	`exception`
	`long`	`globalSequenceNumber`
	`String`	`info`
	`int`	`level`
	`String`	`methodName`
	`long`	`sequenceNumber`
	`int`	`type`
Constructors		
		`TraceNotification(Object, long, long, int, int, String, String, String, Throwable)`

Inherited Member Summary

Fields inherited from class java.util.EventObject

`source`

Methods inherited from class `Notification`

`getMessage()`, `getSequenceNumber()`, `getSource()`, `getTimeStamp()`, `getType()`, `getUserData()`, `setSource(Object)`, `setUserData(Object)`

Methods inherited from class java.util.EventObject

`toString`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Fields

---

className

`public java.lang.String className`

The name of the class from which the information comes

---

exception

`public java.lang.Throwable exception`

The exception sent out by the class Trace.

It can be null if a string is provided as information.

---

globalSequenceNumber

`public long globalSequenceNumber`

Global sequence number representing the place of this notification in all notification sequence.

---

info

`public java.lang.String info`

The information sent out by the class Trace.

It can be null if an exception is provided as information.

---

level

`public int level`

The level of information.

---

methodName

`public java.lang.String methodName`

The name of the method from which the information comes.

---

sequenceNumber

`public long sequenceNumber`

Sequence number representing the place of this notification in the sequence of all same type notifications

---

type

```
public int type
```

The type of information.

Constructors

TraceNotification(Object, long, long, int, int, String, String, String, Throwable)

```
public  TraceNotification(java.lang.Object source, long sequenceNumber, long
            globalSequenceNumber, int level, int type, java.lang.String className,
            java.lang.String methodName, java.lang.String info, java.lang.Throwable
            exception)
```

Construct a TraceNotification object.

Parameters:

`source` - the emitter of the notification.

`sequenceNumber` - the sequence number representing the place of this notification in the sequence of all same type notifications.

`globalSequenceNumber` - the global sequence number representing the place of this notification in all notifications sent out by the class Trace.

`level` - the level of information.

`type` - the type of the information.

`className` - the name of the class from which the information is from.

`methodName` - the name of the method from which the information is from.

`info` - an string as the trace information.

`exception` - an exception as the trace information.

TraceNotification(Object, long, long, int, int, String, String, String, Throwable)

package
javax.management

Description

Provides the **core JMX classes**.

Class Summary	
Interfaces	
DynamicMBean	This interface defines the methods that should be implemented by an MBean having a dynamic management interface.
MBeanRegistration	This interface can optionally be implemented by an MBean in order to carry out operations before and after being registered or de-registered from the MBeanServer.
MBeanServerDelegateMBean	This interface defines the management interface of an object of class MBeanServerDelegate.
NotificationBroadcaster	This interface should be implemented by a registered MBean.
NotificationFilter	This interface should be implemented by a any MBean acting as a notification filter.
NotificationListener	This interface listener should to receive JMX notification.
Classes	
Attribute	This class is used for representing a pair (attribute name, attribute value).
AttributeChangeNotification	This class provides definitions of the attribute change notifications sent by MBeans.
AttributeChangeNotificationFilter	This class provides an implementation of the NotificationFilter interface for the AttributeChangeNotifications.
AttributeList	This class is used to represent a list of attribute values of an object.
AttributeValueExp	The AttributeValueExp class represents attributes used as arguments to relational constraints.
DefaultLoaderRepository	
MBeanAttributeInfo	The MbeanAttributeInfo object describes an MBean attribute exposed for management.
MBeanConstructorInfo	The MBeanConstructorInfo object describes a constructor exposed by an MBean.
MBeanFeatureInfo	The MBeanFeatureInfo object provides general information for an MBean feature: The feature can be :
MBeanInfo	The MBeanInfo object allows to discover the management interface exposed by an MBean; that is, the set of attributes and operations which are available for management operations.
MBeanNotificationInfo	The MBeanNotificationInfo object describes describes a notification emitted by an MBean.
MBeanOperationInfo	The MBeanOperationInfo object describes a management operation exposed by an MBean.

Class Summary	
MBeanParameterInfo	The MBeanParameterInfo object describes an argument of an operation exposed ny an MBean.
MBeanServer	This is the base class for MBean manipulation on the agent side.
MBeanServerDelegate	This class is a MBeanServer representation from the management point of view.
MBeanServerNotification	The MBeanServerNotification class represents an notification emitted by the MBean Server.
Notification	The Notification class represents a notification emitted by an MBean.
NotificationBroadcasterSupport	This class provides an implementation of NotificationBroadcaster.
NotificationFilterSupport	This class provides an implementation of the NotificationFilter interface.
ObjectInstance	This class is used for representing an object instance; that is, an object name and class.
ObjectName	This class is used for representing an object name.
Query	The Query class supports construction of query object constraints.
QueryEval	This class allows a query to be accessed in the context of a specific MBeanServer.
QueryExp	The QueryExp class represents relational constraints that can be used in database query "where clauses." Instances of QueryExp are returned by the static methods of the Query class.
StringValueExp	This class represents strings that are arguments to relational constraints.
ValueExp	The ValueExp class represents values that can be passed as arguments to relational expressions.
Exceptions	
AttributeNotFoundException	The specified attribute does not exist or cannot be retrieved.
BadAttributeValueExpException	The BadAttributeValueExpException is thrown when an invalid MBean attribute is passed to a query constructing method.
BadBinaryOpValueExpException	This exception is thrown when an invalid expression is passed to a method for constructing a query.
BadStringOperationException	This exception is thrown when an invalid string operation is passed to a method for constructing a query.
InstanceAlreadyExistsException	The MBean is already registered in the repository.
InstanceNotFoundException	The specified MBean does not exist in the repository.
IntrospectionException	An exception occured during the introspection of an MBean.
InvalidApplicationException	This exception is thrown when an attempt is made to apply either of the following: A subquery expression to an MBean A qualified attribute expression to an MBean of the wrong class
InvalidAttributeValueException	The specified value is not a valid value for the attribute.
JMException	This class represents exceptions thrown by JMX implementations.

Class Summary	
`JMRuntimeException`	This class represents runtime exceptions emitted by JMX implementations.
`ListenerNotFoundException`	The specified MBean listener does not exist in the repository.
`MalformedObjectNameException`	The format of the string does not correspond to a valid ObjectName
`MBeanException`	This class represents "user defined" exceptions thrown by MBean methods in the agent.
`MBeanRegistrationException`	This class wraps exceptions thrown by the preRegister(), preDeregister() methods of the MBeanRegistration.
`NotCompliantMBeanException`	This exception occurs when trying to register in the MBeanServer an object that is not a JMX compliant MBean.
`OperationsException`	This class represents exceptions thrown in the MBeanServer when performing operations on MBeans.
`ReflectionException`	This class represents exceptions thrown in the MBeanServer when using the java.lang.reflect classes to invoke methods on MBeans.
`RuntimeErrorException`	When a java.lang.Error occurs in the agent it should be caught and re-thrown as a RuntimeErrorException.
`RuntimeMBeanException`	his class represents runtime exceptions thrown by MBean methods in the agent.
`RuntimeOperationsException`	This class represents runtime exceptions thrown in the agent when performing operations on MBeans.
`ServiceNotFoundException`	This class represents exceptions raised when a requested service is not supported.

javax.management
Attribute

Syntax

```
public class Attribute extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.Attribute
```

All Implemented Interfaces:
java.io.Serializable

Description

This class is used for representing a pair (attribute name, attribute value).

Member Summary

Constructors

	`Attribute(String, Object)`

Methods

`boolean`	`equals(Object)`
`String`	`getName()`
`Object`	`getValue()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

Attribute(String, Object)

```
public  Attribute(java.lang.String name, java.lang.Object value)
```

Allows an attribute to be created for an attribute.

Parameters:

`name` - The name of the attribute to be created.

`value` - The value assigned to the attribute.

Methods

---

equals(Object)

```
public boolean equals(java.lang.Object object)
```

Compares the current Attribute with another Attribute

Overrides: java.lang.Object.equals(java.lang.Object) in class java.lang.Object

Parameters:
`object` - The Attribute that the current Attribute is to be compared with.

Returns: True if the two Attribute objects are equal, otherwise false.

---

getName()

```
public java.lang.String getName()
```

Returns the name of the attribute.

---

getValue()

```
public java.lang.Object getValue()
```

Returns the value.

javax.management
AttributeChangeNotification

Syntax

```
public class AttributeChangeNotification extends Notification

java.lang.Object
  |
  +--java.util.EventObject
        |
        +--Notification
              |
              +--javax.management.AttributeChangeNotification
```

All Implemented Interfaces: java.io.Serializable

Description

This class provides definitions of the attribute change notifications sent by MBeans.

Member Summary

Fields

`String`	`ATTRIBUTE_CHANGE`

Constructors

	`AttributeChangeNotification(String, Object, long, Date, String, String, String, Object, Object)`

Methods

`String`	`getAttributeName()`
`String`	`getAttributeType()`
`Object`	`getNewValue()`
`Object`	`getOldValue()`

Inherited Member Summary

Fields inherited from class java.util.EventObject

`source`

Methods inherited from class `Notification`

`getMessage()`, `getSequenceNumber()`, `getSource()`, `getTimeStamp()`, `getType()`, `getUserData()`, `setSource(Object)`, `setUserData(Object)`

Methods inherited from class java.util.EventObject

`toString`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

Fields

ATTRIBUTE_CHANGE

```
public static final java.lang.String ATTRIBUTE_CHANGE
```

Notification type denoting that the observed MBean attribute value has changed. The value of this notification type is `jmx.attribute.change`.

Constructors

AttributeChangeNotification(String, Object, long, Date, String, String, String, Object, Object)

```
public  AttributeChangeNotification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.util.Date timeStamp, java.lang.String msg,
            java.lang.String attributeName, java.lang.String attributeType,
            java.lang.Object oldValue, java.lang.Object newValue)
```

Creates an attribute change notification object.

Parameters:

`type` - The notification type.

`source` - The notification producer, that is the MBean the attribute belongs to.

`sequenceNumber` - The notification sequence number within the source object.

`timeStamp` - The notification emission date.

`msg` - The notification message.

`attributeName` - The MBean attribute name.

`attributeType` - The MBean attribute type.

`oldValue` - The MBean attribute old value.

`newValue` - The MBean attribute new value.

Methods

---

getAttributeName()

```
public java.lang.String getAttributeName()
```

Gets the MBean attribute name.

Returns: The MBean attribute name.

---

getAttributeType()

```
public java.lang.String getAttributeType()
```

Gets the MBean attribute type.

Returns: The MBean attribute type.

---

getNewValue()

`public java.lang.Object getNewValue()`

Gets the MBean attribute new value.

Returns: The MBean attribute new value.

---

getOldValue()

`public java.lang.Object getOldValue()`

Gets the MBean attribute old value.

Returns: The MBean attribute old value.

javax.management
AttributeChangeNotificationFilter

Syntax

```
public class AttributeChangeNotificationFilter extends java.lang.Object implements
   NotificationFilter, java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.AttributeChangeNotificationFilter
```

All Implemented Interfaces: `NotificationFilter`, java.io.Serializable

Description

This class provides an implementation of the NotificationFilter interface for the AttributeChangeNotifications. The filtering is performed on the name of the observed attribute.

This class manages a list of enabled attribute names. A method allows users to enable/disable as many attribute names as required.

Member Summary

Constructors

	`AttributeChangeNotificationFilter()`

Methods

`void`	`disableAllAttributes()`
`void`	`disableAttribute(String)`
`void`	`enableAttribute(String)`
`Vector`	`getEnabledAttributes()`
`boolean`	`isNotificationEnabled(Notification)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

AttributeChangeNotificationFilter()

```
public  AttributeChangeNotificationFilter()
```

Methods

---

disableAllAttributes()

`public synchronized void disableAllAttributes()`

Disables all the attribute names.

---

disableAttribute(String)

`public synchronized void disableAttribute(java.lang.String name)`

Disables all the notifications whose MBean attribute name equals the specified attribute name to be sent to the listener. If the specified prefix is not in the list of enabled attribute names, this method has no effect.

Parameters:
`name` - The MBean attribute name.

---

enableAttribute(String)

`public synchronized void enableAttribute(java.lang.String name)`

Enables all the notifications whose MBean attribute name equals the specified attribute name to be sent to the listener. If the specified name is already in the list of enabled attribute names, this method has no effect.

Parameters:
`name` - The MBean attribute name.

Throws: `java.lang.IllegalArgumentException` - The MBean attribute name parameter is null.

---

getEnabledAttributes()

`public synchronized java.util.Vector getEnabledAttributes()`

Gets all the enabled attribute names for this filter.

Returns: The list containing all the enabled attribute names.

---

isNotificationEnabled(Notification)

`public synchronized boolean isNotificationEnabled(Notification notification)`

Invoked before sending the specified notification to the listener. This filter compares the attribute name with each enabled attribute name. If the MBean attribute name equals one of the enabled attribute name, the notification has to be sent to the listener and this method returns true.

Specified By: `isNotificationEnabled(Notification)` in interface `NotificationFilter`

Parameters:
`notification` - The notification to be sent.

Returns: True if the notification has to be sent to the listener, false otherwise.

javax.management
AttributeList

Syntax

```
public class AttributeList extends java.util.ArrayList
```

```
java.lang.Object
  |
  +--java.util.AbstractCollection
        |
        +--java.util.AbstractList
              |
              +--java.util.ArrayList
                    |
                    +--javax.management.AttributeList
```

All Implemented Interfaces: java.lang.Cloneable, java.util.Collection, java.util.List, java.io.Serializable

Description

This class is used to represent a list of attribute values of an object. The methods used for the insertion of Attribute objects in the AttributeList overrides the corresponding methods in the superclass ArrayList. This is needed in order to insure that the objects contained in the AttributeList are only Attribute objects. This avoids getting an exception when retrieving elements from the AttributeList.

Member Summary

Constructors

	`AttributeList()`
	`AttributeList(AttributeList)`
	`AttributeList(int)`

Methods

`void`	`add(Attribute)`
`void`	`add(int, Attribute)`
`boolean`	`addAll(AttributeList)`
`boolean`	`addAll(int, AttributeList)`
`void`	`set(int, Attribute)`

Inherited Member Summary

Fields inherited from class java.util.AbstractList

`modCount`

Methods inherited from class java.util.ArrayList

`add, add, addAll, addAll, clear, clone, contains, ensureCapacity, get, indexOf, isEmpty, lastIndexOf, remove, removeRange, set, size, toArray, toArray, trimToSize`

Inherited Member Summary

Methods inherited from class java.util.AbstractList

`equals, hashCode, iterator, listIterator, listIterator, subList`

Methods inherited from class java.util.AbstractCollection

`containsAll, remove, removeAll, retainAll, toString`

Methods inherited from class java.lang.Object

`finalize, getClass, notify, notifyAll, wait, wait, wait`

Methods inherited from interface java.util.List

`add, add, addAll, addAll, clear, contains, containsAll, equals, get, hashCode, indexOf, isEmpty, iterator, lastIndexOf, listIterator, listIterator, remove, remove, removeAll, retainAll, set, size, subList, toArray, toArray`

Constructors

AttributeList()

```
public  AttributeList()
```

Constructs an empty AttributeList.

AttributeList(AttributeList)

```
public  AttributeList(AttributeList list)
```

Constructs an AttributeList containing the elements of the AttributeList specified, in the order in which they are returned by the AttributeList's iterator. The AttributeList instance has an initial capacity of 110% of the size of the AttributeList specified.

AttributeList(int)

```
public  AttributeList(int initialCapacity)
```

Constructs an empty AttributeList with the initial capacity specified.

Methods

---

add(Attribute)

```
public void add(Attribute object)
```

Adds the Attribute specified as the last element of the list.

Parameters:

`object` - The attribute to be added.

---

add(int, Attribute)

`public void add(int index, Attribute object)`

Inserts the attribute specified as an element at the position specified. Elements with an index greater than or equal to the current position are shifted up. If the index is out of range (index < 0 || index > size() a RuntimeOperationsException should be raised, wrapping the java.lang.IndexOutOfBoundsException thrown.

Parameters:

`object` - The Attribute object to be inserted.

`index` - The position in the list where the new Attribute object is to be inserted.

---

addAll(AttributeList)

`public boolean addAll(AttributeList list)`

Appends all the elements in the AttributeList specified to the end of the list, in the order in which they are returned by the Iterator of the AttributeList specified.

Parameters:

`list` - Elements to be inserted into the list.

---

addAll(int, AttributeList)

`public boolean addAll(int index, AttributeList list)`

Inserts all of the elements in the AttributeList specified into this list, starting at the specified position, in the order in which they are returned by the Iterator of the AttributeList specified. If the index is out of range (index < 0 || index > size() a RuntimeOperationsException should be raised, wrapping the java.lang.IndexOutOfBoundsException thrown.

Parameters:

`list` - Elements to be inserted into the list.

`index` - Position at which to insert the first element from the AttributeList specified.

---

set(int, Attribute)

`public void set(int index, Attribute object)`

Sets the element at the position specified to be the attribute specified. The previous element at that position is discarded. If the index is out of range (index < 0 || index > size() a RuntimeOperationsException should be raised, wrapping the java.lang.IndexOutOfBoundsException thrown.

Parameters:

`object` - The value to which the attribute element should be set.

`index` - The position specified.

javax.management
AttributeNotFoundException

Syntax

```
public class AttributeNotFoundException extends OperationsException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.AttributeNotFoundException
```

All Implemented Interfaces: java.io.Serializable

Description

The specified attribute does not exist or cannot be retrieved.

Member Summary

Constructors

`AttributeNotFoundException()`
`AttributeNotFoundException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

AttributeNotFoundException()

```
public AttributeNotFoundException()
```

Default constructor.

AttributeNotFoundException(String)

```
public  AttributeNotFoundException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
AttributeValueExp

Syntax

```
public class AttributeValueExp extends ValueExp
```

```
java.lang.Object
  |
  +--QueryEval
        |
        +--ValueExp
              |
              +--javax.management.AttributeValueExp
```

All Implemented Interfaces: java.io.Serializable

Description

The AttributeValueExp class represents attributes used as arguments to relational constraints. An AttributeValueExp may be used anywhere a ValueExp is required.

Member Summary

Constructors		
		`AttributeValueExp()`
		`AttributeValueExp(String)`
Methods		
	`ValueExp`	`apply(Object)`
	`Object`	`getAttribute(Object)`
	`String`	`getAttributeName()`
	`String`	`toString()`

Inherited Member Summary

Methods inherited from class `QueryEval`

`setMBeanServer(MBeanServer)`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

AttributeValueExp()

`public  AttributeValueExp()`

Basic Constructor.

AttributeValueExp(String)

`public  AttributeValueExp(java.lang.String attr)`

Creates a new AttributeValueExp representing the specified object attribute, named attr.

Methods

apply(Object)

`public ValueExp apply(java.lang.Object object)`

Applies the AttributeValueExp on an MBean.

Overrides: `apply(Object)` in class `ValueExp`

Parameters:
`object` - The MBean on which the AttributeValueExp will be applied.

Returns: The ValueExp.

Throws:

getAttribute(Object)

`protected java.lang.Object getAttribute(java.lang.Object mo)`

getAttributeName()

`public java.lang.String getAttributeName()`

Returns a string representation of the name of the attribute.

Returns: A string representation of the name of the attribute.

toString()

`public java.lang.String toString()`

Returns the string representing its value

Overrides: java.lang.Object.toString() in class java.lang.Object

javax.management
BadAttributeValueExpException

Syntax

```
public class BadAttributeValueExpException extends java.lang.Exception
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.management.BadAttributeValueExpException
```

All Implemented Interfaces:

java.io.Serializable

Description

The BadAttributeValueExpException is thrown when an invalid MBean attribute is passed to a query constructing method.

Member Summary

Constructors		
		`BadAttributeValueExpException(Object)`
Methods		
	`String`	`toString()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

BadAttributeValueExpException(Object)

```
public  BadAttributeValueExpException(java.lang.Object val)
```

Constructs an BadAttributeValueExpException with the specified Object.

Methods

---

toString()

`public java.lang.String toString()`

Returns the string representing the object.

Overrides: java.lang.Throwable.toString() in class java.lang.Throwable

javax.management
BadBinaryOpValueExpException

Syntax

`public class BadBinaryOpValueExpException extends java.lang.Exception`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.management.BadBinaryOpValueExpException
```

All Implemented Interfaces: java.io.Serializable

Description

This exception is thrown when an invalid expression is passed to a method for constructing a query.

Member Summary	
Constructors	
	`BadBinaryOpValueExpException(ValueExp)`
Methods	
`ValueExp`	`getExp()`
`String`	`toString()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

BadBinaryOpValueExpException(ValueExp)

`public BadBinaryOpValueExpException(ValueExp exp)`

Constructs an BadBinaryOpValueExpException with the specified ValueExp.

Methods

---

getExp()

```
public ValueExp getExp()
```

Returns the ValueExp that originates the exception.

---

toString()

```
public java.lang.String toString()
```

Returns the string representing the object

Overrides: java.lang.Throwable.toString() in class java.lang.Throwable

javax.management
BadStringOperationException

Syntax

```
public class BadStringOperationException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.management.BadStringOperationException
```

All Implemented Interfaces: java.io.Serializable

Description

This exception is thrown when an invalid string operation is passed to a method for constructing a query.

Member Summary	
Constructors	
	`BadStringOperationException(String)`
Methods	
`String`	`toString()`

Inherited Member Summary
Methods inherited from class java.lang.Throwable
`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace`
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

BadStringOperationException(String)

```
public  BadStringOperationException(java.lang.String op)
```

Constructs an BadStringOperationException with the specified detail message.

Methods

---

toString()

```
public java.lang.String toString()
```

Returns the string representing the object.

Overrides: java.lang.Throwable.toString() in class java.lang.Throwable

javax.management
DefaultLoaderRepository

Syntax

```
public class DefaultLoaderRepository extends java.lang.Object implements java.io.Serializable

java.lang.Object
  |
  +--javax.management.DefaultLoaderRepository
```

All Implemented Interfaces: java.io.Serializable

Description

Member Summary

Constructors

	`DefaultLoaderRepository()`

Methods

`Class`	`loadClass(String)`
`Class`	`loadClassWithout(ClassLoader, String)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

DefaultLoaderRepository()

```
public  DefaultLoaderRepository()
```

Methods

---

loadClass(String)

```
public static java.lang.Class loadClass(java.lang.String className)
```

Go through the list of class loaders and try to load the requested class. The method will stop as soon as the class is found. If the class is not found the method will throw a `ClassNotFoundException` exception.

Parameters:
`className` - The name of the class to be loaded.

Throws: `ClassNotFoundException` - The specified class could not be found.

---

loadClassWithout(ClassLoader, String)

```
public static java.lang.Class loadClassWithout(java.lang.ClassLoader loader,
            java.lang.String className)
```

Go through the list of class loaders but exclude the given class loader, then try to load the requested class. The method will stop as soon as the class is found. If the class is not found the method will throw a `ClassNotFoundException` exception.

Parameters:
`className` - The name of the class to be loaded.

`loader` - The class loader to be excluded.

Throws: `ClassNotFoundException` - The specified class could not be found.

javax.management
DynamicMBean

Syntax

`public interface DynamicMBean`

All Known Implementing Classes:

com.sun.jdmk.comm.HtmlAdaptorServer

Description

This interface defines the methods that should be implemented by an MBean having a dynamic management interface.

Member Summary

Methods

`Object`	`getAttribute(String)`
`AttributeList`	`getAttributes(String[])`
`MBeanInfo`	`getMBeanInfo()`
`Object`	`invoke(String, Object[], String[])`
`void`	`setAttribute(Attribute)`
`AttributeList`	`setAttributes(AttributeList)`

Methods

---

getAttribute(String)

```
public java.lang.Object getAttribute(java.lang.String attribute)
```

Allows the value of a specific attribute of the Dynamic MBean to be obtained.

Parameters:
`attribute` - The name of the attribute to be retrieved

Returns: The value of the attribute retrieved.

Throws: `MBeanException` - Wraps an exception thrown by the MBean's getter.

`ReflectionException` - Wraps an java.lang.Exception thrown while trying to invoke the getter.

---

getAttributes(String[])

```
public AttributeList getAttributes(java.lang.String[] attributes)
```

Enables the values of several attributes of the Dynamic MBean.

Parameters:
`attributes` - A list of the attributes to be retrieved.

Returns: The list of the retrieved attributes.

---

getMBeanInfo()

```
public MBeanInfo getMBeanInfo()
```

This method provides the exposed attributes and actions of the Dynamic MBean. It provides this information using an MBeanInfo object.

Returns: An instance of MBeanInfo allowing all attributes and actions exposed by this Dynamic MBean to be retrieved.

---

invoke(String, Object[], String[])

```
public java.lang.Object invoke(java.lang.String actionName, java.lang.Object[] params,
            java.lang.String[] signature)
```

Allows an action to be invoked on the Dynamic MBean.

Parameters:

`actionName` - The name of the action to be invoked.

`params` - An array containing the parameters to be set when the action is invoked.

`signature` - An array containing the signature of the action. The class objects will be loaded through the same class loader as the one used for loading the MBean on which the action is invoked.

Returns: The object returned by the action, which represents the result of invoking the action on the MBean specified.

Throws: `MBeanException` - Wraps an exception thrown by the MBean's invoked method.

`ReflectionException` - Wraps an java.lang.Exception thrown while trying to invoke the method

---

setAttribute(Attribute)

```
public void setAttribute(Attribute attribute)
```

Sets the value of a specific attribute of the Dynamic MBean

Parameters:

`attribute` - The identification of the attribute to be set and the value it is to be set to.

Throws: `MBeanException` - Wraps an exception thrown by the MBean's setter.

`ReflectionException` - Wraps an exception thrown while trying to invoke the MBean's setter.

---

setAttributes(AttributeList)

```
public AttributeList setAttributes(AttributeList attributes)
```

Sets the values of several attributes of the Dynamic MBean

Parameters:

`name` - The object name of the MBean within which the attributes are to be set.

`attributes` - A list of attributes: The identification of the attributes to be set and the values they are to be set to.

Returns: The list of attributes that were set, with their new values.

javax.management
InstanceAlreadyExistsException

Syntax

```
public class InstanceAlreadyExistsException extends OperationsException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.InstanceAlreadyExistsException
```

All Implemented Interfaces: java.io.Serializable

Description

The MBean is already registered in the repository.

Member Summary	
Constructors	
	`InstanceAlreadyExistsException()`
	`InstanceAlreadyExistsException(String)`

Inherited Member Summary
Methods inherited from class java.lang.Throwable
`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

InstanceAlreadyExistsException()

```
public  InstanceAlreadyExistsException()
```

Default constructor.

InstanceAlreadyExistsException(String)

```
public  InstanceAlreadyExistsException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
InstanceNotFoundException

Syntax

`public class InstanceNotFoundException extends OperationsException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.InstanceNotFoundException
```

All Implemented Interfaces: java.io.Serializable

Description

The specified MBean does not exist in the repository.

Member Summary

Constructors	
	`InstanceNotFoundException()`
	`InstanceNotFoundException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

InstanceNotFoundException()

```
public  InstanceNotFoundException()
```

Default constructor.

InstanceNotFoundException(String)

```
public  InstanceNotFoundException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
IntrospectionException

Syntax

```
public class IntrospectionException extends OperationsException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.IntrospectionException
```

All Implemented Interfaces: java.io.Serializable

Description

An exception occured during the introspection of an MBean.

Member Summary

Constructors

`IntrospectionException()`
`IntrospectionException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

IntrospectionException()

```
public  IntrospectionException()
```

Default constructor.

IntrospectionException(String)

```
public  IntrospectionException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
InvalidApplicationException

Syntax

`public class InvalidApplicationException extends java.lang.Exception`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.management.InvalidApplicationException
```

All Implemented Interfaces: java.io.Serializable

Description

This exception is thrown when an attempt is made to apply either of the following: A subquery expression to an MBean A qualified attribute expression to an MBean of the wrong class

Member Summary

Constructors		
		`InvalidApplicationException(Object)`
Methods		
	`String`	`toString()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

InvalidApplicationException(Object)

`public InvalidApplicationException(java.lang.Object val)`

Constructs an InvalidApplicationException with the specified Object.

Methods

---

toString()

```
public java.lang.String toString()
```

Returns the string representing the object

Overrides: java.lang.Throwable.toString() in class java.lang.Throwable

javax.management
InvalidAttributeValueException

Syntax

```
public class InvalidAttributeValueException extends OperationsException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.InvalidAttributeValueException
```

All Implemented Interfaces: java.io.Serializable

Description

The specified value is not a valid value for the attribute.

Member Summary

Constructors

`InvalidAttributeValueException()`
`InvalidAttributeValueException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

InvalidAttributeValueException()

```
public  InvalidAttributeValueException()
```

Default constructor.

InvalidAttributeValueException(String)

```
public  InvalidAttributeValueException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
JMException

Syntax

```
public class JMException extends java.lang.Exception

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--javax.management.JMException
```

Direct Known Subclasses: com.sun.jdmk.comm.MalformedHttpException, `MBeanException`, `OperationsException`, `ReflectionException`

All Implemented Interfaces: java.io.Serializable

Description

This class represents exceptions thrown by JMX implementations. It does not include the runtime exceptions.

Member Summary	
Constructors	
	`JMException()`
	`JMException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

JMException()

```
public JMException()
```

Default constructor.

JMException(String)

```
public  JMException(java.lang.String msg)
```

Constructor that allows a specific error message to be specified.

javax.management
JMRuntimeException

Syntax

```
public class JMRuntimeException extends java.lang.RuntimeException

java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--javax.management.JMRuntimeException
```

Direct Known Subclasses: com.sun.jdmk.comm.CommunicationException, `MonitorSettingException`, `RuntimeErrorException`, `RuntimeMBeanException`, `RuntimeOperationsException`

All Implemented Interfaces: java.io.Serializable

Description

This class represents runtime exceptions emitted by JMX implementations.

Member Summary

Constructors	
	`JMRuntimeException()`
	`JMRuntimeException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

JMRuntimeException()

```
public  JMRuntimeException()
```

Default constructor.

JMRuntimeException(String)

```
public  JMRuntimeException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
ListenerNotFoundException

Syntax

`public class ListenerNotFoundException extends OperationsException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.ListenerNotFoundException
```

All Implemented Interfaces: java.io.Serializable

Description

The specified MBean listener does not exist in the repository.

Member Summary	
Constructors	
	`ListenerNotFoundException()`
	`ListenerNotFoundException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

ListenerNotFoundException()

```
public  ListenerNotFoundException()
```

Default constructor.

ListenerNotFoundException(String)

```
public  ListenerNotFoundException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
MalformedObjectNameException

Syntax

`public class MalformedObjectNameException extends OperationsException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.MalformedObjectNameException
```

All Implemented Interfaces: java.io.Serializable

Description

The format of the string does not correspond to a valid ObjectName

Member Summary	
Constructors	
	`MalformedObjectNameException()`
	`MalformedObjectNameException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

MalformedObjectNameException()

```
public  MalformedObjectNameException()
```

Default constructor.

MalformedObjectNameException(String)

```
public  MalformedObjectNameException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
MBeanAttributeInfo

Syntax

`public class MBeanAttributeInfo extends MBeanFeatureInfo implements java.io.Serializable`

```
java.lang.Object
  |
  +--MBeanFeatureInfo
        |
        +--javax.management.MBeanAttributeInfo
```

All Implemented Interfaces: java.io.Serializable

Description

The MbeanAttributeInfo object describes an MBean attribute exposed for management.

Member Summary

Constructors

	`MBeanAttributeInfo(String, String, Method, Method)`
	`MBeanAttributeInfo(String, String, String, boolean, boolean)`

Methods

`String`	`getType()`
`boolean`	`isIs()`
`boolean`	`isReadable()`
`boolean`	`isWritable()`

Inherited Member Summary

Fields inherited from class `MBeanFeatureInfo`

`description`, `name`

Methods inherited from class `MBeanFeatureInfo`

`getDescription()`, `getName()`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanAttributeInfo(String, String, Method, Method)

```
public  MBeanAttributeInfo(java.lang.String name, java.lang.String description,
            java.lang.reflect.Method getter, java.lang.reflect.Method setter)
```

This constructor takes the name of a simple attribute, and Method objects for reading and writing the attribute.

Parameters:

`name` - The programmatic name of the attribute.

`description` - A human readable description of the attribute.

`getter` - The method used for reading the attribute value. May be null if the property is write-only.

`setter` - The method used for writing the attribute value. May be null if the attribute is read-only.

Throws: `IntrospectionException` - There is a consistency problem in the definition of this attribute.

MBeanAttributeInfo(String, String, String, boolean, boolean)

```
public  MBeanAttributeInfo(java.lang.String name, java.lang.String type,
            java.lang.String description, boolean isReadable, boolean isWritable)
```

Constructs a MbeanAttributeInfo object.

Parameters:

`name` - The name of the attribute

`type` - The type or class name of the attribute

`description` - A human readable description of the attribute.

`isReadable` - True if the attribute has a getter method, false otherwise.

`isWritable` - True if the attribute has a setter method, false otherwise.

Methods

getType()

```
public java.lang.String getType()
```

Returns the class name of the attribute.

isIs()

```
public boolean isIs()
```

Indicates if this attribute has an "is" getter

isReadable()

```
public boolean isReadable()
```

Whether the value of the attribute can be read.

Returns: True if the attribute can be read, false otherwise.

---

isWritable()

```
public boolean isWritable()
```

Whether new values can be written to the attribute.

Returns: True if the attribute can be written, false otherwise.

javax.management
MBeanConstructorInfo

Syntax

`public class MBeanConstructorInfo extends MBeanFeatureInfo implements java.io.Serializable`

```
java.lang.Object
  |
  +--MBeanFeatureInfo
        |
        +--javax.management.MBeanConstructorInfo
```

All Implemented Interfaces:

java.io.Serializable

Description

The MBeanConstructorInfo object describes a constructor exposed by an MBean.

Member Summary

Constructors	
	`MBeanConstructorInfo(String, Constructor)`
	`MBeanConstructorInfo(String, String, MBeanParameterInfo[])`
Methods	
`MBeanParameterInfo[]`	`getSignature()`

Inherited Member Summary

Fields inherited from class `MBeanFeatureInfo`

`description`, `name`

Methods inherited from class `MBeanFeatureInfo`

`getDescription()`, `getName()`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanConstructorInfo(String, Constructor)

```
public  MBeanConstructorInfo(java.lang.String description, java.lang.reflect.Constructor
            constructor)
```

Constructs a MBeanConstructorInfo object.

Parameters:

`method` - The java.lang.reflect.Method object describing the MBean operation.

`description` - A human readable description of the operation.

MBeanConstructorInfo(String, String, MBeanParameterInfo[])

```
public  MBeanConstructorInfo(java.lang.String name, java.lang.String description,
            MBeanParameterInfo signature)
```

Constructs a MBeanConstructorInfo object.

Parameters:

`name` - The name of the constructor.

`description` - A human readable description of the constructor.

`signature` - MBeanParameterInfo objects describing the parameters(arguments) of the constructor.

Methods

getSignature()

```
public MBeanParameterInfo getSignature()
```

Returns the signature of the method, that is, information on the operations arguments.

javax.management
MBeanException

Syntax

```
public class MBeanException extends JMException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--javax.management.MBeanException
```

Direct Known Subclasses: `MBeanRegistrationException`

All Implemented Interfaces: java.io.Serializable

Description

This class represents "user defined" exceptions thrown by MBean methods in the agent. It "wraps" the actual "user defined" exception thrown. This exception will be built by the MBeanServer when a call to an MBean method results in an unknown exception.

Member Summary

Constructors	
	`MBeanException(Exception)`
	`MBeanException(Exception, String)`
Methods	
`Exception`	`getTargetException()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

MBeanException(Exception)

```
public  MBeanException(java.lang.Exception e)
```

Creates a MBeanException that wraps the actual java.lang.Exception.

MBeanException(Exception, String)

```
public  MBeanException(java.lang.Exception e, java.lang.String message)
```

Creates a MBeanException that wraps the actual java.lang.Exception with a detail message.

Methods

---

getTargetException()

```
public java.lang.Exception getTargetException()
```

Returns the actual java.lang.Exception thrown

javax.management
MBeanFeatureInfo

Syntax

```
public class MBeanFeatureInfo extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.MBeanFeatureInfo
```

Direct Known Subclasses: `MBeanAttributeInfo`, `MBeanConstructorInfo`, `MBeanNotificationInfo`, `MBeanOperationInfo`, `MBeanParameterInfo`

All Implemented Interfaces: java.io.Serializable

Description

The MBeanFeatureInfo object provides general information for an MBean feature: The feature can be :

An attribute, an operation, a parameter, a returned value or a notification.

Member Summary

Fields		
	`String`	`description`
	`String`	`name`
Constructors		
		`MBeanFeatureInfo(String, String)`
Methods		
	`String`	`getDescription()`
	`String`	`getName()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

description

```
protected java.lang.String description
```

The human readable description of the feature.

---

name

```
protected java.lang.String name
```

The name of the feature.

Constructors

MBeanFeatureInfo(String, String)

```
public  MBeanFeatureInfo(java.lang.String name, java.lang.String description)
```

Constructs a MBeanFeatureInfo object.

Parameters:

`name` - The name of the feature.

`description` - A human readable description of the feature.

Methods

---

getDescription()

```
public java.lang.String getDescription()
```

Returns the human readable description of the feature.

---

getName()

```
public java.lang.String getName()
```

Returns the name of the feature.

javax.management
MBeanInfo

Syntax

```
public class MBeanInfo extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.MBeanInfo
```

All Implemented Interfaces:

java.io.Serializable

Description

The MBeanInfo object allows to discover the management interface exposed by an MBean; that is, the set of attributes and operations which are available for management operations.

Member Summary

Constructors	
	`MBeanInfo(String, String, MBeanAttributeInfo[], MBeanConstructorInfo[], MBeanOperationInfo[], MBeanNotificationInfo[])`
Methods	
`MBeanAttributeInfo[]`	`getAttributes()`
`String`	`getClassName()`
`MBeanConstructorInfo[]`	`getConstructors()`
`String`	`getDescription()`
`MBeanNotificationInfo[]`	`getNotifications()`
`MBeanOperationInfo[]`	`getOperations()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanInfo(String, String, MBeanAttributeInfo[], MBeanConstructorInfo[],

MBeanOperationInfo[], MBeanNotificationInfo[])

```
public  MBeanInfo(java.lang.String className, java.lang.String description,
            MBeanAttributeInfo attributes, MBeanConstructorInfo constructors,
            MBeanOperationInfo operations, MBeanNotificationInfo notifications)
```

Constructs an `MBeanInfo`.

Parameters:

`name` - The name of the Java class of the MBean described by this MBeanInfo.

`description` - A human readable description of the MBean (optional).

`attributes` - The list of exposed attributes of the MBean.

`constructors` - The list of public constructors of the MBean.

`operations` - The list of operations of the MBean.

`notifications` - The list of notifications emitted by an MBean.

Methods

---

getAttributes()

```
public MBeanAttributeInfo getAttributes()
```

Rreturns the list of attributes exposed for management. Each attribute is described by an MBeanAttributeInfo object.

Returns: An array of MBeanAttributeInfo objects.

---

getClassName()

```
public java.lang.String getClassName()
```

Returns the name of the Java class of the MBean described by this MBeanInfo.

---

getConstructors()

```
public MBeanConstructorInfo getConstructors()
```

Returns the list of the public constructors of the MBean. Each constructor is described by an MBeanConstructorInfo object.

Returns: An array of MBeanConstructorInfo objects.

---

getDescription()

```
public java.lang.String getDescription()
```

Returns a human readable description of the MBean. Optional.

---

getNotifications()

`public MBeanNotificationInfo getNotifications()`

Rreturns the list of the notifications emitted by the MBean. Each notification is described by an MBeanNotificationInfo object.

Returns: An array of MBeanNotificationInfo objects.

---

getOperations()

`public MBeanOperationInfo getOperations()`

Returns the list of operations of the MBean. Each operation is described by an MBeanOperationInfo object.

Returns: An array of MBeanOperationInfo objects.

javax.management
MBeanNotificationInfo

Syntax

```
public class MBeanNotificationInfo extends MBeanFeatureInfo implements java.io.Serializable
```

```
java.lang.Object
  |
  +--MBeanFeatureInfo
        |
        +--javax.management.MBeanNotificationInfo
```

All Implemented Interfaces: java.io.Serializable

Description

The MBeanNotificationInfo object describes describes a notification emitted by an MBean.

Member Summary	
Constructors	
	`MBeanNotificationInfo(String[], String, String)`
Methods	
`String[]`	`getNotifTypes()`

Inherited Member Summary
Fields inherited from class `MBeanFeatureInfo`
`description`, `name`
Methods inherited from class `MBeanFeatureInfo`
`getDescription()`, `getName()`
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanNotificationInfo(String[], String, String)

```
public  MBeanNotificationInfo(java.lang.String[] notifsType, java.lang.String name,
            java.lang.String description)
```

Constructs a MBeanNotificationInfo object.

Parameters:

`notifType` - The event type string (in dot notation).

`name` - The name of the Notification class.

`description` - A human readable description of the data. Optional.

Methods

---

getNotifTypes()

```
public java.lang.String[] getNotifTypes()
```

Returns the notification type string (in dot notation) of the notification.

javax.management
MBeanOperationInfo

Syntax

`public class MBeanOperationInfo extends MBeanFeatureInfo implements java.io.Serializable`

```
java.lang.Object
  |
  +--MBeanFeatureInfo
        |
        +--javax.management.MBeanOperationInfo
```

All Implemented Interfaces: java.io.Serializable

Description

The MBeanOperationInfo object describes a management operation exposed by an MBean.

Member Summary

Fields	
int	ACTION
int	ACTION_INFO
int	INFO
int	UNKNOWN
Constructors	
	MBeanOperationInfo(String, Method)
	MBeanOperationInfo(String, String, MBeanParameterInfo[], String, int)
Methods	
int	getImpact()
String	getReturnType()
MBeanParameterInfo[]	getSignature()

Inherited Member Summary

Fields inherited from class `MBeanFeatureInfo`

`description`, `name`

Methods inherited from class `MBeanFeatureInfo`

`getDescription()`, `getName()`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

ACTION

```
public static final int ACTION
```

Indicates that the operation is a write-like in nature, and would modify the MBean in some way, typically by writing some value or changing a configuration.

ACTION_INFO

```
public static final int ACTION_INFO
```

Indicates that the operation is a read/write-like in nature.

INFO

```
public static final int INFO
```

Indicates that the operation is a read-like in nature, it basically returns some information.

UNKNOWN

```
public static final int UNKNOWN
```

Indicates that the operation has an "unknown" nature.

Constructors

MBeanOperationInfo(String, Method)

```
public  MBeanOperationInfo(java.lang.String description, java.lang.reflect.Method
            method)
```

Constructs a MBeanOperationInfo object.

Parameters:

`method` - The java.lang.reflect.Method object describing the MBean operation.

`description` - A human readable description of the operation.

MBeanOperationInfo(String, String, MBeanParameterInfo[], String, int)

```
public  MBeanOperationInfo(java.lang.String name, java.lang.String description,
            MBeanParameterInfo signature, java.lang.String type, int impact)
```

Constructs a MBeanOperationInfo object.

Parameters:

`name` - The name of the method.

`description` - A human readable description of the operation.

`signature` - MBeanParameterInfo objects describing the parameters(arguments) of the method.

`type` - The type of the method's return value.

`impact` - The impact of the method, one of INFO, ACTION, ACTION_INFO, UNKNOWN.

Methods

---

getImpact()

```
public int getImpact()
```

Returns the impact of the method, one of `INFO`, `ACTION`, `ACTION_INFO`, `UNKNOWN`.

---

getReturnType()

```
public java.lang.String getReturnType()
```

Returns the description of the method's return value.

---

getSignature()

```
public MBeanParameterInfo getSignature()
```

Returns the signature of the method, that is, information on the operations arguments.

javax.management
MBeanParameterInfo

Syntax

```
public class MBeanParameterInfo extends MBeanFeatureInfo implements java.io.Serializable
```

```
java.lang.Object
  |
  +--MBeanFeatureInfo
        |
        +--javax.management.MBeanParameterInfo
```

All Implemented Interfaces: java.io.Serializable

Description

The MBeanParameterInfo object describes an argument of an operation exposed ny an MBean.

Member Summary	
Constructors	
	`MBeanParameterInfo(String, String, String)`
Methods	
`String`	`getType()`

Inherited Member Summary
Fields inherited from class `MBeanFeatureInfo`
`description`, `name`
Methods inherited from class `MBeanFeatureInfo`
`getDescription()`, `getName()`
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanParameterInfo(String, String, String)

```
public  MBeanParameterInfo(java.lang.String name, java.lang.String type,
            java.lang.String description)
```

Constructs a MBeanParameterInfo object.

Parameters:

`name` - The name of the data

`type` - The type or class name of the data

`description` - A human readable description of the data. Optional.

Methods

---

getType()

```
public java.lang.String getType()
```

Returns the type or class name of the data.

javax.management
MBeanRegistration

Syntax

`public interface MBeanRegistration`

All Known Implementing Classes:

com.sun.jdmk.comm.CommunicatorServer, com.sun.jdmk.comm.HtmlAdaptorServer, `Monitor`, `MLet`

Description

This interface can optionally be implemented by an MBean in order to carry out operations before and after being registered or de-registered from the MBeanServer.

Member Summary

Methods

`void`	`postDeregister()`
`void`	`postRegister(Boolean)`
`void`	`preDeregister()`
`ObjectName`	`preRegister(MBeanServer, ObjectName)`

Methods

---

postDeregister()

`public void postDeregister()`

Allows the MBean to perform any operations needed after having been de-registered in the MBeanServer.

---

postRegister(Boolean)

`public void postRegister(java.lang.Boolean registrationDone)`

Allows the MBean to perform any operations needed after having been registered in the MBeanServer or after the registration has failed.

Parameters:

`registrationDone` - Indicates whether or not the MBean has been successfully registered in the MBeanServer. The value false means that either the registration phase has failed.

---

preDeregister()

`public void preDeregister()`

Allows the MBean to perform any operations it needs before being de-registered by the MBeanServer.

Throws: `java.langException` - This exception should be caught by the MBeanServer and re-thrown as an MBeanRegistrationException.

---

preRegister(MBeanServer, ObjectName)

`public ObjectName preRegister(MBeanServer server, ObjectName name)`

Allows the MBean to perform any operations it needs before being registered in the MBeanServer. If the name of the MBean is not specified, the MBean can provide a name for its registration. If any exception is raised, the MBean will not be registered in the MBeanServer.

Parameters:

`server` - The MBeanServer in which the MBean will be registered.

`name` - The object name of the MBean.

Returns: The name of the MBean registered.

Throws: `java.langException` - This exception should be caught by the MBeanServer and re-thrown as an MBeanRegistrationException.

javax.management
MBeanRegistrationException

Syntax

`public class MBeanRegistrationException extends MBeanException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--MBeanException
                          |
                          +--javax.management.MBeanRegistrationException
```

All Implemented Interfaces:

java.io.Serializable

Description

This class wraps exceptions thrown by the preRegister(), preDeregister() methods of the MBeanRegistration.

Member Summary

Constructors	
	`MBeanRegistrationException(Exception)`
	`MBeanRegistrationException(Exception, String)`

Inherited Member Summary

Methods inherited from interface `MBeanException`

`getTargetException()`

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

MBeanRegistrationException(Exception)

```
public  MBeanRegistrationException(java.lang.Exception e)
```

Creates a MBeanRegistrationException that wraps the actual java.lang.Exception.

MBeanRegistrationException(Exception, String)

```
public  MBeanRegistrationException(java.lang.Exception e, java.lang.String message)
```

Creates a MBeanRegistrationException that wraps the actual java.lang.Exception with a detail message.

javax.management
MBeanServer

Syntax

```
public class MBeanServer extends java.lang.Object
```

```
java.lang.Object
  |
  +--javax.management.MBeanServer
```

Description

This is the base class for MBean manipulation on the agent side. It contains the necessary methods for the creation, registration, and deletion of MBeans as well as the access methods for registered MBeans. This is the core component of the JMX infrastructure.

Every MBean which is added to the MBeanServer becomes manageable: its attributes and operations become remotely accessible through the connectors/adaptors connected to that MBeanServer. An Java object cannot be registered in the MBeanServer unless it is a JMX compliant MBean.

Member Summary

Constructors	
	`MBeanServer()`
	`MBeanServer(String)`
Methods	
`void`	`addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)`
`void`	`addNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)`
`ObjectInstance`	`createMBean(String, ObjectName)`
`ObjectInstance`	`createMBean(String, ObjectName, Object[], String[])`
`ObjectInstance`	`createMBean(String, ObjectName, ObjectName)`
`ObjectInstance`	`createMBean(String, ObjectName, ObjectName, Object[], String[])`
`ObjectInputStream`	`deserialize(ObjectName, byte[])`
`ObjectInputStream`	`deserialize(String, byte[])`
`ObjectInputStream`	`deserialize(String, ObjectName, byte[])`
`ArrayList`	`findJMXAgent(String)`
`Object`	`getAttribute(ObjectName, String)`
`AttributeList`	`getAttributes(ObjectName, String[])`
`String`	`getDefaultDomain()`
`Integer`	`getMBeanCount()`
`MBeanInfo`	`getMBeanInfo(ObjectName)`
`ObjectInstance`	`getObjectInstance(ObjectName)`
`Object`	`instantiate(String)`
`Object`	`instantiate(String, Object[], String[])`
`Object`	`instantiate(String, ObjectName)`
`Object`	`instantiate(String, ObjectName, Object[], String[])`
`Object`	`invoke(ObjectName, String, Object[], String[])`
`boolean`	`isRegistered(ObjectName)`

Member Summary

`Set`	queryMBeans(ObjectName, QueryExp)
`Set`	queryNames(ObjectName, QueryExp)
`ObjectInstance`	registerMBean(Object, ObjectName)
`void`	removeNotificationListener(ObjectName, NotificationListener)
`void`	removeNotificationListener(ObjectName, ObjectName)
`void`	setAttribute(ObjectName, Attribute)
`AttributeList`	setAttributes(ObjectName, AttributeList)
`void`	unregisterMBean(ObjectName)

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanServer()

```
public  MBeanServer()
```

Creates an MBeanServer with a standard default domain name. The default domain name is used as the domain part in the ObjectName of MBeans, if no domain is specified by the user.

The standard default domain name is defined in `DOMAIN`

MBeanServer(String)

```
public  MBeanServer(java.lang.String domain)
```

Creates an MBeanServer with the specified default domain name. The default domain name is used as the domain part in the ObjectName of MBeans if no domain is specified by the user.

Methods

addNotificationListener(ObjectName, NotificationListener, NotificationFilter, Object)

```
public void addNotificationListener(ObjectName name, NotificationListener listener,
            NotificationFilter filter, java.lang.Object handback)
```

Enables a couple (listener,handback) for a registered MBean to be added.

Parameters:
`name` - The name of the MBean on which the listener should be added.

`listener` - The listener object which will handles notifications emitted by the registered MBean.

`filter` - The filter object. If null, no filtering will be performed before handling notifications.

`handback` - The context to be sent to the listener when a notification is emitted.

Throws: `InstanceNotFoundException` - The MBean name doesn't correspond to a registered MBean.

---

addNotificationListener(ObjectName, ObjectName, NotificationFilter, Object)

```
public void addNotificationListener(ObjectName name, ObjectName listener,
            NotificationFilter filter, java.lang.Object handback)
```

Enables a couple (listener,handback) for a registered MBean to be added.

Parameters:

`name` - The name of the MBean on which the listener should be added.

`listener` - The listener name which will handles notifications emitted by the registered MBean.

`filter` - The filter object. If null, no filtering will be performed before handling notifications.

`handback` - The context to be sent to the listener when a notification is emitted.

Throws: `InstanceNotFoundException` - The MBean name or the listener name doesn't correspond to a registered MBean.

---

createMBean(String, ObjectName)

```
public ObjectInstance createMBean(java.lang.String className, ObjectName name)
```

Instantiates and registers a MBean in the MBeanServer. The MBean server will use the `DefaultLoaderRepository` to load the class of the MBean. An object name is associated to the MBean. If the object name given is null, the MBean can automatically provide its own name by implementing the `MBeanRegistration` interface. The call returns an ObjectInstance object representing the newly created MBean.

Parameters:

`className` - The class name of the MBean to be instantiated.

`name` - The object name of the MBean. May be null.

Returns: An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean.

Throws: `ReflectionException` - Wraps Wraps a ClassNotFoundException or a java.lang.Exception that occured trying to invoke the MBean's constructor.

`InstanceAlreadyExistsException` - The MBean is already under the control of the MBeanServer.

`MBeanRegistrationException` - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered.

`MBeanException` - The constructor of the MBean has thrown an exception

`NotCompliantMBeanException` - This class is not an JMX compliant MBean

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean.

---

createMBean(String, ObjectName, Object[], String[])

```
public ObjectInstance createMBean(java.lang.String className, ObjectName name,
            java.lang.Object[] params, java.lang.String[] signature)
```

Instantiates and registers a MBean in the MBeanServer. The MBean server will use the `DefaultLoaderRepository` to load the class of the MBean. An object name is associated to the MBean. If the object name given is null, the MBean can automatically provide its own name by implementing the `MBeanRegistration` interface. The call returns an ObjectInstance object representing the newly created MBean.

Parameters:

`className` - The class name of the MBean to be instantiated.

`name` - The object name of the MBean. May be null.

`params` - An array containing the parameters of the constructor to be invoked.

`signature` - An array containing the signature of the constructor to be invoked.

Returns: An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean.

Throws: `ReflectionException` - Wraps Wraps a ClassNotFoundException or a java.lang.Exception that occured trying to invoke the MBean's constructor.

`InstanceAlreadyExistsException` - The MBean is already under the control of the MBeanServer.

`MBeanRegistrationException` - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered.

`MBeanException` - The constructor of the MBean has thrown an exception

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean.

---

createMBean(String, ObjectName, ObjectName)

```
public ObjectInstance createMBean(java.lang.String className, ObjectName name,
            ObjectName loaderName)
```

Instantiates and registers a MBean in theMBeanServer. The class loader to be used is identified by its object name. An object name is associated to the MBean. If the object name of the loader is null, the ClassLoader that loaded the MBeanServer will be used. If the MBean's object name given is null, the MBean can automatically provide its own name by implementing the `MBeanRegistration` interface. The call returns an ObjectInstance object representing the newly created MBean.

Parameters:

`className` - The class name of the MBean to be instantiated.

`name` - The object name of the MBean. May be null.

`loaderName` - The object name of the class loader to be used.

Returns: An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean.

Throws: `ReflectionException` - Wraps Wraps a ClassNotFoundException or a java.lang.Exception that occured trying to invoke the MBean's constructor.

`InstanceAlreadyExistsException` - The MBean is already under the control of the MBeanServer.

`MBeanRegistrationException` - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered.

`MBeanException` - The constructor of the MBean has thrown an exception

`NotCompliantMBeanException` - This class is not an JMX compliant MBean

`InstanceNotFoundException` - The specified class loader is not registered in the MBeanServer.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean.

---

createMBean(String, ObjectName, ObjectName, Object[], String[])

```
public ObjectInstance createMBean(java.lang.String className, ObjectName name,
            ObjectName loaderName, java.lang.Object[] params, java.lang.String[]
            signature)
```

Instantiates and registers a MBean in the MBeanServer. The class loader to be used is identified by its object name. An object name is associated to the MBean. If the object name of the loader is not specified, the ClassLoader that loaded the MBeanServer will be used. If the MBean object name given is null, the MBean can automatically provide its own name by implementing the `MBeanRegistration` interface. The call returns an ObjectInstance object representing the newly created MBean.

Parameters:

`className` - The class name of the MBean to be instantiated.

`name` - The object name of the MBean. May be null.

`params` - An array containing the parameters of the constructor to be invoked.

`signature` - An array containing the signature of the constructor to be invoked.

`loaderName` - The object name of the class loader to be used.

Returns: An ObjectInstance, containing the ObjectName and the Java class name of the newly instantiated MBean.

Throws: `ReflectionException` - Wraps Wraps a ClassNotFoundException or a java.lang.Exception that occured trying to invoke the MBean's constructor.

`InstanceAlreadyExistsException` - The MBean is already under the control of the MBeanServer.

`MBeanRegistrationException` - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered.

`MBeanException` - The constructor of the MBean has thrown an exception

`InstanceNotFoundException` - The specified class loader is not registered in the MBeanServer.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null, the ObjectName passed in parameter contains a pattern or no ObjectName is specified for the MBean.

---

deserialize(ObjectName, byte[])

```
public java.io.ObjectInputStream deserialize(ObjectName name, byte[] data)
```

This method de-serializes a byte array in the context of the class loader of an MBean.

Parameters:

`name` - The name of the MBean whose class loader should be used for the de-serialization.

`data` - The byte array to be de-sererialized.

Returns: The de-serialized object stream.

Throws: `InstanceNotFoundException` - The specified MBean is not found.

`OperationsException` - Any of the usual Input/Output related exceptions.

---

deserialize(String, byte[])

```
public java.io.ObjectInputStream deserialize(java.lang.String className, byte[] data)
```

This method de-serializes a byte array in the context of a given MBean class loader. The class loader is the one that loaded the class with name "className".

Parameters:

`name` - The name of the class whose class loader should be used for the de-serialization.

`data` - The byte array to be de-sererialized.

Returns: The de-serialized object stream.

Throws: `OperationsException` - Any of the usual Input/Output related exceptions.

`ReflectionException` - The specified class could not be loaded by the default loader repository

---

deserialize(String, ObjectName, byte[])

```
public java.io.ObjectInputStream deserialize(java.lang.String className, ObjectName
            loaderName, byte[] data)
```

This method de-serializes a byte array in the context of a given MBean class loader. The class loader is the one that loaded the class with name "className". The name of the class loader to be used for loading the specified class is specified. If null, the MBean Server's class loader will be used.

Parameters:

`name` - The name of the class whose class loader should be used for the de-serialization.

`data` - The byte array to be de-sererialized.

`loaderName` - The name of the class loader to be used for loading the specified class. If null, the MBean Server's class loader will be used.

Returns: The de-serialized object stream.

Throws: `InstanceNotFoundException` - The specified class loader MBean is not found.

`OperationsException` - Any of the usual Input/Output related exceptions.

`ReflectionException` - The specified class could not be loaded by the specified class loader.

---

findJMXAgent(String)

`public static synchronized java.util.ArrayList findJMXAgent(java.lang.String AgentId)`

Return a list of MBeanServer objects. This static method allows a user to retrieve references on MBeanServer which have been instantiated in the Java Virtual Machine.

Parameters:

`AgentId` - The agent identifier of the MBeanServer to retrieve. If this parameter is null, all MBeanServers present in the JVM are returned.

Returns: A list of MBeanServer objects.

---

getAttribute(ObjectName, String)

`public java.lang.Object getAttribute(ObjectName name, java.lang.String attribute)`

Gets the value of a specific attribute of a named MBean. The MBean is identified by its object name.

Parameters:

`name` - The object name of the MBean from which the attribute is to be retrieved.

`attribute` - A String specifying the name of the attribute to be retrieved.

Returns: The value of the retrieved attribute.

Throws: `AttributeNotFoundException` - The specified attribute is not accessible in the MBean.

`MBeanException` - Wraps an exception thrown by the MBean's getter.

`InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`ReflectionException` - Wraps an java.lang.Exception thrown while trying to invoke the setter.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null or the attribute in parameter is null.

---

getAttributes(ObjectName, String[])

`public AttributeList getAttributes(ObjectName name, java.lang.String[] attributes)`

Enables the values of several attributes of a named MBean. The MBean is identified by its object name.

Parameters:

`name` - The object name of the MBean from which the attributes are to be retrieved.

`attributes` - A list of the attributes to be retrieved.

Returns: The list of the retrieved attributes.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`ReflectionException` - An exception occured trying to invoke the getAttributes method of a Dynamic MBean.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null or attributes in parameter is null.

getDefaultDomain()

`public java.lang.String getDefaultDomain()`

Returns the default domain used for the MBean naming. The default domain name is used as the domain part in the ObjectName of MBeans if no domain is specified by the user.

getMBeanCount()

`public java.lang.Integer getMBeanCount()`

Returns the number of MBeans registered in the MBeanServer.

getMBeanInfo(ObjectName)

`public MBeanInfo getMBeanInfo(ObjectName name)`

This method discovers the attributes and operations that an MBean exposes for management.

Parameters:
`name` - The name of the MBean to analyze

Returns: An instance of MBeanInfo allowing to retrieve all attributes and operations of this MBean.

Throws: `IntrospectionException` - An exception occurs during introspection.

`InstanceNotFoundException` - The specified MBean is not found.

`ReflectionException` - An exception occured trying to invoke the getMBeanInfo of a Dynamic MBean.

getObjectInstance(ObjectName)

`public ObjectInstance getObjectInstance(ObjectName name)`

Gets the ObjectInstance for a given MBean registered with the MBeanServer.

Parameters:
`name` - The object name of the MBean.

Returns: The ObjectInstance associated to the MBean specified by name.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

instantiate(String)

`public java.lang.Object instantiate(java.lang.String className)`

Instantiates an object using the list of all class loaders registered in the MBeanServer (`DefaultLoaderRepository`). The object's class should have a public constructor. It returns a reference to the newly created object. The newly created object is not be registered in the MBeanServer.

Parameters:
`className` - The class name of the object to be instantiated.

Returns: The newly instantiated object.

Throws: `ReflectionException` - Wraps a ClassNotFoundException or the java.lang.Exception that occured trying to invoke the object's constructor.

`MBeanException` - The constructor of the object has thrown an exception

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null.

---

instantiate(String, Object[], String[])

```
public java.lang.Object instantiate(java.lang.String className, java.lang.Object[]
            params, java.lang.String[] signature)
```

Instantiates an object using the list of all class loaders registered in the MBeanServer (`DefaultLoaderRepository`). The object's class should have a public constructor. The call returns a reference to the newly created object. The newly created object is not be registered in the MBeanServer.

Parameters:

`className` - The class name of the object to be instantiated.

`params` - An array containing the parameters of the constructor to be invoked.

`signature` - An array containing the signature of the constructor to be invoked.

Returns: The newly instantiated object.

Throws: `ReflectionException` - Wraps a ClassNotFoundException or the java.lang.Exception that occured trying to invoke the object's constructor.

`MBeanException` - The constructor of the object has thrown an exception

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null.

---

instantiate(String, ObjectName)

```
public java.lang.Object instantiate(java.lang.String className, ObjectName loaderName)
```

Instantiates an object using the class Loader specified by its ObjectName. If the loader name is null, the ClassLoader that loaded the MBean Server will be used. The object's class should have a public constructor. It returns a reference to the new created object. The newly created object is not be registered in the MBeanServer.

Parameters:

`className` - The class name of the MBean to be instantiated.

`loaderName` - The object name of the class loader to be used.

Returns: The newly instantiated object.

Throws: `ReflectionException` - Wraps a ClassNotFoundException or the java.lang.Exception that occured trying to invoke the object's constructor.

`MBeanException` - The constructor of the object has thrown an exception.

`InstanceNotFoundException` - The specified class loader is not registered in the MBaenServer.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null.

instantiate(String, ObjectName, Object[], String[])

```
public java.lang.Object instantiate(java.lang.String className, ObjectName loaderName,
            java.lang.Object[] params, java.lang.String[] signature)
```

Instantiates an object. The class loader to be used is identified by its object name. If the object name of the loader is null, the ClassLoader that loaded the MBeanServer will be used. The object's class should have a public constructor. The call returns a reference to the newly created object. The newly created object is not be registered in the MBeanServer.

Parameters:

`className` - The class name of the object to be instantiated.

`params` - An array containing the parameters of the constructor to be invoked.

`signature` - An array containing the signature of the constructor to be invoked.

`loaderName` - The object name of the class loader to be used.

Returns: The newly instantiated object.

Throws: `ReflectionException` - Wraps a ClassNotFoundException or the java.lang.Exception that occured trying to invoke the object's constructor.

`MBeanException` - The constructor of the object has thrown an exception

`InstanceNotFoundException` - The specified class loader is not registered in the MBeanServer.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The className passed in parameter is null.

invoke(ObjectName, String, Object[], String[])

```
public java.lang.Object invoke(ObjectName name, java.lang.String actionName,
            java.lang.Object[] params, java.lang.String[] signature)
```

Invokes an action on an MBean.

Parameters:

`name` - The object name of the MBean on which the method is to be invoked.

`actionName` - The name of the action to be invoked.

`params` - An array containing the parameters to be set when the action is invoked

`signature` - An array containing the signature of the action. The class objects will be loaded using the same class loader as the one used for loading the MBean on which the action was invoked.

Returns: The object returned by the action, which represents the result ofinvoking the action on the specified MBean.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`MBeanException` - Wraps an exception thrown by the MBean's invoked method.

`ReflectionException` - Wraps an java.lang.Exception thrown while trying to invoke the method.

isRegistered(ObjectName)

```
public boolean isRegistered(ObjectName name)
```

Checks whether an MBean, identified by its object name, is already registered with the MBeanServer.

Parameters:

`name` - The object name of the MBean to be checked.

Returns: True if the MBean is already registered in the MBeanServer, false otherwise.

Throws: `RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null.

---

queryMBeans(ObjectName, QueryExp)

`public java.util.Set queryMBeans(ObjectName name, QueryExp query)`

Gets MBeans controlled by the MBeanServer. This method allows any of the following to be obtained: All MBeans, a set of MBeans specified by pattern matching on the ObjectName and/or a Query expression, a specific MBean. When the object name is null or empty, all objects are to be selected (and filtered if a query is specified). It returns the set of ObjectInstance objects (containing the ObjectName and the Java Class name) for the selected MBeans.

Parameters:

`name` - The object name pattern identifying the MBeans to be retrieved. If null or empty all the MBeans registered will be retrieved.

`query` - The query expression to be applied for selecting MBeans.

Returns: A set containing the ObjectInstance objects for the selected MBeans. If no MBean satisfies the query an empty list is returned.

---

queryNames(ObjectName, QueryExp)

`public java.util.Set queryNames(ObjectName name, QueryExp query)`

Gets the names of MBeans controlled by the MBeanServer. This method enables any of the following to be obtained: The names of all MBeans, the names of a set of MBeans specified by pattern matching on the ObjectName and/or a Query expression, a specific MBean name (equivalent to testing whether an MBean is registered). When the object name is null or empty, all objects are to be selected (and filtered if a query is specified). It returns the set of ObjectNames for the MBeans selected.

Parameters:

`name` - The object name pattern identifying the MBean names to be retrieved. If null or empty, the name of all registered MBeans will be retrieved.

`query` - The query expression to be applied for selecting MBeans.

Returns: A set containing the ObjectNames for the MBeans selected. If no MBean satisfies the query an empty list is returned.

---

registerMBean(Object, ObjectName)

`public ObjectInstance registerMBean(java.lang.Object object, ObjectName name)`

Registers a pre-existing object as an MBean with the MBeanServer. If the object name given is null, the MBean may automatically provide its own name by implementing the `MBeanRegistration` interface. The call returns an ObjectInstance object representing the registered MBean.

Parameters:

`object` - The MBean to be registered as an MBean.

`name` - The object name of the MBean. May be null.

Returns: The ObjectInstance for the MBean that has been registered.

Throws: `InstanceAlreadyExistsException` - The MBean is already under the control of the MBeanServer.

`MBeanRegistrationException` - The preRegister (MBeanRegistration interface) method of the MBean has thrown an exception. The MBean will not be registered.

`NotCompliantMBeanException` - This object is not an JMX compliant MBean

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object passed in parameter is null or no object name is specified.

---

removeNotificationListener(ObjectName, NotificationListener)

```
public void removeNotificationListener(ObjectName name, NotificationListener listener)
```

Enables a listener for an MBean to be removed.

Parameters:

`name` - The name of the MBean on which the listener should be removed.

`listener` - The listener object which will handles notifications emitted by the registered MBean.This method will remove all information related to this listener.

Throws: `InstanceNotFoundException` - The MBean name doesn't correspond to a registered MBean.

`ListenerNotFoundException` - The couple (listener,handback) is not registered in the MBean. The exception message contains either "listener", "handback" or the object name depending on which object cannot be found.

---

removeNotificationListener(ObjectName, ObjectName)

```
public void removeNotificationListener(ObjectName name, ObjectName listener)
```

Enables a listener for an MBean to be removed.

Parameters:

`name` - The name of the MBean on which the listener should be removed.

`listener` - The listener name which will handles notifications emitted by the registered MBean. This method will remove all information related to this listener.

Throws: `InstanceNotFoundException` - The MBean name or the listener name doesn't correspond to a registered MBean

`ListenerNotFoundException` - The couple (listener,handback) is not registered in the MBean. The exception message contains either "listener", "handback" or the object name depending on which object cannot be found.

---

setAttribute(ObjectName, Attribute)

```
public void setAttribute(ObjectName name, Attribute attribute)
```

Sets the value of a specific attribute of a named MBean. The MBean is identified by its object name.

Parameters:

`name` - The name of the MBean within which the attribute is to be set.

`attribute` - The identification of the attribute to be set and the value it is to be set to.

Returns: The value of the attribute that has been set.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`AttributeNotFoundException` - The specified attribute is not accessible in the MBean.

`InvalidAttributeValueException` - The specified value for the attribute is not valid.

`MBeanException` - Wraps an exception thrown by the MBean's setter.

`ReflectionException` - Wraps an java.lang.Exception thrown while trying to invoke the setter.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null or the attribute in parameter is null.

---

setAttributes(ObjectName, AttributeList)

```
public AttributeList setAttributes(ObjectName name, AttributeList attributes)
```

Sets the values of several attributes of a named MBean. The MBean is identified by its object name.

Parameters:

`name` - The object name of the MBean within which the attributes are to be set.

`attributes` - A list of attributes: The identification of the attributes to be set and the values they are to be set to.

Returns: The list of attributes that were set, with their new values.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`ReflectionException` - An exception occured trying to invoke the getAttributes method of a Dynamic MBean.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null or attributes in parameter is null.

---

unregisterMBean(ObjectName)

```
public void unregisterMBean(ObjectName name)
```

De-registers an MBean from the MBeanServer. The MBean is identified by its object name. Once the method has been invoked, the MBean may no longer be accessed by its object name.

Parameters:

`name` - The object name of the MBean to be de-registered.

Throws: `InstanceNotFoundException` - The specified MBean is not registered in the MBeanServer.

`MBeanRegistrationException` - The preDeregister (MBeanRegistration interface) method of the MBean has thrown an exception.

`RuntimeOperationsException` - Wraps an IllegalArgumentException: The object name in parameter is null or the MBean you are trying to de-register is the `MBeanServerDelegate` MBean.

javax.management
MBeanServerDelegate

Syntax

`public class MBeanServerDelegate extends java.lang.Object implements MBeanServerDelegateMBean, NotificationBroadcaster`

```
java.lang.Object
  |
  +--javax.management.MBeanServerDelegate
```

All Implemented Interfaces: `MBeanServerDelegateMBean`, `NotificationBroadcaster`

Description

This class is a MBeanServer representation from the management point of view.

Member Summary

Constructors

	`MBeanServerDelegate()`

Methods

`void`	`addNotificationListener(NotificationListener, NotificationFilter, Object)`
`String`	`getMBeanServerId()`
`String`	`getMBeanServerVersion()`
`MBeanNotificationInfo[]`	`getNotificationInfo()`
`void`	`removeNotificationListener(NotificationListener)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

MBeanServerDelegate()

`public MBeanServerDelegate()`

Create a MBeanServerDelegate object.

Methods

---

addNotificationListener(NotificationListener, NotificationFilter, Object)

```
public synchronized void addNotificationListener(NotificationListener listener,
            NotificationFilter filter, java.lang.Object handback)
```

Enables a couple (listener,handback) for a registered MBean to be added.

Specified By: `addNotificationListener(NotificationListener, NotificationFilter, Object)` in interface `NotificationBroadcaster`

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

`filter` - The filter object. If not specified, no filtering will be performed before handling notifications.

`handback` - The context to be sent to the listener when a notification is emitted.

Throws: `java.lang.IllegalArgumentException` - Listener parameter is null.

---

getMBeanServerId()

```
public java.lang.String getMBeanServerId()
```

Get the MBeanServer agent identification.

Specified By: `getMBeanServerId()` in interface `MBeanServerDelegateMBean`

Returns: The MBeanServer agent identification.

---

getMBeanServerVersion()

```
public java.lang.String getMBeanServerVersion()
```

Get the MBeanServer version.

Specified By: `getMBeanServerVersion()` in interface `MBeanServerDelegateMBean`

Returns: The MBeanServer version.

---

getNotificationInfo()

```
public MBeanNotificationInfo getNotificationInfo()
```

Returns a NotificationInfo object contaning the name of the Java class of the notification and the notification types sent.

Specified By: `getNotificationInfo()` in interface `NotificationBroadcaster`

---

removeNotificationListener(NotificationListener)

```
public synchronized void removeNotificationListener(NotificationListener listener)
```

Enables a listener for an MBean to be removed. All couple (listener, handback) are removed.

Specified By: `removeNotificationListener(NotificationListener)` in interface `NotificationBroadcaster`

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

Throws: `ListenerNotFoundException` **-** The listener is not registered in the MBean.

javax.management
MBeanServerDelegateMBean

Syntax

`public interface MBeanServerDelegateMBean`

All Known Implementing Classes: `MBeanServerDelegate`

Description

This interface defines the management interface of an object of class MBeanServerDelegate.

Member Summary		
Methods		
	`String`	`getMBeanServerId()`
	`String`	`getMBeanServerVersion()`

Methods

---

getMBeanServerId()

`public java.lang.String getMBeanServerId()`

Get the MBeanServer agent identification.

Returns: The MBeanServer agent identification.

---

getMBeanServerVersion()

`public java.lang.String getMBeanServerVersion()`

Get the MBeanServer version.

Returns: The MBeanServer version.

javax.management
MBeanServerNotification

Syntax

`public class MBeanServerNotification extends Notification`

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--Notification
              |
              +--javax.management.MBeanServerNotification
```

All Implemented Interfaces: java.io.Serializable

Description

The MBeanServerNotification class represents an notification emitted by the MBean Server. The MBean Server emits these types of notifications: MBean registration, MBean de-registration.

Member Summary

Fields		
	`String`	`REGISTRATION_NOTIFICATION`
	`String`	`UNREGISTRATION_NOTIFICATION`
Constructors		
		`MBeanServerNotification(String, Object, long, Vector)`
Methods		
	`Vector`	`getMBeanNames()`

Inherited Member Summary

Fields inherited from class java.util.EventObject

`source`

Methods inherited from class `Notification`

`getMessage()`, `getSequenceNumber()`, `getSource()`, `getTimeStamp()`, `getType()`, `getUserData()`, `setSource(Object)`, `setUserData(Object)`

Methods inherited from class java.util.EventObject

`toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Fields

REGISTRATION_NOTIFICATION

```
public static final java.lang.String REGISTRATION_NOTIFICATION
```

Notification type denoting that a MBean has been registered. Value is "JMX.mbean.registered".

UNREGISTRATION_NOTIFICATION

```
public static final java.lang.String UNREGISTRATION_NOTIFICATION
```

Notification type denoting that a MBean has been uregsitered. Value is "JMX.mbean.unregistered".

Constructors

MBeanServerNotification(String, Object, long, Vector)

```
public  MBeanServerNotification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.util.Vector objectNames)
```

Creates a MBeanServerNotification object with the specified object names of the MBeans that caused the notification and the specified notification type.

Parameters:

`type` - A string denoting the type of the notification. Set it to one these values: REGISTRATION_NOTIFICATION, UNREGISTRATION_NOTIFICATION

`source` - The MBeanServerNotification object responsible to forward MBeanServer notification.

`objectNames` - A list of the object names of the MBeans that caused the notification.

Methods

---

getMBeanNames()

```
public java.util.Vector getMBeanNames()
```

Returns a vector of object names of the MBeans that caused the notification

javax.management
NotCompliantMBeanException

Syntax

`public class NotCompliantMBeanException extends OperationsException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.NotCompliantMBeanException
```

All Implemented Interfaces: java.io.Serializable

Description

This exception occurs when trying to register in the MBeanServer an object that is not a JMX compliant MBean.

Member Summary	
Constructors	
	`NotCompliantMBeanException()`
	`NotCompliantMBeanException(String)`

Inherited Member Summary
Methods inherited from class java.lang.Throwable
`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

NotCompliantMBeanException()

```
public  NotCompliantMBeanException()
```

Default constructor.

NotCompliantMBeanException(String)

```
public  NotCompliantMBeanException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
Notification

Syntax

```
public class Notification extends java.util.EventObject
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--javax.management.Notification
```

Direct Known Subclasses: `AttributeChangeNotification`, `MBeanServerNotification`, `MonitorNotification`, `TraceNotification`

All Implemented Interfaces: java.io.Serializable

Description

The Notification class represents a notification emitted by an MBean.

Member Summary

Constructors

	`Notification(String, Object, long)`
	`Notification(String, Object, long, Date)`
	`Notification(String, Object, long, Date, String)`
	`Notification(String, Object, long, String)`

Methods

`String`	`getMessage()`
`long`	`getSequenceNumber()`
`Object`	`getSource()`
`Date`	`getTimeStamp()`
`String`	`getType()`
`Object`	`getUserData()`
`void`	`setSource(Object)`
`void`	`setUserData(Object)`

Inherited Member Summary

Fields inherited from class java.util.EventObject

`source`

Methods inherited from class java.util.EventObject

`toString`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

Notification(String, Object, long)

```
public  Notification(java.lang.String type, java.lang.Object source, long
            sequenceNumber)
```

Creates a Notification object. The notification timeStamp is set to the curent date.

Parameters:

`type` - The notification type.

`source` - The notification source.

`sequenceNumber` - The notification sequence number within the source object.

Notification(String, Object, long, Date)

```
public  Notification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.util.Date timeStamp)
```

Creates a Notification object.

Parameters:

`type` - The notification type.

`source` - The notification source.

`sequenceNumber` - The notification sequence number within the source object.

`date` - The notification emission date.

Notification(String, Object, long, Date, String)

```
public  Notification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.util.Date timeStamp, java.lang.String message)
```

Creates a Notification object.

Parameters:

`type` - The notification type.

`source` - The notification source.

`sequenceNumber` - The notification sequence number within the source object.

`date` - The notification emission date.

`message` - the detail message.

Notification(String, Object, long, String)

```
public  Notification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.lang.String message)
```

Creates a Notification object. The notification timeStamp is set to the curent date.

Parameters:

`type` - The notification type.

`source` - The notification source.

`sequenceNumber` - The notification sequence number within the source object.

`message` - the detail message.

Methods

---

getMessage()

```
public java.lang.String getMessage()
```

Get the notification message.

Returns: The message string of this notification object.

---

getSequenceNumber()

```
public long getSequenceNumber()
```

Get the notification sequence number.

Returns: The notification sequence number within the source object.

---

getSource()

```
public java.lang.Object getSource()
```

get the source object name

Overrides: java.util.EventObject.getSource() in class java.util.EventObject

Returns: The MBean object name on which the notification initially occurred.

---

getTimeStamp()

```
public java.util.Date getTimeStamp()
```

Get the notification timestamp.

Returns: The notification timestamp.

---

getType()

```
public java.lang.String getType()
```

Get the notification type.

Returns: The notification type.

getUserData()

`public java.lang.Object getUserData()`

Get the user data.

Returns: The user data object

setSource(Object)

`public void setSource(java.lang.Object source)`

Set the source object name

Returns: The notification sequence number within the source object.

Throws: `java.lang.IllegalArgumentException` - The source is not a ObjectName

setUserData(Object)

`public void setUserData(java.lang.Object userData)`

Set the user data.

Parameters:
`userData` - The user data object

javax.management
NotificationBroadcaster

Syntax

`public interface NotificationBroadcaster`

All Known Implementing Classes: `MBeanServerDelegate`, `NotificationBroadcasterSupport`

Description

This interface should be implemented by a registered MBean. It allows a listener to be registered within the MBean as event listener.

Member Summary

Methods

`void`	`addNotificationListener(NotificationListener, NotificationFilter, Object)`
`MBeanNotificationInfo[]`	`getNotificationInfo()`
`void`	`removeNotificationListener(NotificationListener)`

Methods

addNotificationListener(NotificationListener, NotificationFilter, Object)

```
public void addNotificationListener(NotificationListener listener, NotificationFilter
            filter, java.lang.Object handback)
```

Enables a couple (listener,handback) for a registered MBean to be added.

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

`filter` - The filter object. If not specified, no filtering will be performed before handling notifications.

`handback` - The context to be sent to the listener when a notification is emitted.

Throws: `java.lang.IllegalArgumentException` - Listener parameter is null.

getNotificationInfo()

```
public MBeanNotificationInfo getNotificationInfo()
```

Returns a NotificationInfo object contaning the name of the Java class of the notification and the notification types sent.

removeNotificationListener(NotificationListener)

`public void removeNotificationListener(NotificationListener listener)`

Enables a listener for an MBean to be removed. All couple (listener, handback) are removed.

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

Throws: `ListenerNotFoundException` - The listener is not registered in the MBean.

javax.management
NotificationBroadcasterSupport

Syntax

```
public class NotificationBroadcasterSupport extends java.lang.Object implements
   NotificationBroadcaster
```

```
java.lang.Object
  |
  +--javax.management.NotificationBroadcasterSupport
```

Direct Known Subclasses: `Monitor`, `Timer`

All Implemented Interfaces: `NotificationBroadcaster`

Description

This class provides an implementation of NotificationBroadcaster. It could be used as a super class of MBean to deal with notification. If inheritance can't be used, the following code can be used as an example for NotificationBroadcaster implementation.

Member Summary	
Constructors	
	`NotificationBroadcasterSupport()`
Methods	
`void`	`addNotificationListener(NotificationListener, NotificationFilter, Object)`
`MBeanNotificationInfo[]`	`getNotificationInfo()`
`void`	`removeNotificationListener(NotificationListener)`
`void`	`sendNotification(Notification)`

Inherited Member Summary
Methods inherited from class java.lang.Object
`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

NotificationBroadcasterSupport()

```
public  NotificationBroadcasterSupport()
```

Methods

---

addNotificationListener(NotificationListener, NotificationFilter, Object)

```
public synchronized void addNotificationListener(NotificationListener listener,
            NotificationFilter filter, java.lang.Object handback)
```

Enables a couple (listener,handback) for a registered MBean to be added.

Specified By: `addNotificationListener(NotificationListener, NotificationFilter, Object)` in interface `NotificationBroadcaster`

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

`filter` - The filter object. If not specified, no filtering will be performed before handling notifications.

`handback` - The context to be sent to the listener when a notification is emitted.

Throws: `java.lang.IllegalArgumentException` - Listener parameter is null.

---

getNotificationInfo()

```
public MBeanNotificationInfo getNotificationInfo()
```

Returns a NotificationInfo object contaning the name of the Java class of the notification and the notification types sent.

Specified By: `getNotificationInfo()` in interface `NotificationBroadcaster`

---

removeNotificationListener(NotificationListener)

```
public synchronized void removeNotificationListener(NotificationListener listener)
```

Enables a listener for an MBean to be removed. All couple (listener, handback) are removed.

Specified By: `removeNotificationListener(NotificationListener)` in interface `NotificationBroadcaster`

Parameters:

`listener` - The listener object which will handles notifications emitted by the registered MBean.

Throws: `ListenerNotFoundException` - The listener is not registered in the MBean.

---

sendNotification(Notification)

```
public synchronized void sendNotification(Notification notification)
```

Enables a MBean to send a notification.

Parameters:

`notification` - The notification to send.

javax.management
NotificationFilter

Syntax

`public interface NotificationFilter extends java.io.Serializable`

All Superinterfaces: java.io.Serializable

All Known Implementing Classes: `AttributeChangeNotificationFilter`, `NotificationFilterSupport`, `TraceFilter`

Description

This interface should be implemented by a any MBean acting as a notification filter. It allows a registered notification listener to filter the notifications of interest.

Member Summary		
Methods		
	`boolean`	`isNotificationEnabled(Notification)`

Methods

isNotificationEnabled(Notification)

`public boolean isNotificationEnabled(Notification notification)`

Invoked before sending the specified notification to the listener.

Parameters:
`notification` - The notification to be sent.

Returns: True if the notification has to be sent to the listener, false otherwise.

javax.management
NotificationFilterSupport

Syntax

```
public class NotificationFilterSupport extends java.lang.Object implements NotificationFilter,
    java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.NotificationFilterSupport
```

All Implemented Interfaces: `NotificationFilter`, java.io.Serializable

Description

This class provides an implementation of the NotificationFilter interface. The filtering is performed on the notification type attribute.

This class manages a list of enabled notification types. A method allows users to enable/disable as many notification types as required.

Member Summary

Constructors

	`NotificationFilterSupport()`

Methods

`void`	`disableAllTypes()`
`void`	`disableType(String)`
`void`	`enableType(String)`
`Vector`	`getEnabledTypes()`
`boolean`	`isNotificationEnabled(Notification)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

NotificationFilterSupport()

```
public  NotificationFilterSupport()
```

Methods

---

disableAllTypes()

`public synchronized void disableAllTypes()`

Disables all the notification types.

---

disableType(String)

`public synchronized void disableType(java.lang.String prefix)`

Disables all the notifications whose type matches the specified prefix to be sent to the listener. If the specified prefix is not in the list of enabled notification types, this method has no effect.

Parameters:
`prefix` - The prefix.

---

enableType(String)

`public synchronized void enableType(java.lang.String prefix)`

Enables all the notifications whose type matches the specified prefix to be sent to the listener. If the specified prefix is already in the list of enabled notification types, this method has no effect.

Parameters:
`prefix` - The prefix.

Throws: `java.lang.IllegalArgumentException` - The prefix parameter is null.

---

getEnabledTypes()

`public synchronized java.util.Vector getEnabledTypes()`

Gets all the enabled notification types for this filter.

Returns: The list containing all the enabled notification types.

---

isNotificationEnabled(Notification)

`public synchronized boolean isNotificationEnabled(Notification notification)`

Invoked before sending the specified notification to the listener. This filter compares the notification type with each enabled type. If the notification type matches one of the enabled type, the notification has to be sent to the listener and this method returns true.

Specified By: `isNotificationEnabled(Notification)` in interface
`NotificationFilter`

Parameters:
`notification` - The notification to be sent.

Returns: True if the notification has to be sent to the listener, false otherwise.

javax.management
NotificationListener

Syntax

`public interface NotificationListener extends java.util.EventListener`

All Superinterfaces: java.util.EventListener

All Known Implementing Classes: `TraceListener`

Description

This interface listener should to receive JMX notification.

Member Summary

Methods	
`void`	`handleNotification(Notification, Object)`

Methods

handleNotification(Notification, Object)

`public void handleNotification(Notification notification, java.lang.Object handback)`

Invoked when an JMAPI notification occurs.

Parameters:

`notification` - The notification.

`handback` - An opaque object which helps the listener to associate information regarding the MBean emitter. This object was passed to the MBean during the addListener call and resend, without modification, to the listener. The MBean object should to use or modify the object.

javax.management
ObjectInstance

Syntax

```
public class ObjectInstance extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.ObjectInstance
```

All Implemented Interfaces:

java.io.Serializable

Description

This class is used for representing an object instance; that is, an object name and class.

Member Summary	
Constructors	
	`ObjectInstance(ObjectName, String)`
	`ObjectInstance(String, String)`
Methods	
`boolean`	`equals(Object)`
`String`	`getClassName()`
`ObjectName`	`getObjectName()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ObjectInstance(ObjectName, String)

```
public  ObjectInstance(ObjectName objectName, java.lang.String className)
```

Allows an object instance to be created given an object name and the full class name, including the package name.

Parameters:

`objectName` - The object name.

`className` - The full class name, including the package name, of the object instance.

ObjectInstance(String, String)

```
public  ObjectInstance(java.lang.String objectName, java.lang.String className)
```

Allows an object instance to be created given a string representation of an object name and the full class name, including the package name.

Parameters:

`objectName` - A string representation of the object name.

`className` - The full class name, including the package name, of the object instance.

Throws: `MalformedObjectNameException` - The string passes in parameter does not have the right format.

Methods

equals(Object)

```
public boolean equals(java.lang.Object object)
```

Compares the current object instance with another object instance.

Overrides: java.lang.Object.equals(java.lang.Object) in class java.lang.Object

Parameters:

`object` - The object instance that the current object instance is to be compared with.

Returns: True if the two object instances are equal, otherwise false.

getClassName()

```
public java.lang.String getClassName()
```

Returns the class part

getObjectName()

```
public ObjectName getObjectName()
```

Returns the object name part.

javax.management

ObjectName

Syntax

```
public class ObjectName extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.ObjectName
```

All Implemented Interfaces:

java.io.Serializable

Description

This class is used for representing an object name. An instance of this class can be used to represent:

- An object name
- An object name pattern, within the context of a query

Member Summary

Constructors	
	`ObjectName(String)`
	`ObjectName(String, Hashtable)`
	`ObjectName(String, String, String)`
Methods	
`boolean`	`equals(Object)`
`String`	`getCanonicalKeyPropertyListString()`
`String`	`getCanonicalName()`
`String`	`getDomain()`
`String`	`getKeyProperty(String)`
`Hashtable`	`getKeyPropertyList()`
`String`	`getKeyPropertyListString()`
`int`	`hashCode()`
`boolean`	`isPattern()`
`String`	`toString()`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, finalize, getClass, notify, notifyAll, wait, wait, wait`

Constructors

ObjectName(String)

```
public  ObjectName(java.lang.String name)
```

Allows an object name to be created from the given string. If the string does not have the right format an MalformedObjectNameException should be raised.

Parameters:
`name` - A string representation of the object name.

Throws: `MalformedObjectNameException` - the string passed in parameter does not have the right format.

ObjectName(String, Hashtable)

```
public  ObjectName(java.lang.String domain, java.util.Hashtable table)
```

Allows an object name with several key properties to be created. If the parameters do not have the right format an MalformedObjectNameException should be raised.

Parameters:
`domain` - The domain part of the object name.

`table` - A hash table containing one or more search keys. Each search key is an attribute-value pair.

Throws: `MalformedObjectNameException` - the string passed in parameter does not have the right format.

ObjectName(String, String, String)

```
public  ObjectName(java.lang.String domain, java.lang.String key, java.lang.String
            value)
```

Allows an object name to be created using only one key property. If the parameters do not have the right format an MalformedObjectNameException should be raised.

Parameters:
`domain` - The domain part of the object name.

`key` - The attribute in the key property of the object name.

`value` - The value in the key property of the object name.

Throws: `MalformedObjectNameException` - the string passed in parameter does not have the right format.

Methods

---

equals(Object)

```
public boolean equals(java.lang.Object object)
```

Compares the current object name with another object name.

Overrides: java.lang.Object.equals(java.lang.Object) in class java.lang.Object

Parameters:
`object` - The object name that the current object name is to be compared with.

Returns: True if the two object names are equal, otherwise false.

---

getCanonicalKeyPropertyListString()

```
public java.lang.String getCanonicalKeyPropertyListString()
```

Returns a string representation of the list of key properties in which the key properties are sorted in lexical order. This is used in lexicographic comparisons performed in order to select MBeans based on their key property list.

---

getCanonicalName()

```
public java.lang.String getCanonicalName()
```

Returns the canonical form of the name; that is, a string representation where the properties are sorted in lexical order.

---

getDomain()

```
public java.lang.String getDomain()
```

Returns the domain part.

---

getKeyProperty(String)

```
public java.lang.String getKeyProperty(java.lang.String property)
```

Obtains the value of a key property in a key property.

Parameters:
`property` - The property whose value is to be obtained.

Returns: The value of the property.

---

getKeyPropertyList()

```
public java.util.Hashtable getKeyPropertyList()
```

Returns a pointer to the list of key properties.

---

getKeyPropertyListString()

```
public java.lang.String getKeyPropertyListString()
```

Returns a string representation of the list of key properties specified at creation time.

---

hashCode()

`public int hashCode()`

Returns a hash code for this object name.

Overrides: java.lang.Object.hashCode() in class java.lang.Object

isPattern()

`public boolean isPattern()`

Checks if a name to be used is a pattern for a query.

Returns: True if the name is a pattern, otherwise false.

toString()

`public java.lang.String toString()`

Returns a string representation of the object name.

Overrides: java.lang.Object.toString() in class java.lang.Object

javax.management
OperationsException

Syntax

```
public class OperationsException extends JMException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--javax.management.OperationsException
```

Direct Known Subclasses: `AttributeNotFoundException`, `InstanceAlreadyExistsException`, `InstanceNotFoundException`, `IntrospectionException`, `InvalidAttributeValueException`, `ListenerNotFoundException`, `MalformedObjectNameException`, `NotCompliantMBeanException`, `ServiceNotFoundException`

All Implemented Interfaces: java.io.Serializable

Description

This class represents exceptions thrown in the MBeanServer when performing operations on MBeans.

Member Summary

Constructors

`OperationsException()`
`OperationsException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

OperationsException()

```
public  OperationsException()
```

Default constructor.

OperationsException(String)

```
public  OperationsException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
Query

Syntax

```
public class Query extends java.lang.Object
```

```
java.lang.Object
  |
  +--javax.management.Query
```

Description

The Query class supports construction of query object constraints. The static methods provided return query expressions that may be used in listing and enumerating MBeans. Individual constraint constructing methods allow only appropriate types as arguments. Composition of calls can construct arbitrary nestings of constraints, as the following example illustrates: QueryExp exp = Query.and(Query.gt(Query.attr("age"),Query.value(5)), Query.match(Query.attr("name"), Query.value("Smith")));

Member Summary

Fields

`int`	`DIV`
`int`	`EQ`
`int`	`GE`
`int`	`GT`
`int`	`LE`
`int`	`LT`
`int`	`MINUS`
`int`	`PLUS`
`int`	`TIMES`

Constructors

	`Query()`

Methods

`QueryExp`	`and(QueryExp, QueryExp)`
`QueryExp`	`anySubString(AttributeValueExp, StringValueExp)`
`AttributeValueExp`	`attr(String)`
`AttributeValueExp`	`attr(String, String)`
`QueryExp`	`between(ValueExp, ValueExp, ValueExp)`
`AttributeValueExp`	`classattr()`
`ValueExp`	`div(ValueExp, ValueExp)`
`QueryExp`	`eq(ValueExp, ValueExp)`
`QueryExp`	`finalSubString(AttributeValueExp, StringValueExp)`
`QueryExp`	`geq(ValueExp, ValueExp)`
`QueryExp`	`gt(ValueExp, ValueExp)`
`QueryExp`	`in(ValueExp, ValueExp[])`
`QueryExp`	`initialSubString(AttributeValueExp, StringValueExp)`
`QueryExp`	`leq(ValueExp, ValueExp)`
`QueryExp`	`lt(ValueExp, ValueExp)`
`QueryExp`	`match(AttributeValueExp, StringValueExp)`

Member Summary

`ValueExp`	`minus(ValueExp, ValueExp)`
`QueryExp`	`not(QueryExp)`
`QueryExp`	`or(QueryExp, QueryExp)`
`ValueExp`	`plus(ValueExp, ValueExp)`
`ValueExp`	`times(ValueExp, ValueExp)`
`ValueExp`	`value(boolean)`
`ValueExp`	`value(double)`
`ValueExp`	`value(float)`
`ValueExp`	`value(int)`
`ValueExp`	`value(long)`
`ValueExp`	`value(Number)`
`StringValueExp`	`value(String)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Fields

DIV

```
public static final int DIV
```

EQ

```
public static final int EQ
```

GE

```
public static final int GE
```

GT

```
public static final int GT
```

LE

```
public static final int LE
```

LT

`public static final int LT`

MINUS

`public static final int MINUS`

PLUS

`public static final int PLUS`

TIMES

`public static final int TIMES`

Constructors

Query()

`public  Query()`

Basic constructor.

Methods

---

and(QueryExp, QueryExp)

`public static QueryExp and(QueryExp q1, QueryExp q2)`

Returns a query expression that is the conjunction of two other query expressions.

Parameters:

`q1` - A query expression

`q2` - Another query expression

Returns: The conjunction of the two arguments.

---

anySubString(AttributeValueExp, StringValueExp)

`public static QueryExp anySubString(AttributeValueExp a, StringValueExp s)`

Returns a query expression that represents a matching constraint on a string argument. The value must contain the given string value.

Parameters:

`a` - An attribute expression

`s` - A string value expression representing the substring.

Returns: The constraint that a matches s.

---

attr(String)

`public static AttributeValueExp attr(java.lang.String name)`

Returns a new attribute expression.

Parameters:
`name` - The name of the attribute.

Returns: An attribute expression for the attribute named name.

---

attr(String, String)

`public static AttributeValueExp attr(java.lang.String className, java.lang.String name)`

Returns a new qualified attribute expression.

Parameters:
`className` - The name of the class possessing the attribute

`name` - The name of the attribute

Returns: An attribute expression for the attribute named name.

---

between(ValueExp, ValueExp, ValueExp)

`public static QueryExp between(ValueExp v1, ValueExp v2, ValueExp v3)`

Returns a query expression that represents the constraint that one value is between two other values.

Parameters:
`v1` - A value expression that is "between" v2 and v3

`v2` - Value expression that represents a boundary of the constraint

`v3` - Value expression that represents a boundary of the constraint

Returns: The constraint that v1 lies between v2 and v3.

---

classattr()

`public static AttributeValueExp classattr()`

Returns a new class attribute expression which can be used in any Query call that expects a ValueExp.

Returns: A class attribute expression.

---

div(ValueExp, ValueExp)

`public static ValueExp div(ValueExp value1, ValueExp value2)`

Returns a binary expression representing the quotient of two numeric values.

Parameters:
`value1` - The first '/' operand.

`value2` - The second '/' operand.

Returns: A ValueExp representing the quotient of two arguments.

---

eq(ValueExp, ValueExp)

`public static QueryExp eq(ValueExp v1, ValueExp v2)`

Returns a query expression that represents an equality constraint on two values.

Parameters:

`v1` - A value expression

`v2` - Another value expression

Returns: A "equal to" constraint on the arguments.

---

finalSubString(AttributeValueExp, StringValueExp)

`public static QueryExp finalSubString(AttributeValueExp a, StringValueExp s)`

Returns a query expression that represents a matching constraint on a string argument. The value must contain the given string value.

Parameters:

`a` - An attribute expression

`s` - A string value expression representing the end of the string value

Returns: The constraint that a matches s.

---

geq(ValueExp, ValueExp)

`public static QueryExp geq(ValueExp v1, ValueExp v2)`

Returns a query expression that represents a "greater than or equal to" constraint on two values.

Parameters:

`v1` - A value expression

`v2` - Another value expression

Returns: A "greater than or equal to" constraint on the arguments.

---

gt(ValueExp, ValueExp)

`public static QueryExp gt(ValueExp v1, ValueExp v2)`

Returns a query expression that represents a "greater than" constraint on two values.

Parameters:

`v1` - A value expression

`v2` - Another value expression

Returns: A "greater than" constraint on the arguments.

---

in(ValueExp, ValueExp[])

`public static QueryExp in(ValueExp val, ValueExp valueList)`

Returns an expression constraining a value to be one of an explicit list.

Parameters:

`val` - A value to be constrained

`valueList` - An array of ValueExps

Returns: A QueryExp that represents the constraint.

---

initialSubString(AttributeValueExp, StringValueExp)

`public static QueryExp initialSubString(AttributeValueExp a, StringValueExp s)`

Returns a query expression that represents a matching constraint on a string argument. The value must start with the given string value.

Parameters:

`a` - An attribute expression

`s` - A string value expression representing the beginning of the string value

Returns: The constraint that a matches s.

---

leq(ValueExp, ValueExp)

`public static QueryExp leq(ValueExp v1, ValueExp v2)`

Returns a query expression that represents a "less than or equal to" constraint on two values.

Parameters:

`v1` - A value expression

`v2` - Another value expression

Returns: A "less than or equal to" constraint on the arguments.

---

lt(ValueExp, ValueExp)

`public static QueryExp lt(ValueExp v1, ValueExp v2)`

Returns a query expression that represents a "less than" constraint on two values.

Parameters:

`v1` - A value expression

`v2` - Another value expression

Returns: A "less than" constraint on the arguments.

---

match(AttributeValueExp, StringValueExp)

`public static QueryExp match(AttributeValueExp a, StringValueExp s)`

Returns a query expression that represents a matching constraint on a string argument. The matching syntax is consistent with file globbing: Supports "?", "*", "[", each of which may be escaped with "\"; Character classes may use "!" for negation and "-" for range. (* for any character sequence ? for a single arbitrary character [...] for a character sequence). For example: a*b?c would match a string starting with the character a, followed by any number of characters, followed by a b, any single character, and a c.

Parameters:

`a` - An attribute expression

`s` - A string value expression representing a matching constraint

Returns: A query expression that represents the matching constraint on the string argument.

---

minus(ValueExp, ValueExp)

`public static ValueExp minus(ValueExp value1, ValueExp value2)`

Returns a binary expression representing the difference of two numeric values.

Parameters:

`value1` - The first '-' operand.

`value2` - The second '-' operand.

Returns: A ValueExp representing the difference of two arguments.

---

not(QueryExp)

`public static QueryExp not(QueryExp queryExp)`

Returns a constraint that is the negation of its argument.

Parameters:

`queryExp` - The constraint to negate

Returns: A negated constraint.

---

or(QueryExp, QueryExp)

`public static QueryExp or(QueryExp q1, QueryExp q2)`

Returns a query expression that is the disjunction of two other query expressions.

Parameters:

`q1` - A query expression

`q2` - Another query expression

Returns: The disjunction of the two arguments.

---

plus(ValueExp, ValueExp)

`public static ValueExp plus(ValueExp value1, ValueExp value2)`

Returns a binary expression representing the sum of two numeric values or the concatenation of two string values.

Parameters:

`value1` - The first '+' operand.

`value2` - The second '+' operand.

Returns: A ValueExp representing the sum or concatenation of the two arguments.

---

times(ValueExp, ValueExp)

`public static ValueExp times(ValueExp value1, ValueExp value2)`

Returns a binary expression representing the product of two numeric values.

Parameters:
`value1` - The first '*' operand.

`value2` - The second '*' operand.

Returns: A ValueExp representing the product.

---

value(boolean)

`public static ValueExp value(boolean val)`

Returns a boolean value expression that can be used in any Query call that expects a ValueExp.

Parameters:
`val` - A boolean value

Returns: A ValueExp object containing the argument.

---

value(double)

`public static ValueExp value(double val)`

Returns a numeric value expression that can be used in any Query call that expects a ValueExp.

Parameters:
`val` - A double value

Returns: A ValueExp object containing the argument.

---

value(float)

`public static ValueExp value(float val)`

Returns a numeric value expression that can be used in any Query call that expects a ValueExp.

Parameters:
`val` - A float value

Returns: A ValueExp object containing the argument.

---

value(int)

`public static ValueExp value(int val)`

Returns a numeric value expression that can be used in any Query call that expects a ValueExp.

Parameters:
`val` - An int value

Returns: A ValueExp object containing the argument.

---

value(long)

`public static ValueExp value(long val)`

Returns a numeric value expression that can be used in any Query call that expects a ValueExp.

Parameters:
`val` - A long value

Returns: A ValueExp object containing the argument.

---

value(Number)

`public static ValueExp value(java.lang.Number val)`

Returns a numeric value expression that can be used in any Query call that expects a ValueExp..

Parameters:
`val` - An instance of Number

Returns: A ValueExp object containing the argument.

---

value(String)

`public static StringValueExp value(java.lang.String val)`

Returns a new string expression.

Parameters:
`val` - The string value

Returns: A ValueExp object containing the string argument.

javax.management
QueryEval

Syntax

```
public abstract class QueryEval extends java.lang.Object implements java.io.Serializable
```

```
java.lang.Object
  |
  +--javax.management.QueryEval
```

Direct Known Subclasses: `QueryExp`, `ValueExp`

All Implemented Interfaces: java.io.Serializable

Description

This class allows a query to be accessed in the context of a specific MBeanServer.

Member Summary	
Constructors	
	`QueryEval()`
Methods	
`void`	`setMBeanServer(MBeanServer)`

Inherited Member Summary

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

QueryEval()

```
public  QueryEval()
```

Methods

---

setMBeanServer(MBeanServer)

```
public static void setMBeanServer(MBeanServer s)
```

Sets the MBeanServer on which the query is to be accessed.

Parameters:

`s` - The MBeanServer on which the query is to be accessed.

javax.management
QueryExp

Syntax

`public abstract class QueryExp extends QueryEval`

```
java.lang.Object
  |
  +--QueryEval
        |
        +--javax.management.QueryExp
```

All Implemented Interfaces:

java.io.Serializable

Description

The QueryExp class represents relational constraints that can be used in database query "where clauses." Instances of QueryExp are returned by the static methods of the Query class.

Member Summary

Constructors		
		`QueryExp()`
Methods		
	`boolean`	`apply(Object)`

Inherited Member Summary

Methods inherited from class `QueryEval`

`setMBeanServer(MBeanServer)`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

QueryExp()

```
public  QueryExp()
```

Methods

---

apply(Object)

`public boolean apply(java.lang.Object object)`

Applies the QueryExp on a MBean.

Parameters:
`object` - The MBean on which the QueryExp will be applied.

Returns: True if the query was successfully applied to the MBean, false otherwise.

Throws:

javax.management

ReflectionException

Syntax

`public class ReflectionException extends JMException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--javax.management.ReflectionException
```

All Implemented Interfaces: java.io.Serializable

Description

This class represents exceptions thrown in the MBeanServer when using the java.lang.reflect classes to invoke methods on MBeans. It "wraps" the actual java.lang.Exception thrown.

Member Summary

Constructors		
		`ReflectionException(Exception)`
		`ReflectionException(Exception, String)`
Methods		
	`Exception`	`getTargetException()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

Inherited Member Summary

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

ReflectionException(Exception)

```
public  ReflectionException(java.lang.Exception e)
```

Creates a ReflectionException that wraps the actual java.lang.Exception.

ReflectionException(Exception, String)

```
public  ReflectionException(java.lang.Exception e, java.lang.String message)
```

Creates a ReflectionException that wraps the actual java.lang.Exception with a detail message.

Methods

---

getTargetException()

```
public java.lang.Exception getTargetException()
```

Returns the actual java.lang.Exception thrown

javax.management
RuntimeErrorException

Syntax

`public class RuntimeErrorException extends JMRuntimeException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--JMRuntimeException
                          |
                          +--javax.management.RuntimeErrorException
```

All Implemented Interfaces: java.io.Serializable

Description

When a java.lang.Error occurs in the agent it should be caught and re-thrown as a RuntimeErrorException.

Member Summary

Constructors	
	`RuntimeErrorException(Error)`
	`RuntimeErrorException(Error, String)`
Methods	
`Error`	`getTargetError()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

RuntimeErrorException(Error)

```
public  RuntimeErrorException(java.lang.Error e)
```

Default constructor.

RuntimeErrorException(Error, String)

```
public  RuntimeErrorException(java.lang.Error e, java.lang.String message)
```

Constructor that allows a specific error message to be specified.

Methods

getTargetError()

```
public java.lang.Error getTargetError()
```

Returns the actual java.lang.Error thrown

javax.management
RuntimeMBeanException

Syntax

`public class RuntimeMBeanException extends JMRuntimeException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--JMRuntimeException
                          |
                          +--javax.management.RuntimeMBeanException
```

All Implemented Interfaces: java.io.Serializable

Description

his class represents runtime exceptions thrown by MBean methods in the agent. It "wraps" the actual java.lang.RuntimeException exception thrown. This exception will be built by the MBeanServer when a call to an MBean method throws a runtime exception.

Member Summary

Constructors

	`RuntimeMBeanException(RuntimeException)`
	`RuntimeMBeanException(RuntimeException, String)`

Methods

`RuntimeException`	`getTargetException()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

RuntimeMBeanException(RuntimeException)

```
public  RuntimeMBeanException(java.lang.RuntimeException e)
```

Creates a RuntimeMBeanException that wraps the actual java.lang.RuntimeException.

RuntimeMBeanException(RuntimeException, String)

```
public  RuntimeMBeanException(java.lang.RuntimeException e, java.lang.String message)
```

Creates a RuntimeMBeanException that wraps the actual java.lang.RuntimeException with a detail message.

Methods

getTargetException()

```
public java.lang.RuntimeException getTargetException()
```

Returns the actual java.lang.RuntimeException thrown

javax.management
RuntimeOperationsException

Syntax

```
public class RuntimeOperationsException extends JMRuntimeException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--JMRuntimeException
                          |
                          +--javax.management.RuntimeOperationsException
```

All Implemented Interfaces:
java.io.Serializable

Description

This class represents runtime exceptions thrown in the agent when performing operations on MBeans. It wraps the actual java.lang.RuntimeException thrown.

Member Summary

Constructors

	`RuntimeOperationsException(RuntimeException)`
	`RuntimeOperationsException(RuntimeException, String)`

Methods

`RuntimeException`	`getTargetException()`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

RuntimeOperationsException(RuntimeException)

```
public  RuntimeOperationsException(java.lang.RuntimeException e)
```

Creates a RuntimeOperationsException that wraps the actual java.lang.RuntimeException.

RuntimeOperationsException(RuntimeException, String)

```
public  RuntimeOperationsException(java.lang.RuntimeException e, java.lang.String
            message)
```

Creates a RuntimeOperationsException that wraps the actual java.lang.RuntimeException with a detail message.

Methods

---

getTargetException()

```
public java.lang.RuntimeException getTargetException()
```

Returns the actual java.lang.RuntimeException thrown

javax.management

ServiceNotFoundException

Syntax

`public class ServiceNotFoundException extends OperationsException`

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--JMException
                    |
                    +--OperationsException
                          |
                          +--javax.management.ServiceNotFoundException
```

All Implemented Interfaces:

java.io.Serializable

Description

This class represents exceptions raised when a requested service is not supported.

Member Summary

Constructors	
	`ServiceNotFoundException()`
	`ServiceNotFoundException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

ServiceNotFoundException()

```
public  ServiceNotFoundException()
```

Default constructor.

ServiceNotFoundException(String)

```
public  ServiceNotFoundException(java.lang.String message)
```

Constructor that allows a specific error message to be specified.

javax.management
StringValueExp

Syntax

`public class StringValueExp extends ValueExp`

```
java.lang.Object
  |
  +--QueryEval
        |
        +--ValueExp
              |
              +--javax.management.StringValueExp
```

All Implemented Interfaces: java.io.Serializable

Description

This class represents strings that are arguments to relational constraints. A StringValueExp may be used anywhere a ValueExp is required.

Member Summary

Constructors

	`StringValueExp()`
	`StringValueExp(String)`

Methods

`String`	`getValue()`
`String`	`toString()`

Inherited Member Summary

Methods inherited from class `ValueExp`

`apply(Object)`

Methods inherited from class `QueryEval`

`setMBeanServer(MBeanServer)`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

StringValueExp()

```
public  StringValueExp()
```

Basic constructor

StringValueExp(String)

```
public  StringValueExp(java.lang.String val)
```

Creates a new StringValueExp representing the string literal val.

Methods

getValue()

```
public java.lang.String getValue()
```

Returns the string represented by the StringValueExp instance.

toString()

```
public java.lang.String toString()
```

Returns the string representing the object.

Overrides: java.lang.Object.toString() in class java.lang.Object

javax.management
ValueExp

Syntax

```
public abstract class ValueExp extends QueryEval
```

```
java.lang.Object
  |
  +--QueryEval
        |
        +--javax.management.ValueExp
```

Direct Known Subclasses: `AttributeValueExp`, `StringValueExp`

All Implemented Interfaces: java.io.Serializable

Description

The ValueExp class represents values that can be passed as arguments to relational expressions. Strings, numbers, attributes are valid values and should be represented by implementations of ValueExp.

Member Summary

Constructors

	`ValueExp()`

Methods

`ValueExp`	`apply(Object)`

Inherited Member Summary

Methods inherited from class `QueryEval`

`setMBeanServer(MBeanServer)`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Constructors

ValueExp()

```
public ValueExp()
```

Methods

---

apply(Object)

`public ValueExp apply(java.lang.Object object)`

Applies the ValueExp on a MBean.

Parameters:

`object` - The MBean on which the ValueExp will be applied.

Returns: The ValueExp.

Throws:

package
# javax.management.loading

Description

Provides the classes which implement **the advanced dynamic loading**.

Class Summary	
Interfaces	
`MLetMBean`	This interface exposes the remote management interface of the MLet MBean.
Classes	
`MLet`	The m-let Service allows you to instantiate and register in the MBeanServer one or several MBeans coming from a remote URL.

javax.management.loading
MLet

Syntax

```
public class MLet extends java.net.URLClassLoader implements MLetMBean, MBeanRegistration
```

```
java.lang.Object
  |
  +--java.lang.ClassLoader
        |
        +--java.security.SecureClassLoader
              |
              +--java.net.URLClassLoader
                    |
                    +--javax.management.loading.MLet
```

All Implemented Interfaces: `MBeanRegistration`, `MLetMBean`

Description

The m-let Service allows you to instantiate and register in the MBeanServer one or several MBeans 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. 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.

Member Summary

Constructors

	`MLet()`
	`MLet(URL[])`
	`MLet(URL[], ClassLoader)`
	`MLet(URL[], ClassLoader, URLStreamHandlerFactory)`

Methods

`void`	`addURL(String)`
`void`	`addURL(URL)`
`Class`	`findClass(String)`
`String`	`findLibrary(String)`
`String`	`getLibraryDirectory()`
`Set`	`getMBeansFromURL(String)`
`Set`	`getMBeansFromURL(URL)`
`URL[]`	`getURLs()`
`void`	`postDeregister()`
`void`	`postRegister(Boolean)`
`void`	`preDeregister()`
`ObjectName`	`preRegister(MBeanServer, ObjectName)`
`void`	`readExternal(ObjectInput)`

Member Summary

`void`	`setLibraryDirectory(String)`
`void`	`writeExternal(ObjectOutput)`

Inherited Member Summary

Methods inherited from class java.net.URLClassLoader

`definePackage, findResource, findResources, getPermissions, newInstance, newInstance`

Methods inherited from class java.security.SecureClassLoader

`defineClass`

Methods inherited from class java.lang.ClassLoader

`defineClass, defineClass, defineClass, definePackage, findLoadedClass, findSystemClass, getPackage, getPackages, getParent, getResource, getResourceAsStream, getResources, getSystemClassLoader, getSystemResource, getSystemResourceAsStream, getSystemResources, loadClass, loadClass, resolveClass, setSigners`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `MLetMBean`

`getResource(String)`, `getResourceAsStream(String)`, `getResources(String)`

Constructors

MLet()

`public MLet()`

Constructs a new MLet using the default delegation parent ClassLoader.

MLet(URL[])

`public MLet(java.net.URL[] 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(URL[], ClassLoader)

`public MLet(java.net.URL[] urls, java.lang.ClassLoader 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. 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

MLet(URL[], ClassLoader, URLStreamHandlerFactory)

```
public  MLet(java.net.URL[] urls, java.lang.ClassLoader parent,
         java.net.URLStreamHandlerFactory 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

Methods

---

addURL(String)

```
public void addURL(java.lang.String url)
```

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

Specified By: `addURL(String)` in interface `MLetMBean`

Throws: `ServiceNotFoundException` - The specified URL is malformed.

---

addURL(URL)

```
public void addURL(java.net.URL url)
```

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

Specified By: `addURL(URL)` in interface `MLetMBean`

Overrides: java.net.URLClassLoader.addURL(java.net.URL) in class java.net.URLClassLoader

---

findClass(String)

```
protected java.lang.Class findClass(java.lang.String name)
```

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

Overrides: java.net.URLClassLoader.findClass(java.lang.String) in class java.net.URLClassLoader

Parameters:
`<VAR>name</VAR>` - The name of the class.

Returns: The resulting Class object.

Throws: `ClassNotFoundException` - The specified class could not be found.

---

findLibrary(String)

```
protected java.lang.String findLibrary(java.lang.String 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

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: java.lang.ClassLoader.findLibrary(java.lang.String) in class java.lang.ClassLoader

Parameters:
`libname` - The library name.

Returns: The absolute path of the native library.

---

getLibraryDirectory()

```
public java.lang.String getLibraryDirectory()
```

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

Specified By: `getLibraryDirectory()` in interface `MLetMBean`

Returns: The current directory used by the library loader.

---

getMBeansFromURL(String)

```
public java.util.Set getMBeansFromURL(java.lang.String url)
```

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

Specified By: `getMBeansFromURL(String)` in interface `MLetMBean`

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` - 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.

getMBeansFromURL(URL)

```
public java.util.Set getMBeansFromURL(java.net.URL url)
```

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

Specified By: `getMBeansFromURL(URL)` in interface `MLetMBean`

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` - 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.

getURLs()

```
public java.net.URL[] 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()` in interface `MLetMBean`

Overrides: java.net.URLClassLoader.getURLs() in class java.net.URLClassLoader

postDeregister()

```
public void postDeregister()
```

Allows the m-let to perform any operations needed after having been de-registered in the MBeanServer.

Specified By: `postDeregister()` in interface `MBeanRegistration`

postRegister(Boolean)

```
public void postRegister(java.lang.Boolean registrationDone)
```

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

Specified By: `postRegister(Boolean)` in interface `MBeanRegistration`

Parameters:

`registrationDone` - Indicates whether or not the m-let has been successfully registered in the MBeanServer. The value false means that either the registration phase has failed.

---

preDeregister()

`public void preDeregister()`

Allows the m-let to perform any operations it needs before being de-registered by the MBeanServer.

Specified By: `preDeregister()` in interface `MBeanRegistration`

Throws: `java.langException` - This exception should be caught by the MBeanServer and re-thrown as an MBeanRegistrationException.

---

preRegister(MBeanServer, ObjectName)

`public ObjectName preRegister(MBeanServer server, ObjectName name)`

Allows the m-let to perform any operations it needs before being registered in the MBeanServer. If the ObjectName is null, the m-let provides a default name for its registration :service=MLet

Specified By: `preRegister(MBeanServer, ObjectName)` in interface `MBeanRegistration`

Parameters:

`server` - The MBeanServer 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: `java.langException` - This exception should be caught by the MBeanServer and re-thrown as an MBeanRegistrationException.

---

readExternal(ObjectInput)

`public void readExternal(java.io.ObjectInput in)`

This method reads the objects contents during Object Serialization.

Parameters:

`<VAR>in</VAR>` - The serialization input stream.

Throws: `IOException` - Signals that an I/O exception of some sort has occurred.

`ClassNotFoundException` - The class for an object being restored cannot be found.

---

setLibraryDirectory(String)

`public void setLibraryDirectory(java.lang.String libdir)`

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

Specified By: `setLibraryDirectory(String)` in interface `MLetMBean`

Parameters:

`<VAR>libdir</VAR>` - The directory used by the library loader.

---

writeExternal(ObjectOutput)

```
public void writeExternal(java.io.ObjectOutput out)
```

This method saves the objects contents during Object Serialization.

Parameters:
`<VAR>out</VAR>` - The serialization output stream.

Throws: `IOException` - Signals that an I/O exception of some sort has occurred.

javax.management.loading
MLetMBean

Syntax

`public interface MLetMBean`

All Known Implementing Classes: `MLet`

Description

This interface exposes the remote management interface of the MLet MBean.

Member Summary

Methods

`void`	`addURL(String)`
`void`	`addURL(URL)`
`String`	`getLibraryDirectory()`
`Set`	`getMBeansFromURL(String)`
`Set`	`getMBeansFromURL(URL)`
`URL`	`getResource(String)`
`InputStream`	`getResourceAsStream(String)`
`Enumeration`	`getResources(String)`
`URL[]`	`getURLs()`
`void`	`setLibraryDirectory(String)`

Methods

---

addURL(String)

```
public void addURL(java.lang.String url)
```

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

Throws: `ServiceNotFoundException` - The specified URL is malformed.

---

addURL(URL)

```
public void addURL(java.net.URL url)
```

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

---

getLibraryDirectory()

```
public java.lang.String getLibraryDirectory()
```

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

Returns: The current directory used by the library loader.

---

getMBeansFromURL(String)

`public java.util.Set getMBeansFromURL(java.lang.String url)`

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

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` - 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 malformed.

---

getMBeansFromURL(URL)

`public java.util.Set getMBeansFromURL(java.net.URL url)`

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

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` - 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.

---

getResource(String)

`public java.net.URL getResource(java.lang.String name)`

Finds the resource with the given name. A resource is some data (images, audio, text, etc) that can be accessed by class code in a way that is independent of the location of the code. The name of a resource is a "/"-separated path name that identifies the resource.

Parameters:
`name` - The resource name

Returns: An URL for reading the resource, or null if the resource could not be found or the caller doesn't have adequate privileges to get the resource.

getResourceAsStream(String)

```
public java.io.InputStream getResourceAsStream(java.lang.String name)
```

Returns an input stream for reading the specified resource. The search order is described in the documentation for getResource(String).

Parameters:
`name` - The resource name

Returns: An input stream for reading the resource, or null if the resource could not be foun

getResources(String)

```
public java.util.Enumeration getResources(java.lang.String name)
```

Finds all the resources with the given name. A resource is some data (images, audio, text, etc) that can be accessed by class code in a way that is independent of the location of the code. The name of a resource is a "/"-separated path name that identifies the resource.

Parameters:
`name` - The resource name.

Returns: An enumeration of URL to the resource. If no resources could be found, the enumeration will be empty. Resources that the doesn't have access to will not be in the enumeration.

getURLs()

```
public java.net.URL[] 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.

setLibraryDirectory(String)

```
public void setLibraryDirectory(java.lang.String libdir)
```

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

Parameters:
`<VAR>libdir</VAR>` - The directory used by the library loader.

package
javax.management.monitor

Description

Provides the definition of the **monitor classes**.

Class Summary

Interfaces	
`CounterMonitorMBean`	This interface exposes the remote management interface of the CounterMonitor MBean.
`GaugeMonitorMBean`	This interface exposes the remote management interface of the GaugeMonitor MBean.
`MonitorMBean`	This interface exposes the remote management interface of the Monitor MBean.
`StringMonitorMBean`	This interface exposes the remote management interface of the StringMonitor MBean.
Classes	
`CounterMonitor`	This class defines a monitor MBean designed to observe the values of a counter attribute.
`GaugeMonitor`	This class defines a monitor MBean designed to observe the values of a gauge attribute.
`Monitor`	This abstract class defines the common part to all monitor MBeans.
`MonitorNotification`	This class provides definitions of the notifications sent by Monitor MBeans.
`StringMonitor`	This class defines a monitor MBean designed to observe the values of a string attribute.
Exceptions	
`MonitorSettingException`	This exception is thrown by the monitor when a monitor setting becomes invalid while the monitor is running.

javax.management.monitor
CounterMonitor

Syntax

```
public class CounterMonitor extends Monitor implements CounterMonitorMBean
```

```
java.lang.Object
  |
  +--NotificationBroadcasterSupport
        |
        +--Monitor
              |
              +--javax.management.monitor.CounterMonitor
```

All Implemented Interfaces: `CounterMonitorMBean`, `MBeanRegistration`, `MonitorMBean`, `NotificationBroadcaster`, java.io.Serializable

Description

This class defines a monitor MBean designed to observe the values of a counter attribute.

A counter monitor sends a `THRESHOLD_VALUE_EXCEEDED` when the value of the counter reaches or exceeds a threshold known as the comparison level. The notify flag must be set to true. In addition, an offset mechanism enables particular counting intervals to be detected.

If the offset value is not zero, whenever the threshold is triggered by the counter value reaching a comparison level, that comparison level is incremented by the offset value. This is regarded as taking place instantaneously, i.e. before the count is incremented. Thus, for each level, the threshold triggers an event notification every time the count increases by an interval equal to the offset value.

If the counter can wrap around its maximum value, then the modulus needs to be specified. The modulus is the value at which the counter is reset to zero.

If the counter difference mode is used, then the value of the derived gauge is calculated as the difference between the observed counter values for two successive observations. If this difference is negative then the value of the derived gauge is incremented by the value of the modulus. The derived gauge value (V[t]) is calculated using the following method:

- if (counter[t] - counter[t-GP]) is positive then V[t] = counter[t] - counter[t-GP]
- if (counter[t] - counter[t-GP]) is negative then V[t] = counter[t] - counter[t-GP] + MODULUS

This implementation of the counter monitor requires the observed attribute to be of type integer.

Member Summary

Constructors

	`CounterMonitor()`

Methods

`Number`	`getDerivedGauge()`
`Date`	`getDerivedGaugeTimeStamp()`
`boolean`	`getDifferenceMode()`
`Number`	`getModulus()`

Member Summary

`MBeanNotificationInfo[]`	`getNotificationInfo()`
`boolean`	`getNotify()`
`Number`	`getOffset()`
`Number`	`getThreshold()`
`void`	`preDeregister()`
`void`	`setDifferenceMode(boolean)`
`void`	`setModulus(Number)`
`void`	`setNotify(boolean)`
`void`	`setOffset(Number)`
`void`	`setThreshold(Number)`
`void`	`start()`
`void`	`stop()`

Inherited Member Summary

Fields inherited from class `Monitor`

`alreadyNotified`, `dbgTag`, `OBSERVED_ATTRIBUTE_ERROR_NOTIFIED`, `OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED`, `OBSERVED_OBJECT_ERROR_NOTIFIED`, `RESET_FLAGS_ALREADY_NOTIFIED`, `RUNTIME_ERROR_NOTIFIED`, `server`

Methods inherited from class `Monitor`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `postDeregister()`, `postRegister(Boolean)`, `preRegister(MBeanServer, ObjectName)`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from class `NotificationBroadcasterSupport`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `removeNotificationListener(NotificationListener)`, `sendNotification(Notification)`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from interface `MBeanRegistration`

`postDeregister()`, `postRegister(Boolean)`, `preRegister(MBeanServer, ObjectName)`

Methods inherited from interface `NotificationBroadcaster`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `removeNotificationListener(NotificationListener)`

Constructors

CounterMonitor()

`public  CounterMonitor()`

Default constructor.

Methods

---

getDerivedGauge()

`public java.lang.Number getDerivedGauge()`

Gets the derived gauge.

Specified By: `getDerivedGauge()` in interface `CounterMonitorMBean`

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

`public java.util.Date getDerivedGaugeTimeStamp()`

Gets the derived gauge timestamp.

Specified By: `getDerivedGaugeTimeStamp()` in interface `CounterMonitorMBean`

Returns: The derived gauge timestamp.

---

getDifferenceMode()

`public boolean getDifferenceMode()`

Gets the difference mode flag value.

Specified By: `getDifferenceMode()` in interface `CounterMonitorMBean`

Returns: True if the difference mode is used, false otherwise.

---

getModulus()

`public java.lang.Number getModulus()`

Gets the modulus value.

Specified By: `getModulus()` in interface `CounterMonitorMBean`

Returns: The modulus value.

---

getNotificationInfo()

`public MBeanNotificationInfo getNotificationInfo()`

Returns a NotificationInfo object contaning the name of the Java class of the notification and the notification types sent.

Specified By: `getNotificationInfo()` in interface `NotificationBroadcaster`

Overrides: `getNotificationInfo()` in class `NotificationBroadcasterSupport`

---

getNotify()

`public boolean getNotify()`

Gets the notifications on/off switch value.

Specified By: `getNotify()` in interface `CounterMonitorMBean`

Returns: True if the CounterMonitor notifies when exceeding the threshold, false otherwise.

---

getOffset()

`public java.lang.Number getOffset()`

Gets the offset value.

Specified By: `getOffset()` in interface `CounterMonitorMBean`

Returns: The offset value.

---

getThreshold()

`public java.lang.Number getThreshold()`

Gets the threshold value.

Specified By: `getThreshold()` in interface `CounterMonitorMBean`

Returns: The threshold value.

---

preDeregister()

`public void preDeregister()`

Allows the Monitor MBean to perform any operations it needs before being de-registered by the MBean server.

Resets the threshold values.

Specified By: `preDeregister()` in interface `MBeanRegistration`

Overrides: `preDeregister()` in class `Monitor`

Throws: **setDifferenceMode(boolean)**

`public void setDifferenceMode(boolean value)`

Sets the difference mode flag value.

Specified By: `setDifferenceMode(boolean)` in interface `CounterMonitorMBean`

Parameters:
`value` - The difference mode flag value.

---

setModulus(Number)

```
public void setModulus(java.lang.Number value)
```

Sets the modulus value.

Specified By: `setModulus(Number)` in interface `CounterMonitorMBean`

Parameters:
`value` - The modulus value.

Throws: `java.lang.IllegalArgumentException` - The specified modulus is null or the modulus value is less than zero.

---

setNotify(boolean)

```
public void setNotify(boolean value)
```

Sets the notifications on/off switch value.

Specified By: `setNotify(boolean)` in interface `CounterMonitorMBean`

Parameters:
`value` - The notifications on/off switch value.

---

setOffset(Number)

```
public void setOffset(java.lang.Number value)
```

Sets the offset value.

Specified By: `setOffset(Number)` in interface `CounterMonitorMBean`

Parameters:
`value` - The offset value.

Throws: `java.lang.IllegalArgumentException` - The specified offset is null or the offset value is less than zero.

---

setThreshold(Number)

```
public void setThreshold(java.lang.Number value)
```

Sets the threshold value.

Specified By: `setThreshold(Number)` in interface `CounterMonitorMBean`

Parameters:
`value` - The threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified threshold is null or the threshold value is less than zero.

---

start()

```
public synchronized void start()
```

Starts the CounterMonitor.

Specified By: `start()` in interface `MonitorMBean`

Specified By: `start()` in interface `MonitorMBean`

Overrides: `start()` in class `Monitor`

---

stop()

```
public synchronized void stop()
```

Stops the CounterMonitor.

Specified By: `stop()` in interface `MonitorMBean`

Specified By: `stop()` in interface `MonitorMBean`

Overrides: `stop()` in class `Monitor`

javax.management.monitor

CounterMonitorMBean

Syntax

`public interface CounterMonitorMBean extends MonitorMBean`

All Superinterfaces: `MonitorMBean`

All Known Implementing Classes: `CounterMonitor`

Description

This interface exposes the remote management interface of the CounterMonitor MBean.

Member Summary

Methods

`Number`	`getDerivedGauge()`
`Date`	`getDerivedGaugeTimeStamp()`
`boolean`	`getDifferenceMode()`
`Number`	`getModulus()`
`boolean`	`getNotify()`
`Number`	`getOffset()`
`Number`	`getThreshold()`
`void`	`setDifferenceMode(boolean)`
`void`	`setModulus(Number)`
`void`	`setNotify(boolean)`
`void`	`setOffset(Number)`
`void`	`setThreshold(Number)`

Inherited Member Summary

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`, `start()`, `stop()`

Methods

getDerivedGauge()

```
public java.lang.Number getDerivedGauge()
```

Gets the derived gauge.

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

```
public java.util.Date getDerivedGaugeTimeStamp()
```

Gets the derived gauge timestamp.

Returns: The derived gauge timestamp.

---

getDifferenceMode()

```
public boolean getDifferenceMode()
```

Gets the difference mode flag value.

Returns: True if the difference mode is used, false otherwise.

---

getModulus()

```
public java.lang.Number getModulus()
```

Gets the modulus value.

Returns: The modulus value.

---

getNotify()

```
public boolean getNotify()
```

Gets the notifications on/off switch value.

Returns: True if the CounterMonitor notifies when exceeding the threshold, false otherwise.

---

getOffset()

```
public java.lang.Number getOffset()
```

Gets the offset value.

Returns: The offset value.

---

getThreshold()

```
public java.lang.Number getThreshold()
```

Gets the threshold value.

Returns: The threshold value.

---

setDifferenceMode(boolean)

```
public void setDifferenceMode(boolean value)
```

Sets the difference mode flag value.

Parameters:
`value` - The difference mode flag value.

---

setModulus(Number)

`public void setModulus(java.lang.Number value)`

Sets the modulus value.

Parameters:
`value` - The modulus value.

Throws: `java.lang.IllegalArgumentException` - The specified modulus is null or the modulus value is less than zero.

---

setNotify(boolean)

`public void setNotify(boolean value)`

Sets the notifications on/off switch value.

Parameters:
`value` - The notifications on/off switch value.

---

setOffset(Number)

`public void setOffset(java.lang.Number value)`

Sets the offset value.

Parameters:
`value` - The offset value.

Throws: `java.lang.IllegalArgumentException` - The specified offset is null or the offset value is less than zero.

---

setThreshold(Number)

`public void setThreshold(java.lang.Number value)`

Sets the threshold value.

Parameters:
`value` - The threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified threshold is null or the threshold value is less than zero.

javax.management.monitor
GaugeMonitor

Syntax

`public class GaugeMonitor extends Monitor implements GaugeMonitorMBean`

```
java.lang.Object
  |
  +--NotificationBroadcasterSupport
        |
        +--Monitor
              |
              +--javax.management.monitor.GaugeMonitor
```

All Implemented Interfaces: `GaugeMonitorMBean`, `MBeanRegistration`, `MonitorMBean`, `NotificationBroadcaster`, java.io.Serializable

Description

This class defines a monitor MBean designed to observe the values of a gauge attribute.

A gauge monitor observes an attribute that is continuously variable with time. A gauge monitor sends notifications as follows:

- if the attribute value is increasing and becomes equal to or greater than the high threshold value, a `THRESHOLD_HIGH_VALUE_EXCEEDED` is sent. The notify high flag must be set to true. Subsequent crossings of the high threshold value do not cause further notifications unless the attribute value becomes equal to or less than the low threshold value.
- if the attribute value is decreasing and becomes equal to or less than the low threshold value, a `THRESHOLD_LOW_VALUE_EXCEEDED` is sent. The notify low flag must be set to true. Subsequent crossings of the low threshold value do not cause further notifications unless the attribute value becomes equal to or greater than the high threshold value.

This provides a hysteresis mechanism to avoid repeated triggering of notifications when the attribute value makes small oscillations around the high or low threshold value.

If the gauge difference mode is used, then the value of the derived gauge is calculated as the difference between the observed gauge values for two successive observations. The derived gauge value (V[t]) is calculated using the following method:

- V[t] = gauge[t] - gauge[t-GP]

This implementation of the gauge monitor requires the observed attribute to be either of type integer or floating-point.

Member Summary

Constructors

	`GaugeMonitor()`

Methods

`Number`	`getDerivedGauge()`
`Date`	`getDerivedGaugeTimeStamp()`
`boolean`	`getDifferenceMode()`

Member Summary

`Number`	`getHighThreshold()`
`Number`	`getLowThreshold()`
`MBeanNotificationInfo[]`	`getNotificationInfo()`
`boolean`	`getNotifyHigh()`
`boolean`	`getNotifyLow()`
`void`	`setDifferenceMode(boolean)`
`void`	`setHighThreshold(Number)`
`void`	`setLowThreshold(Number)`
`void`	`setNotifyHigh(boolean)`
`void`	`setNotifyLow(boolean)`
`void`	`start()`
`void`	`stop()`

Inherited Member Summary

Fields inherited from class `Monitor`

`alreadyNotified`, `dbgTag`, `OBSERVED_ATTRIBUTE_ERROR_NOTIFIED`, `OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED`, `OBSERVED_OBJECT_ERROR_NOTIFIED`, `RESET_FLAGS_ALREADY_NOTIFIED`, `RUNTIME_ERROR_NOTIFIED`, `server`

Methods inherited from class `Monitor`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `postDeregister()`, `postRegister(Boolean)`, `preDeregister()`, `preRegister(MBeanServer, ObjectName)`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from class `NotificationBroadcasterSupport`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `removeNotificationListener(NotificationListener)`, `sendNotification(Notification)`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from interface `MBeanRegistration`

`postDeregister()`, `postRegister(Boolean)`, `preDeregister()`, `preRegister(MBeanServer, ObjectName)`

Methods inherited from interface `NotificationBroadcaster`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `removeNotificationListener(NotificationListener)`

Constructors

GaugeMonitor()

`public  GaugeMonitor()`

Default constructor.

Methods

---

getDerivedGauge()

`public java.lang.Number getDerivedGauge()`

Gets the derived gauge.

Specified By: `getDerivedGauge()` in interface `GaugeMonitorMBean`

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

`public java.util.Date getDerivedGaugeTimeStamp()`

Gets the derived gauge timestamp.

Specified By: `getDerivedGaugeTimeStamp()` in interface `GaugeMonitorMBean`

Returns: The derived gauge timestamp.

---

getDifferenceMode()

`public boolean getDifferenceMode()`

Gets the difference mode flag value.

Specified By: `getDifferenceMode()` in interface `GaugeMonitorMBean`

Returns: True if the difference mode is used, false otherwise.

---

getHighThreshold()

`public java.lang.Number getHighThreshold()`

Gets the high threshold value.

Specified By: `getHighThreshold()` in interface `GaugeMonitorMBean`

Returns: The high threshold value.

---

getLowThreshold()

`public java.lang.Number getLowThreshold()`

Gets the low threshold value.

Specified By: `getLowThreshold()` in interface `GaugeMonitorMBean`

Returns: The low threshold value.

---

getNotificationInfo()

`public MBeanNotificationInfo getNotificationInfo()`

Returns a NotificationInfo object contaning the name of the Java class of the notification and the notification types sent.

Specified By: `getNotificationInfo()` in interface `NotificationBroadcaster`

Overrides: `getNotificationInfo()` in class `NotificationBroadcasterSupport`

---

getNotifyHigh()

`public boolean getNotifyHigh()`

Gets the high notifications on/off switch value.

Specified By: `getNotifyHigh()` in interface `GaugeMonitorMBean`

Returns: True if the GaugeMonitor notifies when exceeding the high threshold, false otherwise.

---

getNotifyLow()

`public boolean getNotifyLow()`

Gets the low notifications on/off switch value.

Specified By: `getNotifyLow()` in interface `GaugeMonitorMBean`

Returns: True if the GaugeMonitor notifies when exceeding the low threshold, false otherwise.

---

setDifferenceMode(boolean)

`public void setDifferenceMode(boolean value)`

Sets the difference mode flag value.

Specified By: `setDifferenceMode(boolean)` in interface `GaugeMonitorMBean`

Parameters:
`value` - The difference mode flag value.

---

setHighThreshold(Number)

`public void setHighThreshold(java.lang.Number value)`

Sets the high threshold value.

Specified By: `setHighThreshold(Number)` in interface `GaugeMonitorMBean`

Parameters:
`value` - The high threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified high threshold is null.

---

setLowThreshold(Number)

```
public void setLowThreshold(java.lang.Number value)
```

Sets the low threshold value.

Specified By: `setLowThreshold(Number)` in interface `GaugeMonitorMBean`

Parameters:
`value` - The low threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified low threshold is null.

---

setNotifyHigh(boolean)

```
public void setNotifyHigh(boolean value)
```

Sets the high notifications on/off switch value.

Specified By: `setNotifyHigh(boolean)` in interface `GaugeMonitorMBean`

Parameters:
`value` - The high notifications on/off switch value.

---

setNotifyLow(boolean)

```
public void setNotifyLow(boolean value)
```

Sets the low notifications on/off switch value.

Specified By: `setNotifyLow(boolean)` in interface `GaugeMonitorMBean`

Parameters:
`value` - The low notifications on/off switch value.

---

start()

```
public synchronized void start()
```

Starts the GaugeMonitor.

Specified By: `start()` in interface `MonitorMBean`

Specified By: `start()` in interface `MonitorMBean`

Overrides: `start()` in class `Monitor`

---

stop()

```
public synchronized void stop()
```

Stops the GaugeMonitor.

Specified By: `stop()` in interface `MonitorMBean`

Specified By: `stop()` in interface `MonitorMBean`

Overrides: `stop()` in class `Monitor`

javax.management.monitor
GaugeMonitorMBean

Syntax

`public interface GaugeMonitorMBean extends MonitorMBean`

All Superinterfaces: `MonitorMBean`

All Known Implementing Classes: `GaugeMonitor`

Description

This interface exposes the remote management interface of the GaugeMonitor MBean.

Member Summary

Methods

`Number`	`getDerivedGauge()`
`Date`	`getDerivedGaugeTimeStamp()`
`boolean`	`getDifferenceMode()`
`Number`	`getHighThreshold()`
`Number`	`getLowThreshold()`
`boolean`	`getNotifyHigh()`
`boolean`	`getNotifyLow()`
`void`	`setDifferenceMode(boolean)`
`void`	`setHighThreshold(Number)`
`void`	`setLowThreshold(Number)`
`void`	`setNotifyHigh(boolean)`
`void`	`setNotifyLow(boolean)`

Inherited Member Summary

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`, `start()`, `stop()`

Methods

---

getDerivedGauge()

```
public java.lang.Number getDerivedGauge()
```

Gets the derived gauge.

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

`public java.util.Date getDerivedGaugeTimeStamp()`

Gets the derived gauge timestamp.

Returns: The derived gauge timestamp.

---

getDifferenceMode()

`public boolean getDifferenceMode()`

Gets the difference mode flag value.

Returns: True if the difference mode is used, false otherwise.

---

getHighThreshold()

`public java.lang.Number getHighThreshold()`

Gets the high threshold value.

Returns: The high threshold value.

---

getLowThreshold()

`public java.lang.Number getLowThreshold()`

Gets the low threshold value.

Returns: The low threshold value.

---

getNotifyHigh()

`public boolean getNotifyHigh()`

Gets the high notifications on/off switch value.

Returns: True if the GaugeMonitor notifies when exceeding the high threshold, false otherwise.

---

getNotifyLow()

`public boolean getNotifyLow()`

Gets the low notifications on/off switch value.

Returns: True if the GaugeMonitor notifies when exceeding the low threshold, false otherwise.

---

setDifferenceMode(boolean)

`public void setDifferenceMode(boolean value)`

Sets the difference mode flag value.

Parameters:
`value` - The difference mode flag value.

---

setHighThreshold(Number)

```
public void setHighThreshold(java.lang.Number value)
```

Sets the high threshold value.

Parameters:
`value` - The high threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified high threshold is null.

---

setLowThreshold(Number)

```
public void setLowThreshold(java.lang.Number value)
```

Sets the low threshold value.

Parameters:
`value` - The low threshold value.

Throws: `java.lang.IllegalArgumentException` - The specified low threshold is null.

---

setNotifyHigh(boolean)

```
public void setNotifyHigh(boolean value)
```

Sets the high notifications on/off switch value.

Parameters:
`value` - The high notifications on/off switch value.

---

setNotifyLow(boolean)

```
public void setNotifyLow(boolean value)
```

Sets the low notifications on/off switch value.

Parameters:
`value` - The low notifications on/off switch value.

javax.management.monitor
Monitor

Syntax

```
public abstract class Monitor extends NotificationBroadcasterSupport implements MonitorMBean,
    MBeanRegistration, java.io.Serializable
```

```
java.lang.Object
  |
  +--NotificationBroadcasterSupport
        |
        +--javax.management.monitor.Monitor
```

Direct Known Subclasses: `CounterMonitor`, `GaugeMonitor`, `StringMonitor`

All Implemented Interfaces: `MBeanRegistration`, `MonitorMBean`, `NotificationBroadcaster`, java.io.Serializable

Description

This abstract class defines the common part to all monitor MBeans. A monitor MBean monitors values of an attribute in an observed MBean. The observed attribute is monitored at intervals specified by the granularity period. A gauge value (derived gauge) is derived from the values of the observed attribute.

Member Summary

Fields	
`int`	`alreadyNotified`
`String`	`dbgTag`
`int`	`OBSERVED_ATTRIBUTE_ERROR_NOTIFIED`
`int`	`OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED`
`int`	`OBSERVED_OBJECT_ERROR_NOTIFIED`
`int`	`RESET_FLAGS_ALREADY_NOTIFIED`
`int`	`RUNTIME_ERROR_NOTIFIED`
`MBeanServer`	`server`
Constructors	
	`Monitor()`
Methods	
`long`	`getGranularityPeriod()`
`String`	`getObservedAttribute()`
`ObjectName`	`getObservedObject()`
`boolean`	`isActive()`
`void`	`postDeregister()`
`void`	`postRegister(Boolean)`
`void`	`preDeregister()`
`ObjectName`	`preRegister(MBeanServer, ObjectName)`
`void`	`setGranularityPeriod(long)`
`void`	`setObservedAttribute(String)`
`void`	`setObservedObject(ObjectName)`

Member Summary

`void`	`start()`
`void`	`stop()`

Inherited Member Summary

Methods inherited from class `NotificationBroadcasterSupport`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`, `sendNotification(Notification)`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

Methods inherited from interface `NotificationBroadcaster`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`

Fields

alreadyNotified

`protected int alreadyNotified`

Selected monitor errors that have already been notified.

dbgTag

`protected java.lang.String dbgTag`

OBSERVED_ATTRIBUTE_ERROR_NOTIFIED

`protected static final int OBSERVED_ATTRIBUTE_ERROR_NOTIFIED`

Flag denoting that a notification has occured after changing the observed attribute. This flag is used to check that the new observed attribute belongs to the observed object at the first notification time.

OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED

`protected static final int OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED`

Flag denoting that a notification has occured after changing the observed object or the observed attribute. This flag is used to check that the observed attribute type is correct (depending on the monitor in use) at the first notification time.

OBSERVED_OBJECT_ERROR_NOTIFIED

`protected static final int OBSERVED_OBJECT_ERROR_NOTIFIED`

Flag denoting that a notification has occured after changing the observed object. This flag is used to check that the new observed object is registered in the MBean server at the first notification time.

RESET_FLAGS_ALREADY_NOTIFIED

`protected static final int RESET_FLAGS_ALREADY_NOTIFIED`

This flag is used to reset the `alreadyNotified` monitor attribute.

RUNTIME_ERROR_NOTIFIED

`protected static final int RUNTIME_ERROR_NOTIFIED`

Flag denoting that a notification has occured after changing the observed object or the observed attribute. This flag is used to notify any exception (except the cases described above) when trying to get the value of the observed attribute at the first notification time.

---

server

`protected MBeanServer server`

Reference on the MBean server. This reference is null when the Monitor MBean is not registered in a MBean server. This reference is initialized before the Monitor MBean is registered in the MBean server.

See Also: `preRegister(MBeanServer, ObjectName)`

Constructors

Monitor()

`public  Monitor()`

Methods

---

getGranularityPeriod()

`public long getGranularityPeriod()`

Gets the granularity period (in milliseconds).

Specified By: `getGranularityPeriod()` in interface `MonitorMBean`

Returns: The granularity period value.

---

getObservedAttribute()

`public java.lang.String getObservedAttribute()`

Gets the attribute being observed.

Specified By: `getObservedAttribute()` in interface `MonitorMBean`

Returns: The attribute being observed.

---

getObservedObject()

`public ObjectName getObservedObject()`

Gets the object name of the object being observed.

Specified By: `getObservedObject()` in interface `MonitorMBean`

Returns: The object being observed.

---

isActive()

`public boolean isActive()`

Tests if the Monitor MBean is active. A Monitor MBean is marked active when the `start()` method is called. It becomes inactive when the `stop()` method is called.

Specified By: `isActive()` in interface `MonitorMBean`

Returns: True if the Monitor MBean is active, false otherwise.

---

postDeregister()

`public void postDeregister()`

Allows the Monitor MBean to perform any operations needed after having been de-registered by the MBean server.

Specified By: `postDeregister()` in interface `MBeanRegistration`

---

postRegister(Boolean)

`public void postRegister(java.lang.Boolean registrationDone)`

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

Specified By: `postRegister(Boolean)` in interface `MBeanRegistration`

Parameters:

`registrationDone` - Indicates whether or not the Monitor MBean has been successfully registered in the MBean server. The value false means that either the registration phase has failed.

---

preDeregister()

`public void preDeregister()`

Allows the Monitor MBean to perform any operations it needs before being de-registered by the MBean server.

Stops the monitor.

Specified By: `preDeregister()` in interface `MBeanRegistration`

Throws: **preRegister(MBeanServer, ObjectName)**

```
public ObjectName preRegister(MBeanServer server, ObjectName name)
```

Allows the Monitor MBean to perform any operations it needs before being registered in the MBean server.

Initializes the reference to the MBean server.

Specified By: `preRegister(MBeanServer, ObjectName)` in interface `MBeanRegistration`

Parameters:
`server` - The MBean server in which the Monitor MBean will be registered.

`name` - The object name of the Monitor MBean.

Returns: The name of the Monitor MBean registered.

Throws: **setGranularityPeriod(long)**

```
public void setGranularityPeriod(long period)
```

Sets the granularity period (in milliseconds).

Specified By: `setGranularityPeriod(long)` in interface `MonitorMBean`

Parameters:
`period` - The granularity period value.

Throws: `java.lang.IllegalArgumentException` - The granularity period is less than or equal to zero.

---

setObservedAttribute(String)

```
public void setObservedAttribute(java.lang.String attribute)
```

Sets the attribute to observe.

Specified By: `setObservedAttribute(String)` in interface `MonitorMBean`

Parameters:
`attribute` - The attribute to observe.

---

setObservedObject(ObjectName)

```
public void setObservedObject(ObjectName object)
```

Sets the object to observe identified by its object name.

Specified By: `setObservedObject(ObjectName)` in interface `MonitorMBean`

Parameters:
`object` - The object to observe.

---

start()

```
public abstract void start()
```

Starts the Monitor.

Specified By: `start()` in interface `MonitorMBean`

---

stop()

```
public abstract void stop()
```

Stops the Monitor.

Specified By: `stop()` in interface `MonitorMBean`

javax.management.monitor
MonitorMBean

Syntax

`public interface MonitorMBean`

All Known Subinterfaces: `CounterMonitorMBean`, `GaugeMonitorMBean`, `StringMonitorMBean`

All Known Implementing Classes: `Monitor`

Description

This interface exposes the remote management interface of the Monitor MBean.

Member Summary

Methods

`long`	`getGranularityPeriod()`
`String`	`getObservedAttribute()`
`ObjectName`	`getObservedObject()`
`boolean`	`isActive()`
`void`	`setGranularityPeriod(long)`
`void`	`setObservedAttribute(String)`
`void`	`setObservedObject(ObjectName)`
`void`	`start()`
`void`	`stop()`

Methods

---

getGranularityPeriod()

```
public long getGranularityPeriod()
```

Gets the granularity period (in milliseconds).

Returns: The granularity period.

---

getObservedAttribute()

```
public java.lang.String getObservedAttribute()
```

Gets the attribute being observed.

Returns: The attribute being observed.

getObservedObject()

`public ObjectName getObservedObject()`

Gets the object name of the object being observed.

Returns: The object being observed.

isActive()

`public boolean isActive()`

Tests if the Monitor MBean is active. A Monitor MBean is marked active when the `start()` method is called. It becomes inactive when the `stop()` method is called.

Returns: True if the Monitor MBean is active, false otherwise.

setGranularityPeriod(long)

`public void setGranularityPeriod(long period)`

Sets the granularity period (in milliseconds).

Parameters:
`period` - The granularity period.

Throws: `java.lang.IllegalArgumentException` - The granularity period is less than or equal to zero.

setObservedAttribute(String)

`public void setObservedAttribute(java.lang.String attribute)`

Sets the attribute to observe.

Parameters:
`attribute` - The attribute to observe.

setObservedObject(ObjectName)

`public void setObservedObject(ObjectName object)`

Sets the object to observe identified by its object name.

Parameters:
`object` - The object to observe.

start()

`public void start()`

Starts the Monitor.

stop()

stop()

`public void stop()`

Stops the Monitor.

javax.management.monitor
MonitorNotification

Syntax

```
public class MonitorNotification extends Notification
```

```
java.lang.Object
  |
  +--java.util.EventObject
        |
        +--Notification
              |
              +--javax.management.monitor.MonitorNotification
```

All Implemented Interfaces: java.io.Serializable

Description

This class provides definitions of the notifications sent by Monitor MBeans.

The notification source and a set of parameters concerning the Monitor MBean's state need to be specified when creating a new object of this class. The list of notifications fired by the Monitor MBeans is the following:

- Common to all kind of monitors:
- The observed object is not registered in the MBean server.
- The observed attribute is not contained in the observed object.
- The type of the observed attribute is not correct.
- Any exception (except the cases described above) occurs when trying to get the value of the observed attribute.

- Common to the counter and the gauge monitors:
- The threshold high or threshold low are not of the same type as the gauge or the threshold high value is less than the threshold low value (gauge monitors).
- The threshold or the offset or the modulus are not of the same type as the counter (counter monitors).

- Counter monitors only:
- The observed attribute has reached the threshold value.

- Gauge monitors only:
- The observed attribute has exceeded the threshold high value.
- The observed attribute has exceeded the threshold low value.

- String monitors only:
- The observed attribute has matched the "string to compare" value.
- The observed attribute has differed from the "string to compare" value.

Member Summary

Fields

`String`	`OBSERVED_ATTRIBUTE_ERROR`
`String`	`OBSERVED_ATTRIBUTE_TYPE_ERROR`
`String`	`OBSERVED_OBJECT_ERROR`
`String`	`RUNTIME_ERROR`
`String`	`STRING_TO_COMPARE_VALUE_DIFFERED`
`String`	`STRING_TO_COMPARE_VALUE_MATCHED`
`String`	`THRESHOLD_ERROR`
`String`	`THRESHOLD_HIGH_VALUE_EXCEEDED`
`String`	`THRESHOLD_LOW_VALUE_EXCEEDED`
`String`	`THRESHOLD_VALUE_EXCEEDED`

Constructors

`MonitorNotification(String, Object, long, Date, String, ObjectName, String, Object, Object)`

Methods

`Object`	`getDerivedGauge()`
`String`	`getObservedAttribute()`
`ObjectName`	`getObservedObject()`
`Object`	`getTrigger()`

Inherited Member Summary

Fields inherited from class java.util.EventObject

`source`

Methods inherited from class `Notification`

`getMessage()`, `getSequenceNumber()`, `getSource()`, `getTimeStamp()`, `getType()`, `getUserData()`, `setSource(Object)`, `setUserData(Object)`

Methods inherited from class java.util.EventObject

`toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Fields

OBSERVED_ATTRIBUTE_ERROR

```
public static final java.lang.String OBSERVED_ATTRIBUTE_ERROR
```

Notification type denoting that the observed attribute is not contained in the observed object. This notification is fired by all kind of monitors. The value of this notification type is `jmx.monitor.error.attribute`.

OBSERVED_ATTRIBUTE_TYPE_ERROR

```
public static final java.lang.String OBSERVED_ATTRIBUTE_TYPE_ERROR
```

Notification type denoting that the type of the observed attribute is not correct. This notification is fired by all kind of monitors. The value of this notification type is `jmx.monitor.error.type`.

OBSERVED_OBJECT_ERROR

```
public static final java.lang.String OBSERVED_OBJECT_ERROR
```

Notification type denoting that the observed object is not registered in the MBean server. This notification is fired by all kind of monitors. The value of this notification type is `jmx.monitor.error.mbean`.

RUNTIME_ERROR

```
public static final java.lang.String RUNTIME_ERROR
```

Notification type denoting that a non-predefined error type has occured when trying to get the value of the observed attribute. This notification is fired by all kind of monitors. The value of this notification type is `jmx.monitor.error.runtime`.

STRING_TO_COMPARE_VALUE_DIFFERED

```
public static final java.lang.String STRING_TO_COMPARE_VALUE_DIFFERED
```

Notification type denoting that the observed attribute has differed from the "string to compare" value. This notification is only fired by string monitors. The value of this notification type is `jmx.monitor.string.differs`.

STRING_TO_COMPARE_VALUE_MATCHED

```
public static final java.lang.String STRING_TO_COMPARE_VALUE_MATCHED
```

Notification type denoting that the observed attribute has matched the "string to compare" value. This notification is only fired by string monitors. The value of this notification type is `jmx.monitor.string.matches`.

THRESHOLD_ERROR

```
public static final java.lang.String THRESHOLD_ERROR
```

Notification type denoting that the type of the threshold, offset or modulus (counter monitors) is not correct or that the threshold high value is less than the threshold low value (gauge monitors). This notification is fired by counter and gauge monitors. The value of this notification type is `jmx.monitor.error.threshold`.

THRESHOLD_HIGH_VALUE_EXCEEDED

```
public static final java.lang.String THRESHOLD_HIGH_VALUE_EXCEEDED
```

Notification type denoting that the observed attribute has exceeded the threshold high value. This notification is only fired by gauge monitors. The value of this notification type is `jmx.monitor.gauge.high`.

THRESHOLD_LOW_VALUE_EXCEEDED

```
public static final java.lang.String THRESHOLD_LOW_VALUE_EXCEEDED
```

Notification type denoting that the observed attribute has exceeded the threshold low value. This notification is only fired by gauge monitors. The value of this notification type is `jmx.monitor.gauge.low`.

THRESHOLD_VALUE_EXCEEDED

```
public static final java.lang.String THRESHOLD_VALUE_EXCEEDED
```

Notification type denoting that the observed attribute has reached the threshold value. This notification is only fired by counter monitors. The value of this notification type is `jmx.monitor.counter.threshold`.

Constructors

MonitorNotification(String, Object, long, Date, String, ObjectName, String, Object, Object)

```
public  MonitorNotification(java.lang.String type, java.lang.Object source, long
            sequenceNumber, java.util.Date timeStamp, java.lang.String msg, ObjectName
            obsObj, java.lang.String obsAtt, java.lang.Object derGauge, java.lang.Object
            trigger)
```

Creates a monitor notification object.

Parameters:

`type` - The notification type.

`source` - The notification producer.

`sequenceNumber` - The notification sequence number within the source object.

`timeStamp` - The notification emission date.

`msg` - The notification message.

`obsObj` - The object observed by the producer of this notification.

`obsAtt` - The attribute observed by the producer of this notification.

`derGauge` - The derived gauge.

`trigger` - The threshold/string (depending on the monitor type) that triggered off the notification.

Methods

---

getDerivedGauge()

`public java.lang.Object getDerivedGauge()`

Gets the derived gauge of this monitor notification.

Returns: The derived gauge.

getObservedAttribute()

`public java.lang.String getObservedAttribute()`

Gets the observed attribute of this monitor notification.

Returns: The observed attribute.

getObservedObject()

`public ObjectName getObservedObject()`

Gets the observed object of this monitor notification.

Returns: The observed object.

getTrigger()

`public java.lang.Object getTrigger()`

Gets the threshold/string (depending on the monitor type) that triggered off this monitor notification.

Returns: The trigger.

javax.management.monitor

MonitorSettingException

Syntax

```
public class MonitorSettingException extends JMRuntimeException
```

```
java.lang.Object
  |
  +--java.lang.Throwable
        |
        +--java.lang.Exception
              |
              +--java.lang.RuntimeException
                    |
                    +--JMRuntimeException
                          |
                          +--javax.management.monitor.MonitorSettingException
```

All Implemented Interfaces: java.io.Serializable

Description

This exception is thrown by the monitor when a monitor setting becomes invalid while the monitor is running.

As the monitor attributes may change at runtime, a check is performed before each observation. If a monitor attribute has become invalid, a MonitorSettingException is thrown.

Member Summary

Constructors

`MonitorSettingException()`
`MonitorSettingException(String)`

Inherited Member Summary

Methods inherited from class java.lang.Throwable

`fillInStackTrace, getLocalizedMessage, getMessage, printStackTrace, printStackTrace, printStackTrace, toString`

Methods inherited from class java.lang.Object

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

Constructors

MonitorSettingException()

`public  MonitorSettingException()`

Default constructor.

MonitorSettingException(String)

`public  MonitorSettingException(java.lang.String message)`

Constructor that allows an error message to be specified.

Parameters:

`message` - The specific error message.

javax.management.monitor

StringMonitor

Syntax

```
public class StringMonitor extends Monitor implements StringMonitorMBean
```

```
java.lang.Object
  |
  +--NotificationBroadcasterSupport
        |
        +--Monitor
              |
              +--javax.management.monitor.StringMonitor
```

All Implemented Interfaces: `MBeanRegistration`, `MonitorMBean`, `NotificationBroadcaster`, java.io.Serializable, `StringMonitorMBean`

Description

This class defines a monitor MBean designed to observe the values of a string attribute.

A string monitor sends notifications as follows:

- if the attribute value is matching the string to compare value, a `STRING_TO_COMPARE_VALUE_MATCHED` is sent. The notify match flag must be set to true. Subsequent matching of the string to compare value do not cause further notifications unless the attribute value differs from the string to compare value.
- if the attribute value is differing from the string to compare value, a `STRING_TO_COMPARE_VALUE_DIFFERED` is sent. The notify differ flag must be set to true. Subsequent differing from the string to compare value do not cause further notifications unless the attribute value matches the string to compare value.

Member Summary

Constructors		
		`StringMonitor()`
Methods		
	`String`	`getDerivedGauge()`
	`Date`	`getDerivedGaugeTimeStamp()`
	`boolean`	`getNotifyDiffer()`
	`boolean`	`getNotifyMatch()`
	`String`	`getStringToCompare()`
	`void`	`setNotifyDiffer(boolean)`
	`void`	`setNotifyMatch(boolean)`
	`void`	`setStringToCompare(String)`
	`void`	`start()`
	`void`	`stop()`

Inherited Member Summary

Fields inherited from class `Monitor`

`alreadyNotified`, `dbgTag`, `OBSERVED_ATTRIBUTE_ERROR_NOTIFIED`, `OBSERVED_ATTRIBUTE_TYPE_ERROR_NOTIFIED`, `OBSERVED_OBJECT_ERROR_NOTIFIED`, `RESET_FLAGS_ALREADY_NOTIFIED`, `RUNTIME_ERROR_NOTIFIED`, `server`

Methods inherited from class `Monitor`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `postDeregister()`, `postRegister(Boolean)`, `preDeregister()`, `preRegister(MBeanServer, ObjectName)`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from class `NotificationBroadcasterSupport`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`, `sendNotification(Notification)`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`

Methods inherited from interface `MBeanRegistration`

`postDeregister()`, `postRegister(Boolean)`, `preDeregister()`, `preRegister(MBeanServer, ObjectName)`

Methods inherited from interface `NotificationBroadcaster`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`

Constructors

StringMonitor()

```
public  StringMonitor()
```

Default constructor.

Methods

getDerivedGauge()

```
public java.lang.String getDerivedGauge()
```

Gets the derived gauge.

Specified By: `getDerivedGauge()` in interface `StringMonitorMBean`

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

`public java.util.Date getDerivedGaugeTimeStamp()`

Gets the derived gauge timestamp.

Specified By: `getDerivedGaugeTimeStamp()` in interface `StringMonitorMBean`

Returns: The derived gauge timestamp.

---

getNotifyDiffer()

`public boolean getNotifyDiffer()`

Gets the differing notifications on/off switch value.

Specified By: `getNotifyDiffer()` in interface `StringMonitorMBean`

Returns: True if the StringMonitor notifies when differing from the string to compare, false otherwise.

---

getNotifyMatch()

`public boolean getNotifyMatch()`

Gets the matching notifications on/off switch value.

Specified By: `getNotifyMatch()` in interface `StringMonitorMBean`

Returns: True if the StringMonitor notifies when matching the string to compare, false otherwise.

---

getStringToCompare()

`public java.lang.String getStringToCompare()`

Gets the string to compare with the observed attribute.

Specified By: `getStringToCompare()` in interface `StringMonitorMBean`

Returns: The string value.

---

setNotifyDiffer(boolean)

`public void setNotifyDiffer(boolean value)`

Sets the differing notifications on/off switch value.

Specified By: `setNotifyDiffer(boolean)` in interface `StringMonitorMBean`

Parameters:

`value` - The differing notifications on/off switch value.

---

setNotifyMatch(boolean)

`public void setNotifyMatch(boolean value)`

Sets the matching notifications on/off switch value.

Specified By: `setNotifyMatch(boolean)` in interface `StringMonitorMBean`

Parameters:
`value` - The matching notifications on/off switch value.

---

setStringToCompare(String)

`public void setStringToCompare(java.lang.String value)`

Sets the string to compare with the observed attribute.

Specified By: `setStringToCompare(String)` in interface `StringMonitorMBean`

Parameters:
`value` - The string value.

Throws: `java.lang.IllegalArgumentException` - The specified string to compare is null.

---

start()

`public synchronized void start()`

Starts the StringMonitor.

Specified By: `start()` in interface `MonitorMBean`

Specified By: `start()` in interface `MonitorMBean`

Overrides: `start()` in class `Monitor`

---

stop()

`public synchronized void stop()`

Stops the StringMonitor.

Specified By: `stop()` in interface `MonitorMBean`

Specified By: `stop()` in interface `MonitorMBean`

Overrides: `stop()` in class `Monitor`

javax.management.monitor
StringMonitorMBean

Syntax

`public interface StringMonitorMBean extends MonitorMBean`

All Superinterfaces: `MonitorMBean`

All Known Implementing Classes: `StringMonitor`

Description

This interface exposes the remote management interface of the StringMonitor MBean.

Member Summary

Methods

`String`	`getDerivedGauge()`
`Date`	`getDerivedGaugeTimeStamp()`
`boolean`	`getNotifyDiffer()`
`boolean`	`getNotifyMatch()`
`String`	`getStringToCompare()`
`void`	`setNotifyDiffer(boolean)`
`void`	`setNotifyMatch(boolean)`
`void`	`setStringToCompare(String)`

Inherited Member Summary

Methods inherited from interface `MonitorMBean`

`getGranularityPeriod()`, `getObservedAttribute()`, `getObservedObject()`, `isActive()`, `setGranularityPeriod(long)`, `setObservedAttribute(String)`, `setObservedObject(ObjectName)`, `start()`, `stop()`

Methods

---

getDerivedGauge()

```
public java.lang.String getDerivedGauge()
```

Gets the derived gauge.

Returns: The derived gauge.

---

getDerivedGaugeTimeStamp()

`public java.util.Date getDerivedGaugeTimeStamp()`

Gets the derived gauge timestamp.

Returns: The derived gauge timestamp.

---

getNotifyDiffer()

`public boolean getNotifyDiffer()`

Gets the differing notifications on/off switch value.

Returns: True if the StringMonitor notifies when differing, false otherwise.

---

getNotifyMatch()

`public boolean getNotifyMatch()`

Gets the matching notifications on/off switch value.

Returns: True if the StringMonitor notifies when matching, false otherwise.

---

getStringToCompare()

`public java.lang.String getStringToCompare()`

Gets the string to compare with the observed attribute.

Returns: The string value.

---

setNotifyDiffer(boolean)

`public void setNotifyDiffer(boolean value)`

Sets the differing notifications on/off switch value.

Parameters:
`value` - The differing notifications on/off switch value.

---

setNotifyMatch(boolean)

`public void setNotifyMatch(boolean value)`

Sets the matching notifications on/off switch value.

Parameters:
`value` - The matching notifications on/off switch value.

---

setStringToCompare(String)

`public void setStringToCompare(java.lang.String value)`

Sets the string to compare with the observed attribute.

Parameters:

`value` - The string value.

Throws: `java.lang.IllegalArgumentException` - The specified string to compare is null.

package

javax.management.timer

Description

Provides the definition of the **Timer MBean**.

Class Summary	
Interfaces	
`TimerMBean`	This interface exposes the management interface of the Timer MBean.
Classes	
`Timer`	This class provides the implementation of the Timer MBean.

javax.management.timer
Timer

Syntax

```
public class Timer extends NotificationBroadcasterSupport implements TimerMBean,
   java.io.Serializable

java.lang.Object
  |
  +--NotificationBroadcasterSupport
        |
        +--javax.management.timer.Timer
```

All Implemented Interfaces: `NotificationBroadcaster`, java.io.Serializable, `TimerMBean`

Description

This class provides the implementation of the Timer MBean. The Timer MBean sends out an alarm at a specified time that wakes up all the listeners registered to receive timer notifications.

This class manages a list of dated timer notifications. A method allows users to add/remove as many notifications as required. Added timer notifications can be made into regularly repeating notifications.

Note:

All notifications before the time when the `addNotification` method is called are ignored irrespective of the `sendPastNotifications` flag.

Member Summary

Fields		
	`long`	`ONE_DAY`
	`long`	`ONE_HOUR`
	`long`	`ONE_MINUTE`
	`long`	`ONE_SECOND`
	`long`	`ONE_WEEK`
Constructors		
		`Timer()`
Methods		
	`void`	`addNotification(Notification, Date)`
	`void`	`addNotification(Notification, Date, long)`
	`void`	`addNotification(Notification, Date, long, long)`
	`Date`	`getDate(int)`
	`Date`	`getDate(Notification)`
	`int`	`getNbNotifications()`
	`Long`	`getNbOccurences(int)`
	`Long`	`getNbOccurences(Notification)`
	`Notification[]`	`getNotifications()`
	`Long`	`getPeriodInMillis(int)`
	`Long`	`getPeriodInMillis(Notification)`
	`boolean`	`getSendPastNotifications()`

Member Summary

`boolean`	`isActive()`
`boolean`	`isEmpty()`
`void`	`removeAllNotifications()`
`void`	`removeNotification(int)`
`void`	`removeNotification(Notification)`
`void`	`setSendPastNotifications(boolean)`
`void`	`start()`
`void`	`stop()`

Inherited Member Summary

Methods inherited from class `NotificationBroadcasterSupport`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`, `sendNotification(Notification)`

Methods inherited from class java.lang.Object

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

Methods inherited from interface `NotificationBroadcaster`

`addNotificationListener(NotificationListener, NotificationFilter, Object)`, `getNotificationInfo()`, `removeNotificationListener(NotificationListener)`

Fields

ONE_DAY

`public static final long ONE_DAY`

Number of milliseconds in one day. Useful constant for the `addNotification` method.

ONE_HOUR

`public static final long ONE_HOUR`

Number of milliseconds in one hour. Useful constant for the `addNotification` method.

ONE_MINUTE

`public static final long ONE_MINUTE`

Number of milliseconds in one minute. Useful constant for the `addNotification` method.

ONE_SECOND

```
public static final long ONE_SECOND
```

Number of milliseconds in one second. Useful constant for the `addNotification` method.

ONE_WEEK

```
public static final long ONE_WEEK
```

Number of milliseconds in one week. Useful constant for the `addNotification` method.

Constructors

Timer()

```
public  Timer()
```

Default constructor.

Methods

---

addNotification(Notification, Date)

```
public synchronized void addNotification(Notification notification, java.util.Date date)
```

Inserts the specified timer notification into the list of notifications with a given date and a null period and number of occurences.

The timer notification will be handled once at the specified date.

Specified By: `addNotification(Notification, Date)` in interface `TimerMBean`

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The date notification is before the current date.

---

addNotification(Notification, Date, long)

```
public synchronized void addNotification(Notification notification, java.util.Date date,
             long periodInMillis)
```

Inserts the specified timer notification into the list of notifications with a given date and period and a null number of occurences.

The timer notification will repeat continuously using the timer period.

Specified By: `addNotification(Notification, Date, long)` in interface `TimerMBean`

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

`periodInMillis` - The period of the timer notification.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The period is negative or the date notification is before the current date.

---

addNotification(Notification, Date, long, long)

```
public synchronized void addNotification(Notification notification, java.util.Date date,
            long periodInMillis, long nbOccurences)
```

Inserts the specified timer notification into the list of notifications with a given date, period and number of occurences.

If the timer notification to be inserted has a date that is before the current date, the method attempts to update the notification using the defined period. If the defined period is null then the method generates an exception.

If the timer notification has a non null period, the date of the notification is updated by adding the period until the notification date is later than the current date. Then the timer notification is inserted into the list of notifications.

If the specified number of occurences is non null, the previous mechanism is employed. The number of times that the period is added is limited by the specified number of occurences. If the notification date remains earlier than the current date, an exception is generated.

A specified timer notification can be added only once.

Note that once the timer notification has been added into the list of notifications, its associated date, period and number of occurences cannot be updated.

Specified By: `addNotification(Notification, Date, long, long)` in interface `TimerMBean`

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

`periodInMillis` - The period of the timer notification.

`nbOccurences` - The total number the timer notification will be emitted.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The period or the number of occurences is negative or the date notification is before the current date.

---

getDate(int)

```
public java.util.Date getDate(int index)
```

Gets a copy of the date of the timer notification at the specified index.

Specified By: `getDate(int)` in interface `TimerMBean`

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the date at the specified index in the list of notifications or null if the specified index is invalid.

---

getDate(Notification)

```
public java.util.Date getDate(Notification notification)
```

Gets a copy of the date of the specified timer notification.

Specified By: `getDate(Notification)` in interface `TimerMBean`

Parameters:
`notification` - The timer notification.

Returns: A copy of the date of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getNbNotifications()

```
public int getNbNotifications()
```

Gets the number of notifications registered into the list of notifications.

Specified By: `getNbNotifications()` in interface `TimerMBean`

Returns: The number of notifications.

---

getNbOccurences(int)

```
public java.lang.Long getNbOccurences(int index)
```

Gets a copy of the remaining number of occurences of the timer notification at the specified index.

Specified By: `getNbOccurences(int)` in interface `TimerMBean`

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the remaining number of occurences at the specified index in the list of notifications or null if the specified index is invalid.

---

getNbOccurences(Notification)

```
public java.lang.Long getNbOccurences(Notification notification)
```

Gets a copy of the remaining number of occurences of the specified timer notification.

Specified By: `getNbOccurences(Notification)` in interface `TimerMBean`

Parameters:
`notification` - The timer notification.

Returns: A copy of the remaining number of occurences of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getNotifications()

`public synchronized Notification getNotifications()`

Gets all the notifications registered into the list of notifications.

Specified By: `getNotifications()` in interface `TimerMBean`

Returns: An array containing all the notifications registered into the list of notifications.

---

getPeriodInMillis(int)

`public java.lang.Long getPeriodInMillis(int index)`

Gets a copy of the period in milliseconds of the timer notification at the specified index.

Specified By: `getPeriodInMillis(int)` in interface `TimerMBean`

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the period at the specified index in the list of notifications or null if the specified index is invalid.

---

getPeriodInMillis(Notification)

`public java.lang.Long getPeriodInMillis(Notification notification)`

Gets a copy of the period in milliseconds of the specified timer notification.

Specified By: `getPeriodInMillis(Notification)` in interface `TimerMBean`

Parameters:
`notification` - The timer notification.

Returns: A copy of the period of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getSendPastNotifications()

`public boolean getSendPastNotifications()`

Gets the flag indicating if the Timer sends past notifications or not.

Specified By: `getSendPastNotifications()` in interface `TimerMBean`

Returns: The past notifications sending on/off flag value.

---

isActive()

`public boolean isActive()`

Tests if the Timer MBean is active. A Timer MBean is marked active when the `start()` method is called. It becomes inactive when the `stop()` method is called.

Specified By: `isActive()` in interface `TimerMBean`

Returns: True if the Timer MBean is active, false otherwise.

---

isEmpty()

```
public boolean isEmpty()
```

Tests if the list of notifications is empty.

Specified By: `isEmpty()` in interface `TimerMBean`

Returns: True if the list of notifications is empty, false otherwise.

---

removeAllNotifications()

```
public synchronized void removeAllNotifications()
```

Removes all the notifications from the list of notifications.

Specified By: `removeAllNotifications()` in interface `TimerMBean`

---

removeNotification(int)

```
public synchronized void removeNotification(int index)
```

Removes the timer notification at the specified index from the list of notifications. Each component in this list with an index greater or equal to the specified index is shifted downward to have an index one smaller than the value it had previously.

Specified By: `removeNotification(int)` in interface `TimerMBean`

Parameters:
`index` - The index of the timer notification to remove.

Throws: `InstanceNotFoundException` - The index is invalid.

---

removeNotification(Notification)

```
public synchronized void removeNotification(Notification notification)
```

Removes the specified timer notification from the list of notifications. If the object is found in this list, each component in the list with an index greater or equal to the object's index is shifted downward to have an index one smaller than the value it had previously.

Specified By: `removeNotification(Notification)` in interface `TimerMBean`

Parameters:
`notification` - The timer notification to be removed.

Throws: `InstanceNotFoundException` - The timer notification to remove is not in the list of notifications.

---

setSendPastNotifications(boolean)

```
public void setSendPastNotifications(boolean value)
```

Sets the flag indicating if the Timer sends past notifications or not.

Specified By: `setSendPastNotifications(boolean)` in interface `TimerMBean`

Parameters:
`value` - The past notifications sending on/off flag value.

---

start()

```
public synchronized void start()
```

Starts the Timer.

Specified By: `start()` in interface `TimerMBean`

---

stop()

```
public synchronized void stop()
```

Stops the Timer.

Specified By: `stop()` in interface `TimerMBean`

javax.management.timer
TimerMBean

Syntax

```
public interface TimerMBean
```

All Known Implementing Classes: `Timer`

Description

This interface exposes the management interface of the Timer MBean.

Member Summary

Methods

`void`	`addNotification(Notification, Date)`
`void`	`addNotification(Notification, Date, long)`
`void`	`addNotification(Notification, Date, long, long)`
`Date`	`getDate(int)`
`Date`	`getDate(Notification)`
`int`	`getNbNotifications()`
`Long`	`getNbOccurences(int)`
`Long`	`getNbOccurences(Notification)`
`Notification[]`	`getNotifications()`
`Long`	`getPeriodInMillis(int)`
`Long`	`getPeriodInMillis(Notification)`
`boolean`	`getSendPastNotifications()`
`boolean`	`isActive()`
`boolean`	`isEmpty()`
`void`	`removeAllNotifications()`
`void`	`removeNotification(int)`
`void`	`removeNotification(Notification)`
`void`	`setSendPastNotifications(boolean)`
`void`	`start()`
`void`	`stop()`

Methods

addNotification(Notification, Date)

```
public void addNotification(Notification notification, java.util.Date date)
```

Inserts the specified timer notification into the list of notifications with a given date and a null period and number of occurences.

The timer notification will be handled once at the specified date.

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The date notification is before the current date.

---

addNotification(Notification, Date, long)

```
public void addNotification(Notification notification, java.util.Date date, long
            periodInMillis)
```

Inserts the specified timer notification into the list of notifications with a given date and period and a null number of occurences.

The timer notification will repeat continuously using the timer period.

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

`periodInMillis` - The period of the timer notification.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The period is negative or the date notification is before the current date.

---

addNotification(Notification, Date, long, long)

```
public void addNotification(Notification notification, java.util.Date date, long
            periodInMillis, long nbOccurences)
```

Inserts the specified timer notification into the list of notifications with a given date, period and number of occurences.

If the timer notification to be inserted has a date that is before the current date, the method attempts to update the notification using the defined period. If the defined period is null then the method generates an exception.

If the timer notification has a non null period, the date of the notification is updated by adding the period until the notification date is later than the current date. Then the timer notification is inserted into the list of notifications.

If the specified number of occurences is non null, the previous mechanism is employed. The number of times that the period is added is limited by the specified number of occurences. If the notification date remains earlier than the current date, an exception is generated.

A specified timer notification can be added only once.

Note that once the timer notification has been added into the list of notifications, its associated date, period and number of occurences cannot be updated.

Parameters:

`notification` - The timer notification to be added.

`date` - The date when the notification occurs.

`periodInMillis` - The period of the timer notification.

`nbOccurences` - The total number the timer notification will be emitted.

Throws: `InstanceAlreadyExistsException` - The timer notification to add is already in the list of notifications.

`java.lang.IllegalArgumentException` - The period or the number of occurences is negative or the date notification is before the current date.

---

getDate(int)

`public java.util.Date getDate(int index)`

Gets a copy of the date of the timer notification at the specified index.

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the date at the specified index in the list of notifications or null if the specified index is invalid.

---

getDate(Notification)

`public java.util.Date getDate(Notification notification)`

Gets a copy of the date of the specified timer notification.

Parameters:
`notification` - The timer notification.

Returns: A copy of the date of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getNbNotifications()

`public int getNbNotifications()`

Gets the number of notifications registered into the list of notifications.

Returns: The number of notifications.

---

getNbOccurences(int)

`public java.lang.Long getNbOccurences(int index)`

Gets a copy of the remaining number of occurences of the timer notification at the specified index.

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the remaining number of occurences at the specified index in the list of notifications or null if the specified index is invalid.

---

getNbOccurences(Notification)

`public java.lang.Long getNbOccurences(Notification notification)`

Gets a copy of the remaining number of occurences of the specified timer notification.

Parameters:
`notification` - The timer notification.

Returns: A copy of the remaining number of occurences of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getNotifications()

`public Notification getNotifications()`

Gets all the notifications registered into the list of notifications.

Returns: An array containing all the notifications registered into the list of notifications.

---

getPeriodInMillis(int)

`public java.lang.Long getPeriodInMillis(int index)`

Gets a copy of the period in milliseconds of the timer notification at the specified index.

Parameters:
`index` - The index of the timer notification.

Returns: A copy of the period at the specified index in the list of notifications or null if the specified index is invalid.

---

getPeriodInMillis(Notification)

`public java.lang.Long getPeriodInMillis(Notification notification)`

Gets a copy of the period in milliseconds of the specified timer notification.

Parameters:
`notification` - The timer notification.

Returns: A copy of the period of the specified timer notification or null if the specified timer notification is not in the list of notifications.

---

getSendPastNotifications()

`public boolean getSendPastNotifications()`

Gets the flag indicating if the Timer sends past notifications or not.

Returns: The past notifications sending on/off flag value.

---

isActive()

`public boolean isActive()`

Tests if the Timer MBean is active. A Timer MBean is marked active when the `start()` method is called. It becomes inactive when the `stop()` method is called.

Returns: True if the Timer MBean is active, false otherwise.

---

isEmpty()

`public boolean isEmpty()`

Tests if the list of notifications is empty.

Returns: True if the list of notifications is empty, false otherwise.

---

removeAllNotifications()

`public void removeAllNotifications()`

Removes all the notifications from the list of notifications.

---

removeNotification(int)

`public void removeNotification(int index)`

Removes the timer notification at the specified index from the list of notifications. Each component in this list with an index greater or equal to the specified index is shifted downward to have an index one smaller than the value it had previously.

Parameters:
`index` - The index of the timer notification to remove.

Throws: `InstanceNotFoundException` - The index is invalid.

---

removeNotification(Notification)

`public void removeNotification(Notification notification)`

Removes the specified timer notification from the list of notifications. If the object is found in this list, each component in the list with an index greater or equal to the object's index is shifted downward to have an index one smaller than the value it had previously.

Parameters:
`notification` - The timer notification to be removed.

Throws: `InstanceNotFoundException` - The timer notification to remove is not in the list of notifications.

---

setSendPastNotifications(boolean)

`public void setSendPastNotifications(boolean value)`

Sets the flag indicating if the Timer sends past notifications or not.

Parameters:
`value` - The past notifications sending on/off flag value.

---

start()

`public void start()`

Starts the Timer.

stop()

```
public void stop()
```

Stops the Timer.

Index