| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Descriptor.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/management/DefaultLoaderRepository.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/DescriptorAccess.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/Descriptor.html)    [**NO FRAMES**](http://docs.google.com/Descriptor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#lnxbz9) | DETAIL: FIELD | CONSTR | [METHOD](#35nkun2) |

## **javax.management**

Interface Descriptor

**All Superinterfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **All Known Implementing Classes:** [DescriptorSupport](http://docs.google.com/javax/management/modelmbean/DescriptorSupport.html), [ImmutableDescriptor](http://docs.google.com/javax/management/ImmutableDescriptor.html)

public interface **Descriptor**extends [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html)

Additional metadata for a JMX element. A Descriptor is associated with a [MBeanInfo](http://docs.google.com/javax/management/MBeanInfo.html), [MBeanAttributeInfo](http://docs.google.com/javax/management/MBeanAttributeInfo.html), etc. It consists of a collection of fields. A field is a name and an associated value.

Field names are not case-sensitive. The names descriptorType, descriptortype, and DESCRIPTORTYPE are all equivalent. However, the case that was used when the field was first set is preserved in the result of the [getFields()](http://docs.google.com/javax/management/Descriptor.html#getFields()) and [getFieldNames()](http://docs.google.com/javax/management/Descriptor.html#getFieldNames()) methods.

Not all field names and values are predefined. New fields can be defined and added by any program.

A descriptor can be mutable or immutable. An immutable descriptor, once created, never changes. The Descriptor methods that could modify the contents of the descriptor will throw an exception for an immutable descriptor. Immutable descriptors are usually instances of [ImmutableDescriptor](http://docs.google.com/javax/management/ImmutableDescriptor.html) or a subclass. Mutable descriptors are usually instances of [DescriptorSupport](http://docs.google.com/javax/management/modelmbean/DescriptorSupport.html) or a subclass.

Certain fields are used by the JMX implementation. This means either that the presence of the field may change the behavior of the JMX API or that the field may be set in descriptors returned by the JMX API. These fields appear in *italics* in the table below, and each one has a corresponding constant in the [JMX](http://docs.google.com/javax/management/JMX.html) class. For example, the field defaultValue is represented by the constant [JMX.DEFAULT\_VALUE\_FIELD](http://docs.google.com/javax/management/JMX.html#DEFAULT_VALUE_FIELD).

Certain other fields have conventional meanings described in the table below but they are not required to be understood or set by the JMX implementation.

Field names defined by the JMX specification in this and all future versions will never contain a period (.). Users can safely create their own fields by including a period in the name and be sure that these names will not collide with any future version of the JMX API. It is recommended to follow the Java package naming convention to avoid collisions between field names from different origins. For example, a field created by example.com might have the name com.example.interestLevel.

Note that the values in the defaultValue, legalValues, maxValue, and minValue fields should be consistent with the type returned by the getType() method for the associated MBeanAttributeInfo or MBeanParameterInfo. For MXBeans, this means that they should be of the mapped Java type, called *opendata*(J) in the [MXBean type mapping rules](http://docs.google.com/MXBean.html#mapping-rules).

| Name | Type | Used in | Meaning |
| --- | --- | --- | --- |
| *defaultValue* | Object | MBeanAttributeInfo  MBeanParameterInfo | Default value for an attribute or parameter. See [javax.management.openmbean](http://docs.google.com/javax/management/openmbean/package-summary.html). |
| deprecated | String | Any | An indication that this element of the information model is no longer recommended for use. A set of MBeans defined by an application is collectively called an *information model*. The convention is for the value of this field to contain a string that is the version of the model in which the element was first deprecated, followed by a space, followed by an explanation of the deprecation, for example "1.3 Replaced by the Capacity attribute". |
| descriptionResource  BundleBaseName | String | Any | The base name for the [ResourceBundle](http://docs.google.com/java/util/ResourceBundle.html) in which the key given in the descriptionResourceKey field can be found, for example "com.example.myapp.MBeanResources". The meaning of this field is defined by this specification but the field is not set or used by the JMX API itself. |
| descriptionResourceKey | String | Any | A resource key for the description of this element. In conjunction with the descriptionResourceBundleBaseName, this can be used to find a localized version of the description. The meaning of this field is defined by this specification but the field is not set or used by the JMX API itself. |
| enabled | String | MBeanAttributeInfo  MBeanNotificationInfo  MBeanOperationInfo | The string "true" or "false" according as this item is enabled. When an attribute or operation is not enabled, it exists but cannot currently be accessed. A user interface might present it as a greyed-out item. For example, an attribute might only be meaningful after the start() method of an MBean has been called, and is otherwise disabled. Likewise, a notification might be disabled if it cannot currently be emitted but could be in other circumstances. |
| *immutableInfo* | String | MBeanInfo | The string "true" or "false" according as this MBean's MBeanInfo is *immutable*. When this field is true, the MBeanInfo for the given MBean is guaranteed not to change over the lifetime of the MBean. Hence, a client can read it once and cache the read value. When this field is false or absent, there is no such guarantee, although that does not mean that the MBeanInfo will necessarily change. |
| infoTimeout | String  Long | MBeanInfo | The time in milli-seconds that the MBeanInfo can reasonably be expected to be unchanged. The value can be a Long or a decimal string. This provides a hint from a DynamicMBean or any MBean that does not define immutableInfo as true that the MBeanInfo is not likely to change within this period and therefore can be cached. When this field is missing or has the value zero, it is not recommended to cache the MBeanInfo unless it has the immutableInfo set to true. |
| *interfaceClassName* | String | MBeanInfo | The Java interface name for a Standard MBean or MXBean, as returned by [Class.getName()](http://docs.google.com/java/lang/Class.html#getName()). A Standard MBean or MXBean registered directly in the MBean Server or created using the [StandardMBean](http://docs.google.com/javax/management/StandardMBean.html) class will have this field in its MBeanInfo Descriptor. |
| *legalValues* | Set<?> | MBeanAttributeInfo  MBeanParameterInfo | Legal values for an attribute or parameter. See [javax.management.openmbean](http://docs.google.com/javax/management/openmbean/package-summary.html). |
| *maxValue* | Object | MBeanAttributeInfo  MBeanParameterInfo | Maximum legal value for an attribute or parameter. See [javax.management.openmbean](http://docs.google.com/javax/management/openmbean/package-summary.html). |
| metricType | String | MBeanAttributeInfo  MBeanOperationInfo | The type of a metric, one of the strings "counter" or "gauge". A metric is a measurement exported by an MBean, usually an attribute but sometimes the result of an operation. A metric that is a *counter* has a value that never decreases except by being reset to a starting value. Counter metrics are almost always non-negative integers. An example might be the number of requests received. A metric that is a *gauge* has a numeric value that can increase or decrease. Examples might be the number of open connections or a cache hit rate or a temperature reading. |
| *minValue* | Object | MBeanAttributeInfo  MBeanParameterInfo | Minimum legal value for an attribute or parameter. See [javax.management.openmbean](http://docs.google.com/javax/management/openmbean/package-summary.html). |
| *mxbean* | String | MBeanInfo | The string "true" or "false" according as this MBean is an [MXBean](http://docs.google.com/javax/management/MXBean.html). A Standard MBean or MXBean registered directly with the MBean Server or created using the [StandardMBean](http://docs.google.com/javax/management/StandardMBean.html) class will have this field in its MBeanInfo Descriptor. |
| *openType* | [OpenType](http://docs.google.com/javax/management/openmbean/OpenType.html) | MBeanAttributeInfo  MBeanOperationInfo  MBeanParameterInfo | The Open Type of this element. In the case of MBeanAttributeInfo and MBeanParameterInfo, this is the Open Type of the attribute or parameter. In the case of MBeanOperationInfo, it is the Open Type of the return value. This field is set in the Descriptor for all instances of [OpenMBeanAttributeInfoSupport](http://docs.google.com/javax/management/openmbean/OpenMBeanAttributeInfoSupport.html), [OpenMBeanOperationInfoSupport](http://docs.google.com/javax/management/openmbean/OpenMBeanOperationInfoSupport.html), and [OpenMBeanParameterInfoSupport](http://docs.google.com/javax/management/openmbean/OpenMBeanParameterInfoSupport.html). It is also set for attributes, operations, and parameters of MXBeans.  This field can be set for an MBeanNotificationInfo, in which case it indicates the Open Type that the [user data](http://docs.google.com/javax/management/Notification.html#getUserData()) will have. |
| *originalType* | String | MBeanAttributeInfo  MBeanOperationInfo  MBeanParameterInfo | The original Java type of this element as it appeared in the [MXBean](http://docs.google.com/javax/management/MXBean.html) interface method that produced this MBeanAttributeInfo (etc). For example, a method  public [MemoryUsage](http://docs.google.com/java/lang/management/MemoryUsage.html) getHeapMemoryUsage();  in an MXBean interface defines an attribute called HeapMemoryUsage of type [CompositeData](http://docs.google.com/javax/management/openmbean/CompositeData.html). The originalType field in the Descriptor for this attribute will have the value "java.lang.management.MemoryUsage".  The format of this string is described in the section [Type Names](http://docs.google.com/MXBean.html#type-names) of the MXBean specification. |
| severity | String  Integer | MBeanNotificationInfo | The severity of this notification. It can be 0 to mean unknown severity or a value from 1 to 6 representing decreasing levels of severity. It can be represented as a decimal string or an Integer. |
| since | String | Any | The version of the information model in which this element was introduced. A set of MBeans defined by an application is collectively called an *information model*. The application may also define versions of this model, and use the "since" field to record the version in which an element first appeared. |
| units | String | MBeanAttributeInfo  MBeanParameterInfo  MBeanOperationInfo | The units in which an attribute, parameter, or operation return value is measured, for example "bytes" or "seconds". |

Some additional fields are defined by Model MBeans. See [ModelMBeanInfo](http://docs.google.com/javax/management/modelmbean/ModelMBeanInfo.html) and related classes and the chapter "Model MBeans" of the  [JMX Specification](http://java.sun.com/products/JavaManagement/download.html).

**Since:** 1.5

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/management/Descriptor.html#clone())()            Returns a descriptor which is equal to this descriptor. |
| boolean | [**equals**](http://docs.google.com/javax/management/Descriptor.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this descriptor to the given object. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getFieldNames**](http://docs.google.com/javax/management/Descriptor.html#getFieldNames())()            Returns all the field names in the descriptor. |
| [String](http://docs.google.com/java/lang/String.html)[] | [**getFields**](http://docs.google.com/javax/management/Descriptor.html#getFields())()            Returns all of the fields contained in this descriptor as a string array. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getFieldValue**](http://docs.google.com/javax/management/Descriptor.html#getFieldValue(java.lang.String))([String](http://docs.google.com/java/lang/String.html) fieldName)            Returns the value for a specific field name, or null if no value is present for that name. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getFieldValues**](http://docs.google.com/javax/management/Descriptor.html#getFieldValues(java.lang.String...))([String](http://docs.google.com/java/lang/String.html)... fieldNames)            Returns all the field values in the descriptor as an array of Objects. |
| int | [**hashCode**](http://docs.google.com/javax/management/Descriptor.html#hashCode())()            Returns the hash code value for this descriptor. |
| boolean | [**isValid**](http://docs.google.com/javax/management/Descriptor.html#isValid())()            Returns true if all of the fields have legal values given their names. |
| void | [**removeField**](http://docs.google.com/javax/management/Descriptor.html#removeField(java.lang.String))([String](http://docs.google.com/java/lang/String.html) fieldName)            Removes a field from the descriptor. |
| void | [**setField**](http://docs.google.com/javax/management/Descriptor.html#setField(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) fieldName, [Object](http://docs.google.com/java/lang/Object.html) fieldValue)            Sets the value for a specific field name. |
| void | [**setFields**](http://docs.google.com/javax/management/Descriptor.html#setFields(java.lang.String%5B%5D,%20java.lang.Object%5B%5D))([String](http://docs.google.com/java/lang/String.html)[] fieldNames, [Object](http://docs.google.com/java/lang/Object.html)[] fieldValues)            Sets all fields in the field names array to the new value with the same index in the field values array. |

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

### getFieldValue

[Object](http://docs.google.com/java/lang/Object.html) **getFieldValue**([String](http://docs.google.com/java/lang/String.html) fieldName)  
 throws [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Returns the value for a specific field name, or null if no value is present for that name.

**Parameters:**fieldName - the field name. **Returns:**the corresponding value, or null if the field is not present. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - if the field name is illegal.

### setField

void **setField**([String](http://docs.google.com/java/lang/String.html) fieldName,  
 [Object](http://docs.google.com/java/lang/Object.html) fieldValue)  
 throws [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Sets the value for a specific field name. This will modify an existing field or add a new field.

The field value will be validated before it is set. If it is not valid, then an exception will be thrown. The meaning of validity is dependent on the descriptor implementation.

**Parameters:**fieldName - The field name to be set. Cannot be null or empty.fieldValue - The field value to be set for the field name. Can be null if that is a valid value for the field. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - if the field name or field value is illegal (wrapped exception is [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)); or if the descriptor is immutable (wrapped exception is [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html)).

### getFields

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

Returns all of the fields contained in this descriptor as a string array.

**Returns:**String array of fields in the format *fieldName=fieldValue*

If the value of a field is not a String, then the toString() method will be called on it and the returned value, enclosed in parentheses, used as the value for the field in the returned array. If the value of a field is null, then the value of the field in the returned array will be empty. If the descriptor is empty, you will get an empty array.**See Also:**[setFields(java.lang.String[], java.lang.Object[])](http://docs.google.com/javax/management/Descriptor.html#setFields(java.lang.String%5B%5D,%20java.lang.Object%5B%5D))

### getFieldNames

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

Returns all the field names in the descriptor.

**Returns:**String array of field names. If the descriptor is empty, you will get an empty array.

### getFieldValues

[Object](http://docs.google.com/java/lang/Object.html)[] **getFieldValues**([String](http://docs.google.com/java/lang/String.html)... fieldNames)

Returns all the field values in the descriptor as an array of Objects. The returned values are in the same order as the fieldNames String array parameter.

**Parameters:**fieldNames - String array of the names of the fields that the values should be returned for. If the array is empty then an empty array will be returned. If the array is null then all values will be returned, as if the parameter were the array returned by [getFieldNames()](http://docs.google.com/javax/management/Descriptor.html#getFieldNames()). If a field name in the array does not exist, including the case where it is null or the empty string, then null is returned for the matching array element being returned. **Returns:**Object array of field values. If the list of fieldNames is empty, you will get an empty array.

### removeField

void **removeField**([String](http://docs.google.com/java/lang/String.html) fieldName)

Removes a field from the descriptor.

**Parameters:**fieldName - String name of the field to be removed. If the field name is illegal or the field is not found, no exception is thrown. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - if a field of the given name exists and the descriptor is immutable. The wrapped exception will be an [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html).

### setFields

void **setFields**([String](http://docs.google.com/java/lang/String.html)[] fieldNames,  
 [Object](http://docs.google.com/java/lang/Object.html)[] fieldValues)  
 throws [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Sets all fields in the field names array to the new value with the same index in the field values array. Array sizes must match.

The field value will be validated before it is set. If it is not valid, then an exception will be thrown. If the arrays are empty, then no change will take effect.

**Parameters:**fieldNames - String array of field names. The array and array elements cannot be null.fieldValues - Object array of the corresponding field values. The array cannot be null. Elements of the array can be null. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - if the change fails for any reason. Wrapped exception is [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) if fieldNames or fieldValues is null, or if the arrays are of different lengths, or if there is an illegal value in one of them. Wrapped exception is [UnsupportedOperationException](http://docs.google.com/java/lang/UnsupportedOperationException.html) if the descriptor is immutable, and the call would change its contents.**See Also:**[getFields()](http://docs.google.com/javax/management/Descriptor.html#getFields())

### clone

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

Returns a descriptor which is equal to this descriptor. Changes to the returned descriptor will have no effect on this descriptor, and vice versa. If this descriptor is immutable, it may fulfill this condition by returning itself.

**Returns:**A descriptor which is equal to this descriptor. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - for illegal value for field names or field values. If the descriptor construction fails for any reason, this exception will be thrown.

### isValid

boolean **isValid**()  
 throws [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html)

Returns true if all of the fields have legal values given their names.

**Returns:**true if the values are legal. **Throws:** [RuntimeOperationsException](http://docs.google.com/javax/management/RuntimeOperationsException.html) - If the validity checking fails for any reason, this exception will be thrown. The method returns false if the descriptor is not valid, but throws this exception if the attempt to determine validity fails.

### equals

boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Compares this descriptor to the given object. The objects are equal if the given object is also a Descriptor, and if the two Descriptors have the same field names (possibly differing in case) and the same associated values. The respective values for a field in the two Descriptors are equal if the following conditions hold:

* If one value is null then the other must be too.
* If one value is a primitive array then the other must be a primitive array of the same type with the same elements.
* If one value is an object array then the other must be too and [Arrays.deepEquals(Object[],Object[])](http://docs.google.com/java/util/Arrays.html#deepEquals(java.lang.Object%5B%5D,%20java.lang.Object%5B%5D)) must return true.
* Otherwise [Object.equals(Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) must return true.

**Overrides:**[equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)) in class [Object](http://docs.google.com/java/lang/Object.html) **Parameters:**obj - the object to compare with. **Returns:**true if the objects are the same; false otherwise.**Since:** 1.6 **See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

int **hashCode**()

Returns the hash code value for this descriptor. The hash code is computed as the sum of the hash codes for each field in the descriptor. The hash code of a field with name n and value v is n.toLowerCase().hashCode() ^ h. Here h is the hash code of v, computed as follows:

* If v is null then h is 0.
* If v is a primitive array then h is computed using the appropriate overloading of java.util.Arrays.hashCode.
* If v is an object array then h is computed using [Arrays.deepHashCode(Object[])](http://docs.google.com/java/util/Arrays.html#deepHashCode(java.lang.Object%5B%5D)).
* Otherwise h is v.hashCode().

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**A hash code value for this object.**Since:** 1.6 **See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Descriptor.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/management/DefaultLoaderRepository.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/DescriptorAccess.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/Descriptor.html)    [**NO FRAMES**](http://docs.google.com/Descriptor.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#lnxbz9) | DETAIL: FIELD | CONSTR | [METHOD](#35nkun2) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).