| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ObjectName.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/ObjectInstance.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/OperationsException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/ObjectName.html)    [**NO FRAMES**](http://docs.google.com/ObjectName.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#lnxbz9) |

## **javax.management**

Class ObjectName

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

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)>, [QueryExp](http://docs.google.com/javax/management/QueryExp.html)

public class **ObjectName**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Comparable](http://docs.google.com/java/lang/Comparable.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)>, [QueryExp](http://docs.google.com/javax/management/QueryExp.html)

Represents the object name of an MBean, or a pattern that can match the names of several MBeans. Instances of this class are immutable.

An instance of this class can be used to represent:

* An object name
* An object name pattern, within the context of a query

An object name consists of two parts, the domain and the key properties.

The *domain* is a string of characters not including the character colon (:). It is recommended that the domain should not contain the string "//", which is reserved for future use.

If the domain includes at least one occurrence of the wildcard characters asterisk (\*) or question mark (?), then the object name is a pattern. The asterisk matches any sequence of zero or more characters, while the question mark matches any single character.

If the domain is empty, it will be replaced in certain contexts by the *default domain* of the MBean server in which the ObjectName is used.

The *key properties* are an unordered set of keys and associated values.

Each *key* is a nonempty string of characters which may not contain any of the characters comma (,), equals (=), colon, asterisk, or question mark. The same key may not occur twice in a given ObjectName.

Each *value* associated with a key is a string of characters that is either unquoted or quoted.

An *unquoted value* is a possibly empty string of characters which may not contain any of the characters comma, equals, colon, or quote.

If the *unquoted value* contains at least one occurrence of the wildcard characters asterisk or question mark, then the object name is a *property value pattern*. The asterisk matches any sequence of zero or more characters, while the question mark matches any single character.

A *quoted value* consists of a quote ("), followed by a possibly empty string of characters, followed by another quote. Within the string of characters, the backslash (\) has a special meaning. It must be followed by one of the following characters:

* Another backslash. The second backslash has no special meaning and the two characters represent a single backslash.
* The character 'n'. The two characters represent a newline ('\n' in Java).
* A quote. The two characters represent a quote, and that quote is not considered to terminate the quoted value. An ending closing quote must be present for the quoted value to be valid.
* A question mark (?) or asterisk (\*). The two characters represent a question mark or asterisk respectively.

A quote may not appear inside a quoted value except immediately after an odd number of consecutive backslashes.

The quotes surrounding a quoted value, and any backslashes within that value, are considered to be part of the value.

If the *quoted value* contains at least one occurrence of the characters asterisk or question mark and they are not preceded by a backslash, then they are considered as wildcard characters and the object name is a *property value pattern*. The asterisk matches any sequence of zero or more characters, while the question mark matches any single character.

An ObjectName may be a *property list pattern*. In this case it may have zero or more keys and associated values. It matches a nonpattern ObjectName whose domain matches and that contains the same keys and associated values, as well as possibly other keys and values.

An ObjectName is a *property value pattern* when at least one of its *quoted* or *unquoted* key property values contains the wildcard characters asterisk or question mark as described above. In this case it has one or more keys and associated values, with at least one of the values containing wildcard characters. It matches a nonpattern ObjectName whose domain matches and that contains the same keys whose values match; if the property value pattern is also a property list pattern then the nonpattern ObjectName can contain other keys and values.

An ObjectName is a *property pattern* if it is either a *property list pattern* or a *property value pattern* or both.

An ObjectName is a pattern if its domain contains a wildcard or if the ObjectName is a property pattern.

If an ObjectName is not a pattern, it must contain at least one key with its associated value.

Examples of ObjectName patterns are:

* \*:type=Foo,name=Bar to match names in any domain whose exact set of keys is type=Foo,name=Bar.
* d:type=Foo,name=Bar,\* to match names in the domain d that have the keys type=Foo,name=Bar plus zero or more other keys.
* \*:type=Foo,name=Bar,\* to match names in any domain that has the keys type=Foo,name=Bar plus zero or more other keys.
* d:type=F?o,name=Bar will match e.g. d:type=Foo,name=Bar and d:type=Fro,name=Bar.
* d:type=F\*o,name=Bar will match e.g. d:type=Fo,name=Bar and d:type=Frodo,name=Bar.
* d:type=Foo,name="B\*" will match e.g. d:type=Foo,name="Bling". Wildcards are recognized even inside quotes, and like other special characters can be escaped with \.

An ObjectName can be written as a String with the following elements in order:

* The domain.
* A colon (:).
* A key property list as defined below.

A key property list written as a String is a comma-separated list of elements. Each element is either an asterisk or a key property. A key property consists of a key, an equals (=), and the associated value.

At most one element of a key property list may be an asterisk. If the key property list contains an asterisk element, the ObjectName is a property list pattern.

Spaces have no special significance in a String representing an ObjectName. For example, the String:

domain: key1 = value1 , key2 = value2

represents an ObjectName with two keys. The name of each key contains six characters, of which the first and last are spaces. The value associated with the key " key1 " also begins and ends with a space.

In addition to the restrictions on characters spelt out above, no part of an ObjectName may contain a newline character ('\n'), whether the domain, a key, or a value, whether quoted or unquoted. The newline character can be represented in a quoted value with the sequence \n.

The rules on special characters and quoting apply regardless of which constructor is used to make an ObjectName.

To avoid collisions between MBeans supplied by different vendors, a useful convention is to begin the domain name with the reverse DNS name of the organization that specifies the MBeans, followed by a period and a string whose interpretation is determined by that organization. For example, MBeans specified by Sun Microsystems Inc., DNS name sun.com, would have domains such as com.sun.MyDomain. This is essentially the same convention as for Java-language package names.

The **serialVersionUID** of this class is 1081892073854801359L.

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

| **Field Summary** | |
| --- | --- |
| static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**WILDCARD**](http://docs.google.com/javax/management/ObjectName.html#WILDCARD)            Defines the wildcard "\*:\*" ObjectName. |

| **Constructor Summary** | |
| --- | --- |
| [**ObjectName**](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Construct an object name from the given string. |
| [**ObjectName**](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.util.Hashtable))([String](http://docs.google.com/java/lang/String.html) domain, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> table)            Construct an object name with several key properties from a Hashtable. |
| [**ObjectName**](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) domain, [String](http://docs.google.com/java/lang/String.html) key, [String](http://docs.google.com/java/lang/String.html) value)            Construct an object name with exactly one key property. |

| **Method Summary** | |
| --- | --- |
| boolean | [**apply**](http://docs.google.com/javax/management/ObjectName.html#apply(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Test whether this ObjectName, which may be a pattern, matches another ObjectName. |
| int | [**compareTo**](http://docs.google.com/javax/management/ObjectName.html#compareTo(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Compares two ObjectName instances. |
| boolean | [**equals**](http://docs.google.com/javax/management/ObjectName.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) object)            Compares the current object name with another object name. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCanonicalKeyPropertyListString**](http://docs.google.com/javax/management/ObjectName.html#getCanonicalKeyPropertyListString())()            Returns a string representation of the list of key properties, in which the key properties are sorted in lexical order. |
| [String](http://docs.google.com/java/lang/String.html) | [**getCanonicalName**](http://docs.google.com/javax/management/ObjectName.html#getCanonicalName())()            Returns the canonical form of the name; that is, a string representation where the properties are sorted in lexical order. |
| [String](http://docs.google.com/java/lang/String.html) | [**getDomain**](http://docs.google.com/javax/management/ObjectName.html#getDomain())()            Returns the domain part. |
| static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getInstance**](http://docs.google.com/javax/management/ObjectName.html#getInstance(javax.management.ObjectName))([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)            Return an instance of ObjectName that can be used anywhere the given object can be used. |
| static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getInstance**](http://docs.google.com/javax/management/ObjectName.html#getInstance(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(name)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String)) can be used. |
| static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getInstance**](http://docs.google.com/javax/management/ObjectName.html#getInstance(java.lang.String,%20java.util.Hashtable))([String](http://docs.google.com/java/lang/String.html) domain, [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> table)            Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(domain, table)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.util.Hashtable)) can be used. |
| static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) | [**getInstance**](http://docs.google.com/javax/management/ObjectName.html#getInstance(java.lang.String,%20java.lang.String,%20java.lang.String))([String](http://docs.google.com/java/lang/String.html) domain, [String](http://docs.google.com/java/lang/String.html) key, [String](http://docs.google.com/java/lang/String.html) value)            Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(domain, key, value)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.lang.String,%20java.lang.String)) can be used. |
| [String](http://docs.google.com/java/lang/String.html) | [**getKeyProperty**](http://docs.google.com/javax/management/ObjectName.html#getKeyProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) property)            Obtains the value associated with a key in a key property. |
| [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> | [**getKeyPropertyList**](http://docs.google.com/javax/management/ObjectName.html#getKeyPropertyList())()            Returns the key properties as a Hashtable. |
| [String](http://docs.google.com/java/lang/String.html) | [**getKeyPropertyListString**](http://docs.google.com/javax/management/ObjectName.html#getKeyPropertyListString())()            Returns a string representation of the list of key properties specified at creation time. |
| int | [**hashCode**](http://docs.google.com/javax/management/ObjectName.html#hashCode())()            Returns a hash code for this object name. |
| boolean | [**isDomainPattern**](http://docs.google.com/javax/management/ObjectName.html#isDomainPattern())()            Checks whether the object name is a pattern on the domain part. |
| boolean | [**isPattern**](http://docs.google.com/javax/management/ObjectName.html#isPattern())()            Checks whether the object name is a pattern. |
| boolean | [**isPropertyListPattern**](http://docs.google.com/javax/management/ObjectName.html#isPropertyListPattern())()            Checks whether the object name is a pattern on the key property list. |
| boolean | [**isPropertyPattern**](http://docs.google.com/javax/management/ObjectName.html#isPropertyPattern())()            Checks whether the object name is a pattern on the key properties. |
| boolean | [**isPropertyValuePattern**](http://docs.google.com/javax/management/ObjectName.html#isPropertyValuePattern())()            Checks whether the object name is a pattern on the value part of at least one of the key properties. |
| boolean | [**isPropertyValuePattern**](http://docs.google.com/javax/management/ObjectName.html#isPropertyValuePattern(java.lang.String))([String](http://docs.google.com/java/lang/String.html) property)            Checks whether the value associated with a key in a key property is a pattern. |
| static [String](http://docs.google.com/java/lang/String.html) | [**quote**](http://docs.google.com/javax/management/ObjectName.html#quote(java.lang.String))([String](http://docs.google.com/java/lang/String.html) s)            Returns a quoted form of the given String, suitable for inclusion in an ObjectName. |
| void | [**setMBeanServer**](http://docs.google.com/javax/management/ObjectName.html#setMBeanServer(javax.management.MBeanServer))([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbs)            Sets the MBean server on which the query is to be performed. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/management/ObjectName.html#toString())()            Returns a string representation of the object name. |
| static [String](http://docs.google.com/java/lang/String.html) | [**unquote**](http://docs.google.com/javax/management/ObjectName.html#unquote(java.lang.String))([String](http://docs.google.com/java/lang/String.html) q)            Returns an unquoted form of the given String. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### WILDCARD

public static final [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **WILDCARD**

Defines the wildcard "\*:\*" ObjectName.

**Since:** 1.6

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

### ObjectName

public **ObjectName**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Construct an object name from the given string.

**Parameters:**name - A string representation of the object name. **Throws:** [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The string passed as a parameter does not have the right format. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - The name parameter is null.

### ObjectName

public **ObjectName**([String](http://docs.google.com/java/lang/String.html) domain,  
 [String](http://docs.google.com/java/lang/String.html) key,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Construct an object name with exactly one key property.

**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](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The domain, key, or value contains an illegal character, or value does not follow the rules for quoting. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - One of the parameters is null.

### ObjectName

public **ObjectName**([String](http://docs.google.com/java/lang/String.html) domain,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> table)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Construct an object name with several key properties from a Hashtable.

**Parameters:**domain - The domain part of the object name.table - A hash table containing one or more key properties. The key of each entry in the table is the key of a key property in the object name. The associated value in the table is the associated value in the object name. **Throws:** [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The domain contains an illegal character, or one of the keys or values in table contains an illegal character, or one of the values in table does not follow the rules for quoting. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - One of the parameters is null.

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

### getInstance

public static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) name)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(name)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String)) can be used. The returned object may be of a subclass of ObjectName. Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

**Parameters:**name - A string representation of the object name. **Returns:**an ObjectName corresponding to the given String. **Throws:** [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The string passed as a parameter does not have the right format. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - The name parameter is null.

### getInstance

public static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) domain,  
 [String](http://docs.google.com/java/lang/String.html) key,  
 [String](http://docs.google.com/java/lang/String.html) value)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(domain, key, value)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.lang.String,%20java.lang.String)) can be used. The returned object may be of a subclass of ObjectName. Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

**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. **Returns:**an ObjectName corresponding to the given domain, key, and value. **Throws:** [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The domain, key, or value contains an illegal character, or value does not follow the rules for quoting. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - One of the parameters is null.

### getInstance

public static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getInstance**([String](http://docs.google.com/java/lang/String.html) domain,  
 [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> table)  
 throws [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Return an instance of ObjectName that can be used anywhere an object obtained with [new ObjectName(domain, table)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String,%20java.util.Hashtable)) can be used. The returned object may be of a subclass of ObjectName. Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

**Parameters:**domain - The domain part of the object name.table - A hash table containing one or more key properties. The key of each entry in the table is the key of a key property in the object name. The associated value in the table is the associated value in the object name. **Returns:**an ObjectName corresponding to the given domain and key mappings. **Throws:** [MalformedObjectNameException](http://docs.google.com/javax/management/MalformedObjectNameException.html) - The domain contains an illegal character, or one of the keys or values in table contains an illegal character, or one of the values in table does not follow the rules for quoting. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - One of the parameters is null.

### getInstance

public static [ObjectName](http://docs.google.com/javax/management/ObjectName.html) **getInstance**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Return an instance of ObjectName that can be used anywhere the given object can be used. The returned object may be of a subclass of ObjectName. If name is of a subclass of ObjectName, it is not guaranteed that the returned object will be of the same class.

The returned value may or may not be identical to name. Calling this method twice with the same parameters may return the same object or two equal but not identical objects.

Since ObjectName is immutable, it is not usually useful to make a copy of an ObjectName. The principal use of this method is to guard against a malicious caller who might pass an instance of a subclass with surprising behavior to sensitive code. Such code can call this method to obtain an ObjectName that is known not to have surprising behavior.

**Parameters:**name - an instance of the ObjectName class or of a subclass **Returns:**an instance of ObjectName or a subclass that is known to have the same semantics. If name respects the semantics of ObjectName, then the returned object is equal (though not necessarily identical) to name. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - The name is null.

### isPattern

public boolean **isPattern**()

Checks whether the object name is a pattern.

An object name is a pattern if its domain contains a wildcard or if the object name is a property pattern.

**Returns:**True if the name is a pattern, otherwise false.

### isDomainPattern

public boolean **isDomainPattern**()

Checks whether the object name is a pattern on the domain part.

**Returns:**True if the name is a domain pattern, otherwise false.

### isPropertyPattern

public boolean **isPropertyPattern**()

Checks whether the object name is a pattern on the key properties.

An object name is a pattern on the key properties if it is a pattern on the key property list (e.g. "d:k=v,\*") or on the property values (e.g. "d:k=\*") or on both (e.g. "d:k=\*,\*").

**Returns:**True if the name is a property pattern, otherwise false.

### isPropertyListPattern

public boolean **isPropertyListPattern**()

Checks whether the object name is a pattern on the key property list.

For example, "d:k=v,\*" and "d:k=\*,\*" are key property list patterns whereas "d:k=\*" is not.

**Returns:**True if the name is a property list pattern, otherwise false.**Since:** 1.6

### isPropertyValuePattern

public boolean **isPropertyValuePattern**()

Checks whether the object name is a pattern on the value part of at least one of the key properties.

For example, "d:k=\*" and "d:k=\*,\*" are property value patterns whereas "d:k=v,\*" is not.

**Returns:**True if the name is a property value pattern, otherwise false.**Since:** 1.6

### isPropertyValuePattern

public boolean **isPropertyValuePattern**([String](http://docs.google.com/java/lang/String.html) property)  
 throws [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html),  
 [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

Checks whether the value associated with a key in a key property is a pattern.

**Parameters:**property - The property whose value is to be checked. **Returns:**True if the value associated with the given key property is a pattern, otherwise false. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If property is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - If property is not a valid key property for this ObjectName.**Since:** 1.6

### getCanonicalName

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

Returns the canonical form of the name; that is, a string representation where the properties are sorted in lexical order.

More precisely, the canonical form of the name is a String consisting of the *domain part*, a colon (:), the *canonical key property list*, and a *pattern indication*.

The *canonical key property list* is the same string as described for [getCanonicalKeyPropertyListString()](http://docs.google.com/javax/management/ObjectName.html#getCanonicalKeyPropertyListString()).

The *pattern indication* is:

* empty for an ObjectName that is not a property list pattern;
* an asterisk for an ObjectName that is a property list pattern with no keys; or
* a comma and an asterisk (,\*) for an ObjectName that is a property list pattern with at least one key.

**Returns:**The canonical form of the name.

### getDomain

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

Returns the domain part.

**Returns:**The domain.

### getKeyProperty

public [String](http://docs.google.com/java/lang/String.html) **getKeyProperty**([String](http://docs.google.com/java/lang/String.html) property)  
 throws [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Obtains the value associated with a key in a key property.

**Parameters:**property - The property whose value is to be obtained. **Returns:**The value of the property, or null if there is no such property in this ObjectName. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - If property is null.

### getKeyPropertyList

public [Hashtable](http://docs.google.com/java/util/Hashtable.html)<[String](http://docs.google.com/java/lang/String.html),[String](http://docs.google.com/java/lang/String.html)> **getKeyPropertyList**()

Returns the key properties as a Hashtable. The returned value is a Hashtable in which each key is a key in the ObjectName's key property list and each value is the associated value.

The returned value may be unmodifiable. If it is modifiable, changing it has no effect on this ObjectName.

**Returns:**The table of key properties.

### getKeyPropertyListString

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

Returns a string representation of the list of key properties specified at creation time. If this ObjectName was constructed with the constructor [ObjectName(String)](http://docs.google.com/javax/management/ObjectName.html#ObjectName(java.lang.String)), the key properties in the returned String will be in the same order as in the argument to the constructor.

**Returns:**The key property list string. This string is independent of whether the ObjectName is a pattern.

### getCanonicalKeyPropertyListString

public [String](http://docs.google.com/java/lang/String.html) **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. Lexical order is the order implied by [String.compareTo(String)](http://docs.google.com/java/lang/String.html#compareTo(java.lang.String)).

**Returns:**The canonical key property list string. This string is independent of whether the ObjectName is a pattern.

### toString

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

Returns a string representation of the object name. The format of this string is not specified, but users can expect that two ObjectNames return the same string if and only if they are equal.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a string representation of this object name.

### equals

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

Compares the current object name with another object name. Two ObjectName instances are equal if and only if their canonical forms are equal. The canonical form is the string described for [getCanonicalName()](http://docs.google.com/javax/management/ObjectName.html#getCanonicalName()).

**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:**object - The object name that the current object name is to be compared with. **Returns:**True if object is an ObjectName whose canonical form is equal to that of this ObjectName.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### hashCode

public int **hashCode**()

Returns a hash code for this object name.

**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.**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)

### quote

public static [String](http://docs.google.com/java/lang/String.html) **quote**([String](http://docs.google.com/java/lang/String.html) s)  
 throws [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Returns a quoted form of the given String, suitable for inclusion in an ObjectName. The returned value can be used as the value associated with a key in an ObjectName. The String s may contain any character. Appropriate quoting ensures that the returned value is legal in an ObjectName.

The returned value consists of a quote ('"'), a sequence of characters corresponding to the characters of s, and another quote. Characters in s appear unchanged within the returned value except:

* A quote ('"') is replaced by a backslash (\) followed by a quote.
* An asterisk ('\*') is replaced by a backslash (\) followed by an asterisk.
* A question mark ('?') is replaced by a backslash (\) followed by a question mark.
* A backslash ('\') is replaced by two backslashes.
* A newline character (the character '\n' in Java) is replaced by a backslash followed by the character '\n'.

**Parameters:**s - the String to be quoted. **Returns:**the quoted String. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if s is null.

### unquote

public static [String](http://docs.google.com/java/lang/String.html) **unquote**([String](http://docs.google.com/java/lang/String.html) q)  
 throws [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html),  
 [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Returns an unquoted form of the given String. If q is a String returned by [quote(s)](http://docs.google.com/javax/management/ObjectName.html#quote(java.lang.String)), then unquote(q).equals(s). If there is no String s for which quote(s).equals(q), then unquote(q) throws an IllegalArgumentException.

These rules imply that there is a one-to-one mapping between quoted and unquoted forms.

**Parameters:**q - the String to be unquoted. **Returns:**the unquoted String. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if q could not have been returned by the [quote(java.lang.String)](http://docs.google.com/javax/management/ObjectName.html#quote(java.lang.String)) method, for instance if it does not begin and end with a quote ("). [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if q is null.

### apply

public boolean **apply**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)  
 throws [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html)

Test whether this ObjectName, which may be a pattern, matches another ObjectName. If name is a pattern, the result is false. If this ObjectName is a pattern, the result is true if and only if name matches the pattern. If neither this ObjectName nor name is a pattern, the result is true if and only if the two ObjectNames are equal as described for the [equals(Object)](http://docs.google.com/javax/management/ObjectName.html#equals(java.lang.Object)) method.

**Specified by:**[apply](http://docs.google.com/javax/management/QueryExp.html#apply(javax.management.ObjectName)) in interface [QueryExp](http://docs.google.com/javax/management/QueryExp.html) **Parameters:**name - The name of the MBean to compare to. **Returns:**True if name matches this ObjectName. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if name is null.

### setMBeanServer

public void **setMBeanServer**([MBeanServer](http://docs.google.com/javax/management/MBeanServer.html) mbs)

**Description copied from interface:** [**QueryExp**](http://docs.google.com/javax/management/QueryExp.html#setMBeanServer(javax.management.MBeanServer)) Sets the MBean server on which the query is to be performed.

**Specified by:**[setMBeanServer](http://docs.google.com/javax/management/QueryExp.html#setMBeanServer(javax.management.MBeanServer)) in interface [QueryExp](http://docs.google.com/javax/management/QueryExp.html) **Parameters:**mbs - The MBean server on which the query is to be performed.

### compareTo

public int **compareTo**([ObjectName](http://docs.google.com/javax/management/ObjectName.html) name)

Compares two ObjectName instances. The ordering relation between ObjectNames is not completely specified but is intended to be such that a sorted list of ObjectNames will appear in an order that is convenient for a person to read.

In particular, if the two ObjectName instances have different domains then their order is the lexicographical order of the domains. The ordering of the key property list remains unspecified.

For example, the ObjectName instances below:

* Shapes:type=Square,name=3
* Colors:type=Red,name=2
* Shapes:type=Triangle,side=isosceles,name=2
* Colors:type=Red,name=1
* Shapes:type=Square,name=1
* Colors:type=Blue,name=1
* Shapes:type=Square,name=2
* JMImplementation:type=MBeanServerDelegate
* Shapes:type=Triangle,side=scalene,name=1

could be ordered as follows:

* Colors:type=Blue,name=1
* Colors:type=Red,name=1
* Colors:type=Red,name=2
* JMImplementation:type=MBeanServerDelegate
* Shapes:type=Square,name=1
* Shapes:type=Square,name=2
* Shapes:type=Square,name=3
* Shapes:type=Triangle,side=scalene,name=1
* Shapes:type=Triangle,side=isosceles,name=2

**Specified by:**[compareTo](http://docs.google.com/java/lang/Comparable.html#compareTo(T)) in interface [Comparable](http://docs.google.com/java/lang/Comparable.html)<[ObjectName](http://docs.google.com/javax/management/ObjectName.html)> **Parameters:**name - the ObjectName to be compared. **Returns:**a negative integer, zero, or a positive integer as this ObjectName is less than, equal to, or greater than the specified ObjectName.**Since:** 1.6

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/ObjectName.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/ObjectInstance.html)   [**NEXT CLASS**](http://docs.google.com/javax/management/OperationsException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/management/ObjectName.html)    [**NO FRAMES**](http://docs.google.com/ObjectName.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#lnxbz9) |

[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).