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

## **javax.naming.directory**

Interface Attribute

**All Superinterfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **All Known Implementing Classes:** [BasicAttribute](http://docs.google.com/javax/naming/directory/BasicAttribute.html)

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

This interface represents an attribute associated with a named object.

In a directory, named objects can have associated with them attributes. The Attribute interface represents an attribute associated with a named object. An attribute contains 0 or more, possibly null, values. The attribute values can be ordered or unordered (see isOrdered()). If the values are unordered, no duplicates are allowed. If the values are ordered, duplicates are allowed.

The content and representation of an attribute and its values is defined by the attribute's *schema*. The schema contains information about the attribute's syntax and other properties about the attribute. See getAttributeDefinition() and getAttributeSyntaxDefinition() for details regarding how to get schema information about an attribute if the underlying directory service supports schemas.

Equality of two attributes is determined by the implementation class. A simple implementation can use Object.equals() to determine equality of attribute values, while a more sophisticated implementation might make use of schema information to determine equality. Similarly, one implementation might provide a static storage structure which simply returns the values passed to its constructor, while another implementation might define get() and getAll(). to get the values dynamically from the directory.

Note that updates to Attribute (such as adding or removing a value) do not affect the corresponding representation of the attribute in the directory. Updates to the directory can only be effected using operations in the DirContext interface.

**Since:** 1.3 **See Also:**[BasicAttribute](http://docs.google.com/javax/naming/directory/BasicAttribute.html)

| **Field Summary** | |
| --- | --- |
| static long | [**serialVersionUID**](http://docs.google.com/javax/naming/directory/Attribute.html#serialVersionUID)            Use serialVersionUID from JNDI 1.1.1 for interoperability. |

| **Method Summary** | |
| --- | --- |
| void | [**add**](http://docs.google.com/javax/naming/directory/Attribute.html#add(int,%20java.lang.Object))(int ix, [Object](http://docs.google.com/java/lang/Object.html) attrVal)            Adds an attribute value to the ordered list of attribute values. |
| boolean | [**add**](http://docs.google.com/javax/naming/directory/Attribute.html#add(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrVal)            Adds a new value to the attribute. |
| void | [**clear**](http://docs.google.com/javax/naming/directory/Attribute.html#clear())()            Removes all values from this attribute. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/naming/directory/Attribute.html#clone())()            Makes a copy of the attribute. |
| boolean | [**contains**](http://docs.google.com/javax/naming/directory/Attribute.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrVal)            Determines whether a value is in the attribute. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**get**](http://docs.google.com/javax/naming/directory/Attribute.html#get())()            Retrieves one of this attribute's values. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**get**](http://docs.google.com/javax/naming/directory/Attribute.html#get(int))(int ix)            Retrieves the attribute value from the ordered list of attribute values. |
| [NamingEnumeration](http://docs.google.com/javax/naming/NamingEnumeration.html)<?> | [**getAll**](http://docs.google.com/javax/naming/directory/Attribute.html#getAll())()            Retrieves an enumeration of the attribute's values. |
| [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) | [**getAttributeDefinition**](http://docs.google.com/javax/naming/directory/Attribute.html#getAttributeDefinition())()            Retrieves the attribute's schema definition. |
| [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) | [**getAttributeSyntaxDefinition**](http://docs.google.com/javax/naming/directory/Attribute.html#getAttributeSyntaxDefinition())()            Retrieves the syntax definition associated with the attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**getID**](http://docs.google.com/javax/naming/directory/Attribute.html#getID())()            Retrieves the id of this attribute. |
| boolean | [**isOrdered**](http://docs.google.com/javax/naming/directory/Attribute.html#isOrdered())()            Determines whether this attribute's values are ordered. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**remove**](http://docs.google.com/javax/naming/directory/Attribute.html#remove(int))(int ix)            Removes an attribute value from the ordered list of attribute values. |
| boolean | [**remove**](http://docs.google.com/javax/naming/directory/Attribute.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrval)            Removes a specified value from the attribute. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**set**](http://docs.google.com/javax/naming/directory/Attribute.html#set(int,%20java.lang.Object))(int ix, [Object](http://docs.google.com/java/lang/Object.html) attrVal)            Sets an attribute value in the ordered list of attribute values. |
| int | [**size**](http://docs.google.com/javax/naming/directory/Attribute.html#size())()            Retrieves the number of values in this attribute. |

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

### serialVersionUID

static final long **serialVersionUID**

Use serialVersionUID from JNDI 1.1.1 for interoperability.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.naming.directory.Attribute.serialVersionUID)

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

### getAll

[NamingEnumeration](http://docs.google.com/javax/naming/NamingEnumeration.html)<?> **getAll**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves an enumeration of the attribute's values. The behaviour of this enumeration is unspecified if the attribute's values are added, changed, or removed while the enumeration is in progress. If the attribute values are ordered, the enumeration's items will be ordered.

**Returns:**A non-null enumeration of the attribute's values. Each element of the enumeration is a possibly null Object. The object's class is the class of the attribute value. The element is null if the attribute's value is null. If the attribute has zero values, an empty enumeration is returned. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception was encountered while retrieving the values.**See Also:**[isOrdered()](http://docs.google.com/javax/naming/directory/Attribute.html#isOrdered())

### get

[Object](http://docs.google.com/java/lang/Object.html) **get**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves one of this attribute's values. If the attribute has more than one value and is unordered, any one of the values is returned. If the attribute has more than one value and is ordered, the first value is returned.

**Returns:**A possibly null object representing one of the attribute's value. It is null if the attribute's value is null. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception was encountered while retrieving the value. [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - If this attribute has no values.

### size

int **size**()

Retrieves the number of values in this attribute.

**Returns:**The nonnegative number of values in this attribute.

### getID

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

Retrieves the id of this attribute.

**Returns:**The id of this attribute. It cannot be null.

### contains

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

Determines whether a value is in the attribute. Equality is determined by the implementation, which may use Object.equals() or schema information to determine equality.

**Parameters:**attrVal - The possibly null value to check. If null, check whether the attribute has an attribute value whose value is null. **Returns:**true if attrVal is one of this attribute's values; false otherwise.**See Also:**[Object.equals(java.lang.Object)](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [BasicAttribute.equals(java.lang.Object)](http://docs.google.com/javax/naming/directory/BasicAttribute.html#equals(java.lang.Object))

### add

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

Adds a new value to the attribute. If the attribute values are unordered and attrVal is already in the attribute, this method does nothing. If the attribute values are ordered, attrVal is added to the end of the list of attribute values.

Equality is determined by the implementation, which may use Object.equals() or schema information to determine equality.

**Parameters:**attrVal - The new possibly null value to add. If null, null is added as an attribute value. **Returns:**true if a value was added; false otherwise.

### remove

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

Removes a specified value from the attribute. If attrval is not in the attribute, this method does nothing. If the attribute values are ordered, the first occurrence of attrVal is removed and attribute values at indices greater than the removed value are shifted up towards the head of the list (and their indices decremented by one).

Equality is determined by the implementation, which may use Object.equals() or schema information to determine equality.

**Parameters:**attrval - The possibly null value to remove from this attribute. If null, remove the attribute value that is null. **Returns:**true if the value was removed; false otherwise.

### clear

void **clear**()

Removes all values from this attribute.

### getAttributeSyntaxDefinition

[DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) **getAttributeSyntaxDefinition**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the syntax definition associated with the attribute. An attribute's syntax definition specifies the format of the attribute's value(s). Note that this is different from the attribute value's representation as a Java object. Syntax definition refers to the directory's notion of *syntax*.

For example, even though a value might be a Java String object, its directory syntax might be "Printable String" or "Telephone Number". Or a value might be a byte array, and its directory syntax is "JPEG" or "Certificate". For example, if this attribute's syntax is "JPEG", this method would return the syntax definition for "JPEG".

The information that you can retrieve from a syntax definition is directory-dependent.

If an implementation does not support schemas, it should throw OperationNotSupportedException. If an implementation does support schemas, it should define this method to return the appropriate information.

**Returns:**The attribute's syntax definition. Null if the implementation supports schemas but this particular attribute does not have any schema information. **Throws:** [OperationNotSupportedException](http://docs.google.com/javax/naming/OperationNotSupportedException.html) - If getting the schema is not supported. [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception occurs while getting the schema.

### getAttributeDefinition

[DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) **getAttributeDefinition**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the attribute's schema definition. An attribute's schema definition contains information such as whether the attribute is multivalued or single-valued, the matching rules to use when comparing the attribute's values. The information that you can retrieve from an attribute definition is directory-dependent.

If an implementation does not support schemas, it should throw OperationNotSupportedException. If an implementation does support schemas, it should define this method to return the appropriate information.

**Returns:**This attribute's schema definition. Null if the implementation supports schemas but this particular attribute does not have any schema information. **Throws:** [OperationNotSupportedException](http://docs.google.com/javax/naming/OperationNotSupportedException.html) - If getting the schema is not supported. [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception occurs while getting the schema.

### clone

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

Makes a copy of the attribute. The copy contains the same attribute values as the original attribute: the attribute values are not themselves cloned. Changes to the copy will not affect the original and vice versa.

**Returns:**A non-null copy of the attribute.

### isOrdered

boolean **isOrdered**()

Determines whether this attribute's values are ordered. If an attribute's values are ordered, duplicate values are allowed. If an attribute's values are unordered, they are presented in any order and there are no duplicate values.

**Returns:**true if this attribute's values are ordered; false otherwise.**See Also:**[get(int)](http://docs.google.com/javax/naming/directory/Attribute.html#get(int)), [remove(int)](http://docs.google.com/javax/naming/directory/Attribute.html#remove(int)), [add(int, java.lang.Object)](http://docs.google.com/javax/naming/directory/Attribute.html#add(int,%20java.lang.Object)), [set(int, java.lang.Object)](http://docs.google.com/javax/naming/directory/Attribute.html#set(int,%20java.lang.Object))

### get

[Object](http://docs.google.com/java/lang/Object.html) **get**(int ix)  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the attribute value from the ordered list of attribute values. This method returns the value at the ix index of the list of attribute values. If the attribute values are unordered, this method returns the value that happens to be at that index.

**Parameters:**ix - The index of the value in the ordered list of attribute values. 0 <= ix < size(). **Returns:**The possibly null attribute value at index ix; null if the attribute value is null. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception was encountered while retrieving the value. [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If ix is outside the specified range.

### remove

[Object](http://docs.google.com/java/lang/Object.html) **remove**(int ix)

Removes an attribute value from the ordered list of attribute values. This method removes the value at the ix index of the list of attribute values. If the attribute values are unordered, this method removes the value that happens to be at that index. Values located at indices greater than ix are shifted up towards the front of the list (and their indices decremented by one).

**Parameters:**ix - The index of the value to remove. 0 <= ix < size(). **Returns:**The possibly null attribute value at index ix that was removed; null if the attribute value is null. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If ix is outside the specified range.

### add

void **add**(int ix,  
 [Object](http://docs.google.com/java/lang/Object.html) attrVal)

Adds an attribute value to the ordered list of attribute values. This method adds attrVal to the list of attribute values at index ix. Values located at indices at or greater than ix are shifted down towards the end of the list (and their indices incremented by one). If the attribute values are unordered and already have attrVal, IllegalStateException is thrown.

**Parameters:**ix - The index in the ordered list of attribute values to add the new value. 0 <= ix <= size().attrVal - The possibly null attribute value to add; if null, null is the value added. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If ix is outside the specified range. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - If the attribute values are unordered and attrVal is one of those values.

### set

[Object](http://docs.google.com/java/lang/Object.html) **set**(int ix,  
 [Object](http://docs.google.com/java/lang/Object.html) attrVal)

Sets an attribute value in the ordered list of attribute values. This method sets the value at the ix index of the list of attribute values to be attrVal. The old value is removed. If the attribute values are unordered, this method sets the value that happens to be at that index to attrVal, unless attrVal is already one of the values. In that case, IllegalStateException is thrown.

**Parameters:**ix - The index of the value in the ordered list of attribute values. 0 <= ix < size().attrVal - The possibly null attribute value to use. If null, 'null' replaces the old value. **Returns:**The possibly null attribute value at index ix that was replaced. Null if the attribute value was null. **Throws:** [IndexOutOfBoundsException](http://docs.google.com/java/lang/IndexOutOfBoundsException.html) - If ix is outside the specified range. [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - If attrVal already exists and the attribute values are unordered.

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

[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).