| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicAttribute.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/naming/directory/Attributes.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/directory/BasicAttributes.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/directory/BasicAttribute.html)    [**NO FRAMES**](http://docs.google.com/BasicAttribute.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#44sinio) |

## **javax.naming.directory**

Class BasicAttribute

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.naming.directory.BasicAttribute**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html)

public class **BasicAttribute**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html)

This class provides a basic implementation of the Attribute interface.

This implementation does not support the schema methods getAttributeDefinition() and getAttributeSyntaxDefinition(). They simply throw OperationNotSupportedException. Subclasses of BasicAttribute should override these methods if they support them.

The BasicAttribute class by default uses Object.equals() to determine equality of attribute values when testing for equality or when searching for values, *except* when the value is an array. For an array, each element of the array is checked using Object.equals(). Subclasses of BasicAttribute can make use of schema information when doing similar equality checks by overriding methods in which such use of schema is meaningful. Similarly, the BasicAttribute class by default returns the values passed to its constructor and/or manipulated using the add/remove methods. Subclasses of BasicAttribute can override get() and getAll() to get the values dynamically from the directory (or implement the Attribute interface directly instead of subclassing BasicAttribute).

Note that updates to BasicAttribute (such as adding or removing a value) does 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.

A BasicAttribute instance is not synchronized against concurrent multithreaded access. Multiple threads trying to access and modify a BasicAttribute should lock the object.

**Since:** 1.3 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#javax.naming.directory.BasicAttribute)

| **Field Summary** | |
| --- | --- |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**attrID**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#attrID)            Holds the attribute's id. |
| protected  boolean | [**ordered**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#ordered)            A flag for recording whether this attribute's values are ordered. |
| protected  [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> | [**values**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#values)            Holds the attribute's values. |

| **Constructor Summary** | |
| --- | --- |
| [**BasicAttribute**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#BasicAttribute(java.lang.String))([String](http://docs.google.com/java/lang/String.html) id)            Constructs a new instance of an unordered attribute with no value. |
| [**BasicAttribute**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#BasicAttribute(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) id, boolean ordered)            Constructs a new instance of a possibly ordered attribute with no value. |
| [**BasicAttribute**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#BasicAttribute(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) id, [Object](http://docs.google.com/java/lang/Object.html) value)            Constructs a new instance of an unordered attribute with a single value. |
| [**BasicAttribute**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#BasicAttribute(java.lang.String,%20java.lang.Object,%20boolean))([String](http://docs.google.com/java/lang/String.html) id, [Object](http://docs.google.com/java/lang/Object.html) value, boolean ordered)            Constructs a new instance of a possibly ordered attribute with a single value. |

| **Method Summary** | |
| --- | --- |
| void | [**add**](http://docs.google.com/javax/naming/directory/BasicAttribute.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/BasicAttribute.html#add(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrVal)            Adds a new value to this attribute. |
| void | [**clear**](http://docs.google.com/javax/naming/directory/BasicAttribute.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/BasicAttribute.html#clone())()            Creates and returns a copy of this object. |
| boolean | [**contains**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#contains(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrVal)            Determines whether a value is in this attribute. |
| boolean | [**equals**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Determines whether obj is equal to this attribute. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**get**](http://docs.google.com/javax/naming/directory/BasicAttribute.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/BasicAttribute.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/BasicAttribute.html#getAll())()            Retrieves an enumeration of this attribute's values. |
| [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) | [**getAttributeDefinition**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#getAttributeDefinition())()            Retrieves this attribute's schema definition. |
| [DirContext](http://docs.google.com/javax/naming/directory/DirContext.html) | [**getAttributeSyntaxDefinition**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#getAttributeSyntaxDefinition())()            Retrieves the syntax definition associated with this attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**getID**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#getID())()            Retrieves the id of this attribute. |
| int | [**hashCode**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#hashCode())()            Calculates the hash code of this attribute. |
| boolean | [**isOrdered**](http://docs.google.com/javax/naming/directory/BasicAttribute.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/BasicAttribute.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/BasicAttribute.html#remove(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) attrval)            Removes a specified value from this attribute. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**set**](http://docs.google.com/javax/naming/directory/BasicAttribute.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/BasicAttribute.html#size())()            Retrieves the number of values in this attribute. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/naming/directory/BasicAttribute.html#toString())()            Generates the string representation of this attribute. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [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** |
| --- |

### attrID

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

Holds the attribute's id. It is initialized by the public constructor and cannot be null unless methods in BasicAttribute that use attrID have been overridden.

### values

protected transient [Vector](http://docs.google.com/java/util/Vector.html)<[Object](http://docs.google.com/java/lang/Object.html)> **values**

Holds the attribute's values. Initialized by public constructors. Cannot be null unless methods in BasicAttribute that use values have been overridden.

### ordered

protected boolean **ordered**

A flag for recording whether this attribute's values are ordered.

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

### BasicAttribute

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

Constructs a new instance of an unordered attribute with no value.

**Parameters:**id - The attribute's id. It cannot be null.

### BasicAttribute

public **BasicAttribute**([String](http://docs.google.com/java/lang/String.html) id,  
 [Object](http://docs.google.com/java/lang/Object.html) value)

Constructs a new instance of an unordered attribute with a single value.

**Parameters:**id - The attribute's id. It cannot be null.value - The attribute's value. If null, a null value is added to the attribute.

### BasicAttribute

public **BasicAttribute**([String](http://docs.google.com/java/lang/String.html) id,  
 boolean ordered)

Constructs a new instance of a possibly ordered attribute with no value.

**Parameters:**id - The attribute's id. It cannot be null.ordered - true means the attribute's values will be ordered; false otherwise.

### BasicAttribute

public **BasicAttribute**([String](http://docs.google.com/java/lang/String.html) id,  
 [Object](http://docs.google.com/java/lang/Object.html) value,  
 boolean ordered)

Constructs a new instance of a possibly ordered attribute with a single value.

**Parameters:**id - The attribute's id. It cannot be null.value - The attribute's value. If null, a null value is added to the attribute.ordered - true means the attribute's values will be ordered; false otherwise.

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

### clone

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

**Description copied from class:** [**Object**](http://docs.google.com/java/lang/Object.html#clone()) Creates and returns a copy of this object. The precise meaning of "copy" may depend on the class of the object. The general intent is that, for any object x, the expression:

x.clone() != x

will be true, and that the expression:

x.clone().getClass() == x.getClass()

will be true, but these are not absolute requirements. While it is typically the case that:

x.clone().equals(x)

will be true, this is not an absolute requirement.

By convention, the returned object should be obtained by calling super.clone. If a class and all of its superclasses (except Object) obey this convention, it will be the case that x.clone().getClass() == x.getClass().

By convention, the object returned by this method should be independent of this object (which is being cloned). To achieve this independence, it may be necessary to modify one or more fields of the object returned by super.clone before returning it. Typically, this means copying any mutable objects that comprise the internal "deep structure" of the object being cloned and replacing the references to these objects with references to the copies. If a class contains only primitive fields or references to immutable objects, then it is usually the case that no fields in the object returned by super.clone need to be modified.

The method clone for class Object performs a specific cloning operation. First, if the class of this object does not implement the interface Cloneable, then a CloneNotSupportedException is thrown. Note that all arrays are considered to implement the interface Cloneable. Otherwise, this method creates a new instance of the class of this object and initializes all its fields with exactly the contents of the corresponding fields of this object, as if by assignment; the contents of the fields are not themselves cloned. Thus, this method performs a "shallow copy" of this object, not a "deep copy" operation.

The class Object does not itself implement the interface Cloneable, so calling the clone method on an object whose class is Object will result in throwing an exception at run time.

**Specified by:**[clone](http://docs.google.com/javax/naming/directory/Attribute.html#clone()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html)**Overrides:**[clone](http://docs.google.com/java/lang/Object.html#clone()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a clone of this instance.**See Also:**[Cloneable](http://docs.google.com/java/lang/Cloneable.html)

### equals

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

Determines whether obj is equal to this attribute. Two attributes are equal if their attribute-ids, syntaxes and values are equal. If the attribute values are unordered, the order that the values were added are irrelevant. If the attribute values are ordered, then the order the values must match. If obj is null or not an Attribute, false is returned.

By default Object.equals() is used when comparing the attribute id and its values except when a value is an array. For an array, each element of the array is checked using Object.equals(). A subclass may override this to make use of schema syntax information and matching rules, which define what it means for two attributes to be equal. How and whether a subclass makes use of the schema information is determined by the subclass. If a subclass overrides equals(), it should also override hashCode() such that two attributes that are equal have the same hash code.

**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 possibly null object to check. **Returns:**true if obj is equal to this attribute; false otherwise.**See Also:**[hashCode()](http://docs.google.com/javax/naming/directory/BasicAttribute.html#hashCode()), [contains(java.lang.Object)](http://docs.google.com/javax/naming/directory/BasicAttribute.html#contains(java.lang.Object))

### hashCode

public int **hashCode**()

Calculates the hash code of this attribute.

The hash code is computed by adding the hash code of the attribute's id and that of all of its values except for values that are arrays. For an array, the hash code of each element of the array is summed. If a subclass overrides hashCode(), it should override equals() as well so that two attributes that are equal have the same hash code.

**Overrides:**[hashCode](http://docs.google.com/java/lang/Object.html#hashCode()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**an int representing the hash code of this attribute.**See Also:**[equals(java.lang.Object)](http://docs.google.com/javax/naming/directory/BasicAttribute.html#equals(java.lang.Object))

### toString

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

Generates the string representation of this attribute. The string consists of the attribute's id and its values. This string is meant for debugging and not meant to be interpreted programmatically.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**The non-null string representation of this attribute.

### getAll

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

Retrieves an enumeration of this attribute's values.

By default, the values returned are those passed to the constructor and/or manipulated using the add/replace/remove methods. A subclass may override this to retrieve the values dynamically from the directory.

**Specified by:**[getAll](http://docs.google.com/javax/naming/directory/Attribute.html#getAll()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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:**[Attribute.isOrdered()](http://docs.google.com/javax/naming/directory/Attribute.html#isOrdered())

### get

public [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.

By default, the value returned is one of those passed to the constructor and/or manipulated using the add/replace/remove methods. A subclass may override this to retrieve the value dynamically from the directory.

**Specified by:**[get](http://docs.google.com/javax/naming/directory/Attribute.html#get()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

### size

public int **size**()

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#size()) Retrieves the number of values in this attribute.

**Specified by:**[size](http://docs.google.com/javax/naming/directory/Attribute.html#size()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **Returns:**The nonnegative number of values in this attribute.

### getID

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

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#getID()) Retrieves the id of this attribute.

**Specified by:**[getID](http://docs.google.com/javax/naming/directory/Attribute.html#getID()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **Returns:**The id of this attribute. It cannot be null.

### contains

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

Determines whether a value is in this attribute.

By default, Object.equals() is used when comparing attrVal with this attribute's values except when attrVal is an array. For an array, each element of the array is checked using Object.equals(). A subclass may use schema information to determine equality.

**Specified by:**[contains](http://docs.google.com/javax/naming/directory/Attribute.html#contains(java.lang.Object)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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)), [equals(java.lang.Object)](http://docs.google.com/javax/naming/directory/BasicAttribute.html#equals(java.lang.Object))

### add

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

Adds a new value to this attribute.

By default, Object.equals() is used when comparing attrVal with this attribute's values except when attrVal is an array. For an array, each element of the array is checked using Object.equals(). A subclass may use schema information to determine equality.

**Specified by:**[add](http://docs.google.com/javax/naming/directory/Attribute.html#add(java.lang.Object)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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

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

Removes a specified value from this attribute.

By default, Object.equals() is used when comparing attrVal with this attribute's values except when attrVal is an array. For an array, each element of the array is checked using Object.equals(). A subclass may use schema information to determine equality.

**Specified by:**[remove](http://docs.google.com/javax/naming/directory/Attribute.html#remove(java.lang.Object)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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

public void **clear**()

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#clear()) Removes all values from this attribute.

**Specified by:**[clear](http://docs.google.com/javax/naming/directory/Attribute.html#clear()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html)

### isOrdered

public boolean **isOrdered**()

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#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.

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

### get

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

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#get(int)) 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.

**Specified by:**[get](http://docs.google.com/javax/naming/directory/Attribute.html#get(int)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

### remove

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

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#remove(int)) 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).

**Specified by:**[remove](http://docs.google.com/javax/naming/directory/Attribute.html#remove(int)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

### add

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

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#add(int,%20java.lang.Object)) 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.

**Specified by:**[add](http://docs.google.com/javax/naming/directory/Attribute.html#add(int,%20java.lang.Object)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

### set

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

**Description copied from interface:** [**Attribute**](http://docs.google.com/javax/naming/directory/Attribute.html#set(int,%20java.lang.Object)) 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.

**Specified by:**[set](http://docs.google.com/javax/naming/directory/Attribute.html#set(int,%20java.lang.Object)) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

### getAttributeSyntaxDefinition

public [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 this attribute.

This method by default throws OperationNotSupportedException. A subclass should override this method if it supports schema.

**Specified by:**[getAttributeSyntaxDefinition](http://docs.google.com/javax/naming/directory/Attribute.html#getAttributeSyntaxDefinition()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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

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

Retrieves this attribute's schema definition.

This method by default throws OperationNotSupportedException. A subclass should override this method if it supports schema.

**Specified by:**[getAttributeDefinition](http://docs.google.com/javax/naming/directory/Attribute.html#getAttributeDefinition()) in interface [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **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.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicAttribute.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/naming/directory/Attributes.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/directory/BasicAttributes.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/directory/BasicAttribute.html)    [**NO FRAMES**](http://docs.google.com/BasicAttribute.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#3rdcrjn) | [METHOD](#44sinio) |

[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).