| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicAttributes.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/BasicAttribute.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/directory/DirContext.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/directory/BasicAttributes.html)    [**NO FRAMES**](http://docs.google.com/BasicAttributes.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

## **javax.naming.directory**

Class BasicAttributes

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

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

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

This class provides a basic implementation of the Attributes interface.

BasicAttributes is either case-sensitive or case-insensitive (case-ignore). This property is determined at the time the BasicAttributes constructor is called. In a case-insensitive BasicAttributes, the case of its attribute identifiers is ignored when searching for an attribute, or adding attributes. In a case-sensitive BasicAttributes, the case is significant.

When the BasicAttributes class needs to create an Attribute, it uses BasicAttribute. There is no other dependency on BasicAttribute.

Note that updates to BasicAttributes (such as adding or removing an attribute) does not affect the corresponding representation in the directory. Updates to the directory can only be effected using operations in the DirContext interface.

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

**Since:** 1.3 **See Also:**[DirContext.getAttributes(javax.naming.Name)](http://docs.google.com/javax/naming/directory/DirContext.html#getAttributes(javax.naming.Name)), [DirContext.modifyAttributes(javax.naming.Name, int, javax.naming.directory.Attributes)](http://docs.google.com/javax/naming/directory/DirContext.html#modifyAttributes(javax.naming.Name,%20int,%20javax.naming.directory.Attributes)), [DirContext.bind(javax.naming.Name, java.lang.Object, javax.naming.directory.Attributes)](http://docs.google.com/javax/naming/directory/DirContext.html#bind(javax.naming.Name,%20java.lang.Object,%20javax.naming.directory.Attributes)), [DirContext.rebind(javax.naming.Name, java.lang.Object, javax.naming.directory.Attributes)](http://docs.google.com/javax/naming/directory/DirContext.html#rebind(javax.naming.Name,%20java.lang.Object,%20javax.naming.directory.Attributes)), [DirContext.createSubcontext(javax.naming.Name, javax.naming.directory.Attributes)](http://docs.google.com/javax/naming/directory/DirContext.html#createSubcontext(javax.naming.Name,%20javax.naming.directory.Attributes)), [DirContext.search(javax.naming.Name, javax.naming.directory.Attributes, java.lang.String[])](http://docs.google.com/javax/naming/directory/DirContext.html#search(javax.naming.Name,%20javax.naming.directory.Attributes,%20java.lang.String%5B%5D)), [Serialized Form](http://docs.google.com/serialized-form.html#javax.naming.directory.BasicAttributes)

| **Constructor Summary** | |
| --- | --- |
| [**BasicAttributes**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#BasicAttributes())()            Constructs a new instance of Attributes. |
| [**BasicAttributes**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#BasicAttributes(boolean))(boolean ignoreCase)            Constructs a new instance of Attributes. |
| [**BasicAttributes**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#BasicAttributes(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) attrID, [Object](http://docs.google.com/java/lang/Object.html) val)            Constructs a new instance of Attributes with one attribute. |
| [**BasicAttributes**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#BasicAttributes(java.lang.String,%20java.lang.Object,%20boolean))([String](http://docs.google.com/java/lang/String.html) attrID, [Object](http://docs.google.com/java/lang/Object.html) val, boolean ignoreCase)            Constructs a new instance of Attributes with one attribute. |

| **Method Summary** | |
| --- | --- |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#clone())()            Creates and returns a copy of this object. |
| boolean | [**equals**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Determines whether this BasicAttributes is equal to another Attributes Two Attributes are equal if they are both instances of Attributes, treat the case of attribute IDs the same way, and contain the same attributes. |
| [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) | [**get**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#get(java.lang.String))([String](http://docs.google.com/java/lang/String.html) attrID)            Retrieves the attribute with the given attribute id from the attribute set. |
| [NamingEnumeration](http://docs.google.com/javax/naming/NamingEnumeration.html)<[Attribute](http://docs.google.com/javax/naming/directory/Attribute.html)> | [**getAll**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#getAll())()            Retrieves an enumeration of the attributes in the attribute set. |
| [NamingEnumeration](http://docs.google.com/javax/naming/NamingEnumeration.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getIDs**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#getIDs())()            Retrieves an enumeration of the ids of the attributes in the attribute set. |
| int | [**hashCode**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#hashCode())()            Calculates the hash code of this BasicAttributes. |
| boolean | [**isCaseIgnored**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#isCaseIgnored())()            Determines whether the attribute set ignores the case of attribute identifiers when retrieving or adding attributes. |
| [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) | [**put**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#put(javax.naming.directory.Attribute))([Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) attr)            Adds a new attribute to the attribute set. |
| [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) | [**put**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#put(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) attrID, [Object](http://docs.google.com/java/lang/Object.html) val)            Adds a new attribute to the attribute set. |
| [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) | [**remove**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#remove(java.lang.String))([String](http://docs.google.com/java/lang/String.html) attrID)            Removes the attribute with the attribute id 'attrID' from the attribute set. |
| int | [**size**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#size())()            Retrieves the number of attributes in the attribute set. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/javax/naming/directory/BasicAttributes.html#toString())()            Generates the string representation of this attribute set. |

| **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)) |

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

### BasicAttributes

public **BasicAttributes**()

Constructs a new instance of Attributes. The character case of attribute identifiers is significant when subsequently retrieving or adding attributes.

### BasicAttributes

public **BasicAttributes**(boolean ignoreCase)

Constructs a new instance of Attributes. If ignoreCase is true, the character case of attribute identifiers is ignored; otherwise the case is significant.

**Parameters:**ignoreCase - true means this attribute set will ignore the case of its attribute identifiers when retrieving or adding attributes; false means case is respected.

### BasicAttributes

public **BasicAttributes**([String](http://docs.google.com/java/lang/String.html) attrID,  
 [Object](http://docs.google.com/java/lang/Object.html) val)

Constructs a new instance of Attributes with one attribute. The attribute specified by attrID and val are added to the newly created attribute. The character case of attribute identifiers is significant when subsequently retrieving or adding attributes.

**Parameters:**attrID - non-null The id of the attribute to add.val - The value of the attribute to add. If null, a null value is added to the attribute.

### BasicAttributes

public **BasicAttributes**([String](http://docs.google.com/java/lang/String.html) attrID,  
 [Object](http://docs.google.com/java/lang/Object.html) val,  
 boolean ignoreCase)

Constructs a new instance of Attributes with one attribute. The attribute specified by attrID and val are added to the newly created attribute. If ignoreCase is true, the character case of attribute identifiers is ignored; otherwise the case is significant.

**Parameters:**attrID - non-null The id of the attribute to add. If this attribute set ignores the character case of its attribute ids, the case of attrID is ignored.val - The value of the attribute to add. If null, a null value is added to the attribute.ignoreCase - true means this attribute set will ignore the case of its attribute identifiers when retrieving or adding attributes; false means case is respected.

| **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/Attributes.html#clone()) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.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)

### isCaseIgnored

public boolean **isCaseIgnored**()

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#isCaseIgnored()) Determines whether the attribute set ignores the case of attribute identifiers when retrieving or adding attributes.

**Specified by:**[isCaseIgnored](http://docs.google.com/javax/naming/directory/Attributes.html#isCaseIgnored()) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Returns:**true if case is ignored; false otherwise.

### size

public int **size**()

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

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

### get

public [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **get**([String](http://docs.google.com/java/lang/String.html) attrID)

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#get(java.lang.String)) Retrieves the attribute with the given attribute id from the attribute set.

**Specified by:**[get](http://docs.google.com/javax/naming/directory/Attributes.html#get(java.lang.String)) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Parameters:**attrID - The non-null id of the attribute to retrieve. If this attribute set ignores the character case of its attribute ids, the case of attrID is ignored. **Returns:**The attribute identified by attrID; null if not found.**See Also:**[Attributes.put(java.lang.String, java.lang.Object)](http://docs.google.com/javax/naming/directory/Attributes.html#put(java.lang.String,%20java.lang.Object)), [Attributes.remove(java.lang.String)](http://docs.google.com/javax/naming/directory/Attributes.html#remove(java.lang.String))

### getAll

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

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#getAll()) Retrieves an enumeration of the attributes in the attribute set. The effects of updates to this attribute set on this enumeration are undefined.

**Specified by:**[getAll](http://docs.google.com/javax/naming/directory/Attributes.html#getAll()) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Returns:**A non-null enumeration of the attributes in this attribute set. Each element of the enumeration is of class Attribute. If attribute set has zero attributes, an empty enumeration is returned.

### getIDs

public [NamingEnumeration](http://docs.google.com/javax/naming/NamingEnumeration.html)<[String](http://docs.google.com/java/lang/String.html)> **getIDs**()

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#getIDs()) Retrieves an enumeration of the ids of the attributes in the attribute set. The effects of updates to this attribute set on this enumeration are undefined.

**Specified by:**[getIDs](http://docs.google.com/javax/naming/directory/Attributes.html#getIDs()) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Returns:**A non-null enumeration of the attributes' ids in this attribute set. Each element of the enumeration is of class String. If attribute set has zero attributes, an empty enumeration is returned.

### put

public [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **put**([String](http://docs.google.com/java/lang/String.html) attrID,  
 [Object](http://docs.google.com/java/lang/Object.html) val)

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#put(java.lang.String,%20java.lang.Object)) Adds a new attribute to the attribute set.

**Specified by:**[put](http://docs.google.com/javax/naming/directory/Attributes.html#put(java.lang.String,%20java.lang.Object)) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Parameters:**attrID - non-null The id of the attribute to add. If the attribute set ignores the character case of its attribute ids, the case of attrID is ignored.val - The possibly null value of the attribute to add. If null, the attribute does not have any values. **Returns:**The Attribute with attrID that was previous in this attribute set; null if no such attribute existed.**See Also:**[Attributes.remove(java.lang.String)](http://docs.google.com/javax/naming/directory/Attributes.html#remove(java.lang.String))

### put

public [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **put**([Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) attr)

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#put(javax.naming.directory.Attribute)) Adds a new attribute to the attribute set.

**Specified by:**[put](http://docs.google.com/javax/naming/directory/Attributes.html#put(javax.naming.directory.Attribute)) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Parameters:**attr - The non-null attribute to add. If the attribute set ignores the character case of its attribute ids, the case of attr's identifier is ignored. **Returns:**The Attribute with the same ID as attr that was previous in this attribute set; null if no such attribute existed.**See Also:**[Attributes.remove(java.lang.String)](http://docs.google.com/javax/naming/directory/Attributes.html#remove(java.lang.String))

### remove

public [Attribute](http://docs.google.com/javax/naming/directory/Attribute.html) **remove**([String](http://docs.google.com/java/lang/String.html) attrID)

**Description copied from interface:** [**Attributes**](http://docs.google.com/javax/naming/directory/Attributes.html#remove(java.lang.String)) Removes the attribute with the attribute id 'attrID' from the attribute set. If the attribute does not exist, ignore.

**Specified by:**[remove](http://docs.google.com/javax/naming/directory/Attributes.html#remove(java.lang.String)) in interface [Attributes](http://docs.google.com/javax/naming/directory/Attributes.html) **Parameters:**attrID - The non-null id of the attribute to remove. If the attribute set ignores the character case of its attribute ids, the case of attrID is ignored. **Returns:**The Attribute with the same ID as attrID that was previous in the attribute set; null if no such attribute existed.

### toString

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

Generates the string representation of this attribute set. The string consists of each attribute identifier and the contents of each attribute. The contents of this string is useful for debugging and is 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:**A non-null string listing the contents of this attribute set.

### equals

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

Determines whether this BasicAttributes is equal to another Attributes Two Attributes are equal if they are both instances of Attributes, treat the case of attribute IDs the same way, and contain the same attributes. Each Attribute in this BasicAttributes is checked for equality using Object.equals(), which may have be overridden by implementations of Attribute). If a subclass overrides equals(), it should override hashCode() as well so that two Attributes instances 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 compare against. **Returns:**true If obj is equal to this BasicAttributes.**See Also:**[hashCode()](http://docs.google.com/javax/naming/directory/BasicAttributes.html#hashCode())

### hashCode

public int **hashCode**()

Calculates the hash code of this BasicAttributes.

The hash code is computed by adding the hash code of the attributes of this object. If this BasicAttributes ignores case of its attribute IDs, one is added to the hash code. If a subclass overrides hashCode(), it should override equals() as well so that two Attributes instances 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 BasicAttributes instance.**See Also:**[equals(java.lang.Object)](http://docs.google.com/javax/naming/directory/BasicAttributes.html#equals(java.lang.Object))

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/BasicAttributes.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/BasicAttribute.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/directory/DirContext.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/directory/BasicAttributes.html)    [**NO FRAMES**](http://docs.google.com/BasicAttributes.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#3rdcrjn) |

[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).