| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Name.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/MalformedLinkException.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/NameAlreadyBoundException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/Name.html)    [**NO FRAMES**](http://docs.google.com/Name.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**

Interface Name

**All Superinterfaces:** [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Object](http://docs.google.com/java/lang/Object.html)>, [Serializable](http://docs.google.com/java/io/Serializable.html) **All Known Implementing Classes:** [CompositeName](http://docs.google.com/javax/naming/CompositeName.html), [CompoundName](http://docs.google.com/javax/naming/CompoundName.html), [LdapName](http://docs.google.com/javax/naming/ldap/LdapName.html)

public interface **Name**extends [Cloneable](http://docs.google.com/java/lang/Cloneable.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Object](http://docs.google.com/java/lang/Object.html)>

The Name interface represents a generic name -- an ordered sequence of components. It can be a composite name (names that span multiple namespaces), or a compound name (names that are used within individual hierarchical naming systems).

There can be different implementations of Name; for example, composite names, URLs, or namespace-specific compound names.

The components of a name are numbered. The indexes of a name with N components range from 0 up to, but not including, N. This range may be written as [0,N). The most significant component is at index 0. An empty name has no components.

None of the methods in this interface accept null as a valid value for a parameter that is a name or a name component. Likewise, methods that return a name or name component never return null.

An instance of a Name may not be synchronized against concurrent multithreaded access if that access is not read-only.

**Since:** 1.3

| **Field Summary** | |
| --- | --- |
| static long | [**serialVersionUID**](http://docs.google.com/javax/naming/Name.html#serialVersionUID)            The class fingerprint that is set to indicate serialization compatibility with a previous version of the class. |

| **Method Summary** | |
| --- | --- |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**add**](http://docs.google.com/javax/naming/Name.html#add(int,%20java.lang.String))(int posn, [String](http://docs.google.com/java/lang/String.html) comp)            Adds a single component at a specified position within this name. |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**add**](http://docs.google.com/javax/naming/Name.html#add(java.lang.String))([String](http://docs.google.com/java/lang/String.html) comp)            Adds a single component to the end of this name. |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**addAll**](http://docs.google.com/javax/naming/Name.html#addAll(int,%20javax.naming.Name))(int posn, [Name](http://docs.google.com/javax/naming/Name.html) n)            Adds the components of a name -- in order -- at a specified position within this name. |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**addAll**](http://docs.google.com/javax/naming/Name.html#addAll(javax.naming.Name))([Name](http://docs.google.com/javax/naming/Name.html) suffix)            Adds the components of a name -- in order -- to the end of this name. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**clone**](http://docs.google.com/javax/naming/Name.html#clone())()            Generates a new copy of this name. |
| int | [**compareTo**](http://docs.google.com/javax/naming/Name.html#compareTo(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Compares this name with another name for order. |
| boolean | [**endsWith**](http://docs.google.com/javax/naming/Name.html#endsWith(javax.naming.Name))([Name](http://docs.google.com/javax/naming/Name.html) n)            Determines whether this name ends with a specified suffix. |
| [String](http://docs.google.com/java/lang/String.html) | [**get**](http://docs.google.com/javax/naming/Name.html#get(int))(int posn)            Retrieves a component of this name. |
| [Enumeration](http://docs.google.com/java/util/Enumeration.html)<[String](http://docs.google.com/java/lang/String.html)> | [**getAll**](http://docs.google.com/javax/naming/Name.html#getAll())()            Retrieves the components of this name as an enumeration of strings. |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**getPrefix**](http://docs.google.com/javax/naming/Name.html#getPrefix(int))(int posn)            Creates a name whose components consist of a prefix of the components of this name. |
| [Name](http://docs.google.com/javax/naming/Name.html) | [**getSuffix**](http://docs.google.com/javax/naming/Name.html#getSuffix(int))(int posn)            Creates a name whose components consist of a suffix of the components in this name. |
| boolean | [**isEmpty**](http://docs.google.com/javax/naming/Name.html#isEmpty())()            Determines whether this name is empty. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**remove**](http://docs.google.com/javax/naming/Name.html#remove(int))(int posn)            Removes a component from this name. |
| int | [**size**](http://docs.google.com/javax/naming/Name.html#size())()            Returns the number of components in this name. |
| boolean | [**startsWith**](http://docs.google.com/javax/naming/Name.html#startsWith(javax.naming.Name))([Name](http://docs.google.com/javax/naming/Name.html) n)            Determines whether this name starts with a specified prefix. |

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

### serialVersionUID

static final long **serialVersionUID**

The class fingerprint that is set to indicate serialization compatibility with a previous version of the class.

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

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

### clone

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

Generates a new copy of this name. Subsequent changes to the components of this name will not affect the new copy, and vice versa.

**Returns:**a copy of this name**See Also:**[Object.clone()](http://docs.google.com/java/lang/Object.html#clone())

### compareTo

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

Compares this name with another name for order. Returns a negative integer, zero, or a positive integer as this name is less than, equal to, or greater than the given name.

As with Object.equals(), the notion of ordering for names depends on the class that implements this interface. For example, the ordering may be based on lexicographical ordering of the name components. Specific attributes of the name, such as how it treats case, may affect the ordering. In general, two names of different classes may not be compared.

**Specified by:**[compareTo](http://docs.google.com/java/lang/Comparable.html#compareTo(T)) in interface [Comparable](http://docs.google.com/java/lang/Comparable.html)<[Object](http://docs.google.com/java/lang/Object.html)> **Parameters:**obj - the non-null object to compare against. **Returns:**a negative integer, zero, or a positive integer as this name is less than, equal to, or greater than the given name **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if obj is not a Name of a type that may be compared with this name**See Also:**[Comparable.compareTo(Object)](http://docs.google.com/java/lang/Comparable.html#compareTo(T))

### size

int **size**()

Returns the number of components in this name.

**Returns:**the number of components in this name

### isEmpty

boolean **isEmpty**()

Determines whether this name is empty. An empty name is one with zero components.

**Returns:**true if this name is empty, false otherwise

### getAll

[Enumeration](http://docs.google.com/java/util/Enumeration.html)<[String](http://docs.google.com/java/lang/String.html)> **getAll**()

Retrieves the components of this name as an enumeration of strings. The effect on the enumeration of updates to this name is undefined. If the name has zero components, an empty (non-null) enumeration is returned.

**Returns:**an enumeration of the components of this name, each a string

### get

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

Retrieves a component of this name.

**Parameters:**posn - the 0-based index of the component to retrieve. Must be in the range [0,size()). **Returns:**the component at index posn **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range

### getPrefix

[Name](http://docs.google.com/javax/naming/Name.html) **getPrefix**(int posn)

Creates a name whose components consist of a prefix of the components of this name. Subsequent changes to this name will not affect the name that is returned and vice versa.

**Parameters:**posn - the 0-based index of the component at which to stop. Must be in the range [0,size()]. **Returns:**a name consisting of the components at indexes in the range [0,posn). **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range

### getSuffix

[Name](http://docs.google.com/javax/naming/Name.html) **getSuffix**(int posn)

Creates a name whose components consist of a suffix of the components in this name. Subsequent changes to this name do not affect the name that is returned and vice versa.

**Parameters:**posn - the 0-based index of the component at which to start. Must be in the range [0,size()]. **Returns:**a name consisting of the components at indexes in the range [posn,size()). If posn is equal to size(), an empty name is returned. **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range

### startsWith

boolean **startsWith**([Name](http://docs.google.com/javax/naming/Name.html) n)

Determines whether this name starts with a specified prefix. A name n is a prefix if it is equal to getPrefix(n.size()).

**Parameters:**n - the name to check **Returns:**true if n is a prefix of this name, false otherwise

### endsWith

boolean **endsWith**([Name](http://docs.google.com/javax/naming/Name.html) n)

Determines whether this name ends with a specified suffix. A name n is a suffix if it is equal to getSuffix(size()-n.size()).

**Parameters:**n - the name to check **Returns:**true if n is a suffix of this name, false otherwise

### addAll

[Name](http://docs.google.com/javax/naming/Name.html) **addAll**([Name](http://docs.google.com/javax/naming/Name.html) suffix)  
 throws [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html)

Adds the components of a name -- in order -- to the end of this name.

**Parameters:**suffix - the components to add **Returns:**the updated name (not a new one) **Throws:** [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html) - if suffix is not a valid name, or if the addition of the components would violate the syntax rules of this name

### addAll

[Name](http://docs.google.com/javax/naming/Name.html) **addAll**(int posn,  
 [Name](http://docs.google.com/javax/naming/Name.html) n)  
 throws [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html)

Adds the components of a name -- in order -- at a specified position within this name. Components of this name at or after the index of the first new component are shifted up (away from 0) to accommodate the new components.

**Parameters:**n - the components to addposn - the index in this name at which to add the new components. Must be in the range [0,size()]. **Returns:**the updated name (not a new one) **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html) - if n is not a valid name, or if the addition of the components would violate the syntax rules of this name

### add

[Name](http://docs.google.com/javax/naming/Name.html) **add**([String](http://docs.google.com/java/lang/String.html) comp)  
 throws [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html)

Adds a single component to the end of this name.

**Parameters:**comp - the component to add **Returns:**the updated name (not a new one) **Throws:** [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html) - if adding comp would violate the syntax rules of this name

### add

[Name](http://docs.google.com/javax/naming/Name.html) **add**(int posn,  
 [String](http://docs.google.com/java/lang/String.html) comp)  
 throws [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html)

Adds a single component at a specified position within this name. Components of this name at or after the index of the new component are shifted up by one (away from index 0) to accommodate the new component.

**Parameters:**comp - the component to addposn - the index at which to add the new component. Must be in the range [0,size()]. **Returns:**the updated name (not a new one) **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html) - if adding comp would violate the syntax rules of this name

### remove

[Object](http://docs.google.com/java/lang/Object.html) **remove**(int posn)  
 throws [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html)

Removes a component from this name. The component of this name at the specified position is removed. Components with indexes greater than this position are shifted down (toward index 0) by one.

**Parameters:**posn - the index of the component to remove. Must be in the range [0,size()). **Returns:**the component removed (a String) **Throws:** [ArrayIndexOutOfBoundsException](http://docs.google.com/java/lang/ArrayIndexOutOfBoundsException.html) - if posn is outside the specified range [InvalidNameException](http://docs.google.com/javax/naming/InvalidNameException.html) - if deleting the component would violate the syntax rules of the name

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Name.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/MalformedLinkException.html)   [**NEXT CLASS**](http://docs.google.com/javax/naming/NameAlreadyBoundException.html) | [**FRAMES**](http://docs.google.com/index.html?javax/naming/Name.html)    [**NO FRAMES**](http://docs.google.com/Name.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).