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

## **javax.naming**

Interface NamingEnumeration<T>

**All Superinterfaces:** [Enumeration](http://docs.google.com/java/util/Enumeration.html)<T>

public interface **NamingEnumeration<T>**extends [Enumeration](http://docs.google.com/java/util/Enumeration.html)<T>

This interface is for enumerating lists returned by methods in the javax.naming and javax.naming.directory packages. It extends Enumeration to allow as exceptions to be thrown during the enumeration.

When a method such as list(), listBindings(), or search() returns a NamingEnumeration, any exceptions encountered are reserved until all results have been returned. At the end of the enumeration, the exception is thrown (by hasMore());

For example, if the list() is returning only a partial answer, the corresponding exception would be PartialResultException. list() would first return a NamingEnumeration. When the last of the results has been returned by the NamingEnumeration's next(), invoking hasMore() would result in PartialResultException being thrown.

In another example, if a search() method was invoked with a specified size limit of 'n'. If the answer consists of more than 'n' results, search() would first return a NamingEnumeration. When the n'th result has been returned by invoking next() on the NamingEnumeration, a SizeLimitExceedException would then thrown when hasMore() is invoked.

Note that if the program uses hasMoreElements() and nextElement() instead to iterate through the NamingEnumeration, because these methods cannot throw exceptions, no exception will be thrown. Instead, in the previous example, after the n'th result has been returned by nextElement(), invoking hasMoreElements() would return false.

Note also that NoSuchElementException is thrown if the program invokes next() or nextElement() when there are no elements left in the enumeration. The program can always avoid this exception by using hasMore() and hasMoreElements() to check whether the end of the enumeration has been reached.

If an exception is thrown during an enumeration, the enumeration becomes invalid. Subsequent invocation of any method on that enumeration will yield undefined results.

**Since:** 1.3 **See Also:**[Context.list(javax.naming.Name)](http://docs.google.com/javax/naming/Context.html#list(javax.naming.Name)), [Context.listBindings(javax.naming.Name)](http://docs.google.com/javax/naming/Context.html#listBindings(javax.naming.Name)), [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)), [Attributes.getAll()](http://docs.google.com/javax/naming/directory/Attributes.html#getAll()), [Attributes.getIDs()](http://docs.google.com/javax/naming/directory/Attributes.html#getIDs()), [Attribute.getAll()](http://docs.google.com/javax/naming/directory/Attribute.html#getAll())

| **Method Summary** | |
| --- | --- |
| void | [**close**](http://docs.google.com/javax/naming/NamingEnumeration.html#close())()            Closes this enumeration. |
| boolean | [**hasMore**](http://docs.google.com/javax/naming/NamingEnumeration.html#hasMore())()            Determines whether there are any more elements in the enumeration. |
| [T](http://docs.google.com/javax/naming/NamingEnumeration.html) | [**next**](http://docs.google.com/javax/naming/NamingEnumeration.html#next())()            Retrieves the next element in the enumeration. |

| **Methods inherited from interface java.util.**[**Enumeration**](http://docs.google.com/java/util/Enumeration.html) |
| --- |
| [hasMoreElements](http://docs.google.com/java/util/Enumeration.html#hasMoreElements()), [nextElement](http://docs.google.com/java/util/Enumeration.html#nextElement()) |

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

### next

[T](http://docs.google.com/javax/naming/NamingEnumeration.html) **next**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Retrieves the next element in the enumeration. This method allows naming exceptions encountered while retrieving the next element to be caught and handled by the application.

Note that next() can also throw the runtime exception NoSuchElementException to indicate that the caller is attempting to enumerate beyond the end of the enumeration. This is different from a NamingException, which indicates that there was a problem in obtaining the next element, for example, due to a referral or server unavailability, etc.

**Returns:**The possibly null element in the enumeration. null is only valid for enumerations that can return null (e.g. Attribute.getAll() returns an enumeration of attribute values, and an attribute value can be null). **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception is encountered while attempting to retrieve the next element. See NamingException and its subclasses for the possible naming exceptions. [NoSuchElementException](http://docs.google.com/java/util/NoSuchElementException.html) - If attempting to get the next element when none is available.**See Also:**[Enumeration.nextElement()](http://docs.google.com/java/util/Enumeration.html#nextElement())

### hasMore

boolean **hasMore**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Determines whether there are any more elements in the enumeration. This method allows naming exceptions encountered while determining whether there are more elements to be caught and handled by the application.

**Returns:**true if there is more in the enumeration ; false otherwise. **Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception is encountered while attempting to determine whether there is another element in the enumeration. See NamingException and its subclasses for the possible naming exceptions.**See Also:**[Enumeration.hasMoreElements()](http://docs.google.com/java/util/Enumeration.html#hasMoreElements())

### close

void **close**()  
 throws [NamingException](http://docs.google.com/javax/naming/NamingException.html)

Closes this enumeration. After this method has been invoked on this enumeration, the enumeration becomes invalid and subsequent invocation of any of its methods will yield undefined results. This method is intended for aborting an enumeration to free up resources. If an enumeration proceeds to the end--that is, until hasMoreElements() or hasMore() returns false-- resources will be freed up automatically and there is no need to explicitly call close().

This method indicates to the service provider that it is free to release resources associated with the enumeration, and can notify servers to cancel any outstanding requests. The close() method is a hint to implementations for managing their resources. Implementations are encouraged to use appropriate algorithms to manage their resources when client omits the close() calls.

**Throws:** [NamingException](http://docs.google.com/javax/naming/NamingException.html) - If a naming exception is encountered while closing the enumeration.**Since:** 1.3

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

[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).