# NSEnumerator Class Reference

**Cocoa > Data Management**

**2007-02-23**

Apple Inc.
© 2007 Apple Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

Apple, the Apple logo, and Cocoa are trademarks of Apple Inc., registered in the United States and other countries.

iPhone is a trademark of Apple Inc.

Simultaneously published in the United States and Canada.

**Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.**

**IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.**

**THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.**

**Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.**

# Contents


## NSEnumerator Class Reference 5

Overview 5
Tasks 6
  Getting the Enumerated Objects 6
Instance Methods 6
  allObjects 6
  nextObject 6

## Document Revision History 9

## Index 11




2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

# NSEnumerator Class Reference

| | |
|---|---|
| **Inherits from** | NSObject |
| **Conforms to** | NSFastEnumeration<br>NSObject (NSObject) |
| **Framework** | /System/Library/Frameworks/Foundation.framework |
| **Availability** | Available in Mac OS X v10.0 and later. |
| **Companion guide** | Collections Programming Topics for Cocoa |
| **Declared in** | NSEnumerator.h |
| **Related sample code** | CoreRecipes<br>iSpend<br>LSMSmartCategorizer<br>SimpleCalendar<br>StickiesExample |

## Overview

`NSEnumerator` is an abstract class, instances of whose subclasses enumerate collections of other objects, such as arrays and dictionaries.

All creation methods are defined in the collection classes—such as `NSArray`, `NSSet`, and `NSDictionary`—which provide special `NSEnumerator` objects with which to enumerate their contents. For example, `NSArray` has two methods that return an `NSEnumerator` object: `objectEnumerator` and `reverseObjectEnumerator`. `NSDictionary` also has two methods that return an `NSEnumerator` object: `keyEnumerator` and `objectEnumerator`. These methods let you enumerate the contents of a dictionary by key or by value, respectively.

You send `nextObject` (page 6) repeatedly to a newly created `NSEnumerator` object to have it return the next object in the original collection. When the collection is exhausted, `nil` is returned. You cannot "reset" an enumerator after it has exhausted its collection. To enumerate a collection again, you need a new enumerator.

The enumerator subclasses used by `NSArray`, `NSDictionary`, and `NSSet` retain the collection during enumeration. When the enumeration is exhausted, the collection is released.


2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

**Note:** It is not safe to modify a mutable collection while enumerating through it. Some enumerators may currently allow enumeration of a collection that is modified, but this behavior is not guaranteed to be supported in the future.

# Tasks

## Getting the Enumerated Objects

- `allObjects` (page 6)
  Returns an array of objects the receiver has yet to enumerate.
- `nextObject` (page 6)
  Returns the next object from the collection being enumerated.

# Instance Methods

## allObjects

Returns an array of objects the receiver has yet to enumerate.

```
- (NSArray *)allObjects
```

**Return Value**
An array of objects the receiver has yet to enumerate.

**Discussion**
Put another way, the array returned by this method does not contain objects that have already been enumerated with previous `nextObject` (page 6) messages.

Invoking this method exhausts the enumerator's collection so that subsequent invocations of `nextObject` return `nil`.

**Availability**
Available in Mac OS X v10.0 and later.

**Declared In**
`NSEnumerator.h`

## nextObject

Returns the next object from the collection being enumerated.

```
- (id)nextObject
```

**Return Value**
The next object from the collection being enumerated, or `nil` when all objects have been enumerated.


2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

**Discussion**
The following code illustrates how this method works using an array:

```
NSArray *anArray = // ... ;
NSEnumerator *enumerator = [anArray objectEnumerator];
id object;

while ((object = [enumerator nextObject])) {
    // do something with object...
}
```

**Availability**
Available in Mac OS X v10.0 and later.

**Related Sample Code**
ColorSyncDevices-Cocoa
Core Animation QuickTime Layer
QTAudioExtractionPanel
SimpleCalendar
StickiesExample

**Declared In**
`NSEnumerator.h`


2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

# Document Revision History

This table describes the changes to *NSEnumerator Class Reference*.

| Date | Notes |
| --- | --- |
| 2007-02-23 | Revised task heading. |
| 2006-05-23 | First publication of this content as a separate document. |



2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

Document Revision History



2007-02-23 | © 2007 Apple Inc. All Rights Reserved.

# Index

## A

`allObjects` instance method 6

## N

`nextObject` instance method 6



2007-02-23 | © 2007 Apple Inc. All Rights Reserved.