NSHashTable Class Reference

Cocoa > Data Management

2007-06-26

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

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

NSHashTable Class Reference 5

Overview 5
Tasks 5
- Initialization 5
- Convenience Constructors 5
- Accessing Content 6
- Manipulating Membership 6
- Comparing Hash Tables 6
- Set Functions 6
- Accessing Pointer Functions 7

Class Methods 7
- hashTableWithOptions: 7
- hashTableWithWeakObjects 7

Instance Methods 8
- addObject: 8
- allObjects 8
- anyObject 8
- containsObject: 9
- count 9
- initWithOptions:capacity: 9
- initWithPointerFunctions:capacity: 10
- intersectHashTable: 10
- intersectsHashTable: 10
- isEqualToHashTable: 11
- isSubsetOfHashTable: 11
- member: 12
- minusHashTable: 12
- objectEnumerator 12
- pointerFunctions 13
- removeAllObjects 13
- removeObject: 13
- setRepresentation 14
- unionHashTable: 14

Constants 14
- NSHashTableOptions 14
- Hash Table Options 15

Document Revision History 17

Index 19

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

NSHashTable Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSCopying NSFastEnumeration NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.5 and later.
Companion guide	Garbage Collection Programming Guide
Declared in	NSHashTable.h

Overview

`NSHashTable` is modeled after `NSSet` but provides different options, in particular to support weak relationships in a garbage-collected environment.

Tasks

Initialization

`- initWithOptions:capacity:` (page 9)
Returns a hash table initialized with the given attributes.

`- initWithPointerFunctions:capacity:` (page 10)
Returns a hash table initialized with the given functions and capacity.

Convenience Constructors

`+ hashTableWithOptions:` (page 7)
Returns a hash table with given pointer functions options.

`+ hashTableWithWeakObjects` (page 7)
Returns a new hash table for storing weak references to its contents.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Accessing Content

- `allObjects` (page 8)
  Returns an array that contains the receiver's members.
- `anyObject` (page 8)
  Returns one of the objects in the receiver.
- `containsObject:` (page 9)
  Returns a Boolean value that indicates whether the receiver contains a given object.
- `count` (page 9)
  Returns the number of elements in the receiver.
- `member:` (page 12)
  Determines whether a given object is an element in the receiver.
- `objectEnumerator` (page 12)
  Returns an enumerator object that lets you access each object in the receiver.
- `setRepresentation` (page 14)
  Returns a set that contains the receiver's members.

Manipulating Membership

- `addObject:` (page 8)
  Adds a given object to the receiver.
- `removeAllObjects` (page 13)
  Removes all objects from the receiver.
- `removeObject:` (page 13)
  Removes a given object from the receiver.

Comparing Hash Tables

- `intersectsHashTable:` (page 10)
  Returns a Boolean value that indicates whether a given hash table intersects with the receiver.
- `isEqualToHashTable:` (page 11)
  Returns a Boolean value that indicates whether a given hash table is equal to the receiver.
- `isSubsetOfHashTable:` (page 11)
  Returns a Boolean value that indicates whether every element in the receiver is also present in another given hash table.

Set Functions

- `intersectHashTable:` (page 10)
  Returns a Boolean value that indicates whether at least one element in the receiver is also present in another given hash table.
- `minusHashTable:` (page 12)
  Removes from the receiver each element contained in another given hash table that is present in the receiver.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

- unionHashTable: (page 14)
  Adds to the receiver each element contained in another given hash table that is not already a member.

Accessing Pointer Functions

- pointerFunctions (page 13)
  Returns the pointer functions for the receiver.

Class Methods

hashTableWithOptions:

Returns a hash table with given pointer functions options.

```
+ (id)hashTableWithOptions:(NSPointerFunctionsOptions)options
```

Parameters

options
  A bit field that specifies the options for the elements in the hash table. For possible values, see "Hash Table Options" (page 15).

Return Value
A hash table with given pointer functions options.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

hashTableWithWeakObjects

Returns a new hash table for storing weak references to its contents.

```
+ (id)hashTableWithWeakObjects
```

Return Value
A new has table that uses the options NSHashTableZeroingWeakMemory (page 15) and NSPointerFunctionsObjectPersonality and has an initial capacity of 0.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Instance Methods

addObject:

Adds a given object to the receiver.

```
- (void)addObject:(id)object
```

Parameters

object

The object to add to the receiver.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

allObjects

Returns an array that contains the receiver's members.

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

Return Value
An array that contains the receiver's members.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

anyObject

Returns one of the objects in the receiver.

```
- (id)anyObject
```

Return Value
One of the objects in the receiver, or `nil` if the receiver contains no objects.

Discussion
The object returned is chosen at the receiver's convenience—the selection is not guaranteed to be random.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

containsObject:

Returns a Boolean value that indicates whether the receiver contains a given object.

```
- (BOOL)containsObject:(id)anObject
```

Parameters

anObject

The object to test for membership in the receiver.

Return Value

`YES` if the receiver contains *anObject*, otherwise `NO`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSHashTable.h`

count

Returns the number of elements in the receiver.

```
- (NSUInteger)count
```

Return Value

The number of elements in the receiver.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSHashTable.h`

initWithOptions:capacity:

Returns a hash table initialized with the given attributes.

```
- (id)initWithOptions:(NSPointerFunctionsOptions)options
    capacity:(NSUInteger)capacity
```

Parameters

options

A bit field that specifies the options for the elements in the hash table. For possible values, see "Hash Table Options" (page 15).

capacity

The initial number of elements the receiver can hold.

Return Value

A hash table initialized with options specified by *options* and initial capacity of *capacity*.

Availability

Available in Mac OS X v10.5 and later.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Declared In
NSHashTable.h

initWithPointerFunctions:capacity:

Returns a hash table initialized with the given functions and capacity.

```
- (id)initWithPointerFunctions:(NSPointerFunctions *)functions
    capacity:(NSUInteger)initialCapacity
```

Parameters

functions

The pointer functions for the new hash table.

initialCapacity

The initial capacity of the hash table.

Return Value
A hash table initialized with the given functions and capacity.

Discussion
Hash tables allocate additional memory as needed, so *initialCapacity* simply establishes the object's initial capacity.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

intersectHashTable:

Returns a Boolean value that indicates whether at least one element in the receiver is also present in another given hash table.

```
- (void)intersectHashTable:(NSHashTable *)other
```

Parameters

other

The hash table with which to compare the receiver.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

intersectsHashTable:

Returns a Boolean value that indicates whether a given hash table intersects with the receiver.

```
- (BOOL)intersectsHashTable:(NSHashTable *)other
```


2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Parameters

other

The hash table with which to compare the receiver.

Return Value

YES if *other* intersects with the receiver, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSHashTable.h

isEqualToHashTable:

Returns a Boolean value that indicates whether a given hash table is equal to the receiver.

```
- (BOOL)isEqualToHashTable:(NSHashTable *)other
```

Parameters

other

The hash table with which to compare the receiver.

Return Value

YES if the contents of *other* are equal to the contents of the receiver, otherwise NO.

Discussion

Two hash tables have equal contents if they each have the same number of members and if each member of one hash table is present in the other.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSHashTable.h

isSubsetOfHashTable:

Returns a Boolean value that indicates whether every element in the receiver is also present in another given hash table.

```
- (BOOL)isSubsetOfHashTable:(NSHashTable *)other
```

Parameters

other

The hash table with which to compare the receiver.

Return Value

YES if every element in the receiver is also present in *other*, otherwise NO.

Availability

Available in Mac OS X v10.5 and later.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Declared In
NSHashTable.h

member:

Determines whether a given object is an element in the receiver.

```
- (id)member:(id)object
```

Parameters

object
The object to test for membership in the receiver.

Return Value
If *object* is a member of the receiver, returns *object*, otherwise returns nil.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

minusHashTable:

Removes from the receiver each element contained in another given hash table that is present in the receiver.

```
- (void)minusHashTable:(NSHashTable *)other
```

Parameters

other
The hash table of elements to remove from the receiver.

Discussion
If any element of *other* isn't present in the receiver, this method has no effect on either the receiver or *other*.

Availability
Available in Mac OS X v10.5 and later.

Declared In
NSHashTable.h

objectEnumerator

Returns an enumerator object that lets you access each object in the receiver.

```
- (NSEnumerator *)objectEnumerator
```

Return Value
An enumerator object that lets you access each object in the receiver.

Discussion
The following code fragment illustrates how you can use this method.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

```
NSEnumerator *enumerator = [myHashTable objectEnumerator];
id value;

while ((value = [enumerator nextObject])) {
    /* code that acts on the hash table's values */
}
```

Note that `NSHashTable` also supports the `NSFastEnumeration` protocol.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

pointerFunctions

Returns the pointer functions for the receiver.

```
- (NSPointerFunctions *)pointerFunctions
```

Return Value
The pointer functions for the receiver.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

removeAllObjects

Removes all objects from the receiver.

```
- (void)removeAllObjects
```

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

removeObject:

Removes a given object from the receiver.

```
- (void)removeObject:(id)object
```

Parameters
object
The object to remove from the receiver.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

setRepresentation

Returns a set that contains the receiver's members.

```
- (NSSet *)setRepresentation
```

Return Value
A set that contains the receiver's members.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

unionHashTable:

Adds to the receiver each element contained in another given hash table that is not already a member.

```
- (void)unionHashTable:(NSHashTable *)other
```

Parameters
other
The hash table of elements to add to the receiver.

Availability
Available in Mac OS X v10.5 and later.

Declared In
`NSHashTable.h`

Constants

NSHashTableOptions

Type to specify a bit-field used to define the behavior of elements in an `NSHashTable` object.

```
typedef NSUInteger NSHashTableOptions;
```

Discussion
For possible values, see "Hash Table Options" (page 15).

Availability
Available in Mac OS X v10.5 and later.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Declared In
NSHashTable.h

Hash Table Options

Components in a bit-field to specify the behavior of elements in an `NSHashTable` object.

```
enum {
    NSHashTableStrongMemory             = 0,
    NSHashTableZeroingWeakMemory        = NSPointerFunctionsZeroingWeakMemory,
    NSHashTableCopyIn                   = NSPointerFunctionsCopyIn,
   NSHashTableObjectPointerPersonality = NSPointerFunctionsObjectPointerPersonality,
};
```

Constants

`NSHashTableStrongMemory`
Equal to `NSPointerFunctionsStrongMemory`.
Available in Mac OS X v10.5 and later.
Declared in `NSHashTable.h`.

`NSHashTableZeroingWeakMemory`
Equal to `NSPointerFunctionsZeroingWeakMemory`.
Available in Mac OS X v10.5 and later.
Declared in `NSHashTable.h`.

`NSHashTableCopyIn`
Equal to `NSPointerFunctionsCopyIn`.
Available in Mac OS X v10.5 and later.
Declared in `NSHashTable.h`.

`NSHashTableObjectPointerPersonality`
Equal to `NSPointerFunctionsObjectPointerPersonality`.
Available in Mac OS X v10.5 and later.
Declared in `NSHashTable.h`.

Declared In
NSHashTable.h

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Document Revision History

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

Date	Notes
2007-06-26	New document that describes the Cocoa class used to contain a set of objects, optionally using weak references.

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Document Revision History

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.

Index

A

addObject: instance method 8
allObjects instance method 8
anyObject instance method 8

C

containsObject: instance method 9
count instance method 9

H

Hash Table Options 15
hashTableWithOptions: class method 7
hashTableWithWeakObjects class method 7

I

initWithOptions:capacity: instance method 9
initWithPointerFunctions:capacity: instance method 10
intersectHashTable: instance method 10
intersectsHashTable: instance method 10
isEqualToHashTable: instance method 11
isSubsetOfHashTable: instance method 11

M

member: instance method 12
minusHashTable: instance method 12

N

NSHashTableCopyIn constant 15
NSHashTableObjectPointerPersonality constant 15
NSHashTableOptions data type 14
NSHashTableStrongMemory constant 15
NSHashTableZeroingWeakMemory constant 15

O

objectEnumerator instance method 12

P

pointerFunctions instance method 13

R

removeAllObjects instance method 13
removeObject: instance method 13

S

setRepresentation instance method 14

U

unionHashTable: instance method 14

2007-06-26 | © 2007 Apple Inc. All Rights Reserved.