

---

# NSMutableDictionary Class Reference

Data Management: Data Types & Collections



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, Cocoa, Mac, Mac OS, and Quartz 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

---

## **NSMutableDictionary Class Reference 5**

---

- Class at a Glance 5
- Overview 6
- Tasks 6
  - Creating and Initializing a Mutable Dictionary 6
  - Adding Entries to a Mutable Dictionary 6
  - Removing Entries From a Mutable Dictionary 7
- Class Methods 7
  - dictionaryWithCapacity: 7
- Instance Methods 8
  - addEntriesFromDictionary: 8
  - initWithCapacity: 8
  - removeAllObjects 9
  - removeObjectForKey: 9
  - removeObjectsForKeys: 10
  - setDictionary: 11
  - setObject:forKey: 11
  - setValue:forKey: 12

---

## **Document Revision History 13**

---

## **Index 15**

---



# NSMutableDictionary Class Reference

---

<b>Inherits from</b>	NSDictionary : NSObject
<b>Conforms to</b>	NSCoding (NSDictionary) NSCopying (NSDictionary) NSMutableCopying (NSDictionary) NSFastEnumeration (NSDictionary) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/Foundation.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Companion guide</b>	Collections Programming Topics for Cocoa
<b>Declared in</b>	NSDictionary.h NSKeyValueCoding.h
<b>Related sample code</b>	From A View to A Movie From A View to A Picture FunHouse GridCalendar Quartz Composer WWDC 2005 TextEdit

## Class at a Glance

An NSDictionary object stores a mutable set of entries.

## Principal Attributes

---

- A count of the number of entries in the dictionary
- The set of keys contained in the dictionary
- The objects that correspond to the keys in the dictionary

[dictionaryWithCapacity:](#) (page 7)

Returns an empty dictionary with enough allocated space to hold a specified number of objects.

## Commonly Used Methods

---

[removeObjectForKey:](#) (page 9)

Removes the specified entry from the dictionary.

[removeObjectsForKeys:](#) (page 10)

Removes multiple entries from the dictionary.

## Overview

The `NSMutableDictionary` class declares the programmatic interface to objects that manage mutable associations of keys and values. With its two efficient primitive methods—[setObject:forKey:](#) (page 11) and [removeObjectForKey:](#) (page 9)—this class adds modification operations to the basic operations it inherits from `NSDictionary`.

The other methods declared here operate by invoking one or both of these primitives. The non-primitive methods provide convenient ways of adding or removing multiple entries at a time.

When an entry is removed from a mutable dictionary, the key and value objects that make up the entry receive `release` messages. If there are no further references to the objects, they're deallocated. Note that if your program keeps a reference to such an object, the reference will become invalid unless you remember to send the object a `retain` message before it's removed from the dictionary. For example, the third statement below would result in a runtime error if `anObject` was not retained before it was removed:

```
id anObject = [[aDictionary objectForKey:theKey] retain];

[aDictionary removeObjectForKey:theKey];
[anObject someMessage];
```

## Tasks

### Creating and Initializing a Mutable Dictionary

+ [dictionaryWithCapacity:](#) (page 7)

Creates and returns a mutable dictionary, initially giving it enough allocated memory to hold a given number of entries.

- [initWithCapacity:](#) (page 8)

Initializes a newly allocated mutable dictionary, allocating enough memory to hold *numItems* entries.

### Adding Entries to a Mutable Dictionary

- [setObject:forKey:](#) (page 11)

Adds a given key-value pair to the receiver.

- [setValue:forKey:](#) (page 12)

Adds a given key-value pair to the receiver.

- [addEntriesFromDictionary:](#) (page 8)  
Adds to the receiver the entries from another dictionary.
- [setDictionary:](#) (page 11)  
Sets the contents of the receiver to entries in a given dictionary.

## Removing Entries From a Mutable Dictionary

- [removeObjectForKey:](#) (page 9)  
Removes a given key and its associated value from the receiver.
- [removeAllObjects](#) (page 9)  
Empties the receiver of its entries.
- [removeObjectsForKeys:](#) (page 10)  
Removes from the receiver entries specified by elements in a given array.

## Class Methods

### **dictionaryWithCapacity:**

Creates and returns a mutable dictionary, initially giving it enough allocated memory to hold a given number of entries.

```
+ (id)dictionaryWithCapacity:(NSUInteger)numItems
```

#### **Parameters**

*numItems*

The initial capacity of the new dictionary.

#### **Return Value**

A new mutable dictionary with enough allocated memory to hold *numItems* entries.

#### **Discussion**

Mutable dictionaries allocate additional memory as needed, so *numItems* simply establishes the object's initial capacity.

#### **Availability**

Available in Mac OS X v10.0 and later.

#### **See Also**

[dictionary](#) (NSMutableDictionary)

[dictionaryWithContentsOfFile:](#) (NSMutableDictionary)

[dictionaryWithContentsOfURL::](#) (NSMutableDictionary)

[dictionaryWithObject:forKey:](#) (NSMutableDictionary)

[dictionaryWithObjects:forKeys::](#) (NSMutableDictionary)

[dictionaryWithObjects:forKeys:count:](#) (NSMutableDictionary)

[dictionaryWithObjectsAndKeys:](#) (NSMutableDictionary)

- [initWithCapacity:](#) (page 8)

**Related Sample Code**

EnhancedAudioBurn

From A View to A Movie

FunHouse

OutputBins2PDE

Quartz Composer WWDC 2005 TextEdit

**Declared In**

NSDictionary.h

## Instance Methods

**addEntriesFromDictionary:**

Adds to the receiver the entries from another dictionary.

- (void)addEntriesFromDictionary:(NSDictionary \*)*otherDictionary***Parameters***otherDictionary*

The dictionary from which to add entries

**Discussion**

Each value object from *otherDictionary* is sent a `retain` message before being added to the receiver. In contrast, each key object is copied (using `copyWithZone:—keys must conform to the NSCopying protocol`), and the copy is added to the receiver.

If both dictionaries contain the same key, the receiver's previous value object for that key is sent a `release` message, and the new value object takes its place.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**- [setObject:forKey:](#) (page 11)**Related Sample Code**

EnhancedDataBurn

From A View to A Movie

From A View to A Picture

QuickLookSketch

Sketch-112

**Declared In**

NSDictionary.h

**initWithCapacity:**Initializes a newly allocated mutable dictionary, allocating enough memory to hold *numItems* entries.

- (id) initWithCapacity:(NSUInteger) *numItems*

#### Parameters

*numItems*

The initial capacity of the initialized dictionary.

#### Return Value

An initialized mutable dictionary, which might be different than the original receiver.

#### Discussion

Mutable dictionaries allocate additional memory as needed, so *numItems* simply establishes the object's initial capacity.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

+ [dictionaryWithCapacity:](#) (page 7)

#### Declared In

NSDictionary.h

## removeAllObjects

Empties the receiver of its entries.

- (void) removeAllObjects

#### Discussion

Each key and corresponding value object is sent a `release` message.

#### Availability

Available in Mac OS X v10.0 and later.

#### See Also

- [removeObjectForKey:](#) (page 9)

- [removeObjectsForKeys:](#) (page 10)

#### Related Sample Code

LSMSmartCategorizer

#### Declared In

NSDictionary.h

## removeObjectForKey:

Removes a given key and its associated value from the receiver.

- (void) removeObjectForKey:(id) *aKey*

#### Parameters

*aKey*

The key to remove.

**Discussion**

Does nothing if *aKey* does not exist.

For example, assume you have an archived dictionary that records the call letters and associated frequencies of radio stations. To remove an entry for a defunct station, you could write code similar to the following:

```
NSMutableDictionary *stations = nil;

stations = [[NSMutableDictionary alloc]
            initWithContentsOfFile: pathToArchive];
[stations removeObjectForKey:@"KIKT"];
```

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [removeAllObjects](#) (page 9)
- [removeObjectsForKeys:](#) (page 10)

**Related Sample Code**

CoreRecipes

DesktopImage

EnhancedAudioBurn

GridCalendar

QuickLookSketch

**Declared In**

NSDictionary.h

**removeObjectsForKeys:**

Removes from the receiver entries specified by elements in a given array.

```
- (void)removeObjectsForKeys:(NSArray *)keyArray
```

**Parameters**

*keyArray*

An array of objects specifying the keys to remove.

**Discussion**

If a key in *keyArray* does not exist, the entry is ignored.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [removeObjectForKey:](#) (page 9)
- [removeObjectForKey:](#) (page 9)

**Related Sample Code**

CoreRecipes

**Declared In**

NSDictionary.h

## setDictionary:

Sets the contents of the receiver to entries in a given dictionary.

- (void)setDictionary:(NSDictionary \*)*otherDictionary*

### Parameters

*otherDictionary*

A dictionary containing the new entries.

### Discussion

All entries are removed from the receiver (with [removeAllObjects](#) (page 9)), then each entry from *otherDictionary* added into the receiver.

### Availability

Available in Mac OS X v10.0 and later.

### Declared In

NSDictionary.h

## setObject:forKey:

Adds a given key-value pair to the receiver.

- (void)setObject:(id)*anObject* forKey:(id)*aKey*

### Parameters

*anObject*

The value for *key*. The object receives a `retain` message before being added to the receiver. This value must not be `nil`.

*aKey*

The key for *value*. The key is copied (using `copyWithZone:`; keys must conform to the `NSCopying` protocol). The key must not be `nil`.

### Discussion

Raises an `NSInvalidArgumentException` if *aKey* or *anObject* is `nil`. If you need to represent a `nil` value in the dictionary, use `NSNull`.

If *aKey* already exists in the receiver, the receiver's previous value object for that key is sent a `release` message and *anObject* takes its place.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [removeObjectForKey:](#) (page 9)

### Related Sample Code

From A View to A Movie

GridCalendar

NSFontAttributeExplorer

Quartz Composer WWDC 2005 TextEdit

Sketch-112

**Declared In**

NSDictionary.h

**setValue:forKey:**

Adds a given key-value pair to the receiver.

```
- (void)setValue:(id)value forKey:(NSString *)key
```

**Parameters***value*

The value for *key*.

*key*

The key for *value*. Note that when using key-value coding, the key must be a string (see Key-Value Coding Fundamentals).

**Discussion**

This method adds *value* and *key* to the receiver using [setObject:forKey:](#) (page 11), unless *value* is `nil` in which case the method instead attempts to remove *key* using [removeObjectForKey:](#) (page 9).

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

[valueForKey:](#) (NSDictionary)

**Related Sample Code**

[CustomAtomicStoreSubclass](#)

[Dicey](#)

[FunHouse](#)

[SimpleCalendar](#)

[SimpleStickies](#)

**Declared In**

NSKeyValueCoding.h

# Document Revision History

---

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

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

**REVISION HISTORY**

Document Revision History

# Index

---

## A

---

addEntriesFromDictionary: [instance method 8](#)

## D

---

dictionaryWithCapacity: [class method 7](#)

## I

---

initWithCapacity: [instance method 8](#)

## R

---

removeAllObjects [instance method 9](#)  
removeObjectForKey: [instance method 9](#)  
removeObjectsForKeys: [instance method 10](#)

## S

---

setDictionary: [instance method 11](#)  
setObject:forKey: [instance method 11](#)  
setValue:forKey: [instance method 12](#)