NSMutableSet Class Reference

Cocoa > Data Management

2006-09-19

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

Simultaneously published in the United States and Canada.

Contents

NSMutableSet Class Reference

Inherits from	NSSet : NSObject
Conforms to	NSCoding (NSSet) NSCopying (NSSet) NSMutableCopying (NSSet) NSFastEnumeration (NSSet) 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	NSPredicate.h NSSet.h
Related sample code	Core Data HTML Store CoreRecipes CustomAtomicStoreSubclass MyPhoto QTMetadataEditor

Overview

The `NSMutableSet` class declares the programmatic interface to an object that manages a mutable set of objects. `NSMutableSet` provides support for the mathematical concept of a set. A set, both in its mathematical sense and in the `NSMutableSet` implementation, is an unordered collection of distinct elements.

The `NSCountedSet` class, which is a concrete subclass of `NSMutableSet`, supports mutable sets that can contain multiple instances of the same element. The `NSSet` class supports creating and managing immutable sets.

You add objects to an `NSMutableSet` object with `addObject:` (page 7), which adds a single object to the set; `addObjectsFromArray:` (page 8), which adds all objects from a specified array to the set; or `unionSet:` (page 11), which adds all the objects from another set. You remove objects from an `NSMutableSet` object using any of the methods `intersectSet:` (page 9), `minusSet:` (page 10), `removeAllObjects` (page 10), or `removeObject:` (page 11).

When an object is added to a set, it receives a `retain` message. When an object is removed from a mutable set, it receives a `release` message. If there are no further references to the object, this means that the object is deallocated. If your program keeps a reference to such an object, the reference will become invalid unless you send the object a `retain` message before it's removed from the array. For example, if `anObject` is not retained before it is removed from the set, the third statement below could result in a runtime error:

```
id anObject = [[aSet anyObject] retain];
[aSet removeObject:anObject];
[anObject someMessage];
```

Tasks

Creating a Mutable Set

`+ setWithCapacity:` (page 7)
Creates and returns a mutable set with a given initial capacity.

`- initWithCapacity:` (page 9)
Returns an initialized mutable set with a given initial capacity.

Adding and Removing Entries

`- addObject:` (page 7)
Adds a given object to the receiver, if it is not already a member.

`- filterUsingPredicate:` (page 8)
Evaluates a given predicate against the receiver's content and removes from the receiver those objects for which the predicate returns false.

`- removeObject:` (page 11)
Removes a given object from the receiver.

`- removeAllObjects` (page 10)
Empties the receiver of all of its members.

`- addObjectsFromArray:` (page 8)
Adds to the receiver each object contained in a given array that is not already a member.

Combining and Recombining Sets

`- unionSet:` (page 11)
Adds to the receiver each object contained in another given set that is not already a member.

`- minusSet:` (page 10)
Removes from the receiver each object contained in another given set that is present in the receiver.

`- intersectSet:` (page 9)
Removes from the receiver each object that isn't a member of another given set.

`- setSet:` (page 11)
Empties the receiver, then adds to the receiver each object contained in another given set.

Class Methods

setWithCapacity:

Creates and returns a mutable set with a given initial capacity.

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

Parameters

numItems

The initial capacity of the new set.

Return Value

A mutable set with initial capacity to hold *numItems* members.

Discussion

Mutable sets 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

- initWithCapacity: (page 9)
+ set (NSSet)
+ setWithObjects:count: (NSSet)

Declared In

NSSet.h

Instance Methods

addObject:

Adds a given object to the receiver, if it is not already a member.

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

Parameters

anObject

The object to add to the receiver.

Discussion

If *anObject* is already present in the set, this method has no effect on either the set or *anObject*.

Availability

Available in Mac OS X v10.0 and later.

See Also

- addObjectsFromArray: (page 8)

- unionSet: (page 11)

Related Sample Code
Core Data HTML Store
CoreRecipes
CustomAtomicStoreSubclass
QTMetadataEditor
Sketch-112

Declared In
NSSet.h

addObjectsFromArray:

Adds to the receiver each object contained in a given array that is not already a member.

```
- (void)addObjectsFromArray:(NSArray *)anArray
```

Parameters

anArray
An array of objects to add to the receiver.

Discussion
If a given element of the array is already present in the set, this method has no effect on either the set or the array element.

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

See Also
- addObject: (page 7)
- unionSet: (page 11)

Related Sample Code
Core Data HTML Store
CoreRecipes

Declared In
NSSet.h

filterUsingPredicate:

Evaluates a given predicate against the receiver's content and removes from the receiver those objects for which the predicate returns false.

```
- (void)filterUsingPredicate:(NSPredicate *)predicate
```

Parameters

predicate
A predicate.

Discussion
The following example illustrates the use of this method.

```
NSMutableSet *mutableSet =
    [NSMutableSet setWithObjects:@"One", @"Two", @"Three", @"Four", nil];
NSPredicate *predicate =
    [NSPredicate predicateWithFormat:@"SELF beginswith 'T'"];
[mutableSet filterUsingPredicate:predicate];
// mutableSet contains (Two, Three)
```

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

Declared In
NSPredicate.h

initWithCapacity:

Returns an initialized mutable set with a given initial capacity.

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

Parameters

numItems
The initial capacity of the set.

Return Value
An initialized mutable set with initial capacity to hold *numItems* members. The returned object might be different than the original receiver.

Discussion
Mutable sets 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
+ setWithCapacity: (page 7)

Declared In
NSSet.h

intersectSet:

Removes from the receiver each object that isn't a member of another given set.

```
- (void)intersectSet:(NSSet *)otherSet
```

Parameters

otherSet
The set with which to perform the intersection.

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

See Also
- `removeObject:` (page 11)
- `removeAllObjects` (page 10)
- `minusSet:` (page 10)

Declared In
`NSSet.h`

minusSet:

Removes from the receiver each object contained in another given set that is present in the receiver.

```
- (void)minusSet:(NSSet *)otherSet
```

Parameters

otherSet
The set of objects to remove from the receiver.

Discussion
If any member of *otherSet* isn't present in the receiving set, this method has no effect on either the receiver or the *otherSet* member.

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

See Also
- `removeObject:` (page 11)
- `removeAllObjects` (page 10)
- `intersectSet:` (page 9)

Related Sample Code
CoreRecipes
MyPhoto

Declared In
`NSSet.h`

removeAllObjects

Empties the receiver of all of its members.

```
- (void)removeAllObjects
```

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

See Also
- `removeObject:` (page 11)
- `minusSet:` (page 10)

- intersectSet: (page 9)

Related Sample Code
CoreRecipes

Declared In
NSSet.h

removeObject:

Removes a given object from the receiver.

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

Parameters

anObject
The object to remove from the receiver.

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

See Also
- removeAllObjects (page 10)
- minusSet: (page 10)
- intersectSet: (page 9)

Related Sample Code
CoreRecipes

Declared In
NSSet.h

setSet:

Empties the receiver, then adds to the receiver each object contained in another given set.

```
- (void)setSet:(NSSet *)otherSet
```

Parameters

otherSet
The set whose members replace the receiver's content.

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

Declared In
NSSet.h

unionSet:

Adds to the receiver each object contained in another given set that is not already a member.

```
- (void)unionSet:(NSSet *)otherSet
```

Parameters

otherSet

The set of objects to add to the receiver.

Discussion

If any member of *otherSet* is already present in the receiver, this method has no effect on either the receiver or the *otherSet* member.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `addObject:` (page 7)
- `addObjectsFromArray:` (page 8)

Declared In

`NSSet.h`

Document Revision History

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

Date	Notes
2006-09-19	Included API introduced in Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

Document Revision History

Index