NSCountedSet Class Reference

Cocoa > Data Management

2009-05-06

Contents

NSCountedSet Class Reference

Inherits from	NSMutableSet : 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	NSSet.h
Related sample code	Dicey

Overview

The `NSCountedSet` class declares the programmatic interface to an object that manages a mutable set of objects. `NSCountedSet` provides support for the mathematical concept of a counted set. A counted set, both in its mathematical sense and in the implementation of `NSCountedSet`, is an unordered collection of elements, just as in a regular set, but the elements of the set aren't necessarily distinct. A counted set is also known as a bag.

Each distinct object inserted into an `NSCountedSet` object has a counter associated with it. `NSCountedSet`keeps track of the number of times objects are inserted and requires that objects be removed the same number of times. Thus, there is only one instance of an object in an `NSSet` object even if the object has been added to the set multiple times. The `count` method defined by the superclass `NSSet` has special significance; it returns the number of distinct objects, not the total number of times objects are represented in the set. The `NSSet` and `NSMutableSet` classes are provided for static and dynamic sets (respectively) whose elements are distinct.

You add objects to or remove objects from a counted set using the `addObject:` (page 6) and `removeObject:` (page 9) methods. You can traverse elements of an `NSCountedSet` object using the enumerator returned by `objectEnumerator` (page 9). The `countForObject:` (page 7) method returns the number of times a given object has been added to this set.

While `NSCountedSet` and `CFBag` are not toll-free bridged, they provide similar functionality. For more information on `CFBag`, consult the *CFBag Reference*.

Tasks

Initializing a Counted Set

`– initWithArray:` (page 7)
Returns a counted set object initialized with the contents of a given array.

`– initWithSet:` (page 8)
Returns a counted set object initialized with the contents of a given set.

`– initWithCapacity:` (page 8)
Returns a counted set object initialized with enough memory to hold a given number of objects.

Adding and Removing Entries

`– addObject:` (page 6)
Adds a given object to the receiver.

`– removeObject:` (page 9)
Removes a given object from the receiver.

Examining a Counted Set

`– countForObject:` (page 7)
Returns the count associated with a given object in the receiver.

`– objectEnumerator` (page 9)
Returns an enumerator object that lets you access each object in the set once, independent of its count.

Instance Methods

addObject:

Adds a given object to the receiver.

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

Parameters

anObject
The object to add to the receiver.

Discussion
If *anObject* is already a member, `addObject:` increments the count associated with the object. If *anObject* is not already a member, it is sent a `retain` message.

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

Declared In
`NSSet.h`

countForObject:

Returns the count associated with a given object in the receiver.

```
- (NSUInteger)countForObject:(id)anObject
```

Parameters

anObject
The object for which to return the count.

Return Value
The count associated with *anObject* in the receiver, which can be thought of as the number of occurrences of *anObject* present in the receiver.

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

See Also
`– count` (NSSet)

Related Sample Code
Dicey

Declared In
`NSSet.h`

initWithArray:

Returns a counted set object initialized with the contents of a given array.

```
- (id)initWithArray:(NSArray *)anArray
```

Parameters

anArray
An array of objects to add to the new set.

Return Value
An initialized counted set object with the contents of *anArray*. The returned object might be different than the original receiver.

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

See Also
`initWithArray:` (NSSet)
`setWithArray:` (NSSet)

Declared In
`NSSet.h`

initWithCapacity:

Returns a counted set object initialized with enough memory to hold a given number of objects.

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

Parameters

numItems

The initial capacity of the new counted set.

Return Value

A counted set object initialized with enough memory to hold `numItems` objects

Discussion

The method is the designated initializer for `NSCountedSet`.

Note that the capacity is simply a hint to help initial memory allocation—the initial count of the object is `0`, and the set still grows and shrinks as you add and remove objects. The hint is typically useful if the set will become large.

Availability

Available in Mac OS X v10.0 and later.

See Also

`initWithCapacity:` (NSMutableSet)
`setWithCapacity:` (NSMutableSet)

Declared In

`NSSet.h`

initWithSet:

Returns a counted set object initialized with the contents of a given set.

```
- (id)initWithSet:(NSSet *)aSet
```

Parameters

aSet

An set of objects to add to the new set.

Return Value

An initialized counted set object with the contents of `aSet`. The returned object might be different than the original receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

`initWithSet:` (NSSet)
`setWithSet:` (NSSet)

Declared In

`NSSet.h`

objectEnumerator

Returns an enumerator object that lets you access each object in the set once, independent of its count.

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

Return Value
An enumerator object that lets you access each object in the set once, independent of its count.

Discussion
If you add a given object to the counted set multiple times, an enumeration of the set will produce that object only once.

You shouldn't modify the set during enumeration. If you intend to modify the set, use the `allObjects` method to create a "snapshot," then enumerate the snapshot and modify the original set.

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

See Also
`nextObject` (NSEnumerator)

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.

Discussion
If *`anObject`* is present in the set, decrements the count associated with it. If the count is decremented to `0`, *`anObject`* is removed from the set and sent a `release` message. `removeObject:` does nothing if *`anObject`* is not present in the receiver.

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

See Also
`- countForObject:` (page 7)

Declared In
`NSSet.h`

Document Revision History

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

Date	Notes
2009-05-06	Referenced CFBag, which implements similar functionality.
2007-01-31	Updated to Mac OS X version 5 and moved minor changes made in v10.4 to v10.5.
2006-11-07	Moved extended description of -count to NSSet reference and emphasized the importance of this method.
2006-05-23	First publication of this content as a separate document.

Index