NSPointerFunctions Class Reference

Cocoa > Data Management

2008-10-15

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

Simultaneously published in the United States and Canada.

Contents

NSPointerFunctions Class Reference

Inherits from	NSObject
Conforms to	NSCopying NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.5 and later.
Declared in	NSPointerFunctions.h
Companion guides	Collections Programming Topics for Cocoa Garbage Collection Programming Guide

Overview

An instance of `NSPointerFunctions` defines callout functions appropriate for managing a pointer reference held somewhere else.

The functions specified by an instance of `NSPointerFunctions` are separated into two clusters—those that define “personality” such as “object” or "C-string", and those that describe memory management issues such as a memory deallocation function. There are constants for common personalities and memory manager selections (see “Memory and Personality Options” (page 10)).

`NSHashTable`, `NSMapTable`, and `NSPointerArray` use an `NSPointerFunctions` object to define the acquisition and retention behavior for the pointers they manage. Note, however, that not all combinations of personality and memory management behavior are valid for these collections. The pointer collection objects copy the `NSPointerFunctions` object on input and output, so you cannot usefully subclass `NSPointerFunctions`.

Tasks

Creating and Initializing an NSPointerFunctions Object

`- initWithOptions:` (page 9)
Returns an `NSPointerFunctions` object initialized with the given options.

`+ pointerFunctionsWithOptions:` (page 9)
Returns a new `NSPointerFunctions` object initialized with the given options.

Personality Functions

`hashFunction` (page 7) *property*
The hash function.

`isEqualFunction` (page 7) *property*
The function used to compare pointers.

`sizeFunction` (page 8) *property*
The function used to determine the size of pointers.

`descriptionFunction` (page 7) *property*
The function used to describe elements.

Memory Configuration

`acquireFunction` (page 6) *property*
The function used to acquire memory.

`relinquishFunction` (page 7) *property*
The function used to relinquish memory.

`usesStrongWriteBarrier` (page 8) *property*
Specifies whether, in a garbage collected environment, pointers should be assigned using a strong write barrier.

`usesWeakReadAndWriteBarriers` (page 8) *property*
Specifies whether, in a garbage collected environment, pointers should use weak read and write barriers.

Properties

For more about Objective-C properties, see "Properties" in *The Objective-C 2.0 Programming Language*.

acquireFunction

The function used to acquire memory.

```
@property void *(*acquireFunction)(const void *src, NSUInteger (*size)(const void
    *item), BOOL shouldCopy)
```

Discussion
This specifies the function to use for copy-in operations.

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

See Also
`@property relinquishFunction` (page 7)

Declared In
`NSPointerFunctions.h`

descriptionFunction

The function used to describe elements.

```
@property NSString *(*descriptionFunction)(const void *item)
```

Discussion
This function is used by description methods for hash and map tables.

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

Declared In
`NSPointerFunctions.h`

hashFunction

The hash function.

```
@property NSUInteger (*hashFunction)(const void *item, NSUInteger (*size)(const
    void *item))
```

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

Declared In
`NSPointerFunctions.h`

isEqualFunction

The function used to compare pointers.

```
@property BOOL (*isEqualFunction)(const void *item1, const void*item2, NSUInteger
     (*size)(const void *item))
```

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

Declared In
`NSPointerFunctions.h`

relinquishFunction

The function used to relinquish memory.

```
@property void (*relinquishFunction)(const void *item, NSUInteger (*size)(const
    void *item))
```

Discussion
This specifies the function to use when an item is removed from a table or pointer array.

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

See Also
`@property acquireFunction` (page 6)

Declared In
`NSPointerFunctions.h`

sizeFunction

The function used to determine the size of pointers.

```
@property NSUInteger (*sizeFunction)(const void *item)
```

Discussion
This function is used for copy-in operations (unless the collection has an object personality).

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

Declared In
`NSPointerFunctions.h`

usesStrongWriteBarrier

Specifies whether, in a garbage collected environment, pointers should be assigned using a strong write barrier.

```
@property BOOL usesStrongWriteBarrier
```

Discussion
If you use garbage collection, read and write barrier functions must be used when pointers are from memory scanned by the collector.

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

See Also
`@property usesWeakReadAndWriteBarriers` (page 8)

Declared In
`NSPointerFunctions.h`

usesWeakReadAndWriteBarriers

Specifies whether, in a garbage collected environment, pointers should use weak read and write barriers.

```
@property BOOL usesWeakReadAndWriteBarriers
```

Discussion
If you use garbage collection, read and write barrier functions must be used when pointers are from memory scanned by the collector.

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

See Also
`@property usesStrongWriteBarrier` (page 8)

Declared In
`NSPointerFunctions.h`

Class Methods

pointerFunctionsWithOptions:

Returns a new `NSPointerFunctions` object initialized with the given options.

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

Parameters
options
The options for the new `NSPointerFunctions` object.

Return Value
A new `NSPointerFunctions` object initialized with the given options.

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

Declared In
`NSPointerFunctions.h`

Instance Methods

initWithOptions:

Returns an `NSPointerFunctions` object initialized with the given options.

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

Parameters
options
The options for the new `NSPointerFunctions` object.

Return Value
The receiver, initialized with the given options.

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

Declared In
`NSPointerFunctions.h`

Constants

NSPointerFunctionsOptions

Defines the memory and personality options for an `NSPointerFunctions` object.

```
typedef NSUInteger NSPointerFunctionsOptions;
```

Discussion
For values, see "Memory and Personality Options" (page 10).

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

Declared In
`NSPointerFunctions.h`

Memory and Personality Options

Specify memory and personality options for an `NSPointerFunctions` object.

```
enum {
    NSPointerFunctionsStrongMemory = (0 << 0),
    NSPointerFunctionsZeroingWeakMemory = (1 << 0),
    NSPointerFunctionsOpaqueMemory = (2 << 0),
    NSPointerFunctionsMallocMemory = (3 << 0),
    NSPointerFunctionsMachVirtualMemory = (4 << 0),
    NSPointerFunctionsObjectPersonality = (0 << 8),
    NSPointerFunctionsOpaquePersonality = (1 << 8),
    NSPointerFunctionsObjectPointerPersonality = (2 << 8),
    NSPointerFunctionsCStringPersonality = (3 << 8),
    NSPointerFunctionsStructPersonality = (4 << 8),
    NSPointerFunctionsIntegerPersonality = (5 << 8),
    NSPointerFunctionsCopyIn = (1 << 16),
};
```

Constants

`NSPointerFunctionsStrongMemory`

Use strong write-barriers to backing store; use garbage-collected memory on copy-in.

This is the default memory value.

As a special case, if you do not use garbage collection and specify this value in conjunction with `NSPointerFunctionsObjectPersonality` (page 11) then the `NSPointerFunctions` object uses `retain` and `release`.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsZeroingWeakMemory`

Use weak read and write barriers; use garbage-collected memory on copyIn.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsOpaqueMemory`
Take no action when pointers are deleted.

This is essentially a no-op relinquish function; the acquire function is only used for copy-in operations. This option is unlikely a to be a good choice for objects.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsMallocMemory`
Use `free()` on removal, `calloc()` on copy in.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsMachVirtualMemory`
Use Mach memory.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsObjectPersonality`
Use `hash` and `isEqual` methods for hashing and equality comparisons, use the `description` method for a description.

This is the default personality value.

As a special case, if you do not use garbage collection and specify this value in conjunction with `NSPointerFunctionsStrongMemory` (page 10) then the `NSPointerFunctions` object uses `retain` and `release`.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsOpaquePersonality`
Use shifted pointer for the hash value and direct comparison to determine equality.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsObjectPointerPersonality`
Use shifted pointer for the hash value and direct comparison to determine equality; use the `description` method for a description.

As a special case, if you do not use garbage collection and specify this value in conjunction with `NSPointerFunctionsStrongMemory` (page 10) then the `NSPointerFunctions` object uses `retain` and `release`.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsCStringPersonality`
Use a string hash and `strcmp`; C-string '`%s`' style description.

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsStructPersonality`
Use a memory hash and `memcmp` (using a size function that you must set—see `sizeFunction` (page 8)).

Available in Mac OS X v10.5 and later.

Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsIntegerPersonality`
: Use unshifted value as hash and equality.

  Available in Mac OS X v10.5 and later.

  Declared in `NSPointerFunctions.h`.

`NSPointerFunctionsCopyIn`
: Use the memory acquire function to allocate and copy items on input (see `acquireFunction` (page 6)).

  Available in Mac OS X v10.5 and later.

  Declared in `NSPointerFunctions.h`.

Discussion
Memory options are mutually exclusive and personality options are mutually exclusive.

Declared In
`NSPointerFunctions.h`

Document Revision History

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

Date	Notes
2008-10-15	Added definition of NSPointerFunctionsOpaqueMemory.
2007-10-31	Clarified definition of NSPointerFunctionsObjectPersonality.
2007-06-26	New document that describes the class used to define callout functions appropriate for managing a pointer reference.

Document Revision History

Index