
NSScriptObjectSpecifier Class Reference

Interapplication Communication





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, AppleScript, 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

NSScriptObjectSpecifier Class Reference 5

Overview	5
Adopted Protocols	6
Tasks	6
Obtaining an Object Specifier for a Descriptor	6
Initializing an Object Specifier	6
Evaluating an Object Specifier	6
Getting, Testing, and Setting Containers	7
Getting and Setting Child References	7
Getting and Setting Object Keys	7
Getting Evaluation Errors	7
Getting a Descriptor for the Object Specifier	8
Class Methods	8
objectSpecifierWithDescriptor:	8
Instance Methods	8
childSpecifier	8
containerClassDescription	9
containerIsObjectBeingTested	9
containerIsRangeContainerObject	10
containerSpecifier	10
descriptor	10
evaluationErrorNumber	11
evaluationErrorSpecifier	11
indicesOfObjectsByEvaluatingWithContainer:count:	12
initWithContainerClassDescription:containerSpecifier:key:	12
initWithContainerSpecifier:key:	12
key	13
keyClassDescription	13
objectsByEvaluatingSpecifier	14
objectsByEvaluatingWithContainers:	14
setChildSpecifier:	15
setContainerClassDescription:	15
setContainerIsObjectBeingTested:	16
setContainerIsRangeContainerObject:	16
setContainerSpecifier:	17
setEvaluationErrorNumber:	17
setKey:	17
Constants	18
NSScriptObjectSpecifier—Specifier Evaluation Errors	18

Document Revision History 21

Index 23

NSScriptObjectSpecifier Class Reference

Inherits from	NSObject
Conforms to	NSCoding NSObject (NSObject)
Framework	/System/Library/Frameworks/Foundation.framework
Availability	Available in Mac OS X v10.0 and later.
Companion guide	Cocoa Scripting Guide
Declared in	NSScriptObjectSpecifiers.h
Related sample code	QuickLookSketch SimpleScriptingObjects SimpleScriptingPlugin Sketch+Accessibility Sketch-112

Overview

`NSScriptObjectSpecifier` is the abstract superclass for classes that instantiate objects called “object specifiers.” An object specifier represents an AppleScript reference form, which is a natural-language expression such as `words 10 through 20 or front document or words whose color is red`.

The scripting system maps these words or phrases to attributes and relationships of scriptable objects. A reference form rarely occurs in isolation; usually a script statement consists of a series of reference forms preceded by a command and typically connected to each other by `of`, such as:

```
get words whose color is blue of paragraph 10 of front document
```

The expression `words whose color is blue of paragraph 10 of front document` specifies a location in the application's AppleScript object model—the objects the application makes available to scripters. The classes of objects in the object model often closely match the classes of actual objects in the application, but they are not required to. An object specifier locates objects in the running application that correspond to the specified object model objects.

Your application typically creates object specifiers when it implements the `objectSpecifier` method for its scriptable classes. That method is defined by the `NSScriptObjectSpecifiers` protocol.

It is unlikely that you would ever need to create your own subclass of `NSScriptObjectSpecifier`; the set of valid AppleScript reference forms is determined by Apple Computer and object specifier classes are already implemented for this set. If for some reason you do need to create a subclass, you must override the primitive

method `indicesOfObjectsByEvaluatingWithContainer:count:` (page 12) to return indices to the elements within the container whose values are matched with the child specifier's key. In addition, you probably need to declare any special instance variables and implement an initializer that invokes super's designated initializer, `initWithContainerClassDescription:containerSpecifier:key:` (page 12), and initializes these variables.

For a comprehensive treatment of object specifiers, including sample code, see Object Specifiers in *Cocoa Scripting Guide*.

Adopted Protocols

NSCoding

- `encodeWithCoder:`
- `initWithCoder:`

Tasks

Obtaining an Object Specifier for a Descriptor

- + `objectSpecifierWithDescriptor:` (page 8)
Returns a new object specifier for an Apple event descriptor.

Initializing an Object Specifier

- `initWithContainerClassDescription:containerSpecifier:key:` (page 12)
Returns an `NSScriptObjectSpecifier` object initialized with the given attributes.
- `initWithContainerSpecifier:key:` (page 12)
Returns an `NSScriptObjectSpecifier` object initialized with a given container specifier and key.

Evaluating an Object Specifier

- `indicesOfObjectsByEvaluatingWithContainer:count:` (page 12)
This primitive method must be overridden by subclasses to return a pointer to an array of indices identifying objects in the key of a given container that are identified by the receiver of the message.
- `objectsByEvaluatingSpecifier` (page 14)
Returns the actual object represented by the nested series of object specifiers.
- `objectsByEvaluatingWithContainers:` (page 14)
Returns the actual object or objects specified by the receiver as evaluated in the context of given container object.

Getting, Testing, and Setting Containers

- [containerClassDescription](#) (page 9)
Returns the class description of the object indicated by the receiver's container specifier.
- [containerIsObjectBeingTested](#) (page 9)
If the receiver's container specifier is `nil`, returns a Boolean value that indicates whether the receiver's container is the object involved in a specifier test.
- [containerIsRangeContainerObject](#) (page 10)
If the receiver's container specifier is `nil`, returns a Boolean value that indicates whether the container for the receiver contains the range of elements represented by an `NSRangeSpecifier`.
- [containerSpecifier](#) (page 10)
Returns the receiver's container specifier.
- [setContainerClassDescription:](#) (page 15)
Sets the class description of the receiver's container specifier to a given specifier.
- [setContainerIsObjectBeingTested:](#) (page 16)
Sets whether the receiver's container should be an object involved in a filter reference or the top-level object.
- [setContainerSpecifier:](#) (page 17)
Sets the container specifier of the receiver.
- [setContainerIsRangeContainerObject:](#) (page 16)
Sets whether the receiver's container is to be the container for a range specifier or a top-level object.

Getting and Setting Child References

- [childSpecifier](#) (page 8)
Returns the receiver's child reference.
- [setChildSpecifier:](#) (page 15)
Sets the receiver's child reference.

Getting and Setting Object Keys

- [key](#) (page 13)
Returns the key of the receiver.
- [keyClassDescription](#) (page 13)
Returns the class description of the objects specified by the receiver.
- [setKey:](#) (page 17)
Sets the key of the receiver.

Getting Evaluation Errors

- [evaluationErrorSpecifier](#) (page 11)
Returns the object specifier in which an evaluation error occurred.

- [evaluationErrorNumber](#) (page 11)
Returns the constant identifying the type of error that caused evaluation to fail.
- [setEvaluationErrorNumber:](#) (page 17)
Sets the value of the evaluation error.

Getting a Descriptor for the Object Specifier

- [descriptor](#) (page 10)
Returns an Apple event descriptor that represents the receiver.

Class Methods

objectSpecifierWithDescriptor:

Returns a new object specifier for an Apple event descriptor.

```
+ (NSScriptObjectSpecifier *)objectSpecifierWithDescriptor:(NSAppleEventDescriptor *)descriptor
```

Parameters

descriptor

An Apple event descriptor. The descriptor must have the type `typeObjectSpecifier`.

Return Value

An object specifier, or `nil` if an error occurs.

Discussion

If `objectSpecifierWithDescriptor:` is invoked and fails during the execution of a script command, information about the error that caused the failure is recorded in `[NSScriptCommand currentCommand]`.

Availability

Available in Mac OS X v10.5 and later.

Declared In

`NSScriptObjectSpecifiers.h`

Instance Methods

childSpecifier

Returns the receiver's child reference.

```
- (NSScriptObjectSpecifier *)childSpecifier
```

Return Value

The receiver's child reference, that is, the object specifier evaluating to the object or objects that the receiver contains.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setChildSpecifier:](#) (page 15)

Declared In

NSScriptObjectSpecifiers.h

containerClassDescription

Returns the class description of the object indicated by the receiver's container specifier.

- (NSScriptClassDescription *)containerClassDescription

Return Value

The class description of the object indicated by the receiver's container specifier.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setContainerClassDescription:](#) (page 15)

Declared In

NSScriptObjectSpecifiers.h

containerIsObjectBeingTested

If the receiver's container specifier is `nil`, returns a Boolean value that indicates whether the receiver's container is the object involved in a specifier test.

- (BOOL)containerIsObjectBeingTested

Return Value

YES if the receiver's container is the object involved in a specifier test, otherwise NO.

Discussion

An example of a specifier test is `whose color is blue`. If the returned value is YES, then the top-level object is the object being tested (that is, the specifier has no container specifier).

Availability

Available in Mac OS X v10.0 and later.

See Also

- `objectBeingTested` (NSScriptExecutionContext)

Declared In

NSScriptObjectSpecifiers.h

containerIsRangeContainerObject

If the receiver's container specifier is `nil`, returns a Boolean value that indicates whether the container for the receiver contains the range of elements represented by an `NSRangeSpecifier`.

- (BOOL)containerIsRangeContainerObject

Return Value

YES if the container for the receiver contains the range of elements represented by an `NSRangeSpecifier`, otherwise NO.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setContainerIsRangeContainerObject:](#) (page 16)

Declared In

`NSScriptObjectSpecifiers.h`

containerSpecifier

Returns the receiver's container specifier.

- (NSScriptObjectSpecifier *)containerSpecifier

Return Value

The receiver's container specifier, which is the object specifier that must be evaluated to provide a context for the evaluation of the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [childSpecifier](#) (page 8)
 - [containerClassDescription](#) (page 9)
 - [setContainerSpecifier:](#) (page 17)

Declared In

`NSScriptObjectSpecifiers.h`

descriptor

Returns an Apple event descriptor that represents the receiver.

- (NSAppleEventDescriptor *)descriptor

Return Value

An Apple event descriptor of type `typeObjectSpecifier`.

Discussion

If the receiver was created with [objectSpecifierWithDescriptor:](#) (page 8), the passed-in descriptor is returned. Otherwise, a new descriptor is created and returned, autoreleased.

Availability

Available in Mac OS X v10.5 and later.

Declared In

NSScriptObjectSpecifiers.h

evaluationErrorNumber

Returns the constant identifying the type of error that caused evaluation to fail.

- (NSInteger)evaluationErrorNumber

Return Value

The constant identifying the type of error that caused evaluation to fail.

Discussion

This error code could be associated with the receiver or any container specifier “above” the receiver. Possible return values are defined in “Constants” (page 18).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [evaluationErrorSpecifier](#) (page 11)

Declared In

NSScriptObjectSpecifiers.h

evaluationErrorSpecifier

Returns the object specifier in which an evaluation error occurred.

- (NSScriptObjectSpecifier *)evaluationErrorSpecifier

Return Value

The object specifier in which an evaluation error occurred.

Discussion

The object specifier failing to evaluate could be the receiver or any container specifier “above” the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [evaluationErrorNumber](#) (page 11)

Declared In

NSScriptObjectSpecifiers.h

indicesOfObjectsByEvaluatingWithContainer:count:

This primitive method must be overridden by subclasses to return a pointer to an array of indices identifying objects in the key of a given container that are identified by the receiver of the message.

```
- (NSInteger *)indicesOfObjectsByEvaluatingWithContainer:(id)aContainer
    count:(NSInteger *)numRefs
```

Discussion

This primitive method must be overridden by subclasses to return a pointer to an array of indices identifying objects in the key of the container *aContainer* that are identified by the receiver of the message. The method uses key-value coding to obtain values based on the receiver's key. It returns the number of such matching objects by indirection in *numRefs*. It returns *nil* directly and *-1* via *numRefs* if all objects in the container (or the sole object) match the value of the receiver's key. This method is invoked by [objectsByEvaluatingWithContainers:](#) (page 14). The default implementation returns *nil* directly and *-1* indirectly via *numRefs*.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScriptObjectSpecifiers.h

initWithContainerClassDescription:containerSpecifier:key:

Returns an `NSScriptObjectSpecifier` object initialized with the given attributes.

```
- (id)initWithContainerClassDescription:(NSScriptClassDescription *)classDescription
    containerSpecifier:(NSScriptObjectSpecifier *)specifier key:(NSString *)key
```

Return Value

An `NSScriptObjectSpecifier` object initialized with container specifier *specifier*, key *key*, and the class description of the object specifier *classDescription*, derived from the value of the specifier's key.

Discussion

You should never pass *nil* for the value of *classDescription*. The receiver's child reference is set to *nil*.

This is the designated initializer for `NSScriptObjectSpecifier`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScriptObjectSpecifiers.h

initWithContainerSpecifier:key:

Returns an `NSScriptObjectSpecifier` object initialized with a given container specifier and key.

```
- (id)initWithContainerSpecifier:(NSScriptObjectSpecifier *)specifier key:(NSString *)key
```

Return Value

An `NSScriptObjectSpecifier` object initialized with container specifier *specifier* and key *key*.

Discussion

The class description of the container is set automatically.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScriptObjectSpecifiers.h

key

Returns the key of the receiver.

- (NSString *)key

Return Value

The key of the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [keyClassDescription](#) (page 13)
- [setKey:](#) (page 17)

Related Sample Code

QuickLookSketch
Sketch+Accessibility
Sketch-112

Declared In

NSScriptObjectSpecifiers.h

keyClassDescription

Returns the class description of the objects specified by the receiver.

- (NSScriptClassDescription *)keyClassDescription

Return Value

The class description of the objects specified by the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [key](#) (page 13)
- [setKey:](#) (page 17)

Related Sample Code

QuickLookSketch
Sketch+Accessibility

Sketch-112

Declared In

NSScriptObjectSpecifiers.h

objectsByEvaluatingSpecifier

Returns the actual object represented by the nested series of object specifiers.

- (id)objectsByEvaluatingSpecifier

Return Value

The actual object represented by the nested series of object specifiers.

Discussion

Recursively obtains the next container in a nested series of object specifiers until it reaches the top-level container specifier (which is either an `NSWhoseSpecifier` or the application object), after which it begins evaluating each object specifier (`objectsByEvaluatingWithContainers:` (page 14)) going in the opposite direction (top-level to innermost) as it unwinds from the stack. Returns the actual object represented by the nested series of object specifiers. Returns `nil` if a container specifier could not be evaluated or if no top-level container specifier could be found. Thus `nil` can be a valid value or can indicate an error; you can use `evaluationErrorNumber` (page 11) to determine if and which error occurred and `evaluationErrorSpecifier` (page 11) to find the container specifier responsible for the error. In the normal course of command processing, this method is invoked by an `NSScriptCommand` object's `evaluatedArguments` and `evaluatedReceivers` methods, which take as message receiver the innermost object specifier.

Availability

Available in Mac OS X v10.0 and later.

See Also- `indicesOfObjectsByEvaluatingWithContainer:count:` (page 12)**Related Sample Code**

Sketch-112

Declared In

NSScriptObjectSpecifiers.h

objectsByEvaluatingWithContainers:

Returns the actual object or objects specified by the receiver as evaluated in the context of given container object.

- (id)objectsByEvaluatingWithContainers:(id)containers

Return ValueThe actual object or objects specified by the receiver as evaluated in the context of its container object or objects (`containers`).

Discussion

Invokes [indicesOfObjectsByEvaluatingWithContainer:count:](#) (page 12) on `self` to get an array of pointers to indices of elements in `containers` that have values paired with the message receiver's key. This method then uses key-value coding to obtain the object or objects associated with the key; it returns these objects or `nil` if there are no matching values in containers. If there are multiple matching values, they are returned in an `NSArray`; if matching values are `nil`, `NSNull` objects are substituted. If `containers` is an `NSArray`, the method recursively evaluates each element in the array and returns an `NSArray` with evaluated objects (including `NSNull`s) in their corresponding slots.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [objectsByEvaluatingSpecifier](#) (page 14)

Related Sample Code

QuickLookSketch

Sketch+Accessibility

Sketch-112

Declared In

`NSScriptObjectSpecifiers.h`

setChildSpecifier:

Sets the receiver's child reference.

```
- (void)setChildSpecifier:(NSScriptObjectSpecifier *)child
```

Parameters

child

The receiver's child reference.

Discussion

Do not invoke this method directly; it is automatically invoked by [setContainerSpecifier:](#) (page 17).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [childSpecifier](#) (page 8)

Declared In

`NSScriptObjectSpecifiers.h`

setContainerClassDescription:

Sets the class description of the receiver's container specifier to a given specifier.

```
- (void)setContainerClassDescription:(NSScriptClassDescription *)classDescription
```

Parameters*classDescription*

The class description of the receiver's container specifier.

Availability

Available in Mac OS X v10.0 and later.

See Also- [containerClassDescription](#) (page 9)**Declared In**

NSScriptObjectSpecifiers.h

setContainerIsObjectBeingTested:

Sets whether the receiver's container should be an object involved in a filter reference or the top-level object.

- (void)setContainerIsObjectBeingTested:(BOOL)flag

Discussion

If the receiver's container specifier is `nil` and *flag* is YES, sets the receiver's container to be an object involved in a filter reference (for example, whose color is blue). If the receiver's container specifier is `nil` and *flag* is NO, sets the receiver's container to be the top-level object.

If *flag* is YES [setContainerIsRangeContainerObject:](#) (page 16) should not also be invoked with an argument of YES.

Availability

Available in Mac OS X v10.0 and later.

See Also- [containerIsObjectBeingTested](#) (page 9)**Declared In**

NSScriptObjectSpecifiers.h

setContainerIsRangeContainerObject:

Sets whether the receiver's container is to be the container for a range specifier or a top-level object.

- (void)setContainerIsRangeContainerObject:(BOOL)flag

Discussion

If the receiver's container specifier is `nil` and *flag* is YES, sets the receiver's container to be the container for a range specifier. If the receiver's container specifier is `nil` and *flag* is NO, sets the receiver's container to be the top-level object.

If *flag* is YES, [setContainerIsObjectBeingTested:](#) (page 16) should not also be invoked with an argument of YES.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [containerIsRangeContainerObject](#) (page 10)

Declared In

NSScriptObjectSpecifiers.h

setContainerSpecifier:

Sets the container specifier of the receiver.

- (void)setContainerSpecifier:(NSScriptObjectSpecifier *)*objSpecifier*

Parameters

objSpecifier

The container specifier for the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [containerSpecifier](#) (page 10)

Declared In

NSScriptObjectSpecifiers.h

setEvaluationErrorNumber:

Sets the value of the evaluation error.

- (void)setEvaluationErrorNumber:(NSInteger)*error*

Parameters

error

The value for the evaluation error.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [evaluationErrorNumber](#) (page 11)

Declared In

NSScriptObjectSpecifiers.h

setKey:

Sets the key of the receiver.

- (void)setKey:(NSString *)*key*

Parameters*key*

The key for the receiver.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [key](#) (page 13)
- [keyClassDescription](#) (page 13)

Declared In

NSScriptObjectSpecifiers.h

Constants

NSScriptObjectSpecifier—Specifier Evaluation Errors

`NSScriptObjectSpecifier` provides the following constants for error codes for specific problems evaluating specifiers:

```
enum {
    NSNoSpecifierError = 0,
    NSNoTopLevelContainersSpecifierError,
    NSContainerSpecifierError,
    NSUnknownKeySpecifierError,
    NSInvalidIndexSpecifierError,
    NSInternalSpecifierError,
    NSOperationNotSupportedForKeySpecifierError
};
```

Constants`NSNoSpecifierError`

No error encountered.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.`NSNoTopLevelContainersSpecifierError`Someone called `evaluate` with `nil`.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.`NSContainerSpecifierError`

Error evaluating container specifier.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.`NSUnknownKeySpecifierError`

Receivers do not understand the key.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.

`NSInvalidIndexSpecifierError`

Index out of bounds.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.

`NSInternalSpecifierError`

Other internal error.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.

`NSOperationNotSupportedForKeySpecifierError`

Attempt made to perform an unsupported operation on some key.

Available in Mac OS X v10.0 and later.

Declared in `NSScriptObjectSpecifiers.h`.

Declared In

`NSScriptObjectSpecifier.h`

Document Revision History

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

Date	Notes
2007-06-29	Added new methods for Mac OS X version 10.5.
	The new methods are descriptor (page 10) and objectSpecifierWithDescriptor: (page 8).
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

C

childSpecifier **instance method** 8
containerClassDescription **instance method** 9
containerIsObjectBeingTested **instance method** 9
containerIsRangeContainerObject **instance method** 10
containerSpecifier **instance method** 10

D

descriptor **instance method** 10

E

evaluationErrorNumber **instance method** 11
evaluationErrorSpecifier **instance method** 11

I

indicesOfObjectsByEvaluatingWithContainer:count: **instance method** 12
initWithContainerClassDescription:
 containerSpecifier:key: **instance method** 12
initWithContainerSpecifier:key: **instance method** 12

K

key **instance method** 13
keyClassDescription **instance method** 13

N

NSContainerSpecifierError **constant** 18
NSInternalSpecifierError **constant** 19
NSInvalidIndexSpecifierError **constant** 19
NSNoSpecifierError **constant** 18
NSNoTopLevelContainersSpecifierError **constant** 18
NSOperationNotSupportedForKeySpecifierError **constant** 19
NSScriptObjectSpecifier—Specifier Evaluation Errors 18
NSUnknownKeySpecifierError **constant** 18

O

objectsByEvaluatingSpecifier **instance method** 14
objectsByEvaluatingWithContainers: **instance method** 14
objectSpecifierWithDescriptor: **class method** 8

S

setChildSpecifier: **instance method** 15
setContainerClassDescription: **instance method** 15
setContainerIsObjectBeingTested: **instance method** 16
setContainerIsRangeContainerObject: **instance method** 16
setContainerSpecifier: **instance method** 17
setEvaluationErrorNumber: **instance method** 17
setKey: **instance method** 17