
NSComparisonMethods Protocol Reference





Apple Inc.
© 2009 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, 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

NSComparisonMethods Protocol Reference 5

Overview	5
Tasks	5
Performing Comparisons	5
Instance Methods	6
doesContain:	6
isCaseInsensitiveLike:	6
isEqualTo:	7
isGreaterThan:	7
isGreaterThanOrEqualTo:	8
isLessThan:	8
isLessThanOrEqualTo:	9
isLike:	10
isNotEqualTo:	10

Document Revision History 11

Index 13

NSComparisonMethods Protocol Reference

(informal protocol)

Framework	/System/Library/Frameworks/Foundation.framework
Companion guide	Cocoa Scripting Guide
Declared in	NSScriptWhoseTests.h

Overview

This informal protocol defines a set of default comparison methods useful for the comparisons in `NSSpecifierTest`.

If you have scriptable objects that need to perform comparisons for scripting purposes, you may need to implement some of the methods declared in `NSScriptingComparisonMethods`. The default implementation provided for many of these methods by `NSObject` is appropriate for objects that implement a single comparison method whose selector, signature, and description match the following:

- (NSComparisonResult)compare:(id)object;

This method should return `NSOrderedAscending` if the receiver is less than *object*, `NSOrderedDescending` if the receiver is greater than *object*, and `NSOrderedSame` if the receiver and *object* are equal. For example, `NSString` does not implement most of the methods declared in this informal protocol, but `NSString` objects still handle messages conforming to this protocol properly because `NSString` implements a `compare:` method that meets the necessary requirements. Cocoa also includes appropriate `compare:` method implementations for the `NSDate`, `NSDecimalNumber`, and `NSNumber` classes.

Tasks

Performing Comparisons

- [doesContain:](#) (page 6)
Returns a Boolean value that indicates whether the receiver contains a given object.
- [isCaseInsensitiveLike:](#) (page 6)
Returns a Boolean value that indicates whether receiver is considered to be “like” a given string when the case of characters in the receiver is ignored.
- [isEqualTo:](#) (page 7)
Returns a Boolean value that indicates whether the receiver is equal to another given object.
- [isGreaterThan:](#) (page 7)
Returns a Boolean value that indicates whether the receiver is greater than another given object.

- [isGreaterThanOrEqualTo:](#) (page 8)
Returns a Boolean value that indicates whether the receiver is greater than or equal to another given object.
- [isLessThan:](#) (page 8)
Returns a Boolean value that indicates whether the receiver is less than another given object.
- [isLessThanOrEqualTo:](#) (page 9)
Returns a Boolean value that indicates whether the receiver is less than or equal to another given object.
- [isLike:](#) (page 10)
Returns a Boolean value that indicates whether the receiver is "like" another given object.
- [isNotEqualTo:](#) (page 10)
Returns a Boolean value that indicates whether the receiver is not equal to another given object.

Instance Methods

doesContain:

Returns a Boolean value that indicates whether the receiver contains a given object.

- (BOOL)doesContain:(id)object

Parameters

object

The object to search for in the receiver.

Return Value

YES if the receiver contains *object*, otherwise NO.

Discussion

Currently, `doesContain:` messages are never sent to any object from within Cocoa itself.

The default implementation for this method provided by `NSObject` returns YES if the receiver is actually an `NSArray` object and an `indexOfObjectIdenticalTo:` message sent to the same object would return something other than `NSNotFound`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isCaseInsensitiveLike:

Returns a Boolean value that indicates whether receiver is considered to be "like" a given string when the case of characters in the receiver is ignored.

- (BOOL)isCaseInsensitiveLike:(NSString *)aString

Parameters*aString*

The string with which to compare the receiver.

Return Value

YES if the receiver is considered to be “like” *aString* when the case of characters in the receiver is ignored, otherwise NO.

Discussion

Currently, `isCaseInsensitiveLike:` messages are never sent to any object from within Cocoa itself.

The default implementation for this method provided by `NSObject` returns NO. `NSString` also provides an implementation of this method, which returns YES if the receiver matches a pattern described by *aString*, ignoring the case of the characters in the receiver.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isEqualTo:

Returns a Boolean value that indicates whether the receiver is equal to another given object.

- (BOOL)isEqualTo:(id)object

Parameters*object*

The object with which to compare the receiver.

Return Value

YES if the receiver is equal to *object*, otherwise NO. In effect returns NO if receiver is nil.

Discussion

During the evaluation of an `NSWhoseSpecifier` object that contains a test whose operator is `NSEqualToComparison`, an `isEqualTo:` message may be sent to each potentially specified object, if neither the potentially specified object nor the object being tested against implements a `scriptingIsEqualTo:` method.

The default implementation for this method provided by `NSObject` returns YES if an `isEqualTo:` message sent to the same object would return YES.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isGreaterThan:

Returns a Boolean value that indicates whether the receiver is greater than another given object.

- (BOOL)isGreaterThan:(id)object

Parameters*object*

The object with which to compare the receiver.

Return ValueYES if the receiver is greater than *object*, otherwise NO.**Discussion**

During the evaluation of an `NSWhoseSpecifier` object that contains a test whose operator is `NSGreaterThanComparison`, an `isGreaterThan:` message may be sent to each potentially specified object, if the potentially specified object does not implement a `scriptingIsGreaterThan:` method and the object being tested against does not implement a `scriptingIsLessThanOrEqualTo:` method.

The default implementation for this method provided by `NSObject` returns YES if a `compare:` message sent to the same object would return `NSOrderedDescending`.

Availability

Available in Mac OS X v10.0 and later.

Declared In`NSScriptWhoseTests.h`**isGreaterThanOrEqualTo:**

Returns a Boolean value that indicates whether the receiver is greater than or equal to another given object.

- (BOOL)isGreaterThanOrEqualTo:(id) *object***Parameters***object*

The object with which to compare the receiver.

Return ValueYES if the receiver is greater than or equal to *object*, otherwise NO.**Discussion**

During the evaluation of an `NSWhoseSpecifier` object that contains a test whose operator is `NSGreaterThanOrEqualToComparison`, an `isGreaterThanOrEqualTo:` message may be sent to each potentially specified object, if the potentially specified object does not implement a `scriptingIsGreaterThanOrEqualTo:` method and the object being tested against does not implement a `scriptingIsLessThan:` method.

The default implementation for this method provided by `NSObject` returns YES if a `compare:` message sent to the same object would return `NSOrderedSame` or `NSOrderedDescending`.

Availability

Available in Mac OS X v10.0 and later.

Declared In`NSScriptWhoseTests.h`**isLessThan:**

Returns a Boolean value that indicates whether the receiver is less than another given object.

- (BOOL)isLessThan:(id)object

Parameters

object

The object with which to compare the receiver.

Return Value

YES if the receiver is less than *object*, otherwise NO.

Discussion

During the evaluation of an `NSWhoseSpecifier` object that contains a test whose operator is `NSLessThanComparison`, an `isLessThan:` message may be sent to each potentially specified object, if the potentially specified object does not implement a `scriptingIsLessThan:` method and the object being tested against does not implement a `scriptingIsGreaterThanOrEqualTo:` method.

The default implementation for this method provided by `NSObject` method returns YES if a `compare:` message sent to the same object would return `NSOrderedAscending`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isLessThanOrEqualTo:

Returns a Boolean value that indicates whether the receiver is less than or equal to another given object.

- (BOOL)isLessThanOrEqualTo:(id)object

Parameters

object

The object with which to compare the receiver.

Return Value

YES if the receiver is less than or equal to *object*, otherwise NO.

Discussion

During the evaluation of an `NSWhoseSpecifier` object that contains a test whose operator is `NSLessThanOrEqualToComparison`, an `isLessThanOrEqualTo:` message may be sent to each potentially specified object, if the potentially specified object does not implement a `scriptingIsLessThanOrEqualTo:` method and the object being tested against does not implement a `scriptingIsGreaterThan:` method.

The default implementation for this method provided by `NSObject` method returns YES if a `compare:` message sent to the same object would return `NSOrderedAscending` or `NSOrderedSame`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isLike:

Returns a Boolean value that indicates whether the receiver is "like" another given object.

- (BOOL)isLike:(NSString *)*object*

Parameters

object

The object with which to compare the receiver.

Return Value

YES if the receiver is considered to be "like" *object*, otherwise NO.

Discussion

Currently, `isLike:` messages are never sent to any object from within Cocoa itself.

The default implementation for this method provided by `NSObject` method returns NO. `NSString` also provides an implementation of this method, which returns YES if the receiver matches a pattern described by *object*.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

isNotEqualTo:

Returns a Boolean value that indicates whether the receiver is not equal to another given object.

- (BOOL)isNotEqualTo:(id)*object*

Parameters

object

The object with which to compare the receiver.

Return Value

YES if the receiver is not equal to *object*, otherwise NO.

Discussion

Currently, `isNotEqualTo:` messages are never sent to any object from within Cocoa itself.

The default implementation for this method provided by `NSObject` method returns YES if an `isEqual:` message sent to the same object would return NO.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptWhoseTests.h`

Document Revision History

This table describes the changes to *NSComparisonMethods Protocol Reference*.

Date	Notes
2009-11-17	Updated status optional of protocol methods.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

D

contains: <NSObject> instance method [6](#)

I

isCaseInsensitiveLike: <NSObject> instance method [6](#)

isEqualTo: <NSObject> instance method [7](#)

isGreaterThan: <NSObject> instance method [7](#)

isGreaterThanOrEqualTo: <NSObject> instance method [8](#)

isLessThan: <NSObject> instance method [8](#)

isLessThanOrEqualTo: <NSObject> instance method [9](#)

isLike: <NSObject> instance method [10](#)

isNotEqualTo: <NSObject> instance method [10](#)