
NSScriptExecutionContext Class Reference

[Cocoa](#) > [Scripting & Automation](#)



2006-05-23



Apple Inc.
© 2006 Apple Computer, 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, Carbon, 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

NSScriptExecutionContext Class Reference 5

Overview	5
Tasks	5
Getting the Current Context	5
Getting and Setting the Container Object	6
Class Methods	6
sharedScriptExecutionContext	6
Instance Methods	6
objectBeingTested	6
rangeContainerObject	7
setObjectBeingTested:	7
setRangeContainerObject:	8
setTopLevelObject:	8
topLevelObject	8

Document Revision History 11

Index 13

NSScriptExecutionContext Class Reference

Inherits from	NSObject
Conforms to	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	NSScriptExecutionContext.h
Related sample code	SimpleCarbonAppleScript

Overview

An `NSScriptExecutionContext` object is a shared instance (there is only one instance of the class) that represents the context in which the current script command is executed. `NSScriptExecutionContext` tracks global state relating to the command being executed, especially the top-level container object (that is, the container implied by a specifier object that specifies no container) used in an evaluation of an `NSScriptObjectSpecifier` object.

In most cases, the top-level container for a complete series of nested object specifiers is automatically set to the application object (`NSApp`), and you can get this object with the `topLevelObject` (page 8) method. But you can also set this top-level container to something else (using `setTopLevelObject:` (page 8)) if the situation warrants it.

It is unlikely that you will need to subclass `NSScriptExecutionContext`.

Tasks

Getting the Current Context

- + [sharedScriptExecutionContext](#) (page 6)
Returns the shared `NSScriptExecutionContext` instance.

Getting and Setting the Container Object

- [topLevelObject](#) (page 8)
Returns the top-level object for an object-specifier evaluation.
- [setTopLevelObject:](#) (page 8)
Sets the top-level object for an object-specifier evaluation.
- [objectBeingTested](#) (page 6)
Returns the top-level container object currently being tested in a “whose” qualifier.
- [setObjectBeingTested:](#) (page 7)
Sets the top-level container object currently being tested in a “whose” qualifier to a given object.
- [rangeContainerObject](#) (page 7)
Returns the top-level container object for an object specifier (encapsulated in an `NSRangeSpecifier` object) that represents the first or last element in a range of elements.
- [setRangeContainerObject:](#) (page 8)
Sets the top-level container object for a range-specifier evaluation to a give object.

Class Methods

sharedScriptExecutionContext

Returns the shared `NSScriptExecutionContext` instance.

```
+ (NSScriptExecutionContext *)sharedScriptExecutionContext
```

Return Value

The shared `NSScriptExecutionContext` instance, creating it first if it doesn't exist.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

SimpleCarbonAppleScript

Declared In

`NSScriptExecutionContext.h`

Instance Methods

objectBeingTested

Returns the top-level container object currently being tested in a “whose” qualifier.

```
- (id)objectBeingTested
```

Return Value

The top-level container object currently being tested in a “whose” qualifier. Returns `nil` if such an object does not exist.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setObjectBeingTested:](#) (page 7)

`containerIsObjectBeingTested` (NSScriptObjectSpecifier)

Declared In

NSScriptExecutionContext.h

rangeContainerObject

Returns the top-level container object for an object specifier (encapsulated in an `NSRangeSpecifier` object) that represents the first or last element in a range of elements.

- (id)rangeContainerObject

Return Value

The top-level container object for an object specifier (encapsulated in an `NSRangeSpecifier` object) that represents the first or last element in a range of elements.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setObjectBeingTested:](#) (page 7)

`containerIsRangeContainerObject` (NSScriptObjectSpecifier)

Declared In

NSScriptExecutionContext.h

setObjectBeingTested:

Sets the top-level container object currently being tested in a “whose” qualifier to a given object.

- (void)setObjectBeingTested:(id)object

Parameters

object

The top-level container object currently being tested.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [objectBeingTested](#) (page 6)

Declared In

NSScriptExecutionContext.h

setRangeContainerObject:

Sets the top-level container object for a range-specifier evaluation to a give object.

```
- (void)setRangeContainerObject:(id)container
```

Parameters

container

The top-level container object for a range-specifier evaluation.

Discussion

Instances of `NSRangeSpecifier` contain object specifiers representing the first or last element in a range of elements, and these specifiers are evaluated in the context of *container*.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [rangeContainerObject](#) (page 7)

Declared In

`NSScriptExecutionContext.h`

setTopLevelObject:

Sets the top-level object for an object-specifier evaluation.

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

Parameters

anObject

The top-level object for an object-specifier evaluation.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [topLevelObject](#) (page 8)

Related Sample Code

`SimpleCarbonAppleScript`

Declared In

`NSScriptExecutionContext.h`

topLevelObject

Returns the top-level object for an object-specifier evaluation.

```
- (id)topLevelObject
```

Return Value

The top-level object for an object-specifier evaluation.

Discussion

For applications, this object is automatically set to the application object, but can be set to some other container object.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTopLevelObject:](#) (page 8)

Declared In

NSScriptExecutionContext.h

Document Revision History

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

Date	Notes
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

O

objectBeingTested **instance method** [6](#)

R

rangeContainerObject **instance method** [7](#)

S

setObjectBeingTested: **instance method** [7](#)
setRangeContainerObject: **instance method** [8](#)
setTopLevelObject: **instance method** [8](#)
sharedScriptExecutionContext **class method** [6](#)

T

topLevelObject **instance method** [8](#)