
NSCreateCommand 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, 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

NSCreateCommand Class Reference 5

Overview 5

Tasks 6

 Getting Information About a Create Command 6

Instance Methods 6

 createClassDescription 6

 resolvedKeyDictionary 6

Document Revision History 9

Index 11

NSCreateCommand Class Reference

Inherits from	NSScriptCommand : NSObject
Conforms to	NSCoding (NSScriptCommand) 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	NSScriptStandardSuiteCommands.h

Overview

An instance of `NSCreateCommand` creates the specified scriptable object (such as a document), optionally supplying the new object with the specified attributes. This command corresponds to AppleScript's `make` command.

`NSCreateCommand` is part of Cocoa's built-in scripting support. Most applications don't need to subclass `NSCreateCommand` or invoke its methods.

When an instance of `NSCreateCommand` is executed, it creates a new object using `[[theClassToBeCreated allocWithZone:NULL] init]` (where `theClassToBeCreated` is the class of the object to be created), unless the command has a `with data` argument. In the latter case, the new object is created by invoking `[[NSScriptCoercionHandler sharedCoercionHandler] coerceValue:theDataAsAnObject toClass:theClassToBeCreated]`. Any properties specified by a `with properties` argument are then set in the new object using `-setScriptingProperties:`.

If an `NSCreateCommand` object with no argument corresponding to the `at` parameter is executed (for example, `tell application "Mail" to make new mailbox with properties {name:"testFolder"};`), and the receiver of the command (not necessarily the application object) has a to-many relationship to objects of the class to be instantiated, and the class description for the receiving class returns `NO` when sent an `isLocationRequiredToCreateForKey: message`, the `NSCreateCommand` object creates a new object and sends the receiver an `insertValue:atIndex:inPropertyWithKey:` message to place the new object in the container. This is part of Cocoa's scripting support for inserting newly-created objects into containers without explicitly specifying a location.

Tasks

Getting Information About a Create Command

- [createClassDescription](#) (page 6)
Returns the class description for the class that is to be created.
- [resolvedKeyDictionary](#) (page 6)
Returns a dictionary that contains the properties that were specified in the `make` Apple event command that has been converted to this `NSCreateCommand` object.

Instance Methods

`createClassDescription`

Returns the class description for the class that is to be created.

```
-(NSScriptClassDescription *)createClassDescription
```

Return Value

The class description for the class that is to be created.

Availability

Available in Mac OS X v10.0 and later.

Declared In

`NSScriptStandardSuiteCommands.h`

`resolvedKeyDictionary`

Returns a dictionary that contains the properties that were specified in the `make` Apple event command that has been converted to this `NSCreateCommand` object.

```
-(NSDictionary *)resolvedKeyDictionary
```

Return Value

A dictionary that contains the properties that were specified in the `make` Apple event script command that has been converted to this `NSCreateCommand` object.

Discussion

The keys in the returned dictionary are the names of properties (attributes or relationships, in the script suite) that have been specified for the command, and the corresponding values in the dictionary are the values that those properties should take. The required and optional arguments for the `make` command are specified in the core suite definition, `NSCoreSuite.scriptSuite`.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSScriptStandardSuiteCommands.h

Document Revision History

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

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

REVISION HISTORY

Document Revision History

Index

C

`createClassDescription` **instance method** [6](#)

R

`resolvedKeyDictionary` **instance method** [6](#)