

---

# NSOpenPanel Class Reference

Data Management: File Management



2009-02-19



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, Mac OS, Quartz, and QuickTime are trademarks of Apple Inc., registered in the United States and other countries.

Shuffle is a trademark of Apple Inc.

OpenGL is a registered trademark of Silicon Graphics, Inc.

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

---

## **NSOpenPanel Class Reference 5**

---

- Overview 5
- Tasks 5
  - Creating Panels 5
  - Configuring Panels 6
  - Running Panels 6
  - Accessing User Selection 7
- Class Methods 7
  - openPanel 7
- Instance Methods 7
  - allowsMultipleSelection 7
  - canChooseDirectories 8
  - canChooseFiles 8
  - resolvesAliases 8
  - setAllowsMultipleSelection: 9
  - setCanChooseDirectories: 9
  - setCanChooseFiles: 10
  - setResolvesAliases: 10
- URLs 11

---

## **Appendix A    Deprecated NSOpenPanel Methods 13**

---

- Deprecated in Mac OS X v10.6 13
  - beginForDirectory:file:types:modelessDelegate:didEndSelector:contextInfo: 13
  - beginSheetForDirectory:file:types:modalForWindow:modalDelegate:didEndSelector: contextInfo: 14
  - filenames 15
  - runModalForDirectory:file:types: 16
  - runModalForTypes: 17

---

## **Document Revision History 19**

---

## **Index 21**

---



# NSOpenPanel Class Reference

---

<b>Inherits from</b>	NSSavePanel : NSPanel : NSWindow : NSResponder : NSObject
<b>Conforms to</b>	NSUserInterfaceValidations (NSWindow) NSAnimatablePropertyContainer (NSWindow) NSCoding (NSResponder) NSObject (NSObject)
<b>Framework</b>	/System/Library/Frameworks/AppKit.framework
<b>Availability</b>	Available in Mac OS X v10.0 and later.
<b>Declared in</b>	NSOpenPanel.h
<b>Companion guides</b>	Application File Management Sheet Programming Topics for Cocoa
<b>Related sample code</b>	ExtractMovieAudioToAIFF FunHouse ImageKitDemo LSMSmartCategorizer QTEExtractAndConvertToAIFF

## Overview

The `NSOpenPanel` class provides the Open panel for the Cocoa user interface. Applications use the Open panel as a convenient way to query the user for the name of a file to open.

## Tasks

### Creating Panels

+ [openPanel](#) (page 7)

Creates and returns a [NSOpenPanel](#) (page 5) object.

## Configuring Panels

- [canChooseFiles](#) (page 8)  
Returns whether the panel allows the user to choose files to open.
- [setCanChooseFiles:](#) (page 10)  
Sets whether the user can select files in the panel's browser.
- [canChooseDirectories](#) (page 8)  
Returns whether the panel allows the user to choose directories to open.
- [setCanChooseDirectories:](#) (page 9)  
Sets whether the user can select directories in the panel's browser.
- [resolvesAliases](#) (page 8)  
Returns whether the panel resolves aliases.
- [setResolvesAliases:](#) (page 10)  
Sets whether the panel resolves aliases.
- [allowsMultipleSelection](#) (page 7)  
Returns whether the panel's browser allows the user to open multiple files (and directories) at a time.
- [setAllowsMultipleSelection:](#) (page 9)  
Sets whether the user can select multiple files (and directories) at one time for opening.

## Running Panels

- [beginForDirectory:file:types:modellessDelegate:didEndSelector:contextInfo:](#) (page 13) **Deprecated in Mac OS X v10.6**  
Presents a modeless Open panel. (**Deprecated.** Use `beginWithCompletionHandler:` instead. You can set `absoluteDirectoryPath` using `setDirectoryURL:`, and you can set `fileTypes` using `setAllowedFileTypes:`.)
- [beginSheetForDirectory:file:types:modalForWindow:modalDelegate:didEndSelector:contextInfo:](#) (page 14) **Deprecated in Mac OS X v10.6**  
Presents an Open panel as a sheet with the directory specified by `absoluteDirectoryPath` and optionally the file specified by `filename` selected. (**Deprecated.** Use `beginSheetModalForWindow:completionHandler:` instead. You can set `absoluteDirectoryPath` using `setDirectoryURL:`, and you can set `fileTypes` using `setAllowedFileTypes:`.)
- [runModalForDirectory:file:types:](#) (page 16) **Deprecated in Mac OS X v10.6**  
Displays the panel and begins a modal event loop that is terminated when the user clicks either OK or Cancel. (**Deprecated.** Use `runModal` instead. You can set `path` using `setDirectoryURL:`, and you can set `fileTypes` using `setAllowedFileTypes:`.)
- [runModalForTypes:](#) (page 17) **Deprecated in Mac OS X v10.6**  
Displays the panel and begins a modal event loop that is terminated when the user clicks either OK or Cancel. (**Deprecated.** Use `runModal` instead. You can set `fileTypes` using `setAllowedFileTypes:`.)

## Accessing User Selection

- [URLs](#) (page 11)  
Returns an array containing the absolute paths of the selected files and directories as URLs.
- [filenames](#) (page 15) **Deprecated in Mac OS X v10.6**  
Returns an array containing the absolute paths (as NSString objects) of the selected files and directories. (**Deprecated**. Use [URLs](#) (page 11) instead.)

## Class Methods

### openPanel

Creates and returns a [NSOpenPanel](#) (page 5) object.

```
+ (NSOpenPanel *)openPanel
```

#### Return Value

The initialized Open panel.

#### Availability

Available in Mac OS X v10.0 and later.

#### Related Sample Code

ContentBurn

FunHouse

ImageKitDemo

LSMSmartCategorizer

Quartz Composer WWDC 2005 TextEdit

#### Declared In

NSOpenPanel.h

## Instance Methods

### allowsMultipleSelection

Returns whether the panel's browser allows the user to open multiple files (and directories) at a time.

```
- (BOOL)allowsMultipleSelection
```

#### Return Value

YES if the panel's browser allows multiple selection; otherwise, NO.

#### Discussion

If multiple files or directories are allowed, then the `filename` method—inherited from `NSSavePanel`—returns a non-`nil` value only if one and only one file is selected. By contrast, `NSOpenPanel`'s [URLs](#) (page 11) method always returns the URLs of the selected files, even if only one file is selected.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [URL \(NSSavePanel\)](#)
- [URLs \(page 11\)](#)
- [setAllowsMultipleSelection: \(page 9\)](#)

### Declared In

NSOpenPanel.h

## canChooseDirectories

Returns whether the panel allows the user to choose directories to open.

- (BOOL)canChooseDirectories

### Return Value

YES if the panel allows the user to choose directories; otherwise, NO.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setCanChooseDirectories: \(page 9\)](#)

### Declared In

NSOpenPanel.h

## canChooseFiles

Returns whether the panel allows the user to choose files to open.

- (BOOL)canChooseFiles

### Return Value

YES if the panel allows the user to choose files; otherwise, NO.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [setCanChooseFiles: \(page 10\)](#)

### Declared In

NSOpenPanel.h

## resolvesAliases

Returns whether the panel resolves aliases.

- (BOOL)resolvesAliases

### Return Value

YES if the panel resolves aliases; otherwise, NO.

### Discussion

If YES, the effect is that dropping an alias on the panel or asking for filenames or URLs returns the resolved aliases. The default is YES.

### Availability

Available in Mac OS X v10.1 and later.

### See Also

- [setResolvesAliases:](#) (page 10)

### Declared In

NSOpenPanel.h

## setAllowsMultipleSelection:

Sets whether the user can select multiple files (and directories) at one time for opening.

```
- (void)setAllowsMultipleSelection:(BOOL)flag
```

### Parameters

*flag*

If YES, the panel's browser allows multiple selection; if NO, it does not.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [allowsMultipleSelection](#) (page 7)

### Related Sample Code

AudioBurn

ContentBurn

FunHouse

ObjectPath

Quartz Composer WWDC 2005 TextEdit

### Declared In

NSOpenPanel.h

## setCanChooseDirectories:

Sets whether the user can select directories in the panel's browser.

```
- (void)setCanChooseDirectories:(BOOL)flag
```

### Parameters

*flag*

If YES, the panel allows the user to choose directories; if NO, it does not.

### Discussion

When a directory is selected, the OK button is enabled only if *flag* is YES.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [canChooseDirectories](#) (page 8)

### Related Sample Code

ContentBurn

ImageKitDemo

LSMSmartCategorizer

MovieAssembler

ObjectPath

### Declared In

NSOpenPanel.h

## setCanChooseFiles:

Sets whether the user can select files in the panel's browser.

```
- (void)setCanChooseFiles:(BOOL)flag
```

### Parameters

*flag*

If YES, the panel allows the user to choose files; if NO, it does not.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

- [canChooseFiles](#) (page 8)

### Related Sample Code

ContentBurn

ImageKitDemo

LSMSmartCategorizer

MovieAssembler

ObjectPath

### Declared In

NSOpenPanel.h

## setResolvesAliases:

Sets whether the panel resolves aliases.

```
- (void)setResolvesAliases:(BOOL)resolvesAliases
```

### Parameters

*resolvesAliases*

If YES, the panel resolves aliases; if NO, it does not.

### Discussion

If YES, the effect is that dropping an alias on the panel or asking for filenames or URLs returns the resolved aliases. Set this value to NO to allow selection of aliases without resolving.

### Availability

Available in Mac OS X v10.1 and later.

### See Also

- [resolvesAliases](#) (page 8)

### Related Sample Code

CIVideoDemoGL

DesktopImage

FunHouse

ObjectPath

Quartz 2D Transformer

### Declared In

NSOpenPanel.h

## URLs

Returns an array containing the absolute paths of the selected files and directories as URLs.

- (NSArray \*)URLs

### Return Value

The array of URLs.

### Discussion

If multiple selections aren't allowed, the array contains a single name.

### Availability

Available in Mac OS X v10.0 and later.

### Related Sample Code

BackgroundExporter

CIAnnotation

CocoaSpeechSynthesisExample

UIKitFrameStepper

SBSetsFinderComment

### Declared In

NSOpenPanel.h



# Deprecated NSOpenPanel Methods

A method identified as deprecated has been superseded and may become unsupported in the future.

## Deprecated in Mac OS X v10.6

### **beginForDirectory:file:types:modelessDelegate:didEndSelector:contextInfo:**

Presents a modeless Open panel. (**Deprecated in Mac OS X v10.6.** Use `beginWithCompletionHandler:` instead. You can set `absoluteDirectoryPath` using `setDirectoryURL:`, and you can set `fileTypes` using `setAllowedFileTypes:`)

```
- (void)beginForDirectory:(NSString *)absoluteDirectoryPath file:(NSString *)filename
    types:(NSArray *)fileTypes modelessDelegate:(id)modelessDelegate
    didEndSelector:(SEL)didEndSelector contextInfo:(void *)contextInfo
```

#### Parameters

*absoluteDirectoryPath*

The directory whose files the panel displays. When `nil`, the directory is the same directory used in the previous invocation of the panel; this is probably the best choice for most situations.

*filename*

Specifies a particular file in *absoluteDirectoryPath* that is selected when the Open panel is presented to the user. When `nil`, no file is initially selected.

*fileTypes*

An array of file extensions and/or HFS file types. Specifies the files the panel allows the user to select. `nil` makes all files in *absoluteDirectoryPath* selectable by the user. An array of types passed in here will override one set using `setAllowedFileTypes:`.

*modelessDelegate*

This is not the same as a delegate assigned to the panel. This delegate is temporary and the relationship only lasts until the panel is dismissed.

*didEndSelector*

The message sent to *modelessDelegate* after the panel's session has ended, but before dismissing the Open panel. *didEndSelector* may dismiss the Open panel itself; otherwise, it will be dismissed on return from the method. The corresponding method should have the following signature:

```
- (void)openPanelDidEnd:(NSOpenPanel *)panel returnCode:(int)returnCode
    contextInfo:(void *)contextInfo
```

The value passed as *returnCode* will be either `NSCancelButton` or `NSOKButton`.

*contextInfo*

Any context information passed to *modelessDelegate* in the *didEndSelector* message.

**Discussion**

Similar to

[beginSheetForDirectory:file:types:modalForWindow:modalDelegate:didEndSelector:contextInfo:](#) (page 14), but allows for modeless operation of the panel.

**Availability**

Available in Mac OS X v10.3 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- [beginWithCompletionHandler:](#)

**Declared In**

NSOpenPanel.h

**beginSheetForDirectory:file:types:modalForWindow:modalDelegate:didEndSelector:contextInfo:**

Presents an Open panel as a sheet with the directory specified by *absoluteDirectoryPath* and optionally the file specified by *filename* selected. (Deprecated in Mac OS X v10.6. Use [beginSheetModalForWindow:completionHandler:](#) instead. You can set *absoluteDirectoryPath* using [setDirectoryURL:](#), and you can set *fileTypes* using [setAllowedFileTypes:](#))

```
- (void)beginSheetForDirectory:(NSString *)absoluteDirectoryPath file:(NSString *)filename types:(NSArray *)fileTypes modalForWindow:(NSWindow *)docWindow modalDelegate:(id)modalDelegate didEndSelector:(SEL)didEndSelector contextInfo:(void *)contextInfo
```

**Parameters**

*absoluteDirectoryPath*

The directory whose files the panel displays. When *nil*, the directory is the same directory used in the previous invocation of the panel; this is probably the best choice for most situations.

*filename*

Specifies a particular file in *absoluteDirectoryPath* that is selected when the Open panel is presented to the user. When *nil*, no file is initially selected.

*fileTypes*

An array of file extensions and/or HFS file types. Specifies the files the panel allows the user to select. *nil* makes all files in *absoluteDirectoryPath* selectable by the user. An array of types passed in here will override one set using [setAllowedFileTypes:](#).

*docWindow*

The window to open the sheet on.

*modalDelegate*

This is not the same as a delegate assigned to the panel. This delegate is temporary and the relationship only lasts until the panel is dismissed..

## Deprecated NSOpenPanel Methods

*didEndSelector*

The message sent to *modalDelegate* after the modal session has ended, but before dismissing the Open panel. *didEndSelector* may dismiss the Open panel itself; otherwise, it will be dismissed on return from the method. The corresponding method should have the following signature:

```
- (void)openPanelDidEnd:(NSOpenPanel *)panel returnCode:(int)returnCode
contextInfo:(void *)contextInfo
```

The value passed as *returnCode* will be either `NSCancelButton` or `NSOKButton`.

*contextInfo*

Any context information passed to *modalDelegate* in the *didEndSelector* message.

**Availability**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- `beginSheetModalForWindow:completionHandler:`

**Related Sample Code**

DemoAssistant

GridCalendar

UIImageViewDemo

JSheets

SillyFrequencyLevels

**Declared In**

`NSOpenPanel.h`

**filenames**

Returns an array containing the absolute paths (as `NSString` objects) of the selected files and directories. (Deprecated in Mac OS X v10.6. Use [URLs](#) (page 11) instead.)

```
- (NSArray *)filenames
```

**Return Value**

The array of filenames.

**Discussion**

If multiple selections aren't allowed, the array contains a single name. The `filenames` method is preferable over `NSSavePanel's filename` to get the name or names of files and directories that the user has selected.

**Availability**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- [URLs](#) (page 11)

**Related Sample Code**

FunHouse

ImageKitDemo

LSMSmartCategorizer  
 ThreadsImporter  
 ThreadsImportMovie

**Declared In**

NSOpenPanel.h

**runModalForDirectory:file:types:**

Displays the panel and begins a modal event loop that is terminated when the user clicks either OK or Cancel. (Deprecated in Mac OS X v10.6. Use `runModal` instead. You can set `path` using `setDirectoryURL:`, and you can set `fileTypes` using `setAllowedFileTypes:`.)

```
- (NSInteger)runModalForDirectory:(NSString *)absoluteDirectoryPath file:(NSString *)filename types:(NSArray *)fileTypes
```

**Parameters**

*absoluteDirectoryPath*

The directory whose files the panel displays. When `nil`, the directory is the same directory used in the previous invocation of the panel; this is probably the best choice for most situations.

*filename*

Specifies a particular file in *absoluteDirectoryPath* that is selected when the Open panel is presented to the user. When `nil`, no file is initially selected.

*fileTypes*

An array of file extensions and/or HFS file types. Specifies the files the panel allows the user to select. `nil` makes all files in *absoluteDirectoryPath* selectable by the user. An array of types passed in here will override one set using `setAllowedFileTypes:`.

**Return Value**

The button clicked to dismiss the dialog: `NSOKButton` for the OK button and `NSCancelButton` for the Cancel button.

**Discussion**

You can control whether directories and files appear in the browser with the [setCanChooseDirectories:](#) (page 9) and [setCanChooseFiles:](#) (page 10) methods.

**Availability**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- `runModal`

**Related Sample Code**

LiveVideoMixer  
 LiveVideoMixer2  
 MatrixMixerTest  
 QTGraphicsImport  
 SBSetsFinderComment

**Declared In**

NSOpenPanel.h

## runModalForTypes:

Displays the panel and begins a modal event loop that is terminated when the user clicks either OK or Cancel. (Deprecated in Mac OS X v10.6. Use `runModal` instead. You can set `fileTypes` using `setAllowedFileTypes:`.)

```
- (NSInteger)runModalForTypes:(NSArray *)fileTypes
```

### Parameters

*fileTypes*

An array of file extensions and/or HFS file types. Specifies the files the panel allows the user to select. `nil` makes all files selectable by the user. An array of types passed in here will override one set using `setAllowedFileTypes:`.

### Return Value

The button used to dismiss the dialog: `NSOKButton` for the OK button and `NSCancelButton` for the Cancel button.

### Discussion

This convenience method sends `runModalForDirectory:nil file:nil types:fileTypes` to the panel. See [runModalForDirectory:file:types:](#) (page 16) for additional details.

### Availability

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

### See Also

- `runModal`

### Related Sample Code

[CIVideoDemoGL](#)

[FunHouse](#)

[ImageKitDemo](#)

[ThreadsImporter](#)

[ThreadsImportMovie](#)

### Declared In

`NSOpenPanel.h`



# Document Revision History

---

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

Date	Notes
2009-02-19	Updated for Mac OS X v10.6.
2009-02-04	Added more specific information regarding <code>didEndSelector</code> behavior in panel-running methods.
2007-04-03	Removed documentation for methods not present in the corresponding header file.
	Removed <code>allowedFileTypes</code> and <code>setAllowedFileTypes:</code> .
	Made editorial improvements.
2006-05-23	First publication of this content as a separate document.

**REVISION HISTORY**

Document Revision History

# Index

---

## A

---

`allowsMultipleSelection` **instance method** [7](#)

## B

---

`beginForDirectory:file:types:modellessDelegate:didEndSelector:contextInfo:` **instance method** [13](#)

`beginSheetForDirectory:file:types:modalForWindow:modalDelegate:didEndSelector:contextInfo:` **instance method** [14](#)

## C

---

`canChooseDirectories` **instance method** [8](#)

`canChooseFiles` **instance method** [8](#)

## F

---

`filenames` **instance method** [15](#)

## O

---

`openPanel` **class method** [7](#)

## R

---

`resolvesAliases` **instance method** [8](#)

`runModalForDirectory:file:types:` **instance method** [16](#)

`runModalForTypes:` **instance method** [17](#)

## S

---

`setAllowsMultipleSelection:` **instance method** [9](#)

`setCanChooseDirectories:` **instance method** [9](#)

`setCanChooseFiles:` **instance method** [10](#)

`setResolvesAliases:` **instance method** [10](#)

## U

---

`URLs` **instance method** [11](#)