
NSSavePanel Class Reference

[Cocoa](#) > [User Experience](#)



2009-02-04



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

Finder 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

NSSavePanel Class Reference 5

Overview	5
Tasks	5
Creating Panels	5
Configuring Panels	6
Configuring Panel Content	6
Running Panels	7
Accessing User Selection	8
Handling Actions	8
Deprecated	8
Class Methods	8
savePanel	8
Instance Methods	9
accessoryView	9
allowedFileTypes	9
allowsOtherFileTypes	10
beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector: contextInfo:	10
cancel:	11
canCreateDirectories	12
canSelectHiddenExtension	12
delegate	12
directory	13
filename	13
isExpanded	14
isExtensionHidden	14
message	14
nameFieldLabel	15
ok:	15
prompt	15
requiredFileType	16
runModal	16
runModalForDirectory:file:	17
setAccessoryView:	18
setAllowedFileTypes:	18
setAllowsOtherFileTypes:	19
setCanCreateDirectories:	19
setCanSelectHiddenExtension:	20
setDelegate:	21
setDirectory:	21
setExtensionHidden:	22
setMessage:	22

- setNameFieldLabel: 23
- setPrompt: 23
- setRequiredFileType: 24
- setTitle: 24
- setTreatsFilePackagesAsDirectories: 25
- title 25
- treatsFilePackagesAsDirectories 26
- URL 26
- validateVisibleColumns 27
- Delegate Methods 27
 - panel:compareFilename:with:caseSensitive: 27
 - panel:directoryDidChange: 28
 - panel:isValidFilename: 28
 - panel:shouldShowFilename: 29
 - panel:userEnteredFilename:confirmed: 29
 - panel:willExpand: 30
 - panelSelectionDidChange: 30
- Constants 31
 - Button tags 31

Appendix A [Deprecated NSSavePanel Methods 33](#)

- Deprecated in Mac OS X v10.3 33
 - selectText: 33

[Document Revision History 35](#)

[Index 37](#)

NSSavePanel Class Reference

Inherits from	NSPanel : NSWindow : NSResponder : NSObject
Conforms to	NSUserInterfaceValidations (NSWindow) NSAnimatablePropertyContainer (NSWindow) NSCoding (NSResponder) NSObject (NSObject)
Framework	/System/Library/Frameworks/AppKit.framework
Availability	Available in Mac OS X v10.0 and later.
Declared in	NSSavePanel.h
Companion guides	Application File Management Sheet Programming Topics for Cocoa
Related sample code	CIVideoDemoGL QTAudioExtractionPanel Quartz Composer WWDC 2005 TextEdit TextEditPlus WhackedTV

Overview

An `NSSavePanel` object creates and manages a Save panel and allows you to run the panel in a modal loop. The Save panel provides a simple way for a user to specify a file to use when saving a document or other data. It can restrict the user to files of a certain type, as specified by an extension.

An `NSSavePanel` object manages a panel that allows users to specify the directory and name under which a file is saved. It supports browsing of the file system, and it accommodates custom accessory views.

Tasks

Creating Panels

+ [savePanel](#) (page 8)

Returns a Save panel that has been initialized with default values.

Configuring Panels

- [accessoryView](#) (page 9)
Returns the custom accessory view for the current application.
- [setAccessoryView:](#) (page 18)
Customizes the panel for the application by adding a custom view to the panel.
- [title](#) (page 25)
Returns the title of the panel.
- [setTitle:](#) (page 24)
Sets the title of the panel.
- [prompt](#) (page 15)
Returns the prompt of the default button.
- [setPrompt:](#) (page 23)
Sets the prompt of the default button.
- [nameFieldLabel](#) (page 15)
Returns the string displayed in front of the filename text field.
- [setNameFieldLabel:](#) (page 23)
Sets the text displayed in front of the text field.
- [message](#) (page 14)
Returns the message displayed in the save panel.
- [setMessage:](#) (page 22)
Sets the message text displayed in the panel.
- [canCreateDirectories](#) (page 12)
Returns a Boolean value that indicates whether the panel allows the user to create directories.
- [setCanCreateDirectories:](#) (page 19)
Sets whether the panel allows the user to create directories.
- [delegate](#) (page 12)
Returns the panel's delegate.
- [setDelegate:](#) (page 21)
Sets an object as the panel's delegate, after verifying which delegate methods are implemented.

Configuring Panel Content

- [isExtensionHidden](#) (page 14)
Returns a Boolean value that indicates whether the extension-hiding checkbox is visible and checked.
- [setExtensionHidden:](#) (page 22)
Sets the value of the extension-hiding checkbox.
- [requiredFileType](#) (page 16)
Returns the required file type (if any).
- [setRequiredFileType:](#) (page 24)
Specifies the type, an extension to be appended to any selected files that don't already have that extension; "nib" and "rtf" are examples.
- [canSelectHiddenExtension](#) (page 12)
Returns a Boolean value that indicates whether the panel allows the user to hide or show extensions.

- [setCanSelectHiddenExtension:](#) (page 20)
Sets whether the panel allows the user to hide or show extensions.
- [allowedFileTypes](#) (page 9)
Returns an array of the allowed file types.
- [setAllowedFileTypes:](#) (page 18)
Specifies the allowed file types for the panel.
- [allowsOtherFileTypes](#) (page 10)
Returns a Boolean value that indicates whether the panel allows the user to save files with an extension that's not in the list of allowed types.
- [setAllowsOtherFileTypes:](#) (page 19)
Sets whether the panel allows the user to save files with an extension that's not in the list of allowed types.
- [treatsFilePackagesAsDirectories](#) (page 26)
Returns a Boolean value that indicates whether the panel displays file packages as directories.
- [setTreatsFilePackagesAsDirectories:](#) (page 25)
Sets the panel's behavior for displaying file packages (for example, `MyApp.app`) to the user.

Running Panels

- [beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector:contextInfo:](#) (page 10)
Presents a Save panel as a sheet with a specified path and, optionally, a specified file in that path.
- [runModal](#) (page 16)
Displays the panel and begins its event loop with the current working (or last selected) directory as the default starting point.
- [runModalForDirectory:file:](#) (page 17)
Initializes the panel to the directory and file specified, if any, then displays it and begins its modal event loop.
- [panel:compareFilename:with:caseSensitive:](#) (page 27) *delegate method*
Controls the ordering of files presented by the `NSSavePanel` object specified.
- [panel:isValidFilename:](#) (page 28) *delegate method*
Gives the delegate the opportunity to validate selected items.
- [panel:userEnteredFilename:confirmed:](#) (page 29) *delegate method*
Tells the delegate that the user confirmed a filename choice by hitting OK or Return in the Save panel.
- [validateVisibleColumns](#) (page 27)
Validates and possibly reloads the browser columns visible in the panel by invoking the delegate method [panel:shouldShowFilename:](#) (page 29).
- [panel:shouldShowFilename:](#) (page 29) *delegate method*
Gives the delegate the opportunity to filter items that it doesn't want the user to choose.
- [panel:willExpand:](#) (page 30) *delegate method*
Tells the delegate that the Save panel is about to expand or collapse because the user clicked the disclosure triangle that displays or hides the file browser.
- [panel:directoryDidChange:](#) (page 28) *delegate method*
Tells the delegate that the user has changed the selected directory in the `NSSavePanel` object specified.

- [panelSelectionDidChange:](#) (page 30) *delegate method*
Tells the delegate that the user changed the selection in the specified Save panel.

Accessing User Selection

- [directory](#) (page 13)
Returns the absolute pathname of the directory currently shown in the panel.
- [setDirectory:](#) (page 21)
Sets the current pathname in the panel's browser.
- [filename](#) (page 13)
Returns the absolute pathname of the file currently shown in the panel.
- [URL](#) (page 26)
Returns the absolute pathname of the file currently shown in the panel as a URL.
- [isExpanded](#) (page 14)
Returns a Boolean value that indicates whether the panel is expanded.

Handling Actions

- [ok:](#) (page 15)
This action method is invoked when the user clicks the panel's OK button.
- [cancel:](#) (page 11)
This action method is invoked when the user clicks the panel's Cancel button.

Deprecated

- [selectText:](#) (page 33) **Deprecated in Mac OS X v10.3**
This method has been deprecated. (**Deprecated.** There is no replacement.)

Class Methods

savePanel

Returns a Save panel that has been initialized with default values.

```
+ (NSSavePanel *)savePanel
```

Return Value

The initialized Save panel.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

CIVideoDemoGL

QTAudioExtractionPanel
 Quartz Composer WWDC 2005 TextEdit
 TextEditPlus
 WhackedTV

Declared In
 NSSavePanel.h

Instance Methods

accessoryView

Returns the custom accessory view for the current application.

- (NSView *)accessoryView

Return Value

The custom accessory view for the current application.

Discussion

In order to free up unused memory after closing the receiver, the accessory view is released after the panel is closed. If you rely on the receiver to hold onto the accessory view until the next time you use it, the accessory view may be deallocated unexpectedly. If you retain the accessory view in your own code, though, this deallocation should not be a problem.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setAccessoryView:](#) (page 18)

Declared In

NSSavePanel.h

allowedFileTypes

Returns an array of the allowed file types.

- (NSArray *)allowedFileTypes

Return Value

An array of the allowed file types.

Discussion

If the user specifies a file whose type is in the array of allowed types, the user is not presented with another dialog (see [allowsOtherFileTypes](#) (page 10) for details about this dialog) when trying to save. Examples of common file types are “rtf”, “tiff”, and “ps”. File type strings encoding HFS file types are not valid values for this attribute. A `nil` return value, which is the default, indicates that the user can save to any ASCII file.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setAllowedFileTypes:](#) (page 18)
- [requiredFileType](#) (page 16)

Declared In

NSSavePanel.h

allowsOtherFileTypes

Returns a Boolean value that indicates whether the panel allows the user to save files with an extension that's not in the list of allowed types.

- (BOOL)allowsOtherFileTypes

Return Value

YES if the panel allows the user to save files with an extension that's not in the list of allowed types; NO otherwise.

Discussion

If the user tries to save a filename with a recognized extension that's not in the list of allowed types they are presented with a dialog. If this method returns YES, then the dialog presents the option of using the extension the user specified.

The default setting is NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setAllowsOtherFileTypes:](#) (page 19)
- [allowedFileTypes](#) (page 9)

Declared In

NSSavePanel.h

beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector:contextInfo:

Presents a Save panel as a sheet with a specified path and, optionally, a specified file in that path.

- (void)beginSheetForDirectory:(NSString *)*path* file:(NSString *)*name* modalForWindow:(NSWindow *)*docWindow* modalDelegate:(id)*modalDelegate* didEndSelector:(SEL)*didEndSelector* contextInfo:(void *)*contextInfo*

Parameters

path

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.

name

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

docWindow

If not `nil`, the Save panel slides down as a sheet running as a document modal window in *docWindow*. If `nil`, the behavior defaults to a standalone modal window.

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. The `NSSavePanel` object does not retain the modal delegate.

didEndSelector

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

```
- (void)savePanelDidEnd:(NSSavePanel *)sheet returnCode:(int)returnCode
contextInfo:(void *)contextInfo;
```

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

contextInfo

Context information passed to *modalDelegate* in the *didEndSelector* message.

Availability

Available in Mac OS X v10.0 and later.

Related Sample Code

Cropped Image

QTAudioExtractionPanel

Quartz Composer WWDC 2005 TextEdit

RGB ValueTransformers

TextEditPlus

Declared In

`NSSavePanel.h`

cancel:

This action method is invoked when the user clicks the panel's Cancel button.

```
- (IBAction)cancel:(id)sender
```

Parameters

sender

The `NSSavePanel` object whose Cancel button was clicked.

Availability

Available in Mac OS X v10.0 and later.

See Also

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

Declared In

`NSSavePanel.h`

canCreateDirectories

Returns a Boolean value that indicates whether the panel allows the user to create directories.

- (BOOL)canCreateDirectories

Return Value

YES when the panel allows the user to create directories; NO otherwise.

Discussion

The default value is YES.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setCanCreateDirectories:](#) (page 19)

Declared In

NSSavePanel.h

canSelectHiddenExtension

Returns a Boolean value that indicates whether the panel allows the user to hide or show extensions.

- (BOOL)canSelectHiddenExtension

Return Value

YES when the panel allows the user to hide or show extensions; NO otherwise.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setCanSelectHiddenExtension:](#) (page 20)

Declared In

NSSavePanel.h

delegate

Returns the panel's delegate.

- (id)delegate

Return Value

The panel's delegate.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setDelegate:](#) (page 21)

Declared In

NSSavePanel.h

directory

Returns the absolute pathname of the directory currently shown in the panel.

- (NSString *)directory

Return Value

The absolute pathname of the directory currently shown in the panel.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setDirectory:](#) (page 21)

Related Sample Code

QTAudioExtractionPanel

Declared In

NSSavePanel.h

filename

Returns the absolute pathname of the file currently shown in the panel.

- (NSString *)filename

Return Value

The absolute pathname of the file.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [URL](#) (page 26)

Related Sample Code

CIVideoDemoGL

Quartz Composer WWDC 2005 TextEdit

RGB ValueTransformers

TextEditPlus

WhackedTV

Declared In

NSSavePanel.h

isExpanded

Returns a Boolean value that indicates whether the panel is expanded.

- (BOOL)isExpanded

Return Value

YES if the panel is expanded; NO otherwise.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

isExtensionHidden

Returns a Boolean value that indicates whether the extension-hiding checkbox is visible and checked.

- (BOOL)isExtensionHidden

Return Value

YES when the extension-hiding checkbox is visible and checked; NO otherwise.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [setCanSelectHiddenExtension:](#) (page 20)
- [setExtensionHidden:](#) (page 22)

Related Sample Code

Quartz Composer WWDC 2005 TextEdit
TextEditPlus

Declared In

NSSavePanel.h

message

Returns the message displayed in the save panel.

- (NSString *)message

Return Value

The message displayed in the save panel.

Discussion

This prompt appears on all `NSSavePanel` objects (or all `NSOpenPanel` objects if the receiver of this message is an `NSOpenPanel` instance) in your application. The default message text is an empty string.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setMessage:](#) (page 22)

Declared In

NSSavePanel.h

nameFieldLabel

Returns the string displayed in front of the filename text field.

- (NSString *)nameFieldLabel

Return Value

The string displayed in front of the filename text field.

Discussion

By default the label is “Save As:”

Availability

Available in Mac OS X v10.3 and later.

See Also

- [setNameFieldLabel:](#) (page 23)

Declared In

NSSavePanel.h

ok:

This action method is invoked when the user clicks the panel’s OK button.

- (IBAction)ok:(id)sender

Parameters

sender

The NSSavePanel object whose OK button was clicked.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [cancel:](#) (page 11)

Declared In

NSSavePanel.h

prompt

Returns the prompt of the default button.

- (NSString *)prompt

Return Value

The prompt of the default button.

Discussion

This prompt appears on all `NSSavePanel` objects (or all `NSOpenPanel` objects if the receiver of this message is an `NSOpenPanel` instance) in your application. By default, the text in the default button is “Open” for an open panel and “Save” for a Save panel.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setPrompt:](#) (page 23)

Declared In

`NSSavePanel.h`

requiredFileType

Returns the required file type (if any).

- (`NSString *`)requiredFileType

Return Value

The required file type (if any).

Discussion

A file specified in the Save panel is saved with the designated filename and this file type as an extension. Examples of common file types are “rtf”, “tiff”, and “ps”. File type strings encoding HFS file types are not valid values for this attribute. An `nil` return value indicates that the user can save to any ASCII file.

This method is equivalent to calling [allowedFileTypes](#) (page 9) and returning the first element of the list of allowed types, or `nil` if there are none.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setRequiredFileType:](#) (page 24)

Declared In

`NSSavePanel.h`

runModal

Displays the panel and begins its event loop with the current working (or last selected) directory as the default starting point.

- (`NSInteger`)runModal

Return Value

`NSFileHandlingPanelOKButton` (if the user clicks the OK button) or `NSFileHandlingPanelCancelButton` (if the user clicks the Cancel button).

Discussion

This method invokes `runModalForDirectory:file:` (page 17) (*file* argument is an empty `NSString` object), which in turn performs `NSApplication's runModalForWindow: method` with `self` as the argument. Do not invoke `filename` (page 13) or `directory` (page 13) within a modal loop because the information these methods fetch is updated only upon return.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `runModalForDirectory:file:` (page 17)
- `runModalForWindow:` (`NSApplication`)

Related Sample Code

`QTKitProgressTester`
`ThreadsExportMovie`

Declared In

`NSSavePanel.h`

runModalForDirectory:file:

Initializes the panel to the directory and file specified, if any, then displays it and begins its modal event loop.

```
- (NSInteger)runModalForDirectory:(NSString *)path file:(NSString *)filename
```

Parameters

path

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 *path* that is selected when the Save panel is presented to a user. When `nil`, no file is initially selected.

Return Value

`NSFileHandlingPanelOKButton` (if the user clicks the OK button) or `NSFileHandlingPanelCancelButton` (if the user clicks the Cancel button).

Discussion

This method invokes `NSApplication's runModalForWindow: method` with `self` as the argument. Do not invoke `filename` (page 13) or `directory` (page 13) within a modal loop because the information these methods fetch is updated only upon return.

Availability

Available in Mac OS X v10.0 and later.

See Also

- `runModal` (page 16)
- `runModalForWindow:` (`NSApplication`)

Related Sample Code

`AlbumToSlideshow`
`CoreRecipes`

Declared In

NSSavePanel.h

setAccessoryView:

Customizes the panel for the application by adding a custom view to the panel.

```
- (void)setAccessoryView:(NSView *)aView
```

Parameters*aView*

View to set as the panel's accessory view.

Discussion

The custom object that is added appears just above the OK and Cancel buttons at the bottom of the panel. The `NSSavePanel` object automatically resizes itself to accommodate *aView*. You can invoke this method repeatedly to change the accessory view as needed. If *aView* is `nil`, the Save panel removes the current accessory view.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [accessoryView](#) (page 9)

Related Sample Code

CustomSave

QTAudioExtractionPanel

QTKitPlayer

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

NSSavePanel.h

setAllowedFileTypes:

Specifies the allowed file types for the panel.

```
- (void)setAllowedFileTypes:(NSArray *)types
```

Parameters*types*

Array to set as the panel's array of allowed file types.

Discussion

The parameter must not be empty. A file type is an extension to be appended to any selected files that don't already have that extension; "nib" and "rtf" are examples. The items in *types* should not include the period that begins the extension. File type strings encoding HFS file types are not valid values. Pass `nil` to allow any file type, which is the default.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [allowedFileTypes](#) (page 9)
- [setRequiredFileType:](#) (page 24)

Related Sample Code

TimelineToTC

Declared In

NSSavePanel.h

setAllowsOtherFileTypes:

Sets whether the panel allows the user to save files with an extension that's not in the list of allowed types.

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

Parameters*flag*

YES specifies that the panel allows the user to save files with an extension that's not in the list of allowed types; NO specifies otherwise.

Discussion

If the user tries to save a filename with a recognized extension that's not in the list of allowed types they are presented with a dialog. If [allowsOtherFileTypes](#) (page 10) is YES, then the dialog presents the option of using the extension the user specified.

The default setting is NO.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [allowsOtherFileTypes](#) (page 10)
- [allowedFileTypes](#) (page 9)

Related Sample Code

MassiveImage

Quartz 2D Transformer

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

NSSavePanel.h

setCanCreateDirectories:

Sets whether the panel allows the user to create directories.

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

Parameters*flag*

YES specifies that the panel allows the user to create directories; NO specifies otherwise.

Availability

Available in Mac OS X v10.3 and later.

See Also- [canCreateDirectories](#) (page 12)**Related Sample Code**

GridCalendar

OpenGLCaptureToMovie

Quartz Composer QCTV

WhackedTV

Declared In

NSSavePanel.h

setCanSelectHiddenExtension:

Sets whether the panel allows the user to hide or show extensions.

- (void)setCanSelectHiddenExtension:(BOOL)*flag***Parameters***flag*

YES specifies that the panel allows the user to hide or show extensions; NO specifies otherwise.

Discussion

This method must be called before the panel is displayed. If set to YES, `isExtensionHidden` and `setExtensionHidden:`, respectively, can be used to get and set the value of the checkbox that hides or shows extensions.

Availability

Available in Mac OS X v10.1 and later.

See Also- [canSelectHiddenExtension](#) (page 12)- [isExtensionHidden](#) (page 14)- [setExtensionHidden:](#) (page 22)**Related Sample Code**

MassiveImage

Quartz Composer QCTV

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

WhackedTV

Declared In

NSSavePanel.h

setDelegate:

Sets an object as the panel's delegate, after verifying which delegate methods are implemented.

- (void)setDelegate:(id)*anObject*

Parameters

anObject

Object to set as the panel's delegate.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [delegate](#) (page 12)

Related Sample Code

AudioBurn

QTKitCreateMovie

QTKitPlayer

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

NSSavePanel.h

setDirectory:

Sets the current pathname in the panel's browser.

- (void)setDirectory:(NSString *)*path*

Parameters

path

String to set as the panel's current pathname.

Discussion

The *path* argument must be an absolute pathname.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [directory](#) (page 13)

Related Sample Code

CustomSave

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In

NSSavePanel.h

setExtensionHidden:

Sets the value of the extension-hiding checkbox.

- (void)setExtensionHidden:(BOOL)flag

Parameters

flag

YES specifies that the extension-hiding checkbox is visible and checked; NO specifies otherwise.

Discussion

This method should rarely be used because the state is saved on a per-application basis. Use this method to set whether a file's extension should be indicated as being shown.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [setCanSelectHiddenExtension:](#) (page 20)
- [isExtensionHidden](#) (page 14)

Related Sample Code

Fiendishthngs

Declared In

NSSavePanel.h

setMessage:

Sets the message text displayed in the panel.

- (void)setMessage:(NSString *)message

Parameters

message

String to set as the panel's message.

Discussion

This prompt appears on all `NSSavePanel` objects (or all `NSOpenPanel` objects if the receiver of this message is an `NSOpenPanel` instance) in your application. The default message text is an empty string.

Availability

Available in Mac OS X v10.3 and later.

See Also

- [message](#) (page 14)

Related Sample Code

CustomSave

NSOperationSample

Quartz 2D Transformer

TrackBall

ViewController

Declared In

NSSavePanel.h

setNameFieldLabel:

Sets the text displayed in front of the text field.

- (void)setNameFieldLabel:(NSString *)label

Parameters*label*

String to set as the text displayed in front of the panel's text field.

Discussion

By default the label is "Save As:"

Availability

Available in Mac OS X v10.3 and later.

See Also- [nameFieldLabel](#) (page 15)**Related Sample Code**

CustomSave

Declared In

NSSavePanel.h

setPrompt:

Sets the prompt of the default button.

- (void)setPrompt:(NSString *)prompt

Parameters*prompt*

String to set as the prompt of the panel's default button.

Discussion

This prompt appears on all NSSavePanel objects (or all NSOpenPanel objects if the receiver of this message is an NSOpenPanel instance) in your application. By default, the text in the default button is "Open" for an Open panel and "Save" for a Save panel.

It is intended that short words or phrases, such as "Open," "Save," "Set," or "Choose," be used on the button. The button is not resized to accommodate long prompts.

Since this method previously affected a title field, any colon at the end of *prompt* is removed.

Availability

Available in Mac OS X v10.0 and later.

See Also- [prompt](#) (page 15)

Related Sample Code

AudioBurn
 CocoaSpeechSynthesisExample
 ContentBurn
 NSOperationSample
 ObjectPath

Declared In

NSSavePanel.h

setRequiredFileType:

Specifies the type, an extension to be appended to any selected files that don't already have that extension; "nib" and "rtf" are examples.

```
- (void)setRequiredFileType:(NSString *)type
```

Parameters

type

String to set as the extension to be appended to any selected files that don't already have that extension.

Discussion

The argument *type* should not include the period that begins the extension. Pass *nil* to indicate any type. File type strings encoding HFS file types are not valid values for this attribute. You need to invoke this method each time the Save panel is used for another file type within the application.

This method is equivalent to calling [setAllowedFileTypes:](#) (page 18) with an array containing only *type* (unless *type* is *nil*, and then it's equivalent to calling [setAllowedFileTypes:](#) with *nil*).

Availability

Available in Mac OS X v10.0 and later.

See Also

- [requiredFileType](#) (page 16)

Related Sample Code

CIVideoDemoGL
 Quartz Composer QCTV
 Quartz Composer WWDC 2005 TextEdit
 TextEditPlus
 WhackedTV

Declared In

NSSavePanel.h

setTitle:

Sets the title of the panel.

```
- (void)setTitle:(NSString *)title
```


Parameters*title*

String to set as the panel's title.

Discussion

By default, "Save" is the title string. If you adapt the `NSSavePanel` object for other uses, its title should reflect the user action that brings it to the screen.

Availability

Available in Mac OS X v10.0 and later.

See Also- [title](#) (page 25)**Related Sample Code**

AudioBurn

ContentBurn

ObjectPath

Quartz Composer WWDC 2005 TextEdit

TextEditPlus

Declared In`NSSavePanel.h`**setTreatsFilePackagesAsDirectories:**

Sets the panel's behavior for displaying file packages (for example, `MyApp.app`) to the user.

- (void)setTreatsFilePackagesAsDirectories:(BOOL)flag

Parameters*flag*

YES specifies that the panel will display file packages as directories; NO specifies otherwise.

Availability

Available in Mac OS X v10.0 and later.

See Also- [treatsFilePackagesAsDirectories](#) (page 26)**Related Sample Code**

MassiveImage

Quartz 2D Transformer

Declared In`NSSavePanel.h`**title**

Returns the title of the panel.

- (NSString *)title

Return Value

The title of the panel.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTitle:](#) (page 24)

Declared In

NSSavePanel.h

treatsFilePackagesAsDirectories

Returns a Boolean value that indicates whether the panel displays file packages as directories.

- (BOOL)treatsFilePackagesAsDirectories

Return Value

YES if the panel displays file packages as directories; NO otherwise.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [setTreatsFilePackagesAsDirectories:](#) (page 25)

Declared In

NSSavePanel.h

URL

Returns the absolute pathname of the file currently shown in the panel as a URL.

- (NSURL *)URL

Return Value

The absolute pathname of the file currently shown in the panel as a URL.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [filename](#) (page 13)

Related Sample Code

MassiveImage

NSOperationSample

PDF Annotation Editor

PDF Calendar

Quartz2DBasics

Declared In

NSSavePanel.h

validateVisibleColumns

Validates and possibly reloads the browser columns visible in the panel by invoking the delegate method `panel:shouldShowFilename:` (page 29).

```
- (void)validateVisibleColumns
```

Discussion

You might use this method if you want the browser to only allow selection of files with certain extensions based on the selection made in an accessory-view pop-up list. When the user changes the selection, you would invoke this method to revalidate the visible columns.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

Delegate Methods

panel:compareFilename:with:caseSensitive:

Controls the ordering of files presented by the NSSavePanel object specified.

```
- (NSComparisonResult)panel:(id)sender compareFilename:(NSString *)fileName1
  with:(NSString *)fileName2 caseSensitive:(BOOL)flag
```

Parameters

sender

Panel requesting the ordering.

fileName1

String representing the first filename to order.

fileName2

String representing the second filename to order.

flag

YES specifies that the ordering is to be case-sensitive; NO specifies otherwise.

Return Value

One of the following:

- NSOrderedAscending if *fileName1* should precede *fileName2*
- NSOrderedSame if the two names are equivalent
- NSOrderedDescending if *fileName2* should precede *fileName1*

Discussion

Don't reorder filenames in the Save panel without good reason, because it may confuse the user to have files in one Save panel or Open panel ordered differently than those in other such panels or in the Finder. The default behavior of Save and Open panels is to order files as they appear in the Finder. Note also that by implementing this method you will reduce the operating performance of the panel.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

panel:directoryDidChange:

Tells the delegate that the user has changed the selected directory in the NSSavePanel object specified.

```
- (void)panel:(id)sender directoryDidChange:(NSString *)path
```

Parameters

sender

Panel whose directory has changed.

path

String representing the new directory's path.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSSavePanel.h

panel:isValidFilename:

Gives the delegate the opportunity to validate selected items.

```
- (BOOL)panel:(id)sender isValidFilename:(NSString *)filename
```

Parameters

sender

Panel requesting filename validation.

filename

String representing the filename to validate.

Return Value

YES if the filename is valid, or NO if the save panel should stay in its modal loop and wait for the user to type in or select a different filename or names.

Discussion

The NSSavePanel object *sender* sends this message just before the end of a modal session for each filename displayed or selected (including filenames in multiple selections). If the delegate refuses a filename in a multiple selection, none of the filenames in the selection is accepted.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

panel:shouldShowFilename:

Gives the delegate the opportunity to filter items that it doesn't want the user to choose.

```
- (BOOL)panel:(id)sender shouldShowFilename:(NSString *)filename
```

Parameters*sender*

Panel that is querying whether it should show a certain file.

filename

String representing the name of the file to be loaded in the browser.

Return Value

YES if *filename* should be selectable, and NO if the save panel should disable the file or directory.

Discussion

The NSSavePanel object *sender* sends this message to the panel's delegate for each file or directory (*filename*) it is about to load in the browser.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

panel:userEnteredFilename:confirmed:

Tells the delegate that the user confirmed a filename choice by hitting OK or Return in the Save panel.

```
- (NSString *)panel:(id)sender userEnteredFilename:(NSString *)filename
  confirmed:(BOOL)okFlag
```

Parameters*sender*

Panel reporting the user's confirmation of a filename choice.

filename

String representing the filename the user chose.

okFlag

YES specifies the user clicked the OK button; NO specifies otherwise.

Return Value

String representing the filename that should be used for saving.

Discussion

You can either leave the filename alone, return a new filename, or return `nil` to cancel the save (and leave the Save panel as is). This method is sent before any required extension is appended to the filename and before the Save panel asks the user whether to replace an existing file.

Note that in the future, this method may be called multiple times in the sessions as the user types. In those cases, *okFlag* will be NO until the user confirms the choice, in which case *okFlag* will become YES. If the delegate does extensive validation or puts up alerts, it should do so only when *okFlag* is YES.

Availability

Available in Mac OS X v10.0 and later.

See Also

- [panel:isValidFilename:](#) (page 28)

Declared In

NSSavePanel.h

panel:willExpand:

Tells the delegate that the Save panel is about to expand or collapse because the user clicked the disclosure triangle that displays or hides the file browser.

```
- (void)panel:(id)sender willExpand:(BOOL)expanding
```

Parameters

sender

Panel that is about to expand or collapse.

expanding

YES specifies that the panel is expanding; NO specifies that it is collapsing.

Availability

Available in Mac OS X v10.0 and later.

Declared In

NSSavePanel.h

panelSelectionDidChange:

Tells the delegate that the user changed the selection in the specified Save panel.

```
- (void)panelSelectionDidChange:(id)sender
```

Parameters

sender

Panel whose selection has changed.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSSavePanel.h

Constants

Button tags

Button tags that refer to items on the panel.

```
enum {
    NSFileHandlingPanelCancelButton = NSCancelButton,
    NSFileHandlingPanelOKButton = NSOKButton
};
```

Constants

`NSFileHandlingPanelCancelButton`

The Cancel button.

Available in Mac OS X v10.0 and later.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelOKButton`

The OK button.

Available in Mac OS X v10.0 and later.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelImageButton`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelTitleField`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelBrowser`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelForm`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelHomeButton`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

`NSFileHandlingPanelDiskButton`

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in `NSSavePanel.h`.

NSFileHandlingPanelDiskEjectButton

Deleted in Mac OS X v10.3.

Available in Mac OS X v10.0 through Mac OS X v10.2.

Declared in NSSavePanel.h.

Declared In

NSSavePanel.h

Deprecated NSSavePanel Methods

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

Deprecated in Mac OS X v10.3

selectText:

This method has been deprecated. (Deprecated in Mac OS X v10.3. There is no replacement.)

- (IBAction)selectText:(id)sender

Availability

Deprecated in Mac OS X v10.3.

Declared In

NSSavePanel.h

Document Revision History

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

Date	Notes
2009-02-04	Updated documentation to use more consistent method formatting and corrected various errors.
2008-07-11	Corrected singleton information in the overview of this class.
2007-04-03	Updated for Mac OS X v10.5.
2006-05-23	First publication of this content as a separate document.

REVISION HISTORY

Document Revision History

Index

A

accessoryView [instance method 9](#)
allowedFileTypes [instance method 9](#)
allowsOtherFileTypes [instance method 10](#)

B

beginSheetForDirectory:file:modalForWindow:
 modalDelegate:didEndSelector:contextInfo:
 [instance method 10](#)
Button tags [31](#)

C

cancel: [instance method 11](#)
canCreateDirectories [instance method 12](#)
canSelectHiddenExtension [instance method 12](#)

D

delegate [instance method 12](#)
directory [instance method 13](#)

F

filename [instance method 13](#)

I

isExpanded [instance method 14](#)
isExtensionHidden [instance method 14](#)

M

message [instance method 14](#)

N

nameFieldLabel [instance method 15](#)
NSFileHandlingPanelBrowser [constant 31](#)
NSFileHandlingPanelCancelButton [constant 31](#)
NSFileHandlingPanelDiskButton [constant 31](#)
NSFileHandlingPanelDiskEjectButton [constant 32](#)
NSFileHandlingPanelForm [constant 31](#)
NSFileHandlingPanelHomeButton [constant 31](#)
NSFileHandlingPanelImageButton [constant 31](#)
NSFileHandlingPanelOKButton [constant 31](#)
NSFileHandlingPanelTextField [constant 31](#)

O

ok: [instance method 15](#)

P

panel:compareFilename:with:caseSensitive:
 <NSObject> [delegate method 27](#)
panel:directoryDidChange: <NSObject> [delegate
method 28](#)
panel:isValidFilename: <NSObject> [delegate
method 28](#)
panel:shouldShowFilename: <NSObject> [delegate
method 29](#)
panel:userEnteredFilename:confirmed:
 <NSObject> [delegate method 29](#)
panel:willExpand: <NSObject> [delegate method 30](#)
panelSelectionDidChange: <NSObject> [delegate
method 30](#)
prompt [instance method 15](#)

R

requiredFileType **instance method** [16](#)
runModal **instance method** [16](#)
runModalForDirectory:file: **instance method** [17](#)

S

savePanel **class method** [8](#)
selectText: **instance method** [33](#)
setAccessoryView: **instance method** [18](#)
setAllowedFileTypes: **instance method** [18](#)
setAllowsOtherFileTypes: **instance method** [19](#)
setCanCreateDirectories: **instance method** [19](#)
setCanSelectHiddenExtension: **instance method** [20](#)
setDelegate: **instance method** [21](#)
setDirectory: **instance method** [21](#)
setExtensionHidden: **instance method** [22](#)
setMessage: **instance method** [22](#)
setNameFieldLabel: **instance method** [23](#)
setPrompt: **instance method** [23](#)
setRequiredFileType: **instance method** [24](#)
setTitle: **instance method** [24](#)
setTreatsFilePackagesAsDirectories: **instance method** [25](#)

T

title **instance method** [25](#)
treatsFilePackagesAsDirectories **instance method** [26](#)

U

URL **instance method** [26](#)

V

validateVisibleColumns **instance method** [27](#)