

---

# NSSavePanel Class Reference

Data Management: File Management





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	6
Creating Panels	6
Configuring Panels	6
Configuring Panel Content	7
Running Panels	7
Accessing User Selection	8
Handling Actions	9
Class Methods	9
savePanel	9
Instance Methods	9
accessoryView	9
allowedFileTypes	10
allowsOtherFileTypes	10
beginSheetModalForWindow:completionHandler:	11
beginWithCompletionHandler:	11
cancel:	12
canCreateDirectories	12
canSelectHiddenExtension	13
delegate	13
directoryURL	13
isExpanded	14
isExtensionHidden	14
message	14
nameFieldLabel	15
nameFieldStringValue	15
ok:	16
prompt	16
runModal	16
setAccessoryView:	17
setAllowedFileTypes:	17
setAllowsOtherFileTypes:	18
setCanCreateDirectories:	19
setCanSelectHiddenExtension:	19
setDelegate:	20
setDirectoryURL:	20
setExtensionHidden:	21
setMessage:	21
setNameFieldLabel:	22
setNameFieldStringValue:	22

- setPrompt: 23
- setShowsHiddenFiles: 23
- setTitle: 24
- setTreatsFilePackagesAsDirectories: 24
- showsHiddenFiles 25
- title 25
- treatsFilePackagesAsDirectories 25
- URL 26
- validateVisibleColumns 26
- Constants 26
  - Button tags 26

**Appendix A      [Deprecated NSSavePanel Methods 29](#)**

---

- Deprecated in Mac OS X v10.3 29
  - selectText: 29
- Deprecated in Mac OS X v10.6 29
  - beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector: contextInfo: 29
  - directory 30
  - filename 31
  - requiredFileType 31
  - runModalForDirectory:file: 32
  - setDirectory: 32
  - setRequiredFileType: 33
  - panel:compareFilename:with:caseSensitive: 34
  - panel:directoryDidChange: 34
  - panel:isValidFilename: 35
  - panel:shouldShowFilename: 36

**[Document Revision History 37](#)**

---

**[Index 39](#)**

---

# NSSavePanel Class Reference

---

<b>Inherits from</b>	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>	NSSavePanel.h
<b>Companion guides</b>	Application File Management Sheet Programming Topics for Cocoa
<b>Related sample code</b>	ImageKitDemo QTAudioContextInsert QTAudioExtractionPanel Quartz Composer WWDC 2005 TextEdit 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.

An `NSSavePanel` object may have a delegate. The methods that delegates of `NSSavePanel` may implement are specified by the `NSOpenSavePanelDelegate` protocol.

## Tasks

### Creating Panels

- + [savePanel](#) (page 9)  
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 17)  
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 16)  
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 22)  
Sets the text displayed in front of the text field.
- [message](#) (page 14)  
Returns the message displayed in the save panel.
- [setMessage:](#) (page 21)  
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.
- [showsHiddenFiles](#) (page 25)  
Returns whether the panel displays files that are normally hidden from the user.
- [setShowsHiddenFiles:](#) (page 23)  
Specifies whether the panel displays files that are normally hidden from the user.
- [delegate](#) (page 13)  
Returns the panel's delegate.
- [setDelegate:](#) (page 20)  
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 21)  
Sets the value of the extension-hiding checkbox.
- [canSelectHiddenExtension](#) (page 13)  
Returns a Boolean value that indicates whether the panel allows the user to hide or show extensions.
- [setCanSelectHiddenExtension:](#) (page 19)  
Sets whether the panel allows the user to hide or show extensions.
- [allowedFileTypes](#) (page 10)  
Returns an array of the allowed file types.
- [setAllowedFileTypes:](#) (page 17)  
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 18)  
Sets whether the panel allows the user to save files with an extension that's not in the list of allowed types.
- [treatsFilePackagesAsDirectories](#) (page 25)  
Returns a Boolean value that indicates whether the panel displays file packages as directories.
- [setTreatsFilePackagesAsDirectories:](#) (page 24)  
Sets the panel's behavior for displaying file packages (for example, `MyApp.app`) to the user.
- [requiredFileType](#) (page 31) **Deprecated in Mac OS X v10.6**  
Returns the required file type (if any). (**Deprecated.** Use [allowedFileTypes](#) (page 10) instead.)
- [setRequiredFileType:](#) (page 33) **Deprecated in Mac OS X v10.6**  
Specifies the type, an extension to be appended to any selected files that don't already have that extension; "nib" and "rtf" are examples. (**Deprecated.** Use [setAllowedFileTypes:](#) (page 17) instead.)

## Running Panels

- [beginSheetModalForWindow:completionHandler:](#) (page 11)  
Presents the panel as a sheet modal to the specified window.
- [beginWithCompletionHandler:](#) (page 11)  
Presents the panel as a modeless window.
- [runModal](#) (page 16)  
Displays the panel and begins its event loop with the current working (or last selected) directory as the default starting point.
- [validateVisibleColumns](#) (page 26)  
Validates and possibly reloads the browser columns visible in the panel by invoking the delegate method [panel:shouldShowFilename:](#) (page 36).

- [beginSheetForDirectory:file:modalForWindow:modalDelegate:didEndSelector:contextInfo:](#) (page 29) **Deprecated in Mac OS X v10.6**  
Presents a Save panel as a sheet with a specified path and, optionally, a specified file in that path. (**Deprecated.** Use [beginSheetModalForWindow:completionHandler:](#) (page 11) instead.)
- [runModalForDirectory:file:](#) (page 32) **Deprecated in Mac OS X v10.6**  
Initializes the panel to the directory and file specified, if any, then displays it and begins its modal event loop. (**Deprecated.** Use [runModal](#) (page 16) instead.)
- [panel:compareFilename:with:caseSensitive:](#) (page 34) *delegate method* **Deprecated in Mac OS X v10.6**  
Controls the ordering of files presented by the NSSavePanel object specified. (**Deprecated.** There is no replacement.)
- [panel:directoryDidChange:](#) (page 34) *delegate method* **Deprecated in Mac OS X v10.6**  
Tells the delegate that the user has changed the selected directory in the NSSavePanel object specified. (**Deprecated.** Use [panel:didChangeToDirectoryURL:](#) (NSOpenSavePanelDelegate) instead.)
- [panel:isValidFilename:](#) (page 35) *delegate method* **Deprecated in Mac OS X v10.6**  
Gives the delegate the opportunity to validate selected items. (**Deprecated.** Use [panel:validateURL:error:](#) (NSOpenSavePanelDelegate) instead. If both methods are implemented, the URL version will be called.)
- [panel:shouldShowFilename:](#) (page 36) *delegate method* **Deprecated in Mac OS X v10.6**  
Gives the delegate the opportunity to filter items that it doesn't want the user to choose. (**Deprecated.** Use [panel:shouldEnableURL:](#) (NSOpenSavePanelDelegate).)

## Accessing User Selection

- [directoryURL](#) (page 13)  
Returns the directory shown in the panel as a URL.
- [setDirectoryURL:](#) (page 20)  
Sets the directory shown in the panel to the directory with the specified URL.
- [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.
- [nameFieldStringValue](#) (page 15)  
Returns the user-editable filename currently shown in the name field.
- [setNameFieldStringValue:](#) (page 22)  
Sets the filename in the name field to the specified value.
- [selectText:](#) (page 29) **Deprecated in Mac OS X v10.3**  
This method has been deprecated. (**Deprecated.** There is no replacement.)
- [directory](#) (page 30) **Deprecated in Mac OS X v10.6**  
Returns the absolute pathname of the directory currently shown in the panel. (**Deprecated.** Use [directoryURL](#) (page 13) instead.)
- [filename](#) (page 31) **Deprecated in Mac OS X v10.6**  
Returns the absolute pathname of the file currently shown in the panel. (**Deprecated.** Use [URL](#) (page 26) instead.)

- [setDirectory:](#) (page 32) **Deprecated in Mac OS X v10.6**  
Sets the current pathname in the panel's browser. (**Deprecated.** Use [setDirectoryURL:](#) (page 20) instead.)

## Handling Actions

- [ok:](#) (page 16)  
This action method is invoked when the user clicks the panel's OK button.
- [cancel:](#) (page 12)  
This action method is invoked when the user clicks the panel's Cancel button.

## 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

ImageKitDemo

QTAudioExtractionPanel

Quartz Composer WWDC 2005 TextEdit

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 panel, the accessory view is released after the panel is closed. If you rely on the panel 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 17)

**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 17)

**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; otherwise, NO.

**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 18)
- [allowedFileTypes](#) (page 10)

**Declared In**

NSSavePanel.h

**beginSheetModalForWindow:completionHandler:**

Presents the panel as a sheet modal to the specified window.

```
- (void)beginSheetModalForWindow:(NSWindow *)window
    completionHandler:(void (^)(NSInteger result))handler
```

**Parameters**

*window*

The window in which the panel will be presented.

*handler*

The block called after the user has closed the panel. The argument passed in will be `NSFileHandlingPanelOKButton` if the user chose the OK button or `NSFileHandlingPanelCancelButton` if the user chose the Cancel button.

**Discussion**

Any properties of the panel you wish to set should be set before calling this method.

**Availability**

Available in Mac OS X v10.6 and later.

**Related Sample Code**

Denoise

DispatchFractal

From A View to A Movie

From A View to A Picture

TextSizingExample

**Declared In**

NSSavePanel.h

**beginWithCompletionHandler:**

Presents the panel as a modeless window.

- (void)beginWithCompletionHandler:(void (^)(NSInteger result))*handler*

#### Parameters

*handler*

The block called after the user has closed the panel. The argument passed in will be `NSFileHandlingPanelOKButton` if the user chose the OK button or `NSFileHandlingPanelCancelButton` if the user chose the Cancel button.

#### Discussion

Any properties of the panel you wish to set should be set before calling this method.

#### Availability

Available in Mac OS X v10.6 and later.

#### 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 16)

#### 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; otherwise, NO.

#### 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; otherwise, NO.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

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

**Declared In**

NSSavePanel.h

## delegate

Returns the panel's delegate.

- (id <NSOpenSavePanelDelegate>)delegate

**Return Value**

The panel's delegate.

**Availability**

Available in Mac OS X v10.3 and later.

**See Also**

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

**Declared In**

NSSavePanel.h

## directoryURL

Returns the directory shown in the panel as a URL.

- (NSURL \*)directoryURL

**Return Value**

The directory's URL.

**Availability**

Available in Mac OS X v10.6 and later.

**See Also**

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

**Related Sample Code**

From A View to A Movie

From A View to A Picture

**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; otherwise, NO.

**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; otherwise, NO.

**Availability**

Available in Mac OS X v10.1 and later.

**See Also**

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

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

**Related Sample Code**

Quartz Composer WWDC 2005 TextEdit

**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 21)

**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 22)

**Declared In**

`NSSavePanel.h`

**nameFieldStringValue**

Returns the user-editable filename currently shown in the name field.

- (`NSString *`)nameFieldStringValue

**Return Value**

The filename currently selected.

**Availability**

Available in Mac OS X v10.6 and later.

**See Also**

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

**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 12)

**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

**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 `NSApplication`'s `runModalForWindow:` method with `self` as the argument.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- `runModalForWindow:` (`NSApplication`)

**Related Sample Code**

AudioDataOutputToAudioUnit

QTKitProgressTester

ThreadsExportMovie

**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**

ImageApp

QTAudioContextInsert

QTAudioExtractionPanel

QTKitPlayer

Quartz Composer WWDC 2005 TextEdit

**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 10)

### Related Sample Code

AudioDataOutputToAudioUnit

ImageApp

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*

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

### 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 10)

### Related Sample Code

MassiveImage

Quartz 2D Transformer

Quartz Composer WWDC 2005 TextEdit

**Declared In**

NSSavePanel.h

**setCanCreateDirectories:**

Sets whether the panel allows the user to create directories.

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

**Parameters**

*flag*

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

**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*

If YES, the panel allows the user to hide or show extensions; if NO, it does not.

**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 13)

- [isExtensionHidden](#) (page 14)

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

**Related Sample Code**

AudioDataOutputToAudioUnit  
 MassivelImage  
 Quartz Composer QCTV  
 Quartz Composer WWDC 2005 TextEdit  
 WhackedTV

**Declared In**

NSSavePanel.h

**setDelegate:**

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

```
- (void)setDelegate:(id <NSOpenSavePanelDelegate>)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 13)

**Related Sample Code**

AudioBurn  
 CIRAWFilterSample  
 QTKitCreateMovie  
 QTKitPlayer  
 Quartz Composer WWDC 2005 TextEdit

**Declared In**

NSSavePanel.h

**setDirectoryURL:**

Sets the directory shown in the panel to the directory with the specified URL.

```
- (void)setDirectoryURL:(NSURL *)url
```

**Parameters**

*url*  
 The URL of the directory to set.

**Availability**

Available in Mac OS X v10.6 and later.

**See Also**

- [directoryURL](#) (page 13)

**Declared In**

NSSavePanel.h

**setExtensionHidden:**

Sets the value of the extension-hiding checkbox.

- (void)setExtensionHidden:(BOOL)flag

**Parameters***flag*

If YES, the extension-hiding checkbox is visible and checked; if NO, it is not.

**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 19)
- [isExtensionHidden](#) (page 14)

**Related Sample Code**

Fiendishthngs

FunHouse

**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

Quartz 2D Transformer  
 SBSetFinderComment  
 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

**setNameFieldStringValue:**

Sets the filename in the name field to the specified value.

```
- (void)setNameFieldStringValue:(NSString *)value
```

**Parameters**

*value*

The filename to set. The value must not be nil.

**Availability**

Available in Mac OS X v10.6 and later.

**See Also**

- [nameFieldStringValue](#) (page 15)

**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 16)

**Related Sample Code**

AudioBurn

CocoaSpeechSynthesisExample

ContentBurn

GridCalendar

ObjectPath

**Declared In**

`NSSavePanel.h`

**setShowsHiddenFiles:**

Specifies whether the panel displays files that are normally hidden from the user.

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

**Parameters**

*flag*

If YES, the panel displays hidden files; if NO, it does not.

**Availability**

Available in Mac OS X v10.6 and later.

**See Also**

- [showsHiddenFiles](#) (page 25)

**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

MovieAssembler

ObjectPath

Quartz Composer WWDC 2005 TextEdit

**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*

If YES, the panel will display file packages as directories; if NO, it will not.

**Availability**

Available in Mac OS X v10.0 and later.

**See Also**

- [treatsFilePackagesAsDirectories](#) (page 25)

**Related Sample Code**

CustomSave

MassiveImage

Quartz 2D Transformer

TextSizingExample

**Declared In**

`NSSavePanel.h`

## showsHiddenFiles

Returns whether the panel displays files that are normally hidden from the user.

- (BOOL)showsHiddenFiles

### Return Value

YES if the panel displays hidden files; otherwise, NO.

### Discussion

The default value is NO.

### Availability

Available in Mac OS X v10.6 and later.

### See Also

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

### 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; otherwise, NO.

### Availability

Available in Mac OS X v10.0 and later.

### See Also

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

**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.

**Related Sample Code**

CocoaSpeechSynthesisExample

MassiveImage

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 36).

- (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

## 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.

NSFileHandlingPanelTextField

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**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.3.

#### **Declared In**

NSSavePanel.h

## Deprecated in Mac OS X v10.6

### **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. (Deprecated in Mac OS X v10.6. Use `beginSheetModalForWindow:completionHandler:` (page 11) instead.)

```
- (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.

## Deprecated NSSavePanel Methods

*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.

Deprecated in Mac OS X v10.6.

**See Also**

- [beginSheetModalForWindow:completionHandler:](#) (page 11)

**Related Sample Code**

Cropped Image

ImageKitDemo

QTAudioExtractionPanel

Quartz Composer WWDC 2005 TextEdit

RGB ValueTransformers

**Declared In**

`NSSavePanel.h`

**directory**

Returns the absolute pathname of the directory currently shown in the panel. (Deprecated in Mac OS X v10.6. Use [directoryURL](#) (page 13) instead.)

```
- (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.

Deprecated in Mac OS X v10.6.

**See Also**

- [directoryURL](#) (page 13)

**Related Sample Code**

QTAudioContextInsert

QTAudioExtractionPanel

## Deprecated NSSavePanel Methods

**Declared In**

NSSavePanel.h

**filename**

Returns the absolute pathname of the file currently shown in the panel. (Deprecated in Mac OS X v10.6. Use [URL](#) (page 26) instead.)

- (NSString \*)filename

**Return Value**

The absolute pathname of the file.

**Availability**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- [URL](#) (page 26)

**Related Sample Code**

CIVideoDemoGL

ImageKitDemo

Quartz Composer WWDC 2005 TextEdit

RGB ValueTransformers

WhackedTV

**Declared In**

NSSavePanel.h

**requiredFileType**

Returns the required file type (if any). (Deprecated in Mac OS X v10.6. Use [allowedFileTypes](#) (page 10) instead.)

- (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 10) 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.

Deprecated in Mac OS X v10.6.

**See Also**

- [allowedFileTypes](#) (page 10)

**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. (Deprecated in Mac OS X v10.6. Use [runModal](#) (page 16) instead.)

```
- (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.

**Availability**

Available in Mac OS X v10.0 and later.

Deprecated in Mac OS X v10.6.

**See Also**

- [runModal](#) (page 16)

- `runModalForWindow:` (`NSApplication`)

**Related Sample Code**

[AlbumToSlideshow](#)

[CoreRecipes](#)

**Declared In**

NSSavePanel.h

**setDirectory:**

Sets the current pathname in the panel's browser. (Deprecated in Mac OS X v10.6. Use [setDirectoryURL:](#) (page 20) instead.)

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

## Deprecated NSSavePanel Methods

**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.

Deprecated in Mac OS X v10.6.

**See Also**

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

**Related Sample Code**

CustomSave

Quartz Composer WWDC 2005 TextEdit

**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. (Deprecated in Mac OS X v10.6. Use [setAllowedFileTypes:](#) (page 17) instead.)

```
- (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 17) 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.

Deprecated in Mac OS X v10.6.

**See Also**

- [setAllowedFileTypes:](#) (page 17)

**Related Sample Code**

CIVideoDemoGL

PDF Calendar

Quartz Composer QCTV

Quartz Composer WWDC 2005 TextEdit

WhackedTV

**Declared In**

NSSavePanel.h

**panel:compareFilename:with:caseSensitive:**

Controls the ordering of files presented by the `NSSavePanel` object specified. (Deprecated in Mac OS X v10.6. There is no replacement.)

```
- (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*

If YES, the ordering is case-sensitive; if NO, it is not.

**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.

Deprecated in Mac OS X v10.6.

**Declared In**

NSSavePanel.h

**panel:directoryDidChange:**

Tells the delegate that the user has changed the selected directory in the `NSSavePanel` object specified. (Deprecated in Mac OS X v10.6. Use `panel:didChangeToDirectoryURL:(NSOpenSavePanelDelegate)` instead.)

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

## Deprecated NSSavePanel Methods

**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.

Deprecated in Mac OS X v10.6.

**See Also**- `panel:didChangeToDirectoryURL:(NSOpenSavePanelDelegate)`**Declared In**

NSSavePanel.h

**panel:isValidFilename:**

Gives the delegate the opportunity to validate selected items. (Deprecated in Mac OS X v10.6. Use `panel:validateURL:error:(NSOpenSavePanelDelegate)` instead. If both methods are implemented, the URL version will be called.)

- (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.

Deprecated in Mac OS X v10.6.

**See Also**- `panel:validateURL:error:(NSOpenSavePanelDelegate)`**Declared In**

NSSavePanel.h

## panel:shouldShowFilename:

Gives the delegate the opportunity to filter items that it doesn't want the user to choose. (Deprecated in Mac OS X v10.6. Use `panel:shouldEnableURL:(NSOpenSavePanelDelegate)`.)

```
- (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.

Deprecated in Mac OS X v10.6.

### See Also

- `panel:shouldEnableURL:(NSOpenSavePanelDelegate)`.

### Declared In

`NSSavePanel.h`

# Document Revision History

---

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

Date	Notes
2009-02-19	Updated for Mac OS X v10.6.
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** 10  
allowsOtherFileTypes **instance method** 10

## B

---

beginSheetForDirectory:file:modalForWindow:  
modalDelegate:didEndSelector:contextInfo:  
**instance method** 29  
beginSheetModalForWindow:completionHandler:  
**instance method** 11  
beginWithCompletionHandler: **instance method** 11  
Button tags 26

## C

---

cancel: **instance method** 12  
canCreateDirectories **instance method** 12  
canSelectHiddenExtension **instance method** 13

## D

---

delegate **instance method** 13  
directory **instance method** 30  
directoryURL **instance method** 13

## F

---

filename **instance method** 31

## I

---

isExpanded **instance method** 14  
isExtensionHidden **instance method** 14

## M

---

message **instance method** 14

## N

---

nameFieldLabel **instance method** 15  
nameFieldStringValue **instance method** 15  
NSFileHandlingPanelBrowser **constant** 27  
NSFileHandlingPanelCancelButton **constant** 27  
NSFileHandlingPanelDiskButton **constant** 27  
NSFileHandlingPanelDiskEjectButton **constant** 27  
NSFileHandlingPanelForm **constant** 27  
NSFileHandlingPanelHomeButton **constant** 27  
NSFileHandlingPanelImageButton **constant** 27  
NSFileHandlingPanelOKButton **constant** 27  
NSFileHandlingPanelTextField **constant** 27

## O

---

ok: **instance method** 16

## P

---

panel:compareFilename:with:caseSensitive:  
<NSObject> **delegate method** 34  
panel:directoryDidChange: <NSObject> **delegate  
method** 34  
panel:isValidFilename: <NSObject> **delegate  
method** 35

panel:shouldShowFilename: <NSObject> delegate  
method 36  
prompt instance method 16

## R

---

requiredFileType instance method 31  
runModal instance method 16  
runModalForDirectory:file: instance method 32

## S

---

savePanel class method 9  
selectText: instance method 29  
setAccessoryView: instance method 17  
setAllowedFileTypes: instance method 17  
setAllowsOtherFileTypes: instance method 18  
setCanCreateDirectories: instance method 19  
setCanSelectHiddenExtension: instance method 19  
setDelegate: instance method 20  
setDirectory: instance method 32  
setDirectoryURL: instance method 20  
setExtensionHidden: instance method 21  
setMessage: instance method 21  
setNameFieldLabel: instance method 22  
setNameFieldStringValue: instance method 22  
setPrompt: instance method 23  
setRequiredFileType: instance method 33  
setShowsHiddenFiles: instance method 23  
setTitle: instance method 24  
setTreatsFilePackagesAsDirectories: instance  
method 24  
showsHiddenFiles instance method 25

## T

---

title instance method 25  
treatsFilePackagesAsDirectories instance method  
25

## U

---

URL instance method 26

## V

---

validateVisibleColumns instance method 26