
Preference Panes Framework Reference

[Cocoa > User Experience](#)



2006-05-23



Apple Inc.
© 2006 Apple Computer, Inc.
All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, electronic, photocopying, recording, or otherwise, without prior written permission of Apple Inc., with the following exceptions: Any person is hereby authorized to store documentation on a single computer for personal use only and to print copies of documentation for personal use provided that the documentation contains Apple's copyright notice.

The Apple logo is a trademark of Apple Inc.

Use of the "keyboard" Apple logo (Option-Shift-K) for commercial purposes without the prior written consent of Apple may constitute trademark infringement and unfair competition in violation of federal and state laws.

No licenses, express or implied, are granted with respect to any of the technology described in this document. Apple retains all intellectual property rights associated with the technology described in this document. This document is intended to assist application developers to develop applications only for Apple-labeled computers.

Every effort has been made to ensure that the information in this document is accurate. Apple is not responsible for typographical errors.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010

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

Simultaneously published in the United States and Canada.

Even though Apple has reviewed this document, APPLE MAKES NO WARRANTY OR REPRESENTATION, EITHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT, ITS QUALITY, ACCURACY, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. AS A RESULT, THIS DOCUMENT IS PROVIDED "AS IS," AND YOU, THE READER, ARE ASSUMING THE ENTIRE RISK AS TO ITS QUALITY AND ACCURACY.

IN NO EVENT WILL APPLE BE LIABLE FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES RESULTING FROM ANY

DEFECT OR INACCURACY IN THIS DOCUMENT, even if advised of the possibility of such damages.

THE WARRANTY AND REMEDIES SET FORTH ABOVE ARE EXCLUSIVE AND IN LIEU OF ALL OTHERS, ORAL OR WRITTEN, EXPRESS OR IMPLIED. No Apple dealer, agent, or employee is authorized to make any modification, extension, or addition to this warranty.

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may also have other rights which vary from state to state.

Contents

Introduction **Introduction** 5

Part I **Classes** 7

NSPreferencePane Class Reference 9

Overview 9

Tasks 10

Instance Methods 11

Constants 19

Notifications 20

Document Revision History 23

Index 25

Introduction

Framework	/System/Library/Frameworks/PreferencePanes.framework
Header file directories	/System/Library/Frameworks/PreferencePanes.framework/Headers
Declared in	NSPreferencePane.h

The Preferences Panes framework defines the interface for subclasses to provide preference panes to System Preferences or other applications.

Classes

NSPreferencePane Class Reference

Inherits from	NSObject
Conforms to	NSObject (NSObject)
Framework	/System/Library/Frameworks/PreferencePanels.framework
Availability	Available in Mac OS X v10.1 and later.
Declared in	PreferencePanels/NSPreferencePane.h
Companion guide	Preference Panes
Related sample code	PrefsPane

Overview

NSPreferencePane is an abstract class that defines the interface for subclasses to provide preference panes to System Preferences or other applications.

Preference panes are subclasses of NSPreferencePane, packaged up in bundles and loaded by a preference application, such as System Preferences. These bundles have a suffix of `.prefPane`. Bundles intended for use by System Preferences are located in the `Library/PreferencePanels` directories of the various file system domains. See the chapter "File-System Domains" in *File System Overview* for information about domains.

The preference pane bundle normally contains a nib file with the user interface for modifying user preferences. The nib file contains a window assigned to the `_window` outlet of the preference pane instance (the nib's File's Owner). The NSPreferencePane implementation of `loadMainView` (page 14), invoked by the preference application, loads the nib file and uses the content view of `_window` as the preference pane's main view. Override this method if you need a different technique for creating the user interface.

The NSPreferencePane subclass is responsible for initializing the user interface with the current preference settings and recording any modifications the user makes. Through a series of `will...`, `did...`, and `should...` delegate methods, the preference application notifies the preference pane when the pane is selected (displayed) and deselected, allowing the pane to perform the necessary actions at the appropriate times. Implement these methods (and any additional target-action methods connected to the interface) as needed to produce the desired behavior for your preference pane.

Preference panes support Help menu items. You specify static help menu items under `NSPrefPaneHelpAnchors` in the bundle's `Info.plist` file; for dynamic items you implement `updateHelpMenuWithArray:` (page 18).

Tasks

Initializing the Preference Pane

- [initWithBundle:](#) (page 14)
Initializes the receiver with the preference pane's bundle, returning `self`.

Obtaining the Preference Pane Bundle

- [bundle](#) (page 12)
Returns the `NSBundle` instance representing the preference pane's bundle.

Setting Up the Main View

- [assignMainView](#) (page 11)
Locates and assigns the receiver's main view from the nib file loaded by [loadMainView](#) (page 14).
- [loadMainView](#) (page 14)
Loads the receiver's user interface into its main view.
- [mainNibName](#) (page 15)
Returns the name of the preference pane's nib file.
- [mainView](#) (page 15)
Returns the main view of the preference pane.
- [mainViewDidLoad](#) (page 16)
Notifies the receiver that the main view is set up and prepared to be displayed.
- [setMainView:](#) (page 17)
Sets the main view of the preference pane.

Handling Keyboard Focus

- [firstKeyView](#) (page 13)
Returns the first view in the keyboard focus chain.
- [initialKeyView](#) (page 13)
Returns the view that should have the keyboard focus when the pane is selected.
- [lastKeyView](#) (page 14)
Returns the last view in the keyboard focus chain.
- [setFirstKeyView:](#) (page 16)
Sets the first view in the keyboard focus chain.
- [setInitialKeyView:](#) (page 17)
Sets the view that should have keyboard focus when the pane is selected.
- [setLastKeyView:](#) (page 17)
Sets the last view in the keyboard focus chain.

- [autoSaveTextFields](#) (page 12)

Handling Preference Pane Selection

- [isSelected](#) (page 14)
Returns YES if the receiver is currently selected by the user, otherwise NO.
- [didSelect](#) (page 12)
Notifies the receiver that the main application has just displayed the preference pane's main view.
- [willSelect](#) (page 19)
Notifies the receiver that the main application is about to display the preference pane's main view.
- [didUnselect](#) (page 13)
Notifies the receiver that the main application has just stopped displaying the preference pane's main view.
- [replyToShouldUnselect:](#) (page 16)
Notifies the main application of the receiver's willingness (or unwillingness) to be deselected.
- [shouldUnselect](#) (page 18)
Returns a value indicating whether the receiver is willing to be deselected.
- [willUnselect](#) (page 19)
Notifies the receiver that the main application is about to stop displaying the preference pane's main view.

Help Menu Support

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

Instance Methods

assignMainView

Locates and assigns the receiver's main view from the nib file loaded by [loadMainView](#) (page 14).

- (void)assignMainView

Discussion

The default implementation sets the receiver's main view to the content view of the window referenced by the `_window` outlet. Before returning, `assignMainView` releases the window and sets the `_window` outlet to `nil`. Returns the main view if successful, `nil` otherwise.

Override this method if your main view is located in the nib file loaded by [loadMainView](#) (page 14), but is not the content view of a window in the file. Call [setMainView:](#) (page 17) to set the main view of the preference pane before returning. Also call [setInitialKeyView:](#) (page 17), [setFirstKeyView:](#) (page 16), and [setLastKeyView:](#) (page 17) to set the initial, first, and last keyboard focus views, respectively.

Availability

Available in Mac OS X v10.1 and later.

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

autoSaveTextFields

- (BOOL)autoSaveTextFields

Discussion

If this method returns YES, text fields are forced to give up their responder status before [shouldUnselect](#) (page 18) is called on the preference pane. If this method returns NO, the preference pane is responsible for forcing text fields to give up their responder status before saving them. The default return value is YES.

Availability

Available in Mac OS X v10.2 and later.

Declared In

NSPreferencePane.h

bundle

Returns the NSBundle instance representing the preference pane's bundle.

- (NSBundle *)bundle

Availability

Available in Mac OS X v10.1 and later.

Declared In

NSPreferencePane.h

didSelect

Notifies the receiver that the main application has just displayed the preference pane's main view.

- (void)didSelect

Discussion

Default implementation does nothing. Override this method to perform actions right after the main view has been placed into a window on the screen.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [willSelect](#) (page 19)

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

didUnselect

Notifies the receiver that the main application has just stopped displaying the preference pane's main view.

```
- (void)didUnselect
```

Discussion

Default implementation does nothing. Override this method to perform actions right after the main view has been removed from the screen.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [willUnselect](#) (page 19)

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

firstKeyView

Returns the first view in the keyboard focus chain.

```
- (NSView *)firstKeyView
```

Availability

Available in Mac OS X v10.1 and later.

See Also

- [setFirstKeyView:](#) (page 16)

Declared In

NSPreferencePane.h

initialKeyView

Returns the view that should have the keyboard focus when the pane is selected.

```
- (NSView *)initialKeyView
```

Availability

Available in Mac OS X v10.1 and later.

See Also

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

Declared In

NSPreferencePane.h

initWithBundle:

Initializes the receiver with the preference pane's bundle, returning `self`.

- (id)initWithBundle:(NSBundle *)*bundle*

Availability

Available in Mac OS X v10.1 and later.

Declared In

NSPreferencePane.h

isSelected

Returns YES if the receiver is currently selected by the user, otherwise NO.

- (BOOL)isSelected

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSPreferencePane.h

lastKeyView

Returns the last view in the keyboard focus chain.

- (NSView *)lastKeyView

Availability

Available in Mac OS X v10.1 and later.

See Also

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

Declared In

NSPreferencePane.h

loadMainView

Loads the receiver's user interface into its main view.

- (NSView *)loadMainView

Discussion

The default implementation loads the main nib file (identified by `mainNibName` (page 15)) and invokes `assignMainView` (page 11) to set the main view of the preference pane. Returns the main view if successful, `nil` otherwise.

Subclasses should rarely need to override this method. Override this method if you need to use a non-nib based technique for creating the main view. Call `setMainView:` (page 17) to set the main view of the preference pane before returning. Also call `setInitialKeyView:` (page 17), `setFirstKeyView:` (page 16), and `setLastKeyView:` (page 17) to set the initial, first, and last keyboard focus views, respectively.

Availability

Available in Mac OS X v10.1 and later.

See Also

- `mainViewDidLoad` (page 16)

Declared In

`NSPreferencePane.h`

mainNibName

Returns the name of the preference pane's nib file.

```
- (NSString *)mainNibName
```

Discussion

The name should not include the `.nib` extension.

The default implementation returns the value of the `NSMainNibFile` key in the bundle's information property list. If the key does not exist, it returns a default value of `@Main`.

Availability

Available in Mac OS X v10.1 and later.

See Also

- `loadMainView` (page 14)

Related Sample Code

`PrefsPane`

Declared In

`NSPreferencePane.h`

mainView

Returns the main view of the preference pane.

```
- (NSView *)mainView
```

Availability

Available in Mac OS X v10.1 and later.

See Also

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

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

mainViewDidLoad

Notifies the receiver that the main view is set up and prepared to be displayed.

- (void)mainViewDidLoad

Discussion

Invoked by the default implementation of [loadMainView](#) (page 14) after the main nib file has been loaded and the main view of the preference pane has been set. The default implementation does nothing. Override this method to initialize the main view with the current preference settings.

Availability

Available in Mac OS X v10.1 and later.

Declared In

NSPreferencePane.h

replyToShouldUnselect:

Notifies the main application of the receiver's willingness (or unwillingness) to be deselected.

- (void)replyToShouldUnselect:(BOOL)*shouldUnselect*

Discussion

If you override [shouldUnselect](#) (page 18) to return `NSUnselectLater`, you must invoke `replyToShouldUnselect:` when you have determined whether or not the preference pane can be deselected.

You should not override this method.

Availability

Available in Mac OS X v10.1 and later.

Declared In

NSPreferencePane.h

setFirstKeyView:

Sets the first view in the keyboard focus chain.

- (void)setFirstKeyView:(NSView *)*view*

Discussion

The first view can be set in the nib file by connecting a view to the receiver's `_firstKeyView` outlet.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [firstKeyView](#) (page 13)

Declared In

NSPreferencePane.h

setInitialKeyView:

Sets the view that should have keyboard focus when the pane is selected.

```
- (void)setInitialKeyView:(NSView *)view
```

Discussion

The initial view can be set in the nib file by connecting a view to the receiver's `_initialKeyView` outlet.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [initialKeyView](#) (page 13)

Declared In

NSPreferencePane.h

setLastKeyView:

Sets the last view in the keyboard focus chain.

```
- (void)setLastKeyView:(NSView *)view
```

Discussion

The last view can be set in the nib file by connecting a view to the receiver's `_lastKeyView` outlet.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [lastKeyView](#) (page 14)

Declared In

NSPreferencePane.h

setMainView:

Sets the main view of the preference pane.

```
- (void)setMainView:(NSView *)view
```

Discussion

You should not need to call this directly unless you override [loadMainView](#) (page 14) or [assignMainView](#) (page 11).

Availability

Available in Mac OS X v10.1 and later.

See Also

- [mainView](#) (page 15)

Declared In

NSPreferencePane.h

shouldUnselect

Returns a value indicating whether the receiver is willing to be deselected.

```
- (NSPreferencePaneUnselectReply)shouldUnselect
```

Discussion

The possible return values are described in “[Constants](#)” (page 19). The default implementation always returns `NSUnselectNow`. Override this method if your pane needs to cancel or delay a deselect action. If you override this method to return `NSUnselectLater`, you must invoke [replyToShouldUnselect:](#) (page 16) when you have determined whether or not the deselection can occur.

Availability

Available in Mac OS X v10.1 and later.

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

updateHelpMenuWithArray:

```
- (void)updateHelpMenuWithArray:(NSArray *)arrayOfMenuItems
```

Discussion

Call this method if you need to update help menu items dynamically. If you have static help menu items, you should not use this method. Specify them under the `NSPrefPane1HelpAnchors` key in the bundle’s `Info.plist` instead.

The array contains dictionaries with two keys. Use `title` for the help menu item title, and `anchor` for the anchor reference for `AHLookupAnchor`.

Availability

Available in Mac OS X v10.3 and later.

Declared In

NSPreferencePane.h

willSelect

Notifies the receiver that the main application is about to display the preference pane's main view.

- (void)willSelect

Discussion

Default implementation does nothing. Override this method to perform actions right before the main view is displayed.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [didSelect](#) (page 12)

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

willUnselect

Notifies the receiver that the main application is about to stop displaying the preference pane's main view.

- (void)willUnselect

Discussion

Default implementation does nothing. Override this method to perform actions right before the main view is removed from the screen.

Availability

Available in Mac OS X v10.1 and later.

See Also

- [didUnselect](#) (page 13)

Related Sample Code

PrefsPane

Declared In

NSPreferencePane.h

Constants

The following constants are defined by NSPreferencePane for use as the return values of [shouldUnselect](#) (page 18):

Constant	Description
<code>NSUnselectCancel</code>	Cancel the deselection Available in Mac OS X v10.1 and later. Declared in <code>NSPreferencePane.h</code> .
<code>NSUnselectNow</code>	Continue the deselection Available in Mac OS X v10.1 and later. Declared in <code>NSPreferencePane.h</code> .
<code>NSUnselectLater</code>	Delay the deselection until the preference pane invokes replyToShouldUnselect: (page 16) Available in Mac OS X v10.1 and later. Declared in <code>NSPreferencePane.h</code> .

The following constants are defined by `NSPreferencePane` for use in the input parameter of [updateHelpMenuWithArray:](#) (page 18):

Constant	Description
<code>title</code>	Key for the help menu item in the dictionary supplied as an argument to updateHelpMenuWithArray: (page 18)
<code>anchor</code>	Key for the anchor reference for <code>AHLookupAnchor</code> in the dictionary supplied as an argument to updateHelpMenuWithArray: (page 18)

Notifications

NSPreferencePaneDoUnselectNotification

Notifies the main application that the preference pane is willing to be deselected. Posted when [replyToShouldUnselect:](#) (page 16) is invoked with an argument of YES after [shouldUnselect](#) (page 18) has returned a value of `NSUnselectLater`.

Availability

Available in Mac OS X v10.1 and later.

Declared In

`NSPreferencePane.h`

NSPreferencePaneCancelUnselectNotification

Notifies the main application that the preference pane is not willing to be deselected. Posted when [replyToShouldUnselect:](#) (page 16) is invoked with an argument of NO after [shouldUnselect](#) (page 18) has returned a value of `NSUnselectLater`.

Availability

Available in Mac OS X v10.1 and later.

Declared In

NSPreferencePane.h

Document Revision History

This table describes the changes to *Preference Panes Framework Reference*.

Date	Notes
2006-05-23	First publication of this content as a collection of separate documents
	First publication of this content as a collection of separate documents
2005-04-29	Added new custom Help menu properties.
2003-08-22	Added new API (autoSaveTextFields, isSelected, updateHelpMenuWithArray); new constants (kNSPrefPaneHelpMenuAnchorKey and kNSPrefPaneHelpMenuTitleKey); and updated introduction to cover Help support. Also added Revision History page.

REVISION HISTORY

Document Revision History

Index

A

anchor **constant** [20](#)
assignMainView **instance method** [11](#)
autoSaveTextFields **instance method** [12](#)

B

bundle **instance method** [12](#)

D

didSelect **instance method** [12](#)
didUnselect **instance method** [13](#)

F

firstKeyView **instance method** [13](#)

I

initialKeyView **instance method** [13](#)
initWithBundle: **instance method** [14](#)
isSelected **instance method** [14](#)

L

lastKeyView **instance method** [14](#)
loadMainView **instance method** [14](#)

M

mainNibName **instance method** [15](#)
mainView **instance method** [15](#)
mainViewDidLoad **instance method** [16](#)

N

NSPreferencePaneCancelUnselectNotification
notification [20](#)
NSPreferencePaneDoUnselectNotification
notification [20](#)
NSUnselectCancel **constant** [20](#)
NSUnselectLater **constant** [20](#)
NSUnselectNow **constant** [20](#)

R

replyToShouldUnselect: **instance method** [16](#)

S

setFirstKeyView: **instance method** [16](#)
setInitialKeyView: **instance method** [17](#)
setLastKeyView: **instance method** [17](#)
setMainView: **instance method** [17](#)
shouldUnselect **instance method** [18](#)

T

title **constant** [20](#)

U

updateHelpMenuWithArray: [instance method 18](#)

W

willSelect [instance method 19](#)

willUnselect [instance method 19](#)