

[AppKit](#) / Windows, Panels, and Screens

API Collection

Windows, Panels, and Screens

Organize your view hierarchies and facilitate their display onscreen.

Topics

Windows

`class` `NSWindow`

A window that an app displays on the screen.

`class` `NSPanel`

A special kind of window that typically performs a function that is auxiliary to the main window.

`protocol` `NSWindowDelegate`

A set of optional methods that a window's delegate can implement to respond to events, such as window resizing, moving, exposing, and minimizing.

`class` `NSWindowTab`

A tab associated with a window that is part of a tabbing group.

`class` `NSWindowTabGroup`

A group of windows that display together as a single tabbed window.

Window Restoration

`protocol` `NSWindowRestoration`

A set of methods that restoration classes must implement to handle the recreation of windows.

`protocol NSUserInterfaceItemIdentification`

A set of methods used to associate a unique identifier with objects in your user interface.

Screens

`class NSScreen`

An object that describes the attributes of a computer's monitor or screen.

Popovers

`class NSPopover`

A means to display additional content related to existing content on the screen.

`protocol NSPopoverDelegate`

A set of optional methods that a popover delegate can implement to provide additional or custom functionality.

Alerts

`class NSAlert`

A modal dialog or sheet attached to a document window.

`protocol NSAlertDelegate`

A set of optional methods implemented by the delegate of an [NSAlert](#) object to respond to a user's request for help.

Open and Save Panels

`class NSOpenPanel`

A panel that prompts the user to select a file to open.

`class NSSavePanel`

A panel that prompts the user for information about where to save a file.

`protocol NSOpenSavePanelDelegate`

A set of methods for managing interactions with an open or save panel.

Share Panel

`class NSSharingServicePicker`

A list of sharing services that the user can choose from.

`protocol NSPreviewRepresentableActivityItem`

An interface you adopt in custom objects that you want to share using the macOS share sheet.

`class NSPreviewRepresentingActivityItem`

A type that adds metadata to an item you share using the macOS share sheet.

Print and PDF Panels

`class NSPDFPanel`

A Save or Export as PDF panel that's consistent with the macOS user interface.

`protocol NSPrintPanelAccessorizing`

A set of methods that a Print panel object can use to get information from a printing accessory controller.

Color Panels

`class NSColorPanel`

A standard user interface for selecting color in an app.

`protocol NSColorPickingCustom`

A set of methods that provides a way to add color pickers—custom user interfaces for color selection—to an app's color panel.

`protocol NSColorPickingDefault`

A set of methods that provides basic behavior for a color picker.

`class NSColorPicker`

An abstract superclass that implements the default color picking protocol.

Font Panels

`class NSFontPanel`

The Font panel—a user interface object that displays a list of available fonts, letting the user preview them and change the font used to display text.

`struct ModeMask`

☰ NSFontPanelValidation

A set of methods you use to tell the Font panel to display some or all of its elements.

protocol NSFontChanging

See Also

User Interface

☰ Views and Controls

Present your content onscreen and handle user input and events.

☰ View Management

Manage your user interface, including the size and position of views in a window.

☰ View Layout

Position and size views using a stack view or Auto Layout constraints.

☰ Appearance Customization

Add Dark Mode support to your app, and use appearance proxies to modify your UI.

☰ Animation

Animate your views and other content to create a more engaging experience for users.

☰ Sound, Speech, and Haptics

Play sounds and haptic feedback, and incorporate speech recognition and synthesis into your interface.

📄 Supporting Continuity Camera in Your Mac App

Incorporate scanned documents and pictures from a user's iPhone, iPad, or iPod touch into your Mac app using Continuity Camera.