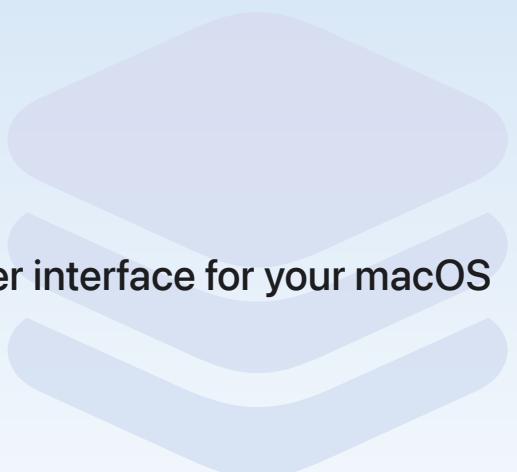


Framework

AppKit

Construct and manage a graphical, event-driven user interface for your macOS app.

Mac Catalyst 13.0+ | macOS 10.0+



Overview

AppKit contains the objects you need to build the user interface for a macOS app. In addition to drawing windows, buttons, panels, and text fields, it handles all the event management and interaction between your app, people, and macOS.



Aside from drawing and managing interactions, AppKit handles printing, animating, as well as creating documents with large amounts of data efficiently. The framework also contains built-in support for localization and accessibility to ensure that your app reaches as many people as possible.

AppKit also works with [SwiftUI](#), so you can implement parts of your AppKit app in SwiftUI or mix interface elements between the two frameworks. For example, you can place AppKit views and view controllers inside SwiftUI views, and vice versa.

Note

For information about bringing your iPad app to Mac, see [Mac Catalyst](#). To build an iOS app, you can use SwiftUI to create an app that works across all of Apple's platforms, or use [UIKit](#) to create an app for iOS only.

Topics

Essentials

Adopting Liquid Glass

Find out how to bring the new material to your app.

AppKit updates

Learn about important changes to AppKit.

Protecting the User's Privacy

Secure personal data, and respect user preferences for how data is used.

Porting your macOS apps to Apple silicon

Create a version of your macOS app that runs on both Apple silicon and Intel-based Mac computers.

App Structure

App and Environment

Learn about the objects that you use to interact with the system.

Documents, Data, and Pasteboard

Organize your app's data and preferences, and share that data on the pasteboard or in iCloud.

- ☰ Cocoa Bindings
 - Automatically synchronize your data model with your app's interface using Cocoa Bindings.
- ☰ Resource Management
 - Manage the storyboards and nib files containing your app's user interface, and learn how to load data that is stored in resource files.
- ☰ App Extensions
 - Extend your app's basic functionality to other parts of the system.

User Interface

Your app's user interface provides visual, audible, and tactile feedback to the user about what your app is doing.

- ☰ 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.
- ☰ Windows, Panels, and Screens
 - Organize your view hierarchies and facilitate their display onscreen.
- ☰ 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.

User Interactions

Mouse, Keyboard, and Trackpad

Handle events related to mouse, keyboard, and trackpad input.

Menus, Cursors, and the Dock

Implement menus and cursors to facilitate interactions with your app, and use your app's Dock tile to convey updated information.

Gestures

Encapsulate your app's event-handling logic in gesture recognizers so that you can reuse that code throughout your app.

Touch Bar

Display interactive content and controls in the Touch Bar.

Drag and Drop

Support the direct manipulation of your app's content using drag and drop.

Accessibility for AppKit

Make your AppKit apps accessible to everyone who uses macOS.

Graphics, Drawing, Color, and Printing

Images and PDF

Create and manage images, in bitmap, PDF, and other formats.

Drawing

Draw shapes, images, and other content on the screen.

Color

Represent colors using built-in or custom formats, and give users options for selecting and applying colors.

Printing

Display the system print panels and manage the printing process.

Text

Text Display

Display text and check spelling.

TextKit

Manage text storage and perform custom layout of text-based content in your app's views.

Fonts

Manage the fonts used to display text.

Writing Tools

Add support for Writing Tools to your app's text views.

Deprecated

Avoid using deprecated classes and protocols in your apps.

Deprecated Symbols

Review symbols that are no longer supported, and find the replacements to use instead.

Reference

Enumerations

Enumerations for use with multiple classes.

Constants

Constants for use with multiple classes.

Data Types

Data types for use with multiple classes.

Macros

Macros for use with multiple classes.

Classes

`class NSScrollViewEdgeEffectStyle`

Styles for a scroll view's edge effect.