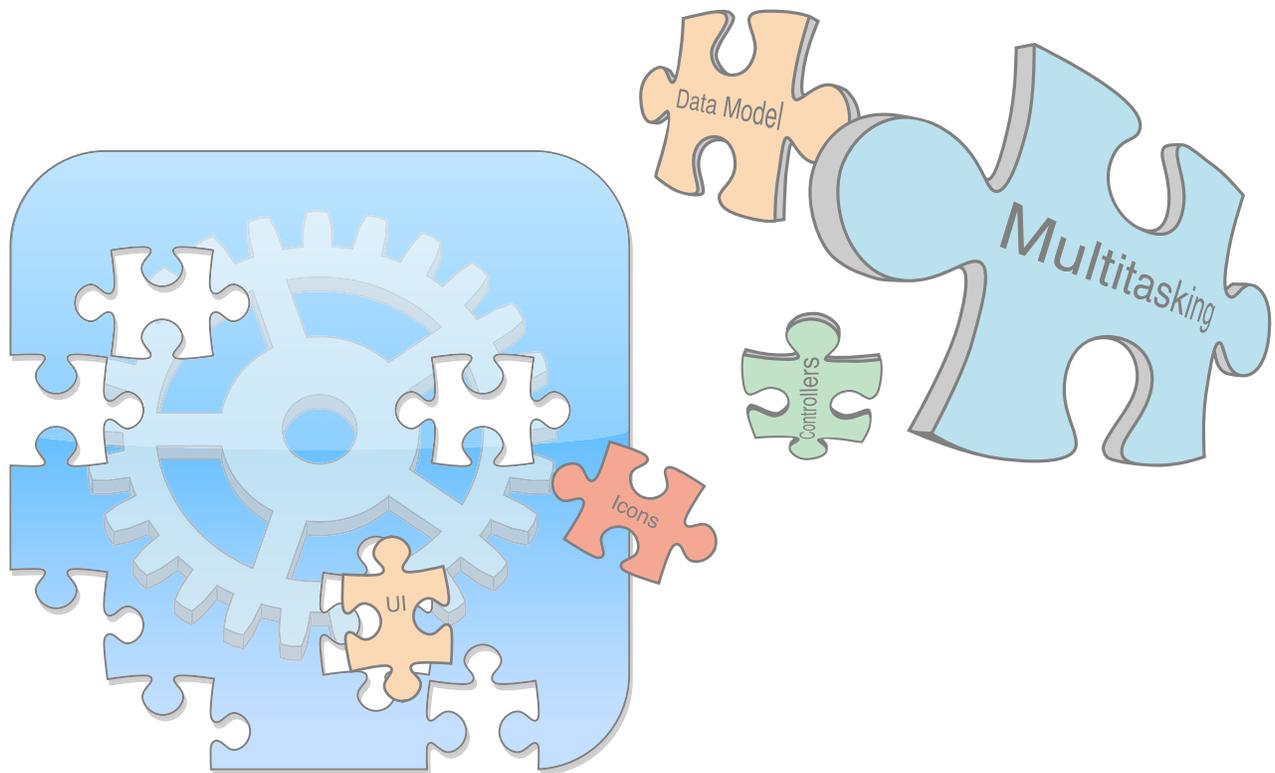


About iOS App Programming

This document is the starting point for creating iOS apps. It describes the fundamental architecture of iOS apps, including how the code you write fits together with the code provided by iOS. This document also offers practical guidance to help you make better choices during your design and planning phase and guides you to the other documents in the iOS developer library that contain more detailed information about how to address a specific task.



The contents of this document apply to all iOS apps running on all types of iOS devices, including iPad, iPhone, and iPod touch.

Note: Development of iOS apps requires an Intel-based Macintosh computer with the iOS SDK installed. For information about how to get the iOS SDK, go to the [iOS Dev Center](#).

At a Glance

The starting point for any new app is identifying the design choices you need to make and understanding how those choices map to an appropriate implementation.

Translate Your Initial Idea into an Implementation Plan

Every great iOS app starts with a great idea, but translating that idea into actions requires some planning. Every iOS app relies heavily on design patterns, and those design patterns influence much of the code you need to write. So before you write any code, take the time to explore the possible techniques and technologies available for writing that code. Doing so can save you a lot of time and frustration.

Relevant Chapter: [“App Design Basics”](#) (page 13)

UIKit Provides the Core of Your App

The core infrastructure of an iOS app is built from objects in the UIKit framework. The objects in this framework provide all of the support for handling events, displaying content on the screen, and interacting with the rest of the system. Understanding the role these objects play, and how you modify them to customize the default app behavior, is therefore very important for writing apps quickly and correctly.

Relevant Chapter: [“Core App Objects”](#) (page 21)

Apps Must Behave Differently in the Foreground and Background

An iOS device runs multiple apps simultaneously but only one app—the foreground app—has the user’s attention at any given time. The current foreground app is the only app allowed to present a user interface and respond to touch events. Other apps remain in the background, usually asleep but sometimes running additional code. Transitioning between the foreground and background states involves changing several aspects of your app’s behavior.

Relevant Chapter: [“App States and Multitasking”](#) (page 37)

iCloud Affects the Design of Your Data Model and UI Layers

iCloud allows you to share the user’s data among multiple instances of your app running on different iOS and Mac OS X devices. Incorporating support for iCloud into your app involves changing many aspects of how you manage your files. Because files in iCloud are accessible by more than just your app, all file operations must be synchronized to prevent data corruption. And depending on your app and how it presents its data, iCloud can also require changes to portions of your user interface.

Relevant Chapter: [“Integrating iCloud Support Into Your App”](#) (page 30)

Apps Require Some Specific Resources

There are some resources that must be present in all iOS apps. Most apps include images, sounds, and other types of resources for presenting the app’s content but the App Store also requires some specific resources be present. The reason is that iOS uses several specific resources when presenting your app to the user and when coordinating interactions with other parts of the system. So these resources are there to improve the overall user experience.

Relevant Chapter: [“App-Related Resources”](#) (page 98)

Apps Should Restore Their Previous UI State at Launch Time

At launch time, your app should restore its user interface to the state it was in when it was last used. During normal use, the system controls when apps are terminated. Normally when this happens, the app displays its default user interface when it is relaunched. With state restoration, UIKit helps your app restore your app’s interface to its previous state, which promotes a consistent user experience.

Relevant Chapter: [“State Preservation and Restoration”](#) (page 72)

Many App Behaviors Can Be Customized

The core architecture of all apps may be the same, but there are still ways for you to tweak the high-level design of your app. Some of these tweaks are how you add specific high-level features, such as data protection and URL handling. Others affect the design of specific types of apps, such as VoIP apps.

Relevant Chapter: [“Advanced App Tricks”](#) (page 113)

Apps Must Be Tuned for Performance

Great apps are always tuned for the best possible performance. For iOS apps, performance means more than just writing fast code. It often means writing better code so that your user interface remains responsive to user input, your app does not degrade battery life significantly, and your app does not impact other system resources. Before you can tune your code, though, learn about the types of changes that are likely to provide the most benefit.

Relevant Chapter: [“Performance Tuning”](#) (page 132)

The iOS Environment Affects Many App Behaviors

There are aspects of iOS itself that impact how you design and write applications. Because iOS is built for mobile devices, it takes a more active role in providing security for apps. Other system behaviors also affect everything from how memory is managed to how the system responds to hardware input. All of these system behaviors affect the way you design your apps.

Relevant Appendix: [“The iOS Environment”](#) (page 142)

How to Use This Document

This document provides important information about the core objects of your app and how they work together. This document does not address the creation of any specific type of iOS app. Instead, it provides a tour of the architecture that is common to all iOS apps and highlights key places where you can modify that architecture to meet your needs. Whenever possible, the document also offers tips and guidance about ways to implement features related to the core app architecture.

Prerequisites

This document is the entry-point guide for designing an iOS app. This guide also covers many of the practical aspects involved with implementing your app. However, this book assumes that you have already installed the iOS SDK and configured your development environment. You must perform those steps before you can start writing and building iOS apps.

If you are new to iOS app development, read *Start Developing iOS Apps Today*. This document offers a step-by-step introduction to the development process to help you get up to speed quickly. It also includes a hands-on tutorial that walks you through the app-creation process from start to finish, showing you how to create a simple app and get it running quickly.

See Also

For additional information related to app design, see the following documents:

- For guidance about how to design an iOS app, read *iOS Human Interface Guidelines*. This book provides you with tips and guidance about how to create a great experience for users of your app. It also conveys the basic design philosophy surrounding iOS apps.
- If you are not sure what is possible in an iOS app, read *iOS Technology Overview*. This book provides a summary of iOS technologies and the situations where you might want to use them. This book is not required reading but is a good reference during the brainstorming phase of your project.