Mac Catalyst Essentials

Turning on Mac Catalyst

Before you create a Mac version of an existing iPad app, you must configure the app's Xcode project to use Mac Catalyst. This tutorial guides you through the configuration steps for Recipes, a sample iOS app serving up simple dishes to prepare and cook.

Download the project files and follow these steps to configure the Recipes project.

10 mins Estimated Time | Project Files | Xcode 12.2 or later

Section 1
Add Support for Mac

In this section, you'll turn on Mac Catalyst and set the deployment target to the latest version of macOS.

Important

Before turning on Mac Catalyst, replace any deprecated method calls your app makes, the deprecated calls may not be available in macOS. Recipes doesn't call any deprecated methods, so it's ready for you to turn on Mac Catalyst.

Step 1

In the Project navigator, select the project to view the Project editor.

Note

The screenshot shows the project name as Recipes, but you may see a different name. Xcode uses the project name you specified for this tutorial and use the starting point project, the name you see is `turning-on-mac-catalyst-start`. Xcode adds the Xcode project extension to a project's project files when the tutorial name will a "-complete" or "-starting point" suffix as the name of the Xcode project.

Step 2

Select Recipes from the target list in the Project editor to view the target settings.

Step 3

In the target settings, click General to view the general settings of the target.

In the Deployment Info section, add Mac as a supported device and enable Mac support.

Step 4

Select the Mac checkbox.

Xcode displays an "Enable Mac support?" confirmation after you select the Mac checkbox.

Important

Enabling Mac support doesn't change your source code, but you may need to make changes before Xcode can build your Mac app. Some iOS-only frameworks and iPad app uses might not be available in macOS, causing compile errors when you try to build. Mac Catalyst automatically adapts the source code from the Mac version of your app. The Recipes app uses only available frameworks, so it's ready for you to enable Mac support.

Step 5

Click Enable to add Mac support to the iOS target.

When you click Enable, Xcode adds the Recipes.entitlements file to the project. The file contains entitlements for App Sandbox and outgoing network connections. Xcode also adds the Camera entitlement to the file. Recipes uses the camera to access the NSCameraUsageDescription key.

Step 6

Select macOS 11 as the deployment target for the Mac version of the app.

Note

The Recipes app targets the latest macOS version to take advantage of new features, but the iPhone and iPad versions of the app will run in earlier versions of iOS.

Tip

Select the option Scale Interface to Match iPad.

Scale Interface to Match iPad is the default option. It provides a simplified approach for bringing an iPad app to the Mac.

Section 2
Code Signing the App

Before you build the app, set the development team and signing certificate that Xcode uses to code sign the app. Code signing assures users that the app comes from you, and no one has modified the app since you last signed it.

Step 1

Select the Recipes project. Then select the Recipes target, and click Signing & Capabilities.

Step 2

In the Signing & Capabilities editor, select your development team.

Tip

If you don't have a development team, select None.

Step 3

Next, select Development as the Signing Certificate.

Tip

If you selected None as the development team, select Sign to Run Locally for the Signing Certificate.

Section 3
Build and Run the Mac App

With the Xcode project set up to use Mac Catalyst, build and run the Recipes app on your Mac, and take a look at the features the new version supports.

Step 1

Choose Product > Destination > My Mac.

Setting the Build destination to My Mac builds a local version of the Recipes app.

Check for build errors before running the app.

Step 2

Choose Product > Build.

After Xcode builds the app, it's time to run it. Launch the app on your Mac.

Step 3

Choose Product > Run.

Important

Recipes requires macOS 11 or later to run. The app doesn't run on earlier versions of macOS.

Step 4

Review the Recipes app.

The app supports many system features of macOS without additional source code changes. Recipes supports the menu bar, resizable window, and support for trackpad, mouse, and keyboard input.

Check Your Understanding

Question 1 of 2

What is the first thing to do when preparing a project for Mac Catalyst?

- Replace calls made to deprecated APIs with calls to the updated APIs.
- Add a Build Phase to the Xcode project that scans for incompatible frameworks.
- Select the Mac checkbox in the Deployment Info section of the iOS target's general settings.

Submit | Next question

Next
Creating a Sidebar

iOS apps often display a tab bar at the bottom of a device's screen, but that design can look out of place on the Mac. To give the Recipes app a layout and appearance more consistent with the Mac, replace the tab bar with a sidebar.

Get started