

[Xcode](#) / Creating an Xcode project for an app

Article

# Creating an Xcode project for an app

Start developing your app by creating an Xcode project from a template.

## Overview

To begin writing code quickly, create an Xcode project from a template that contains the essential project configuration and files for the platform and type of app you want to create. For example, select a template and options for developing a multiplatform SwiftUI app, macOS command-line tool, iOS augmented reality app, or visionOS immersive environment app.

## Prepare configuration information

Before you create a project, collect the information that Xcode needs to identify your app and you as a developer:

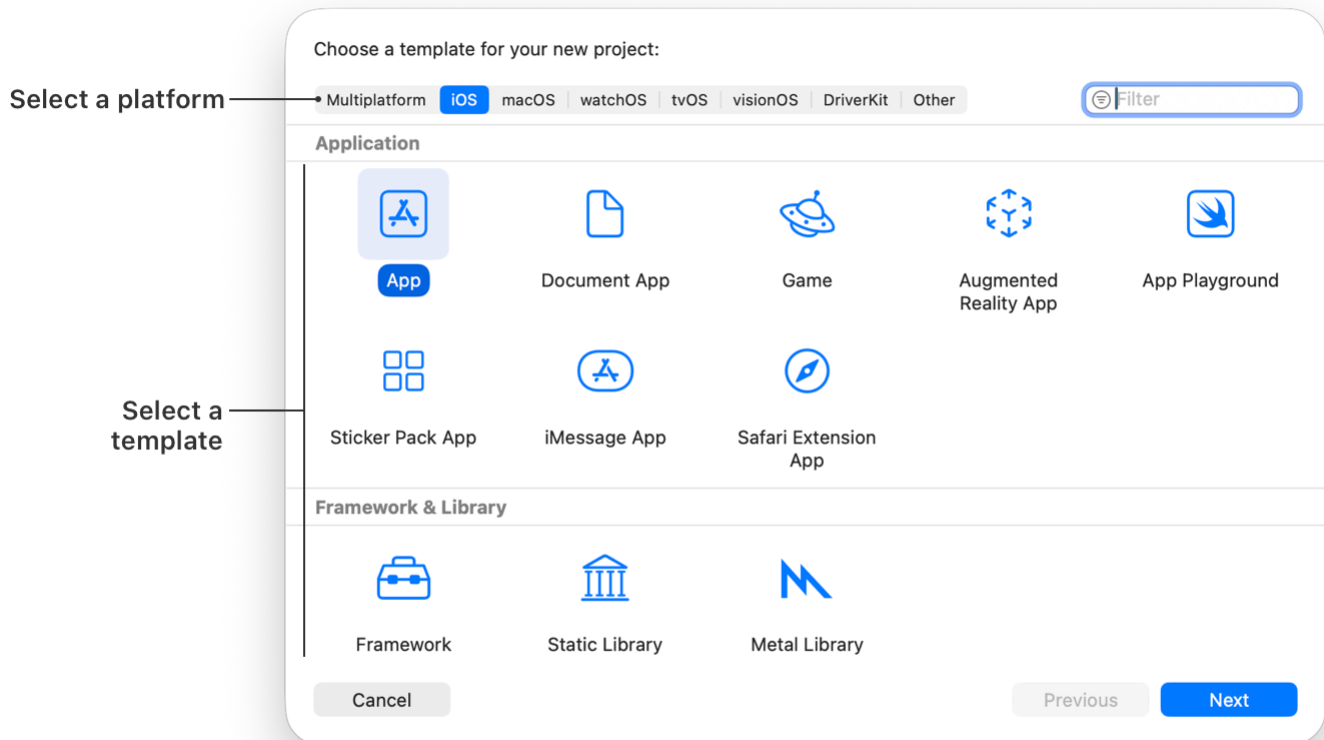
- **Product name.** The name of your app as you want it to appear in the App Store and on a device when a person installs it. Choose a product name that has a minimum length of 2 and no more than 30 characters, and is similar to the app name that you enter later in App Store Connect. For more information, see [App information](#) in App Store Connect Help.
- **Team.** The email or phone number for the Apple account that you want to use as a developer if you haven't already added it to Xcode > Settings > Apple Accounts.
- **Organization identifier.** A reverse DNS string that uniquely identifies your organization. If you don't have a company identifier, use `com.example.` followed by your organization name, and replace it before you distribute your app.

## Important

The organization identifier is part of the bundle ID (`CFBundleIdentifier`) by default. Xcode uses the bundle ID to register an App ID when you first run your app on a device. The number of App IDs are limited if you are not a member of the Apple Developer Program, and you can't change the App ID after you upload a build to App Store Connect, so choose the organization identifier carefully.

## Create a project

Launch Xcode, then click Create New Project in the Xcode window or choose File > New > Project. In the sheet that appears, select a specific platform or Multiplatform for an app that runs on all platforms. Then select a template under Application depending on the platform you select.



If you see a banner that says you don't have support for a platform, you can create the project, but you can't build and run it on a device. To install the platform now, click the Get button on the right of the banner. Otherwise, you can manage downloads in the Components settings later (see [Downloading and installing additional Xcode components](#)).

In the following sheet, enter a *product name* and *organization identifier* that Xcode uses to create the *bundle identifier* that identifies your app throughout the system. Optionally, choose an account from the Team pop-up menu that Xcode uses to code sign your app.

Choose options for your new project:

Product Name: HelloWorld

Team: Ravi Patel

Organization Identifier: com.example.rpatel

Bundle Identifier: com.example.rpatel.HelloWorld

Interface: SwiftUI

Language: Swift

Testing System: Swift Testing with XCTest UI Tests

Storage: None

☐ Host in CloudKit

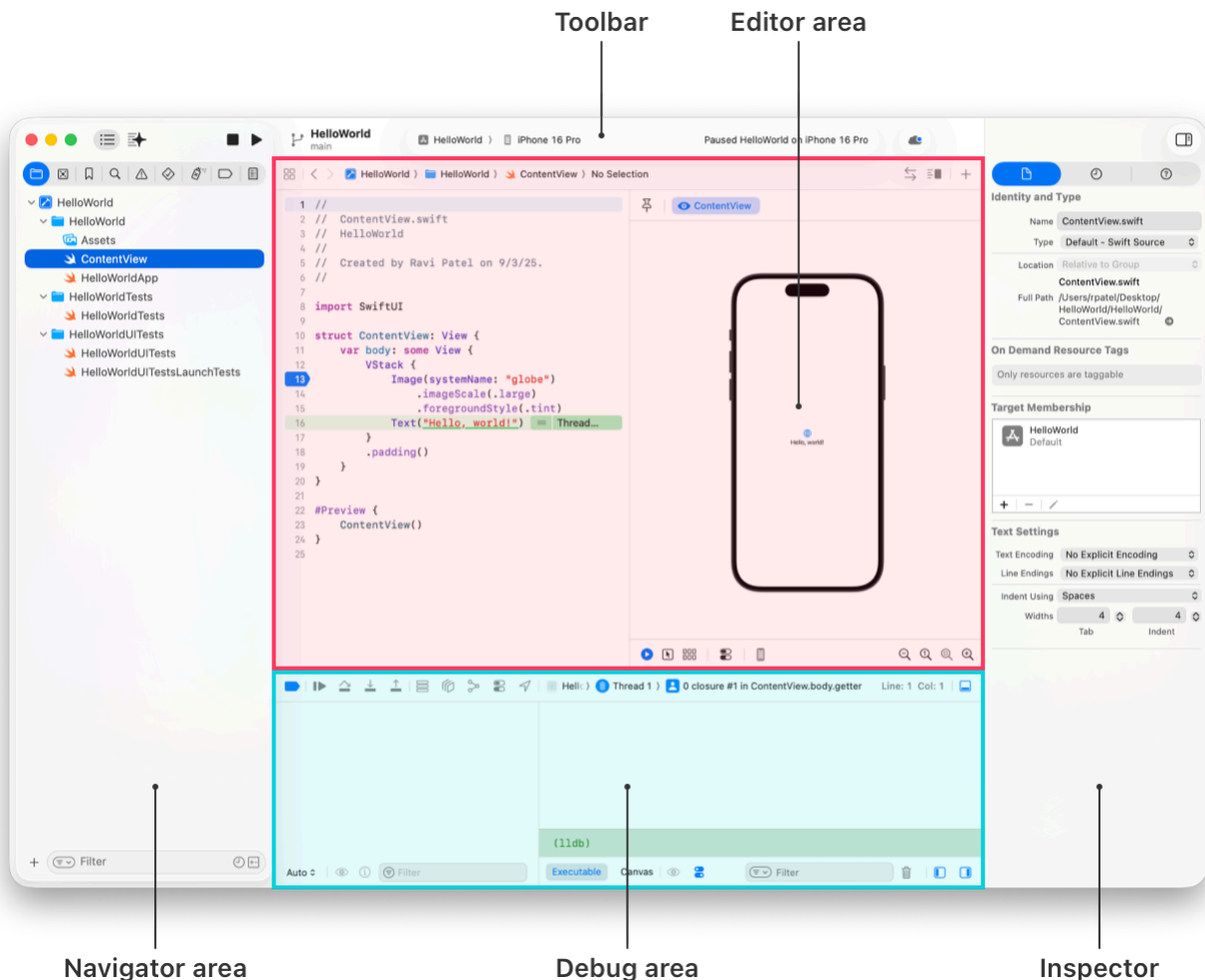
Cancel Previous Next

In the sheets that appear, choose other options from pop-up menus depending on the template you select, such as testing systems, storage, interface, and language. In the last sheet, choose a location for your project, select other options, and click Create.

## Manage files in the main window

When you create a project or open an existing project, the *main window* appears, showing the necessary files and resources for developing your app.

You can access different parts of your project from the *navigator area* on the left. Use the *Project navigator* to select files you want to edit in the *editor area*. For example, when you select a Swift file in the Project navigator, the file opens in the *source editor*, where you can modify the code and set breakpoints.



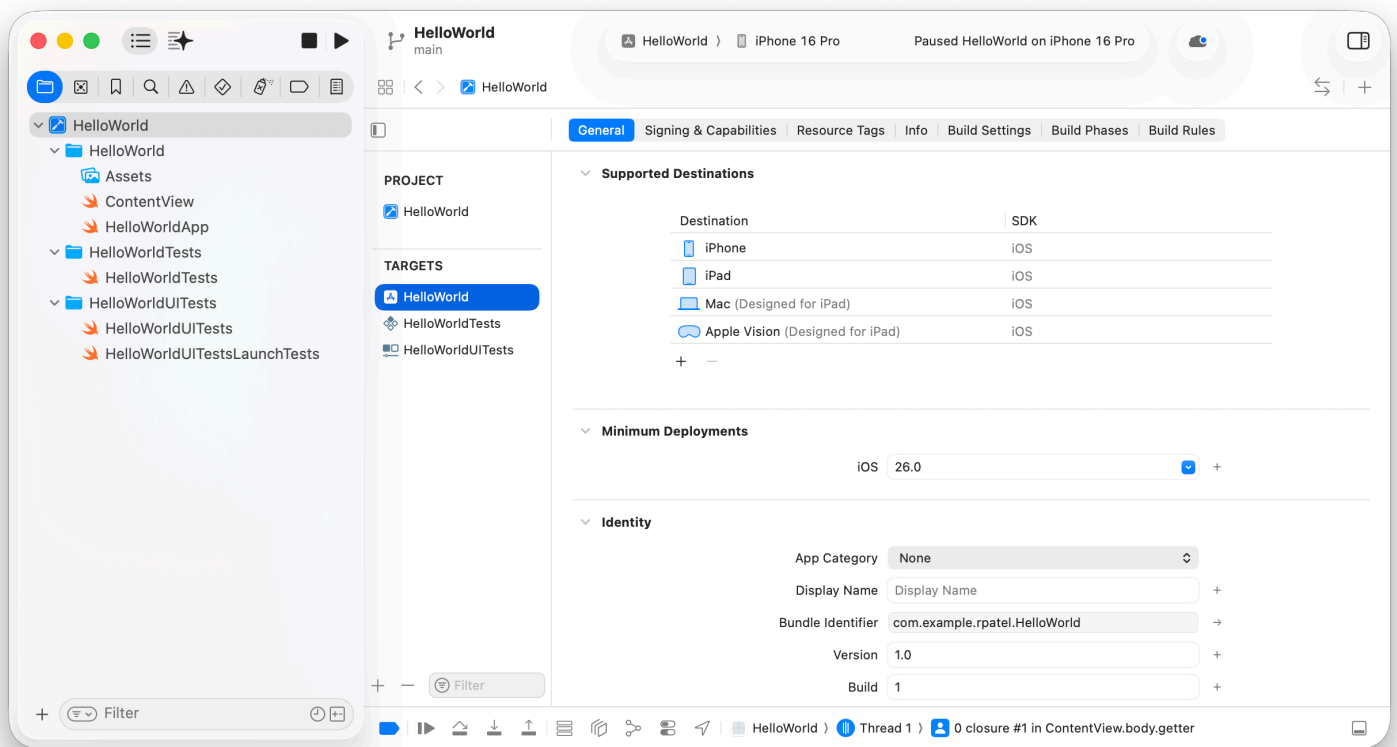
Details about the selected file also appear in the *inspector area* on the right. In the inspector area, you can select the File inspector to edit properties of a file. If you want to hide the inspector to make more room for the editor, click the “Hide or show the Inspectors” button in the upper-right corner of the toolbar.

You use the *toolbar* to build and run your app on a simulated or real device. For iOS apps, choose the app target and a simulator or connected device from the run destination menu in the toolbar, then click the Run button. For macOS apps, just click the Run button.

When your app launches, the *debug area* opens, where you can control the execution of your app and inspect variables. When the app stops at the breakpoint, use the controls in the debug area to step through the code or continue execution. When you are done running the app, click the Stop button in the toolbar.

If you use SwiftUI, you can see an interactive preview of the user interface while you create your app. Xcode keeps the changes you make in the source file, the canvas on the right, and the inspector in sync. You can use the controls in the preview to run the app with the debugger, too. For more information, see [Creating your app’s interface with SwiftUI](#).

To change properties you enter when creating your project, select the project name in the Project navigator that appears at the top, then the *project editor* opens in the editor area. Most of the properties you entered appear on the *General pane* of the project editor.







## Note

Most templates are preconfigured with an asset catalog for the app icon, but you can use a multilayer Icon Composer file that supports [Liquid Glass](#) instead. For more information, see [Creating your app icon using Icon Composer](#).

# See Also

## Essentials

-  **Creating your app's interface with SwiftUI**  
Develop apps in SwiftUI with an interactive preview that keeps the code and layout in sync.
-  **Previewing your app's interface in Xcode**  
Iterate designs quickly and preview your apps' displays across different Apple devices.
-  **Building and running an app**  
Compile your source files and assemble an app bundle to run on a device or simulator.
-  **Xcode updates**

Learn about important changes to Xcode.