**iOS Development on MAC with XCode**

<https://help.apple.com/xcode/mac/current/#/dev42b289fbc>

**Create XCode project**

1. Start XCode and create a project with the same name as the HTML project.

|  |  |
| --- | --- |
|  | 1. Choose the programming language as Javascript. 2. Enter Tean, Organization and the rest. |

1. To get started with PhoneGap, you first need to download phonegap.
2. When the download completes, unarchive it to your desktop. PhoneGap download contains a bunch of device-specific directories (e.g., android, iphone, blackberry, windows mobile), and some library and utility files and directories. The only one we’ll be looking at is the iphone directory.
3. The iphone directory contains the starter files for an Xcode project.

|  |  |
| --- | --- |
|  |  |

1. Inside the iphone directory, there is a directory named www. You can think of this as the web root of the application. By default, it contains two sample files named index.html and master.css. These are used as the demo PhoneGap application. We don’t need them, so you can delete them both.

|  |  |
| --- | --- |
|  | 1. Next, copy all of your HTML project files into the www directory (on the Mac, hold down Option while dragging files to make a copy). 2. Don’t change your folder structure or naming; just drop everything in there as is. 3. If you have added a manifest link to the html tag in index.html as described in Chapter 6, you must remove it. It’s NOT necessary when using PhoneGap and may cause performance problems. |

1. Next, go into your index.html file, add the following line to the section, and save the file:

*<script type="text/javascript" src="phonegap.js" charset="utf-8"></script>*

1. You don’t need to copy the phonegap.js file into your www directory. When you build your app, Xcode takes care of this for you.

|  |  |
| --- | --- |
|  | 1. Copy your entire web app into the www directory. Make sure the main page for your app is named index.html; otherwise, PhoneGap won’t know what file to launch. 2. Open the project in Xcode by double-clicking the PhoneGap.xcodeproj file in the Finder. 3. Once the project window is open, make sure you have the most recent version of the iPhone Simulator (3.1.2 as of this writing) selected as your active SDK and then click the Build and Run button (Figure 7-7). 4. After about 10 seconds, the iPhone Simulator should appear and launch your app |

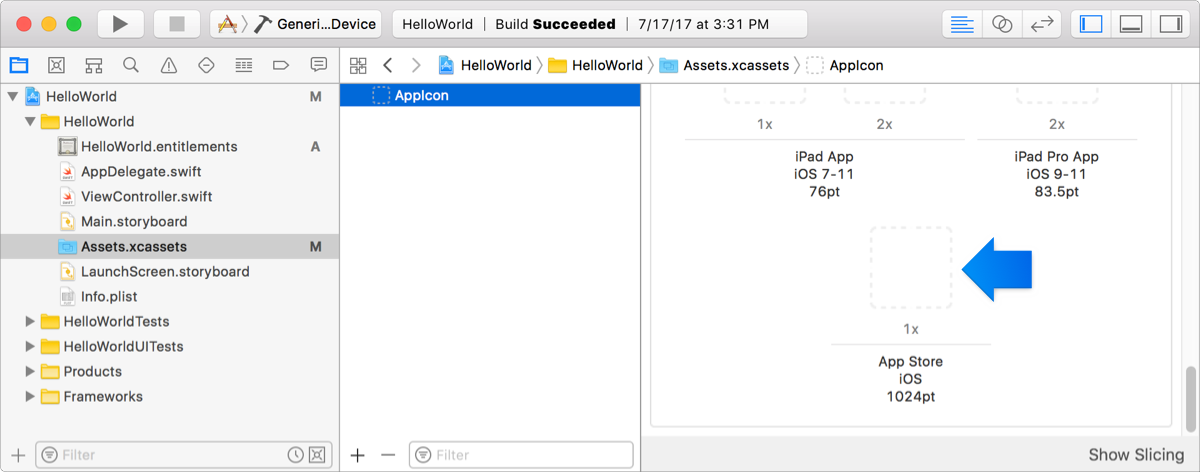
|  |  |
| --- | --- |
|  | 1. If the simulator does not launch, it means there is an error in your project. Look for a red number in the bottom right corner of the Xcode window; this is the number of errors encountered. 2. Click the number for details about the error, and then review these steps to figure out where things went wrong. If you run into a problem you can’t resolve, visit the PhoneGap community resources at <http://phonegap.com/community>. 3. Search through the wiki and Google Group for answers to your problem before posting a question. If you do post a question, include as much information as possible about the error. |

1. Your app should now be running in the iPhone Simulator as a native app. This may seem like no big deal, because the app will look and feel just like the full-screen web app that we had running in Chapter 6.
2. However, there is a profound difference: namely, that we can now start accessing device features that were previously unavailable. Before we get to that, though, we need to do a bit of cleanup.

**Add an App Store icon to your project**

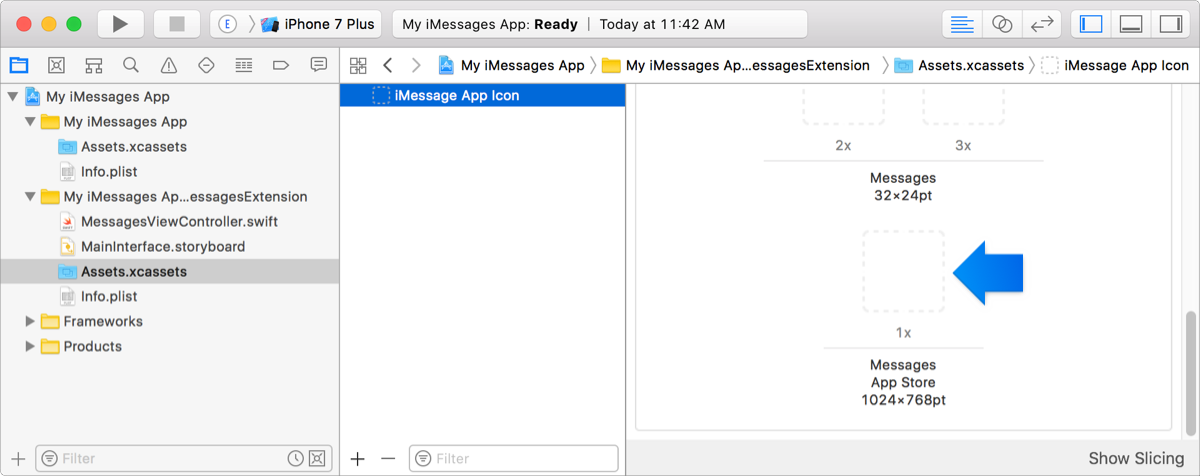
For Xcode 9, place the required [App Store icon](https://help.apple.com/xcode/mac/current/#/dev470df9e03) in an asset catalog located in the app bundle.

* For iOS apps, drag an icon to the App Store iOS well located in the AppIcon image set.



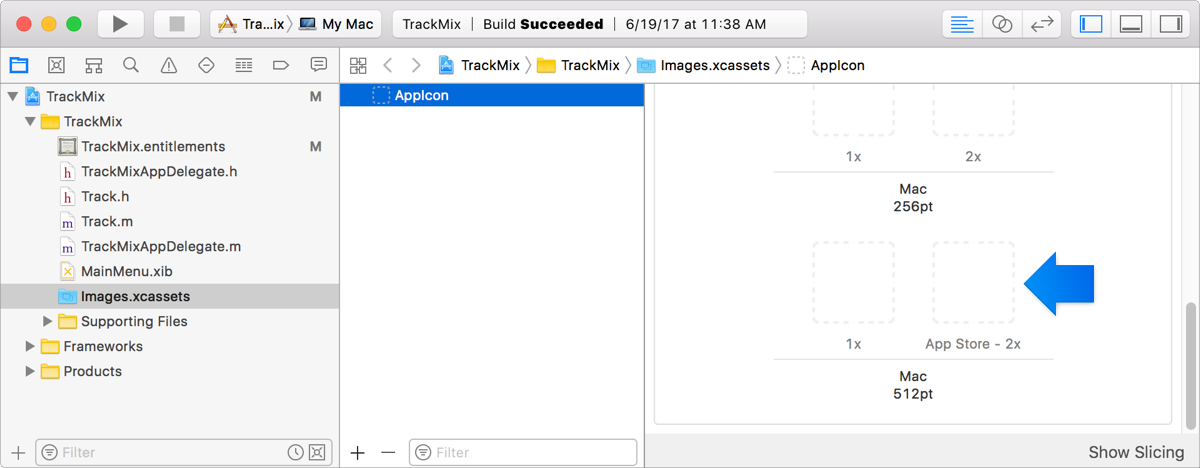
* For iMessage and Sticker Pack apps, provide both an App Store icon and a Messages App Store icon.

For the iMessage apps, drag an icon to the App Store iOS well located in the AppIcon image set of the iOS target. Then drag an icon to the Messages App Store well in the iMessage App Icon image set of the MessagesExtension target.



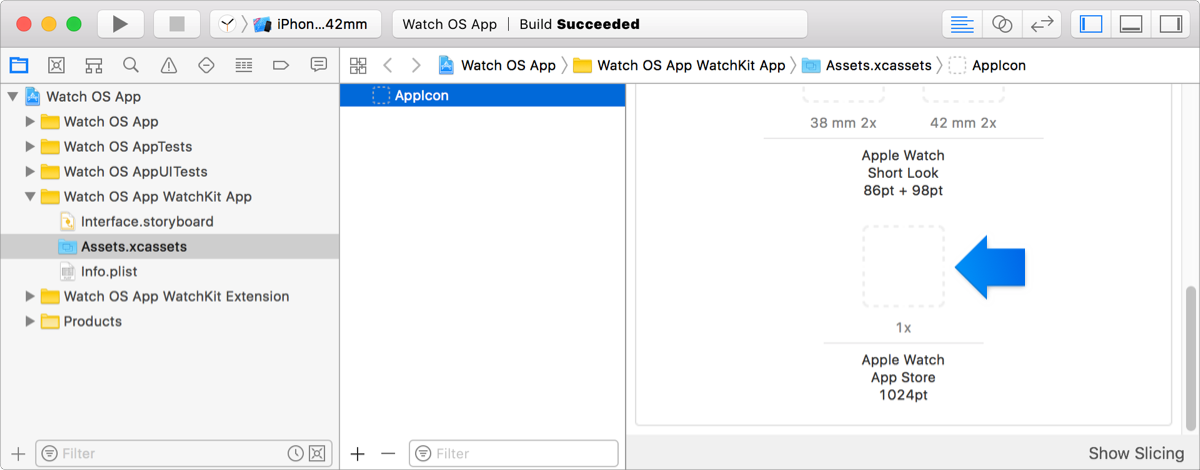
For Sticker Pack apps, the App Store iOS and the Messages App Store wells are both located in the iMessage App Icon image set in the Stickers.xcstickers file.

* For macOS apps, drag an icon to the App Store - 2x well in the AppIcon image set.



* For tvOS apps, drag the front, middle, and back icons to the App Icon - App Store image set located in the App Icon & Top Shelf Image folder.
* For watchOS apps, provide an App Store icon for the iOS and the watchOS targets.

For the iOS target, drag an icon to the App Store iOS well located in the AppIcon image set. For the WatchKit App target, drag an icon to the Apple Watch App Store well located in the AppIcon image set.



**Configure Targets and Builds**

1. Use the *project editor* to view and edit your project and [target](https://help.apple.com/xcode/mac/current/#/dev7d429ef73) settings by clicking the Project navigator button (https://help.apple.com/xcode/mac/current/en.lproj/Art/IDEStructureNavigatorTemplate.png) in the navigator bar 🡺 then select the project in the content area below. The project editor appears to the right of the navigator
2. In the project editor, select a project or target to edit its settings. To select from a list of projects and targets, click the “Show project and targets list” icon (https://help.apple.com/xcode/mac/current/en.lproj/Art/bs_buildsettings_showprojectsandtargets_button.png) at the top of the project editor. Otherwise, choose a target from the pop-up menu at the top of the project editor.
3. To show the different project or target settings, click the tabs at the top of the project editor.
4. The different settings are:

|  |  |
| --- | --- |
| General | Adjust the most commonly modified target settings, such as identity, signing, and deployment options. |
| Signing & Capabilities | Code sign your app and enable app services, such as [Push Notifications](https://help.apple.com/xcode/mac/current/#/devd573b503b), [CloudKit](https://help.apple.com/xcode/mac/current/#/devecab2d791), [Game Center](https://help.apple.com/xcode/mac/current/#/devac341811d), and [In-App Purchase](https://help.apple.com/xcode/mac/current/#/dev6e098884c). |
| Resource Tags | Identify and manage sets of [on-demand resources](https://help.apple.com/xcode/mac/current/#/devaf48822a6). |
| Info | Edit the properties of a project or target, including Info.plist values, supported document types, and localizations. |
| Build Settings | Customize options that affect the behavior of the build system while building your project. See [Build settings reference](https://help.apple.com/xcode/mac/current/#/itcaec37c2a6). |
| Build Phases | Edit and reorder tasks performed by the build system while building your project, such as [running a script](https://help.apple.com/xcode/mac/current/#/devc8c930575), copying files, or [linking to frameworks](https://help.apple.com/xcode/mac/current/#/dev51a648b07). |
| Build Rules | Customize rules the build system uses when processing certain types of files during the build process. |

1. Edit General settings, signing settings, Info settings, build settings.
2. Configure build phases.
3. Strip Swift symbols
4. Manage schemes
5. Configure capabilities
6. Work with Swift packages
7. Create a Mac version of an iPad app
8. Create and distribute a watch-only app
9. Localize your app
10. Manage devices
11. Maintain signing assets
12. Use source control
13. Distribute your app
14. Distribute binary frameworks

**Steps 1:**

Create an XCFramework.You can create an [XCFramework](https://help.apple.com/xcode/mac/current/#/dev8d5c361d4) that contains multiple platform-specific variants to be used by clients on different platforms, including Simulator builds. XCFramework can also contain a macOS variant of your framework built for AppKit and another macOS variant of your framework built for UIKit.

In the Build Setting pane of the [project editor](https://help.apple.com/xcode/mac/current/#/devdab46c612), set the Build Libraries for Distribution build setting to Yes and the [Skip Install (SKIP\_INSTALL)](https://help.apple.com/xcode/mac/current/#/itcaec37c2a6?sub=devfeb7a0695) build setting to No.

Ensure that you have a scheme which builds only the framework target and its dependencies.

**Step 2:**

Create an archive of the framework or library for each platform you wish to support by entering one xcodebuild command for each platform’s generic run destination:

xcodebuild archive [-project <project name>] -scheme <scheme name> -destination "generic/platform=<platform name>[,arch=<architecture name>][,variant=<variant name>]" [-configuration <configuration name>] [-archivePath <archive output path>]

To build a macOS variant of your framework built for UIKit, pass Mac Catalyst as the variant argument.

**Step 3:**

Export the framework or library from each archive and include any headers.

**Step 4:**

Create an XCFramework which includes each variant of the framework or library by entering an xcodebuild command with the -create-xcframework option in Terminal:

xcodebuild -create-xcframework -framework <path> [-framework <path>...] -output <path>xcodebuild -create-xcframework -library <path> [-headers <path>] [-library <path> [-headers <path>]...] -output <path>

To see all of the command options, enter xcodebuild -help or xcodebuild -create-xcframework -help in Terminal.

1. Submit (Distribute) apps to the App Store

NOTE: Before you distribute your app through the App Store, test it on simulated and real devices in Xcode, then

distribute the final build to real-world users. The final build you upload to [App Store Connect](https://help.apple.com/xcode/mac/current/#/devec4355ec5) should be product quality and pass all validation tests. If you [distribute your app using TestFlight](https://help.apple.com/xcode/mac/current/#/dev2539d985f), perform some of the steps below before you distribute the final build. Then the final build is already uploaded to [App Store Connect](https://help.apple.com/xcode/mac/current/#/devec4355ec5) and if you distribute to external testers, approved by [Beta App Review](https://help.apple.com/xcode/mac/current/#/devfc7555948).

## Step 1: Prepare your app for submission

Go to [App Review](https://developer.apple.com/app-store/review/) to review the App Store and human interface guidelines. For watchOS apps, also read [Preparing Your watchOS App for Submission](https://developer.apple.com/app-store/watch). For example, you must provide an [App Store icon](https://help.apple.com/xcode/mac/current/#/dev470df9e03). If you haven’t already done so, [add an App Store icon](https://help.apple.com/xcode/mac/current/#/dev4b0ebb1bb) to your Xcode project.

## Step 2: Enter additional information in App Store Connect

You may need to enter additional information in App Store Connect before you can submit your app to App Review. After your app is uploaded or released, you can’t change some of this metadata, so it’s important to choose your settings carefully. For the metadata that is required (or can’t be changed later), go to [Required, localizable, and editable properties](https://help.apple.com/app-store-connect/#/devfc3066644) in [App Store Connect Help](https://help.apple.com/app-store-connect).

For a Mac app built with Mac Catalyst, create separate app records for the Mac and iPad versions, then enter all the required information.

## Step 3: Archive, validate, and upload your app

If you didn’t [distributed the final build using TestFlight](https://help.apple.com/xcode/mac/current/#/dev2539d985f), [prepare your app for distribution](https://help.apple.com/xcode/mac/current/#/dev91fe7130a) and [create an archive of your app](https://help.apple.com/xcode/mac/current/#/devf37a1db04) now. [Validate the archive](https://help.apple.com/xcode/mac/current/#/dev37441e273) and fix any validation errors before continuing. Then [upload it to App Store Connect](https://help.apple.com/xcode/mac/current/#/dev442d7f2ca) and wait for it to pass App Store Connect validation tests.

For a Mac app built with Mac Catalyst, create separate archives for the iPad and Mac app. When creating the archive for the Mac app, choose My Mac as the run destination.

## Step 4: Submit your app to App Review

To submit the build to App Review, go to [Publish your app](https://help.apple.com/app-store-connect/#/dev34e9bbb5a) in [App Store Connect Help](https://help.apple.com/app-store-connect).

## Validate the App

Before you [upload an app to App Store Connect](https://help.apple.com/xcode/mac/current/#/dev442d7f2ca), validate the [archive](https://help.apple.com/xcode/mac/current/#/devdaa3662ee) to determine whether it meets minimum App Store requirements and ensure that it passes standard App Store Connect checks.

In the [Archives organizer](https://help.apple.com/xcode/mac/current/#/devea503dbda), select the [archive](https://help.apple.com/xcode/mac/current/#/devdaa3662ee) you want to validate, then click Validate App.

*Note:* If the Validate App button is disabled, verify that the archive contains a single top-level app.

1. In the sheet that appears, choose App Store distribution options, then click Next.

To learn about the distribution options, go to [Distribution options](https://help.apple.com/xcode/mac/current/#/devde46df08a).

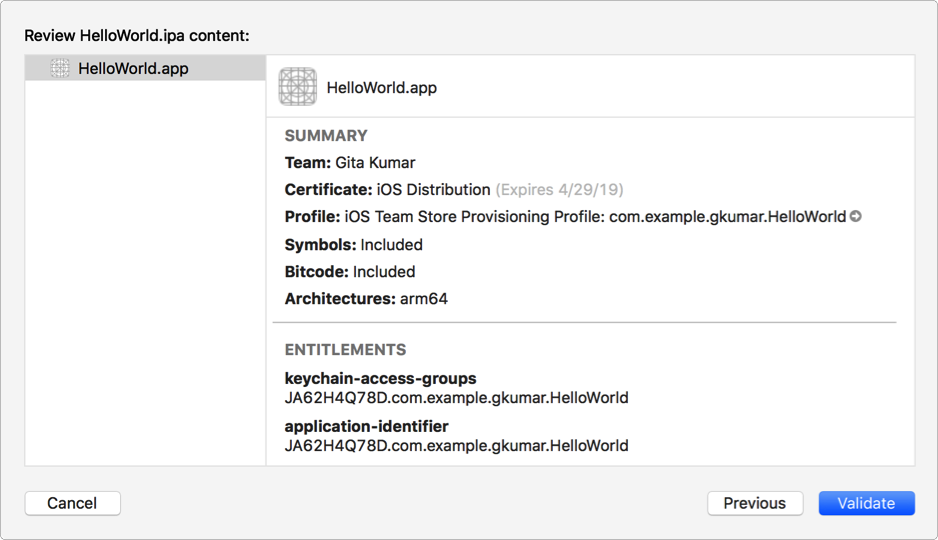
1. In the sheet that appears, choose a signing option, then click Next.

To learn about the signing options, go to [Distribution signing options](https://help.apple.com/xcode/mac/current/#/devff5ececf8). If you select “Manually manage signing”, go to [Manually manage distribution signing](https://help.apple.com/xcode/mac/current/#/devcac6ab5b3) for alternate steps.

1. If you are missing a required distribution certificate, follow the instructions in the next sheet to create it.

To delete or create a certificate, click Manage Certificates and go to [Manage distribution certificates](https://help.apple.com/xcode/mac/current/#/devcac6ab5b3?sub=dev52693cead).

1. Review the [signing certificate](https://help.apple.com/xcode/mac/current/#/dev1c7c2c67d), [provisioning profile](https://help.apple.com/xcode/mac/current/#/dev46a99ba04), and [entitlements](https://help.apple.com/xcode/mac/current/#/dev5f8bfd349), then click Validate.



1. Review and correct any validation issues found, then click Done.
2. If no issues are found, a green checkmark appears.
3. If Xcode doesn’t find an App Store Connect record for your app, the validation fails with an App Store Connection Operation Error. In App Store Connect, [Add an app to your account](https://help.apple.com/app-store-connect/#/dev2cd126805) and ensure that the [bundle ID](https://help.apple.com/xcode/mac/current/#/dev9b66ae7df) matches the bundle ID in your Xcode project.

## Verify the target and project build settings

If you changed the default build settings, verify some of the settings before you [distribute an app through the App Store](https://help.apple.com/xcode/mac/current/#/dev067853c94). You do this in the Build Settings pane of the [project editor](https://help.apple.com/xcode/mac/current/#/devdab46c612).

The Architectures [build setting](https://help.apple.com/xcode/mac/current/#/dev382dac089) identifies the architectures for which your app is built. An iOS device uses a set of architectures, which includes arm64 and for deployment targets earlier than iOS 11, armv7. You have two options for specifying the value of this setting:

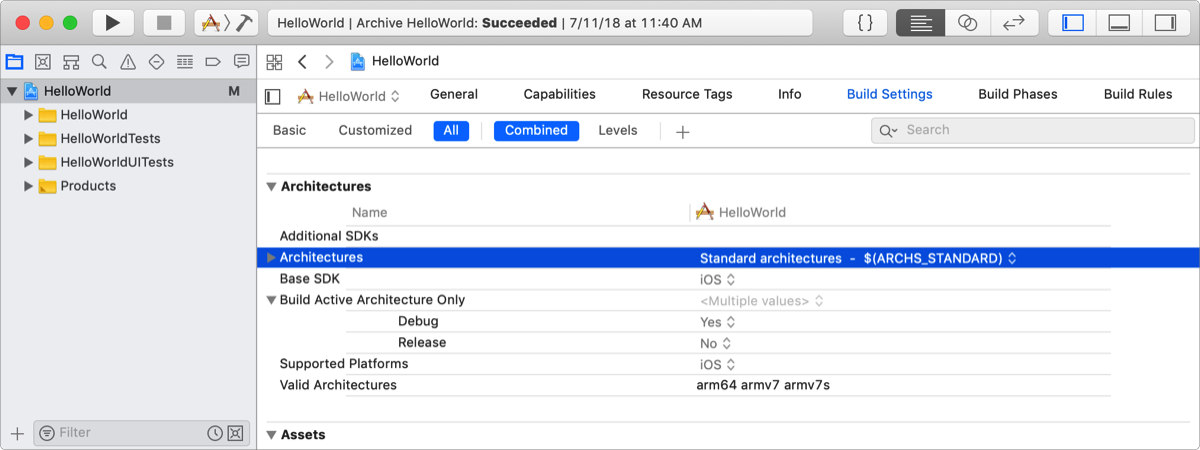
**Standard:** Produces an [app binary](https://help.apple.com/xcode/mac/current/#/dev7af8b1d18) for a set of architectures appropriate for the deployment target.

**Other:** Produces an app binary for a specified set of architectures.

1. In the [project editor](https://help.apple.com/xcode/mac/current/#/devdab46c612), choose the project, click Build Settings, and [find the build setting](https://help.apple.com/xcode/mac/current/#/dev04b3a04ba?sub=dev03f45e585) called Architectures.

All of the targets in your project need to support the same architectures, so select the project, not the target, in this step. Also, verify that no individual targets override this setting.

1. Select Standard or Other from the build-setting value list.



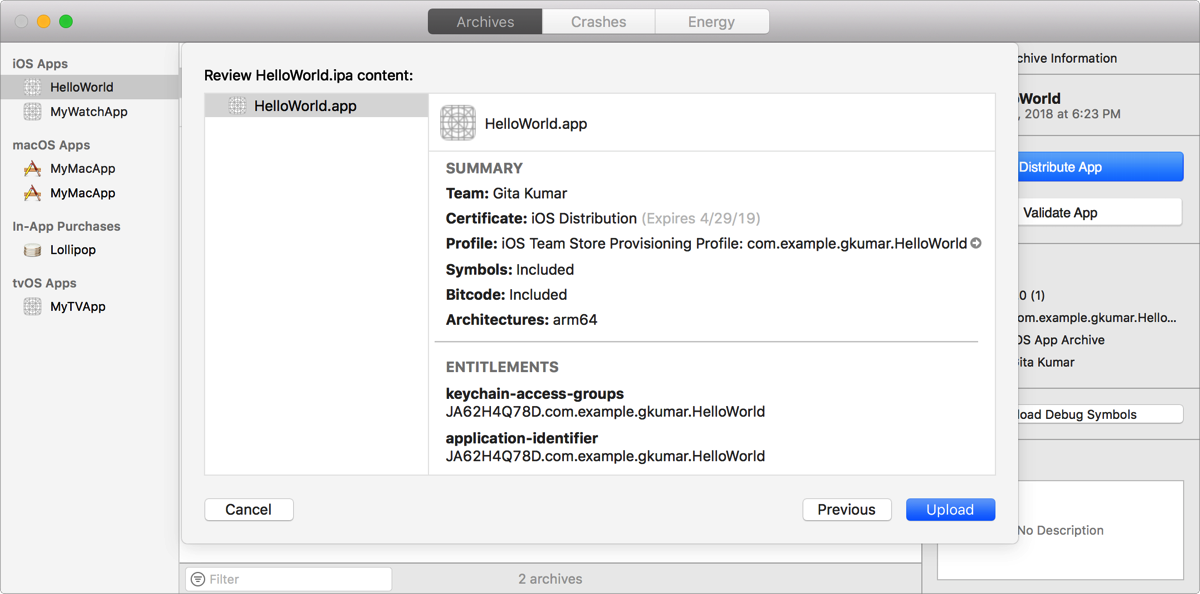
1. If you select Other, click the Add button (+) to enter the custom device architecture names you support.

**Important:** The App Store rejects a build that supports only armv7s. If armv7s is included in the Valid Architectures list, armv7 must also be included.

## Upload the App

Upload your app to [App Store Connect](https://help.apple.com/xcode/mac/current/#/devec4355ec5) to distribute it to users through the App Store, or optionally, to distribute your beta app to testers using [TestFlight](https://help.apple.com/xcode/mac/current/#/dev14db04cfa).

* 1. In the [Archives organizer](https://help.apple.com/xcode/mac/current/#/devea503dbda), select the [archive](https://help.apple.com/xcode/mac/current/#/devdaa3662ee) you want to upload, then click Distribute App.
  2. In the sheet that appears, select App Store Connect as the distribution method, then click Next.
  3. In the next sheet, select Upload, then click Next.
  4. Alternatively, to export the app before uploading it, click Export, then click Next. Select a location for the files, then click Export. A folder containing the [archive export files](https://help.apple.com/xcode/mac/current/#/deva1f2ab5a2) appears in Finder.
  5. In the next sheet, choose distribution options, then click Next. To learn about the distribution options, go to [Distribution options](https://help.apple.com/xcode/mac/current/#/devde46df08a).
  6. In the next sheet, choose a signing option, then click Next. To learn about the signing options, go to [Distribution signing options](https://help.apple.com/xcode/mac/current/#/devff5ececf8). If you select “Manually manage signing”, go to [Manually manage distribution signing](https://help.apple.com/xcode/mac/current/#/devcac6ab5b3) for alternate steps.
  7. If you are missing a required distribution certificate, follow the instructions in the next sheet to create it.
  8. To delete or create a certificate, click Manage Certificates and go to [Manage distribution certificates](https://help.apple.com/xcode/mac/current/#/devcac6ab5b3?sub=dev52693cead).
  9. Review the [signing certificate](https://help.apple.com/xcode/mac/current/#/dev1c7c2c67d), [provisioning profile](https://help.apple.com/xcode/mac/current/#/dev46a99ba04), and [entitlements](https://help.apple.com/xcode/mac/current/#/dev5f8bfd349).



* 1. Click Upload.

## Distribute to registered devices (iOS, tvOS, watchOS)

Before uploading your app to [App Store Connect](https://help.apple.com/xcode/mac/current/#/devec4355ec5), optionally distribute it for testing (beta) on registered devices using an [ad hoc provisioning profile](https://help.apple.com/xcode/mac/current/#/dev4335bfd3d) or [development provisioning profile](https://help.apple.com/xcode/mac/current/#/deva2614151b).

These export methods allow you to test variants of your app that are built locally by Xcode. Users don’t need to be on your team or App Store Connect users to run the app, but their devices need to be registered in your [developer account](https://help.apple.com/xcode/mac/current/#/devbbdb0a993). You can register a limited number of devices per product family per year that your team uses for development and testing. Choose one of these methods if you can allocate a portion of these devices for testing and can collect device IDs from your users.

Otherwise, members of the [Apple Developer Program](https://help.apple.com/xcode/mac/current/#/dev13f7a8004) may [distribute an app using TestFlight](https://help.apple.com/xcode/mac/current/#/dev2539d985f), which doesn’t impact the development device quota.

## Step 1: Register all test devices

Register one or more test devices before creating an ad hoc or development provisioning profile. Collect [device IDs](https://help.apple.com/xcode/mac/current/#/devc478fed96) from external users and then sign in to your developer account to [register multiple devices](https://help.apple.com/developer-account/#/devebd34abb1).

Beta testers can [locate a device ID using Finder (iOS, tvOS)](https://help.apple.com/xcode/mac/current/#/dev93ef696c6?sub=devdfa32588f) if they don’t have Xcode installed.

## Step 2: Archive your app

[Prepare your app for distribution](https://help.apple.com/xcode/mac/current/#/dev91fe7130a) and then [create an archive of your app](https://help.apple.com/xcode/mac/current/#/devf37a1db04).

## Step 3: Export the app using an ad hoc or development provisioning profile

When you [export the app](https://help.apple.com/xcode/mac/current/#/dev23ea8b877), Xcode re-signs the app using signing assets that depend on the options you select. To use an ad hoc provisioning profile, select Ad Hoc and to use a development provisioning profile, select Development as the [distribution method](https://help.apple.com/xcode/mac/current/#/dev31de635e5). If you select “Automatically manage signing”, Xcode creates and manages the signing certificates and provisioning profiles for you.

## Step 4: Install the app on user devices

Give the exported [iOS App (IPA) file](https://help.apple.com/xcode/mac/current/#/dev0460aa98a) to users along with instructions on how to install the app. They can [install an app on a device using Xcode](https://help.apple.com/xcode/mac/current/#/devade83d1d7?sub=dev103e8473e) or [install an app on a device using Apple Configurator 2](https://help.apple.com/xcode/mac/current/#/devade83d1d7?sub=dev87a955931). Otherwise, [install an iOS or tvOS app using Apple Configurator 2](https://help.apple.com/xcode/mac/current/#/devade83d1d7?sub=dev87a955931).

## Step 5: Solicit crash reports from users

Ask the users to [send you the app’s crash logs from their devices](https://help.apple.com/xcode/mac/current/#/dev0f3181c2c). Then [import and view the crash logs](https://help.apple.com/xcode/mac/current/#/dev85c64ec79?sub=devc8ddd72c5) in the Devices and Simulators window.

## Step 6: Use Xcode Server to distribute the app to testers

Optionally, use [continuous integration with Xcode Server](https://help.apple.com/xcode/mac/current/#/dev466720061) to build, analyze, test, and archive your app. Xcode Server hosts a website that facilitates the distribution of product builds and archives to testers and other team members.

**See also**[Distribution overview](https://help.apple.com/xcode/mac/current/#/devac02c5ab8)