

Submitting Your App to the Store

After you upload a build to iTunes Connect and sufficiently test that build, submit the build to the store using iTunes Connect. It's recommended that you submit the last archive you distribute for testing. Use TestFlight to distribute your app for testing, described in [Distributing Your App Using TestFlight](#) (iOS, tvOS, watchOS). For Mac apps, export your app and distribute it for testing yourself, described in [Exporting Your App for Testing \(Mac\)](#). The last build you test should be product quality and pass iTunes Connect validation tests.

Prepare Your App for Submission

Before submitting your final candidate, review the human interface and store guidelines again, enter information in iTunes Connect, and verify your Xcode project settings.

Review Human Interface and Store Guidelines

To streamline the approval process, review the following guidelines and fix any problems and retest your app before continuing.

- Follow the user interface guidelines in:
 - iOS Human Interface Guidelines*
 - Apple TV Human Interface Guidelines*
 - Apple Watch Human Interface Guidelines*
 - OS X Human Interface Guidelines*
- Review the store guidelines in:
 - App Store Review Guidelines*
 - Mac App Store Review Guidelines*

For WatchKit apps, go to [Preparing Your App Submission for Apple Watch](#). For more information on the app review process, go to [App Review](#).

Enter Additional Information in iTunes Connect

iTunes Connect is a web tool you use to enter information about your app for sale in the store. iTunes Connect stores all the metadata about your app including the app versions and builds that you upload using Xcode. Before you submit your app, enter all the required information, described in [Viewing and Changing Your App's Metadata](#), for your type of app. For descriptions of the metadata, see [iTunes Connect App Properties](#).

Verify Your Xcode Project

Verify that your Xcode project is configured correctly:

- Review your Xcode project configuration. Read [Configuring Your Xcode Project for Distribution](#).
- The bundle ID in the Xcode project, described in [Setting the Bundle ID](#), should match the bundle ID you enter in iTunes Connect.

Note: If you previously uploaded a build to iTunes Connect, you can't change your bundle ID in iTunes Connect.

- Use the same App ID to code sign your archive as you used for development.

If you don't use any app services that require an explicit App ID, you can use the Xcode wildcard App ID. If you want to create a new App ID, read [Registering App IDs](#). However, if you change your App ID, retest your app using the new App ID before uploading a build to iTunes Connect.

- Review the version and build settings, described in [Setting the Version Number and Build String](#). iTunes Connect extracts its prerelease version number and build number from the archive.
- To ensure that your app enables the key app services you want to use, review your App ID settings. Read [Adding Capabilities](#).

For Mac apps, review these additional tasks:

- Select Mac App Store as the signing method during development, as described in [Choosing a Signing Identity \(Mac\)](#).
- All apps need to have App Sandbox enabled, as described in [Configuring App Sandbox \(Mac\)](#).
- All apps and their installer packages need to be signed to submit them to the Mac App Store. If you use a helper app, read *Daemons and Services Programming Guide* to learn how to configure the helper app.

Upload Your Final Candidate

If you have never uploaded your app to iTunes Connect, read [Uploading Your App to iTunes Connect](#) for how to create, validate, and upload an archive. If you used TestFlight to distribute your app to testers (described in [Distributing Your App Using TestFlight \(iOS, tvOS, watchOS\)](#)), you can submit the last build you distributed for testing directly to App Review.

Submit Your App

To submit the build of your app to App Review, read [Submitting the App in iTunes Connect Developer Guide](#).

Recap

In this chapter, you learned how to prepare your app and submit it to App Review. To learn how to view the status of your app after you submit it to App Review, read [Viewing the Status of Your App](#).