Your First App Store Submission

Developer

Contents

About Your First App Store Submission 4
At a Glance 5
Enroll in the Program 5
Provision Devices 5
Create an App Record in iTunes Connect 5
Submit the App 6
Solve Problems and Choose Your Next Steps 6
Prerequisites 6
See Also 6

Enrolling in the iOS Developer Program 8
How to Administer Your Account Using Web Tools 8
How Apple Email Tracks Your Enrollment Process 8
Accessing the iOS Developer Program Tools 9

Creating an Xcode Project 12

Provisioning Your Devices for Development 16
Request a Development Certificate 16
Provision Your Device 18
Code Sign Your App 19
Launch Your App on the Device 20

Testing Your App on Many Devices and iOS Versions 22
Configure Your App for Distribution 22
Test Your App Locally 24
Register Device IDs 26
Create Your Distribution Certificate 28
Create an Ad Hoc Provisioning Profile 29
Create an iOS App Store Package for Testing 31
Install the Ad Hoc Provisioning Profile and App on the Device 33
Send Crash Reports to Developers 35

Creating Your App Record in iTunes Connect 37

Before You Begin 37
Capture Screenshots 38
Set the Launch Image 39
Set the Bundle ID to Match the App ID 40
Create the iTunes Connect App Record 41

Submitting Your App 45
Before You Begin 45
Create a Distribution Provisioning Profile 45
Create and Validate the Archive 47
Submit and Ship Your App 49
Submit the Archive 50
Ship Your App 51
Respond to User Issues 52

Review and Troubleshooting 54
Review: How Provisioning Works 54
Verify Your Certificates Using Keychain Access 56
Export and Import Your Digital Identities 58
Verify the iOS App Target Settings 59

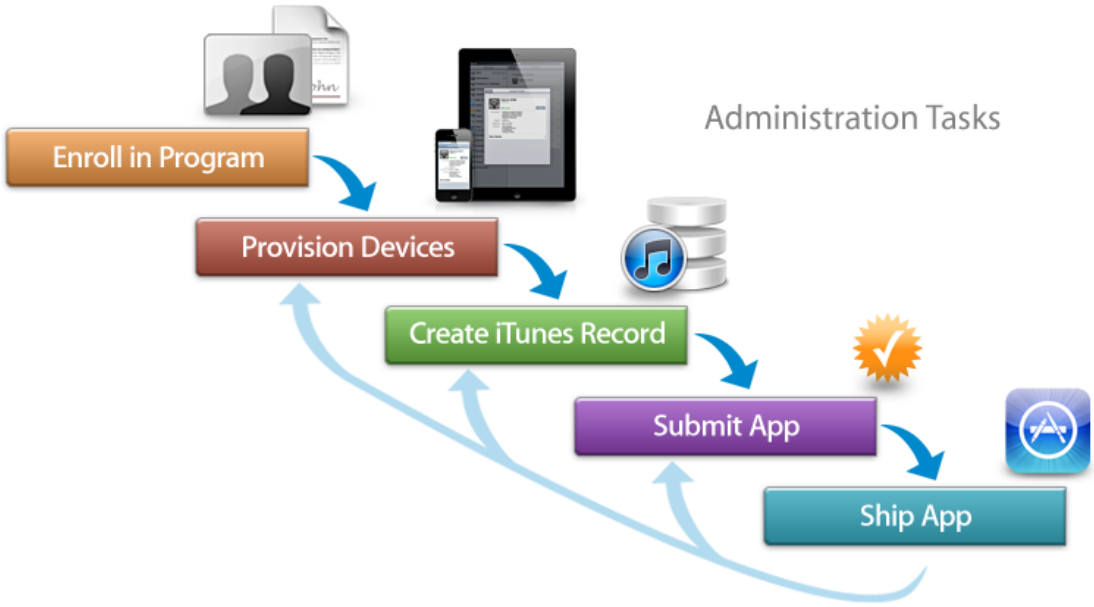
Next Steps 60
Using iCloud Storage 60
Using Other Technologies 60
Managing Your Team 60
Getting Your App Approved 61
Marketing Your App and Managing Your Account 61

Document Revision History 62

About Your First App Store Submission

This tutorial teaches you the process of provisioning devices and submitting your iOS app to the App Store. (You can start some of these tasks as soon as you are ready to run your app on a device.) You learn this process best using your own Xcode project. But if you created a HelloWorld project in one of the other tutorials, it will work well too.

Most of your time is spent on coding tasks. But to develop for the App Store, you must also perform several administrative tasks, using Xcode and other tools, throughout the lifetime of your app. The App Store is a curated store, which means that only Apple-approved apps are available for purchase. Apple does this to provide the best experience possible for our users. For example, apps that are sold on the App Store must not crash or exhibit other major bugs.



Apple provides the tools you need to develop, test, and submit your iOS app to the App Store. To run an app on a device, the device needs to be provisioned for development, and later provisioned for testing. You also need to provide information about your app that the App Store displays to customers and upload screenshots. Then you submit the app to Apple for approval. After the app is approved, you set a date the app should appear in the App Store as well as its price. Finally, you use Apple's tools to monitor the sales of the app, customer reviews, and crash reports. Then you repeat the entire process again to submit updates to your app.

If you use certain technologies—such as iCloud storage or In-App Purchase—there are additional configuration and administration tasks you need to perform. There are also tasks you perform to manage a team of developers. This tutorial doesn't cover these additional tasks.

At a Glance

This tutorial teaches you the process of developing an iOS app, testing it on a device, and submitting it to the App Store. You can use the same HelloWorld app you created in *Your First iOS App* in this tutorial or follow the steps using your own app or a new Xcode project you create in "Creating an Xcode Project" (page 12). This tutorial is about the *process*, not about how you design your user interface or write your code.

Enroll in the Program

Before you can access any of the tools you need to run your app on a device or begin the submission process, you must join the iOS Developer Program as either an individual or a company.

Relevant chapters "Enrolling in the iOS Developer Program" (page 8)

Provision Devices

The first step after creating a working app is to run it on a device. Fortunately, Xcode simplifies this process by creating default signing certificates and provisioning profiles for you.

Relevant chapters "Creating an Xcode Project" (page 12), "Provisioning Your Devices for Development" (page 16), "Testing Your App on Many Devices and iOS Versions" (page 22)

Create an App Record in iTunes Connect

Before you can submit an app for approval, you need to provide some important information to set up your iTunes Connect account. iTunes Connect is the marketing and business tool you use to check the status of your contracts, set up tax and banking information, obtain sales and finance reports, manage developers, and manage metadata about your app. At a minimum, you need to create an app record and complete forms to validate and submit your app.

Relevant chapters "Creating Your App Record in iTunes Connect" (page 37)

Submit the App

Submitting your app to the App Store is a multistep process involving several tools. First, create a distribution provisioning profile using iOS Provisioning Portal. Then create an archive, validate it, and submit it to the App Store using Xcode. When your app is approved, set the date when the app will be available to customers using iTunes Connect. Finally, don't forget to respond to user issues after you ship your first version.

Relevant chapters "Submitting Your App" (page 45)

Solve Problems and Choose Your Next Steps

As you complete the tasks in this tutorial, you may encounter problems that you don't know how to solve. In this book, you'll learn some troubleshooting steps that every developer should know about provisioning devices.

After you finish this tutorial, consider using some of the specialized technologies—for example, iCloud storage and push notifications—that require additional configuration and provisioning. And if your development team grows, you need to learn how to manage your team.

Relevant chapters "Review and Troubleshooting" (page 54), "Next Steps" (page 60)

Prerequisites

To benefit from this tutorial, you should read and work through the example in *Your First iOS App* to learn the basics of app development before reading this tutorial. This tutorial uses the HelloWorld app you created in *Your First iOS App*, but you can use your own app to follow along.

See Also

Provisioning your devices and submitting your app to the App Store is more complicated if you have a team of developers and use specialized technologies, such as iCloud storage and push notifications. There are also numerous administrative tasks not covered in this tutorial that you need to perform to market and maintain your app on the App Store. Your app also needs to follow the guidelines in order to be approved by Apple. Read these books next:

- To learn more about using iOS Provisioning Portal, see *iOS Team Administration Guide*.
- To learn more about using Xcode to provision your devices and submit your app, see *Tools Workflow Guide for iOS*.
- To learn more about using iTunes Connect to manage your app, see *iTunes Connect Developer Guide*.
- To learn more about the user interface guidelines and get your app approved, see *iOS Human Interface Guidelines* and *App Store Review Guidelines for iOS Apps*.

Enrolling in the iOS Developer Program

To develop for the App Store, you first need to join the iOS Developer Program. After you enroll in the program, you have access to all the resources and tools you need to manage your account and begin testing your app on a device.

When you enroll, you become the primary contact for Apple, sign legal agreements, create your assets, and market your app. You'll be asked whether you are an individual developer or a company. If you are a company, you may add other persons to your team and grant some of them privileges to manage your account. Individuals who need to run the app on a device during development must be added to your team before they can test on a device.

How to Administer Your Account Using Web Tools

You administer your account with these iOS Developer Program web tools:

- Member Center—The primary tool used to manage developer program accounts, invite team members, purchase technical support, and sign up for compatibility labs. The Member Center is also a gateway to the other resources and tools.
- iOS Provisioning Portal—A web tool used to register the app ID, register devices, create signing certificates, and create provisioning profiles. These steps are necessary for security and to ensure your app is not distributed prematurely.
- iTunes Connect—The marketing and business tool used to check the status of your contracts, set up tax and banking information, obtain sales and finance reports, and manage metadata about the app.

To enroll in the iOS Developer Program, go to Apple Developer Program Enrollment where a web assistant guides you through the entire process of enrolling. If you have not registered as an Apple Developer yet, you can do so as part of enrolling in the iOS Developer Program. When you are prompted to select a program, select the iOS Developer Program.

How Apple Email Tracks Your Enrollment Process

During the enrollment process, you receive a series of emails from Apple containing further instructions and links to various web tools. Read and follow the email instructions carefully to complete the enrollment:

- If this is the first time you have registered as an Apple Developer, you receive an email thanking you for registering that contains your Apple ID. You may receive a separate email asking you to verify your email address. You must verify your address before the iOS Developer Program registration can be completed.
- After submitting your iOS Developer Program enrollment request, you receive a confirmation email with an invitation to visit the Member Center.

  While waiting for your request to be processed further, you can visit the Member Center to explore the resources, including documentation, you use to develop your app.
- After your enrollment request is processed by Apple, you receive an email requesting that you sign the license agreement.

  Follow the instructions in the email to sign the license agreement.
- After signing the license agreement and completing the enrollment process online, you receive an email containing your activation code.

  Click the activation code in the email to complete the purchase of your iOS Developer Program.
- After the enrollment is successful, you receive an email welcoming you to the iOS Developer Program.

  Click the "Log in now" button in the email or go directly to the Member Center. The Member Center contains links to all the web tools you'll need to manage your account.
- Finally, you receive an email inviting you to use iTunes Connect to set up your app for purchase on the App Store.

After you have successfully enrolled in the iOS Developer Program, you can follow the rest of the steps in this tutorial. You can perform some of the iOS Provisioning Portal administration tasks using Xcode and return to the web tools as needed.

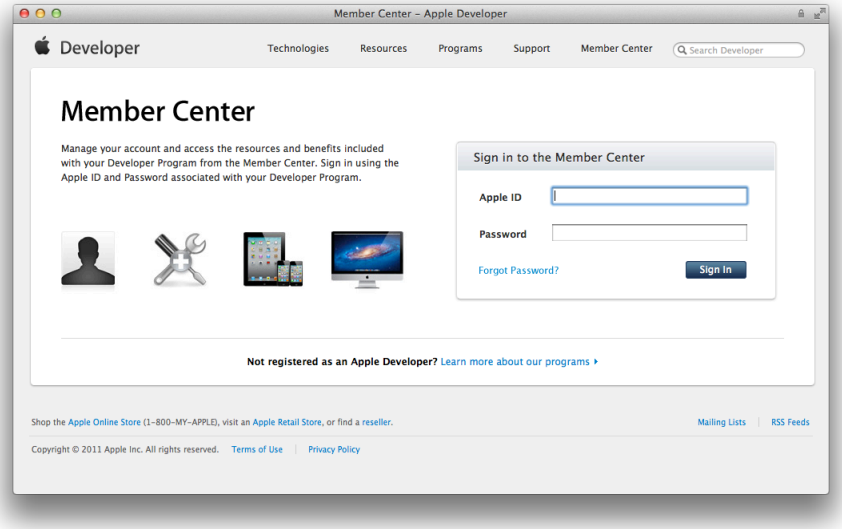
Accessing the iOS Developer Program Tools

To follow the steps in this tutorial, you need to know how to log in to the Member Center and find the web tools.

To log in to the Member Center . . .

1. Open a browser and go to http://developer.apple.com.
2. Select Member Center in the toolbar.

3. Enter your Apple ID and Password and click Sign In.



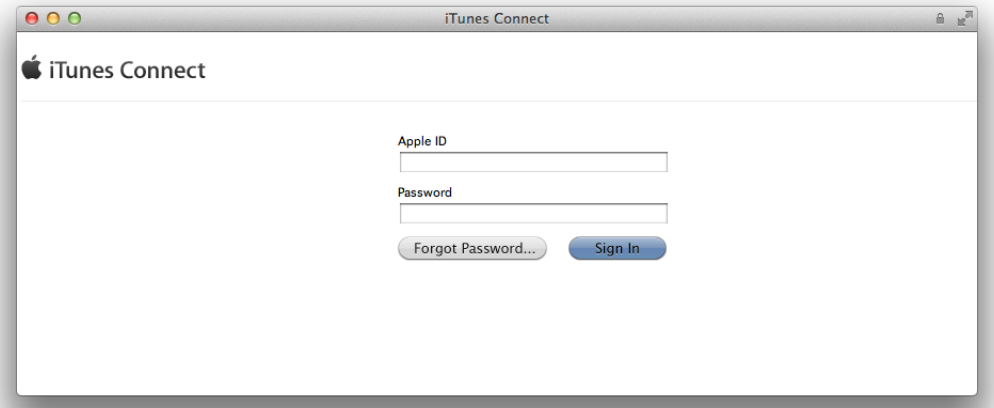
To go to iOS Provisioning Profile . . .

1. Log in to the Member Center.

2. Click the icon or text next to iOS Provisioning Portal under "Technical Resources and Tools."

To go to iTunes Connect . . .

1. Log in to the Member Center.
2. Click the icon or text next to iTunes Connect under App Store Distribution.
3. Enter your Apple ID and Password and click Sign In.



Creating an Xcode Project

Some of the App Store configuration is completed for you when you create an Xcode project from a template. Xcode prompts you to enter the product name and company identifier. The bundle ID is derived from these two properties but you can change this default value for the bundle ID after you create the project. The product name, company identifier, and bundle ID should match the values you set in your iTunes Connect app record that you create later. Xcode uses reasonable defaults for other values as well but you nevertheless need to configure additional settings later before you submit it to the App Store. When you create your Xcode project, you should carefully consider the template you choose; starting with the right template helps speed the development process.

To get started, open an existing Xcode project, such as the HelloWorld app you created in *Your First iOS App*, or create a new Xcode project. If you don't have an existing Xcode project, create a simple Xcode project from a template now, following the steps in this tutorial.

> **Note** If you already have an Xcode project, skip to "Provisioning Your Devices for Development" (page 16).

To create a new project . . .

1. Open Xcode.

2. Choose File > New > New Project or click “Create a new Xcode project” on the Welcome to Xcode window.

3. In the iOS section, select Application, select a template from the list of templates, and then click Next.

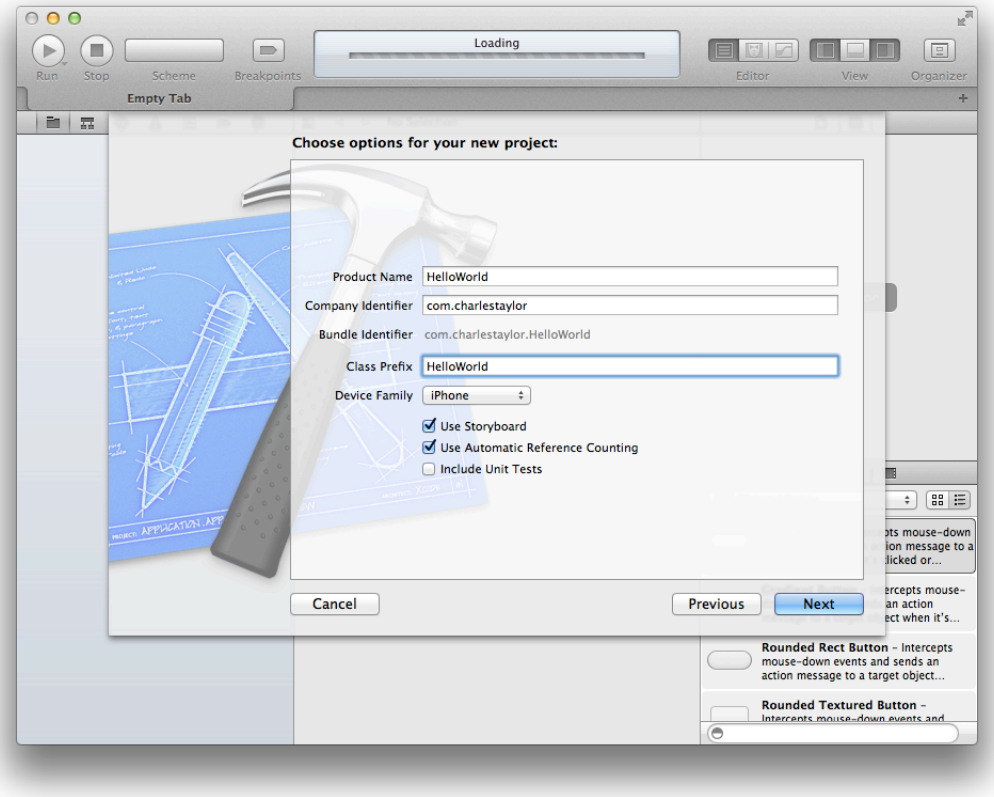
   A dialog appears, prompting you to name your app and choose additional options for your project.

4. Fill in the Product Name, Company Identifier, and Class Prefix fields.

   You can use the following values:

   - Product Name: `HelloWorld`
   - Company Identifier: Your company identifier, if you have one. If you don’t have a company identifier, you can use `edu.self`.

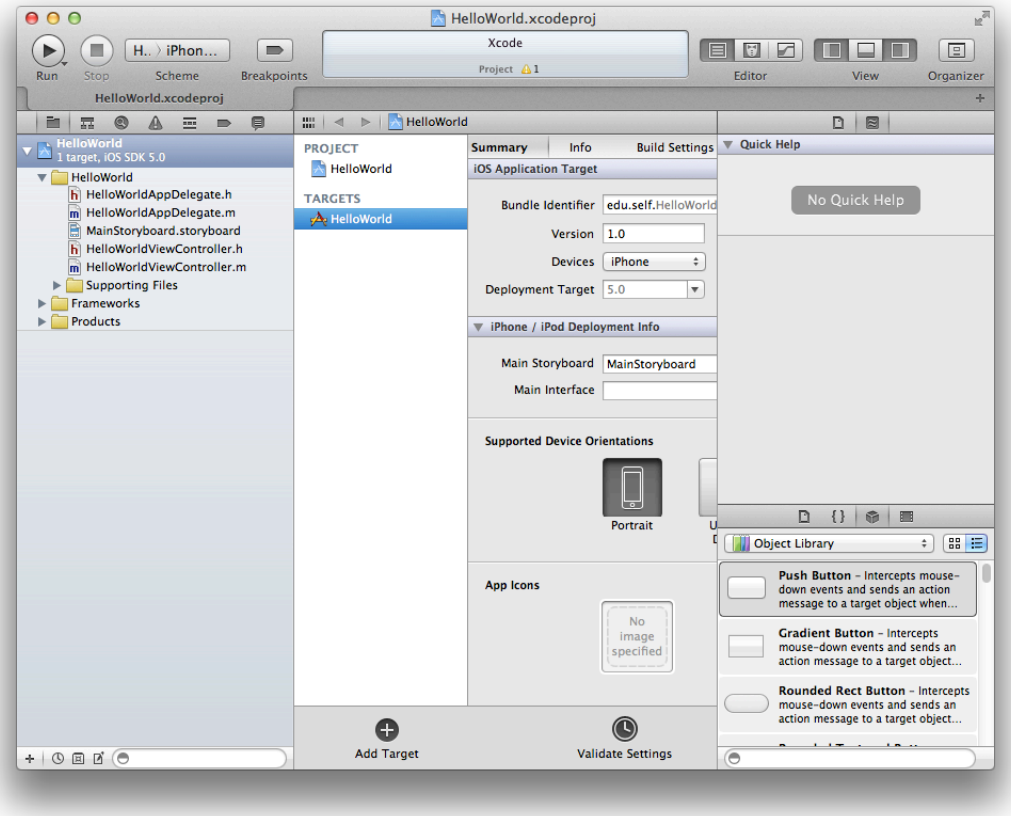
- Class Prefix: `HelloWorld`



5. In the Device Family pop-up menu, make sure that iPhone is selected.
6. Make sure that the Use Storyboard and Use Automatic Reference Counting options are selected and that the Include Unit Tests option is unselected.
7. Click Next.

   Another dialog appears, asking you to specify where you want to save your project.
8. Specify a location for your project (leave the Source Control option unselected) and then click Create.

You should see a new project window similar to this:



Provisioning Your Devices for Development

Although you can test an app you're developing in a simulator, there's nothing like running your app on an actual device. To run your app on an iOS device during development, it must be connected to your Mac, enabled for development, and recognized by Apple. You do this by providing some information about the app, yourself, and the device. You create a type of signing certificate, called a *development certificate* , to identify yourself. All of this information is incorporated into a development provisioning profile that is installed and allows the app to launch on the device.

The easiest way to provision your devices for development is with Xcode. You can log in to your account and view all the provisioning profiles and signing certificates for your account. Xcode provides a default iOS Team Provisioning Profile and iOS Wildcard App ID for you. Xcode automatically updates this provisioning profile when new development certificates or device IDs are added to your account. The iOS Team Provisioning Profile uses the iOS Wildcard App ID that matches all apps developed by you or your team. The iOS Team Provisioning Profile allows you to begin running the app on a device immediately. However, if you use iCloud storage, push notifications, In-App Purchase, or Game Center, you need to create a specialized provisioning profile.

To run your app on a device, follow this process:

1. Request a development certificate.
2. Add your device to the portal.
3. Code sign your app.
4. Launch your app on the device.

Request a Development Certificate

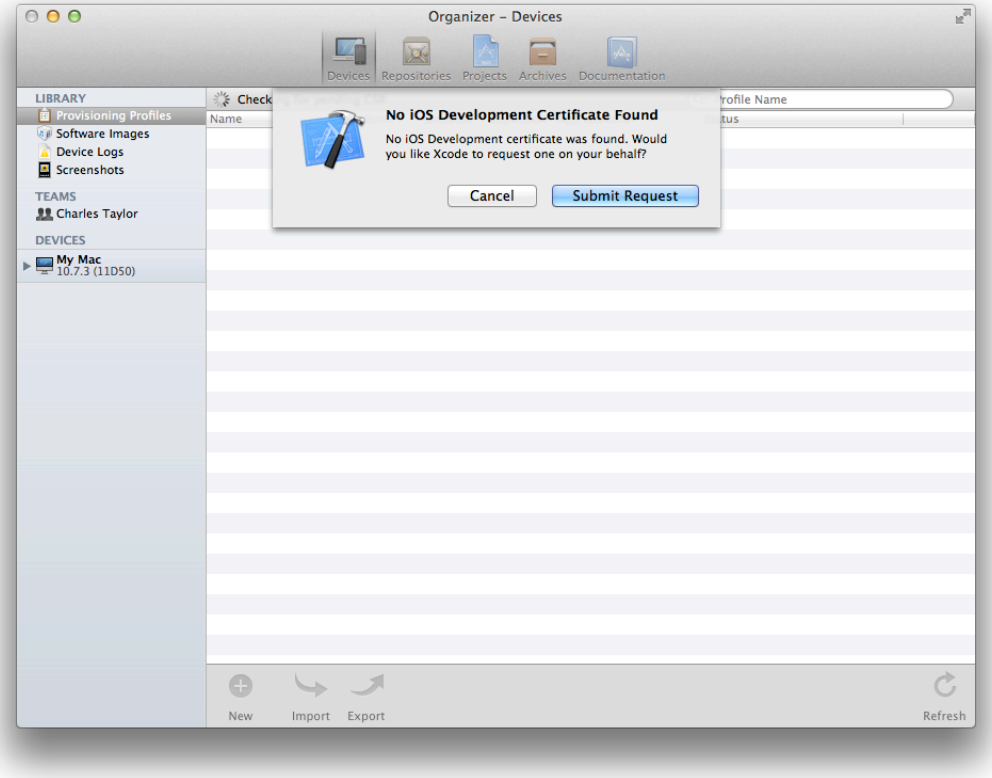
When you refresh the provisioning profiles, Xcode creates your signing certificates. Xcode creates both development and distribution certificates on your behalf and automatically adds them to your keychain. A distribution certificate is needed later for testing and submitting your app to the App Store.

To request your development certificate . . .

1. Choose Window > Organizer.
2. Click Devices.
3. In the Library section, select Provisioning Profiles.

4. Click the Refresh button at the bottom of the window.
5. Enter your user name and password and click Log in.

   After you log in to your account, a prompt appears, asking whether Xcode should request your development certificate.



6. Click the Submit Request button.

   The development certificate is added to your keychain and later added to the iOS Team Provisioning Profile.

   (More prompts may appear, asking whether Xcode should request other types of certificates. Click the Submit Request button for each prompt that appears.)

7. If a prompt appears, at the end of the refresh process, asking if you want to export your developer profile, click Export.

   The private keys for your certificates are stored in your keychain, and the public keys are stored in the portal. For this reason, you can't refresh your provisioning profiles and certificates in Xcode to replace a missing private key in your keychain. Instead, you should back up your certificates after you create them and import them when you are missing a private key or move to another Mac.

8. Enter a filename and password, and click Save.

   Because the file contains your digital identity which can be used to sign apps in your name, it is encrypted and password protected. (You will need the password later to import your digital assets to another system.)

Provision Your Device

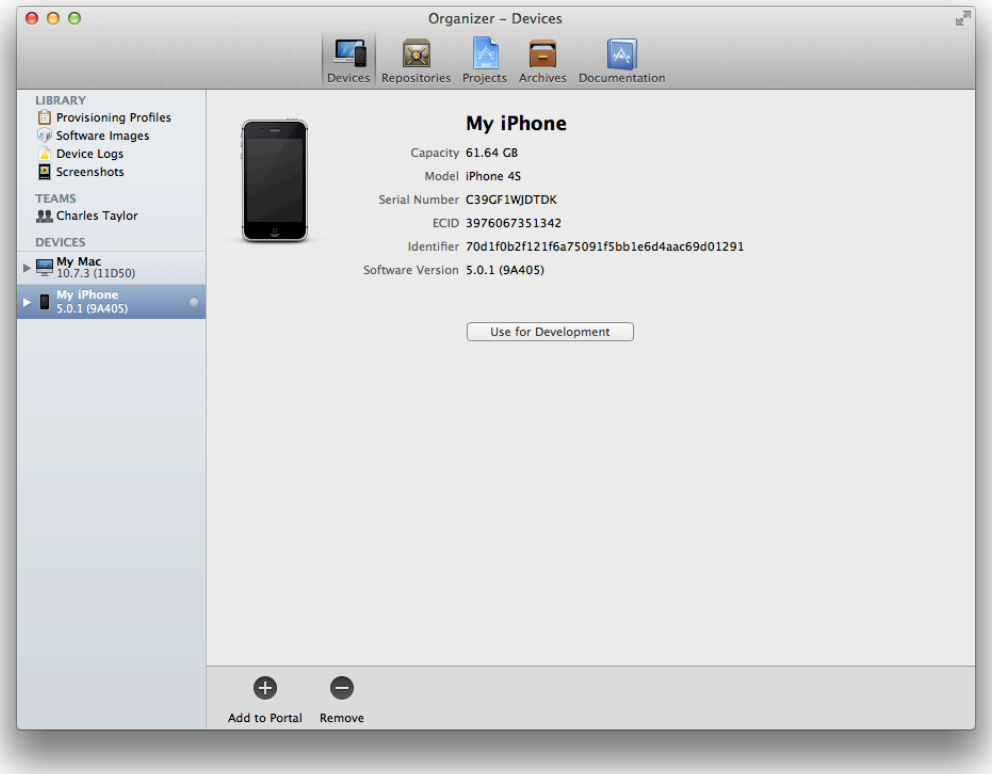
The first time you add a device, Xcode creates the iOS Team Provisioning Profile and automatically installs it on the device. You simply connect your iOS device to your Mac and click the "Use for Development" button to add the device to the iOS Team Provisioning Profile.

To provision your device . . .

1. Connect your device to your Mac.
2. Open the Devices organizer (Window > Organizer > Devices).
3. In the Devices section, select your iOS device.
4. Click the "Use for Development" button.

   The first time you add a device ID to your account, Xcode creates the iOS Team Provisioning Profile using the iOS Wildcard App ID, your development certificate, and the device ID. The iOS Team Provisioning Profile is also installed on your iOS device.

If the device was used for development in the past, the “Use for Development” button may not appear. If this happens, click “Add to Portal” at the bottom of the screen instead.



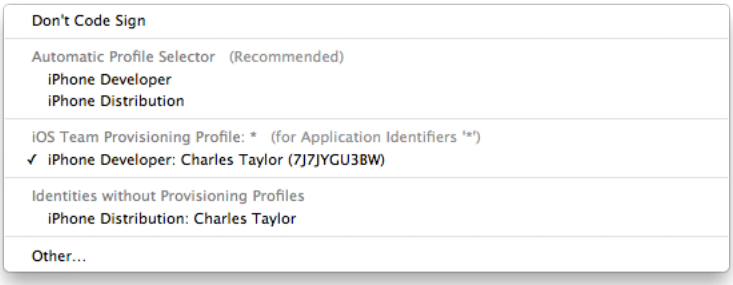
Code Sign Your App

When you build the app, you code sign it with the signing certificate contained in the provisioning profile you want to use. A menu item appears (in the Code Signing Identity Build setting pop-up menu) for each provisioning profile your development certificate belongs to. The default setting is iPhone Developer in the Automatic Profile Selector menu, which matches your iOS Team Provisioning Profile developer certificate. If you previously changed this build setting or created other provisioning profiles that use your developer certificate, set the Code Signing Identity to your developer certificate contained in the iOS Team Provisioning Profile.

To set the code signing identity to your iOS Team Provisioning Profile development certificate . . .

1. Select the project.
2. Click Build Settings.

3. Click All.
4. Type `Code Signing` in the search field in the Build Settings pane of the project editor.
5. From the Code Signing Identity pop-up menu, in the iOS Team Provisioning Profile section, choose the certificate that begins with "iPhone Developer:" followed by your name.



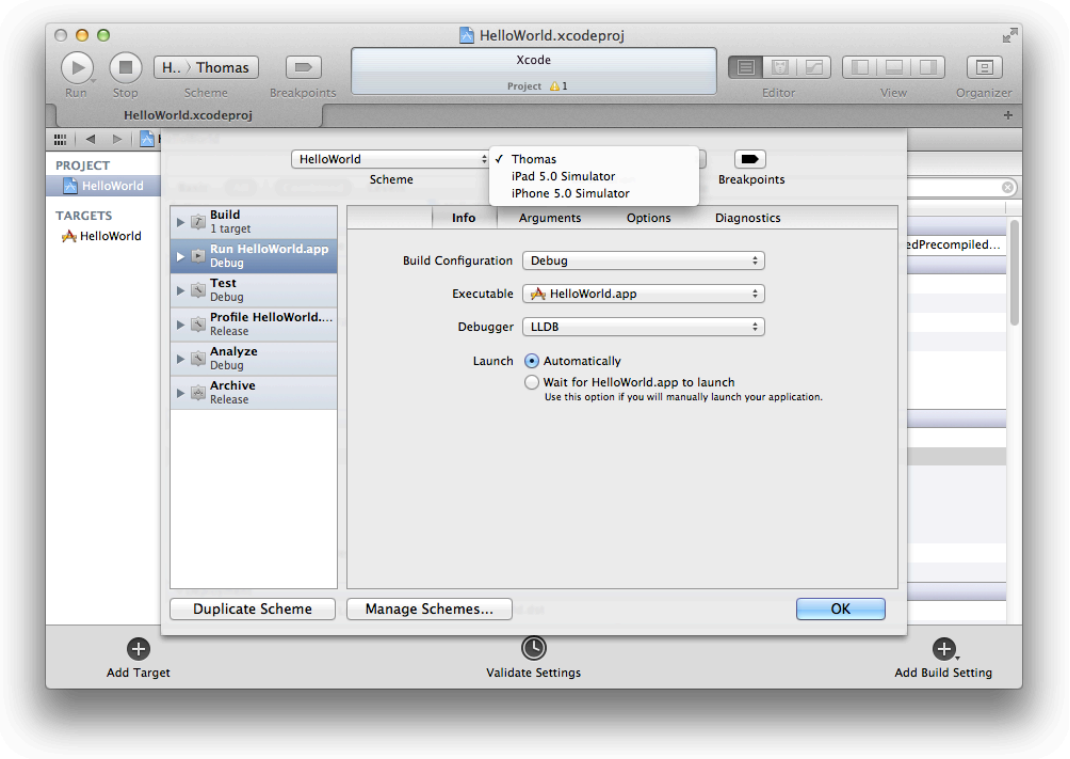
Launch Your App on the Device

After you provision your device for development, you can tell Xcode to launch the app on the device. You do this by changing the *run destination* setting in the Scheme pop-up menu before you build the app. When you connect an iOS device with a valid provisioning profile into your Mac, its name and the iOS release it's running appear as an option in the destination Scheme pop-up menu.

To launch the app on the device . . .

1. Choose Product > Edit Scheme to open the scheme editor.
2. Select your device from the Destination pop-up menu.

When you connect an iOS device with a valid provisioning profile into your Mac, its name appears as an option in the destination Scheme pop-up menu.



3. Click OK to close the scheme editor.
4. Click the Run button.

   When the app is built, it is signed. If a prompt appears asking whether codesign can sign the app using a key in your keychain, click Allow or Always Allow.

Testing Your App on Many Devices and iOS Versions

It's a good idea to have a plan for rigorously testing the app on a variety of devices and iOS versions. It's not sufficient to test the app using a simulator and on a device provisioned for development. A simulator doesn't run all threads that run on devices, and launching apps on devices using Xcode disables some of the watchdog timers. At a minimum, test the app on all the devices you have available. Ideally, test the app on all the devices and iOS versions you intend to support. For example, if you have a game that's intended to run only on iPhone and iPad touch, you should test on those devices and not on iPad.

You do this by creating a special distribution provisioning profile, called an **ad hoc provisioning profile**, and send it, along with the app, to testers. An ad hoc provisioning profile doesn't require testers to enroll in the developer program, be added to your team, create signing certificates, or use Xcode to run your app. Instead, app testers simply install the app and the ad hoc provisioning profile on their device to launch the app. You can then collect and analyze crash reports or logs from these testers to resolve problems before you ship your app.

Testing your app consists of these tasks:

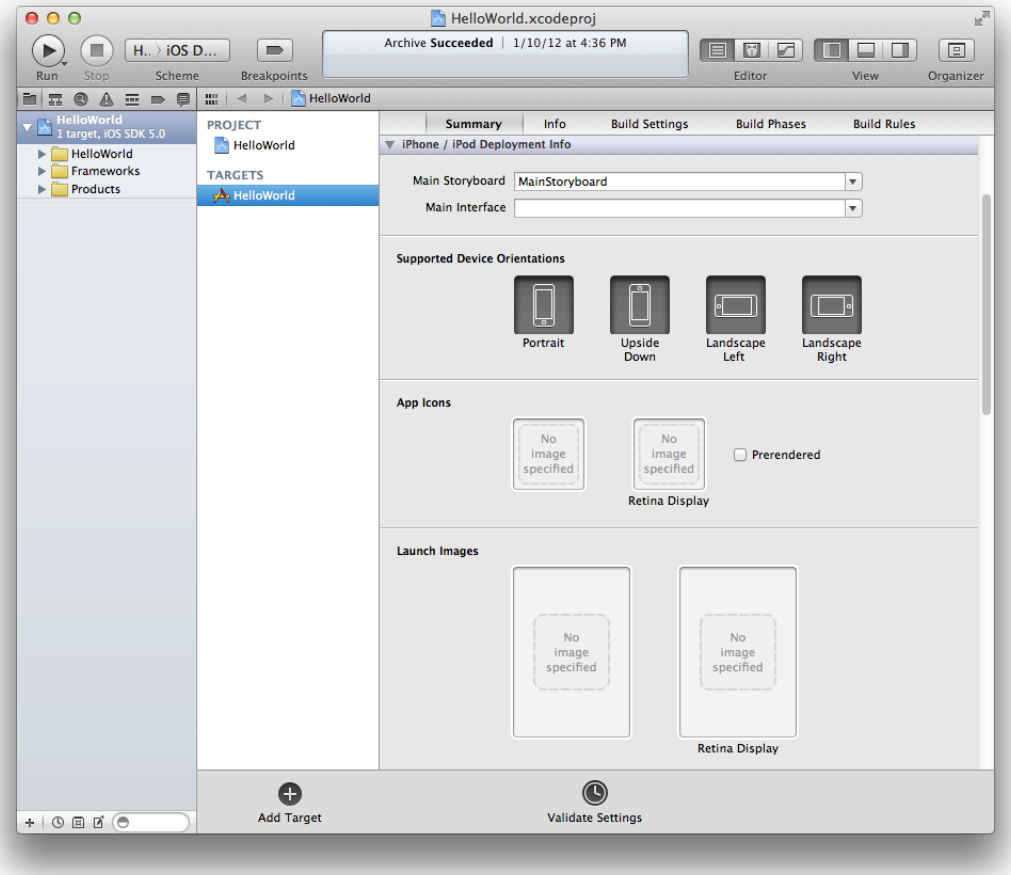
1. Configure your app for distribution.
2. Test your app locally.
3. Register all the testing unit device IDs.
4. Create a distribution certificate.
5. Create an ad hoc provisioning profile.
6. Create an iOS App Store Package.
7. Install the ad hoc provisioning profile and app on test devices.
8. Send crash reports to developers.

Configure Your App for Distribution

Most of the settings you need to configure your app for distribution and submission to the App Store are located in the target's Summary pane in Xcode. The default values for these settings are sufficient to start developing an app, but before distributing it, you should change some of those values. For example, test versions of your app should contain artwork that iTunes uses to identify your app on the device. Otherwise, when testers add your app to their iTunes library, iTunes uses generic artwork to represent it. Later, you'll need

iTunes artwork to pass validation tests before you submit your app to the App Store. So now is a good time to add the iTunes artwork to your Xcode project and set other distribution options. Later, you'll set more values for submission to the App Store.

These deployment settings are located in the project editor pane's Summary tab in iPhone/iPod Deployment Info.



First, set the device orientations your app supports by selecting the Portrait, Upside Down, Landscape Left, and Landscape Right icons as appropriate. In the HelloWorld app project, select all the orientations.

App icons are nice to have for testing but are required later to submit to the App Store (you need to set the app icons to pass validation tests). However, launch images—the images iOS displays to the user while your app is launching on the device—are optional. At a minimum, create two versions of your app icons—non-Retina 57 x 57 pixels and Retina 114 x 114 pixels—and drag them to the corresponding wells in the Summary pane. If your app supports the iPad, create a 72 x 72 pixels version of your app icon and drag it to the corresponding well in the Summary pane.

To add an app icon . . .

1. In the project navigator, select your project.
2. Select your target in the Targets section of the second sidebar.
3. Select the Summary tab.
4. Drag the icon file to the appropriate App Icons box.

Test Your App Locally

Before you distribute your app for testing, you should finish configuring and testing your app locally. You need to set the target devices and version of iOS that you intend to support. For example, if you want to test your app on both iPhone and iPad devices, set the target devices to Universal. Set the target iOS version as well, and be sure to test your app on the corresponding simulators and devices before creating the archive. For example, if you use storyboards, your minimum iOS version is 5.0. The selection of simulators changes depending on the values you set.

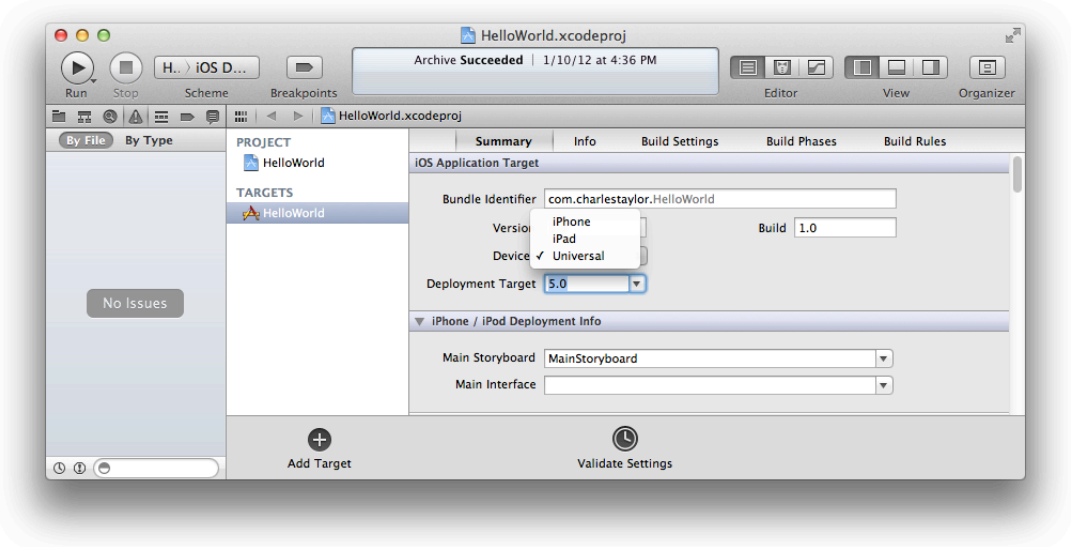
> **Important** Your app won't successfully install on a test device that is not supported by your app.

For this tutorial, set the HelloWorld target devices to Universal, and the deployment target to 5.0. Then test HelloWorld in the iPad 5.0 Simulator and iPhone 5.0 Simulator. If you have corresponding iOS devices, connect them to your Mac and run the app on them too.

To set the device families you want to support . . .

1. In the project navigator, select the project.
2. From the target list in the project editor, select the target that builds your app, and click Summary.
3. From the Devices pop-up menu, choose iPhone, iPad, or Universal (to target both device types).

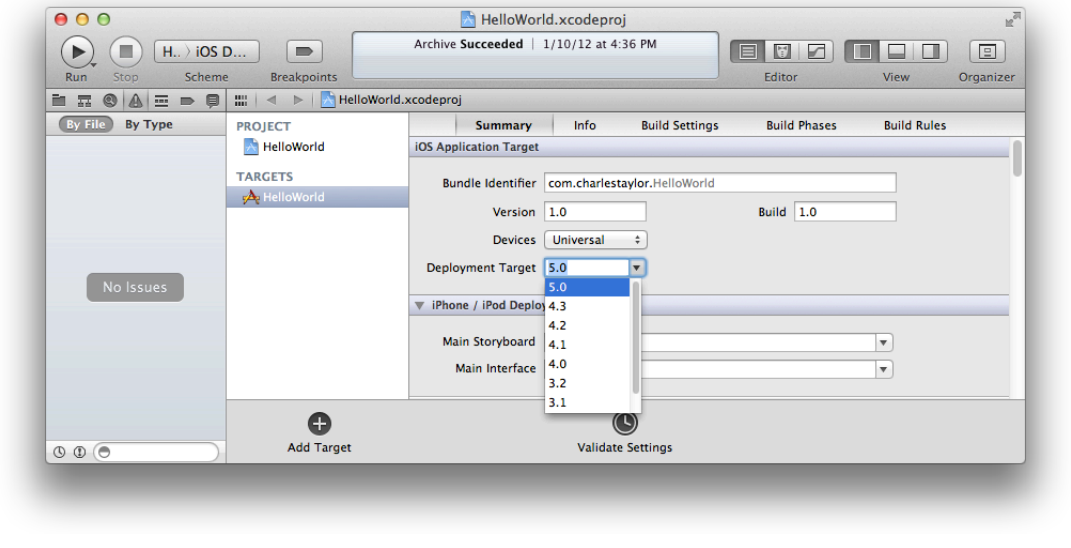
The scheme toolbar menu changes to include the devices you have selected.



To set the deployment target . . .

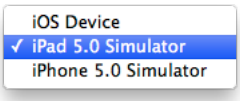
1. In the project navigator, select the project.
2. From the target list in the project editor, select the target that builds your app, and click Summary.
3. From the Deployment Target pop-up menu, choose the minimum iOS release you want to target.

   The scheme toolbar menu changes to include the iOS releases you selected.



To test your app on a simulator . . .

1. Choose a simulator from the scheme toolbar menu.



2. Choose Product > Run.

Register Device IDs

Before you can create an ad hoc provisioning profile, you need to collect device IDs from testers and add them to the iOS Provisioning Portal.

Testers can get their device ID using iTunes. (They do not need to install Xcode to do this.) Send these instructions to testers and ask them to send their device IDs to you.

To simulate a person testing your app, use a device and Mac that you are not using for development to get the device ID. Then follow the steps to add the device ID to the portal.

To get a device ID using iTunes . . .

1. Launch iTunes.
2. Connect your device to your Mac.
3. In the Devices section, select the device.
4. In the Summary pane, click the Serial Number label under Software Version.

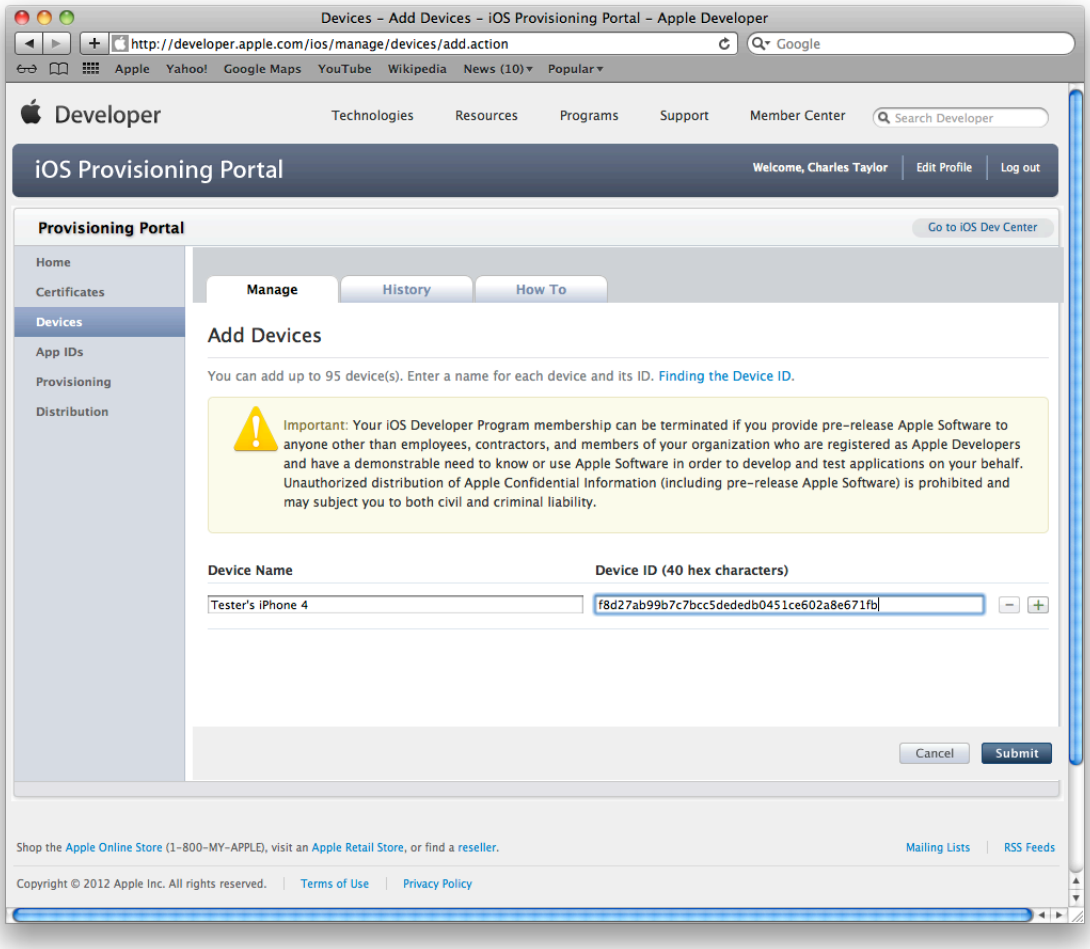
The label Serial Number changes to Identifier and displays the device ID.

5. To copy the device ID, position the pointer over the device ID and press Command-C.

To add a tester's device ID . . .

1. Log in to the iOS Provisioning Portal from the Member Center.
2. Click Add Devices.

3. Enter a device name and the device ID.



4. Click Submit.

Create Your Distribution Certificate

If you don't already have a valid distribution certificate in your keychain, refresh the provisioning profiles in Xcode again to create one. If you created your distribution certificate in the course of creating your development certificate, you can skip this step.

To request your distribution certificate . . .

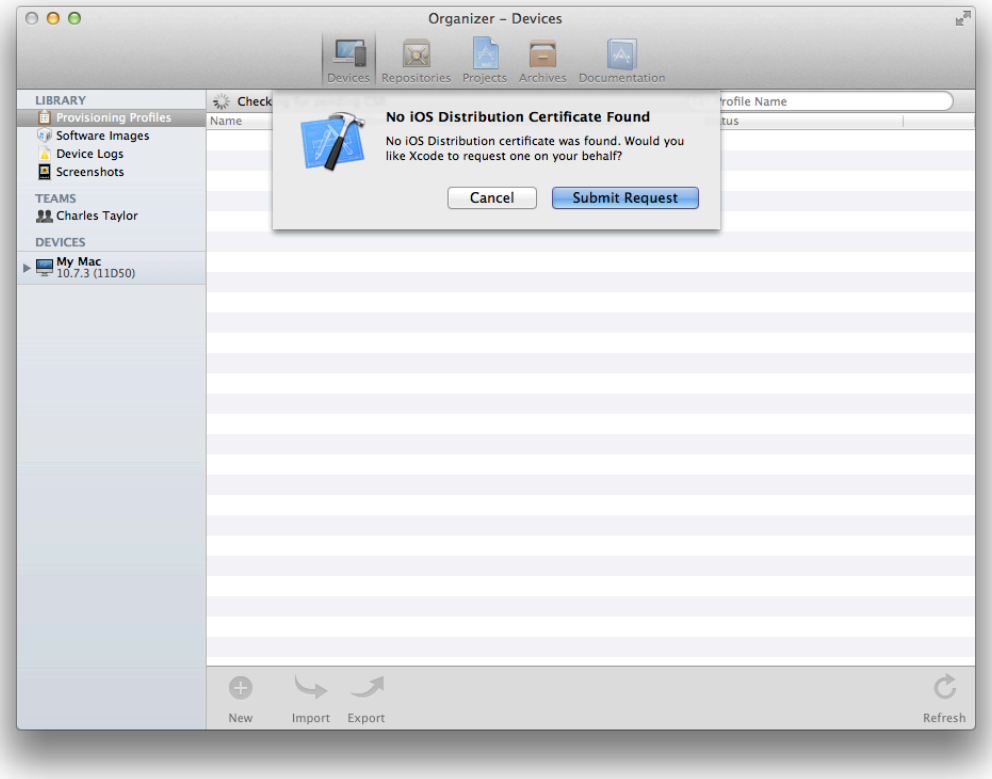
1. Choose Window > Organizer.
2. Click Devices.

3. In the Library section, select Provisioning Profiles.
4. Click the Refresh button at the bottom of the window.
5. Enter your user name and password, and click Log in.

   After you log in to your account, a prompt appears, asking whether Xcode should request your distribution certificate.

6. Click the Submit Request button.

   The distribution certificate is added to your keychain.



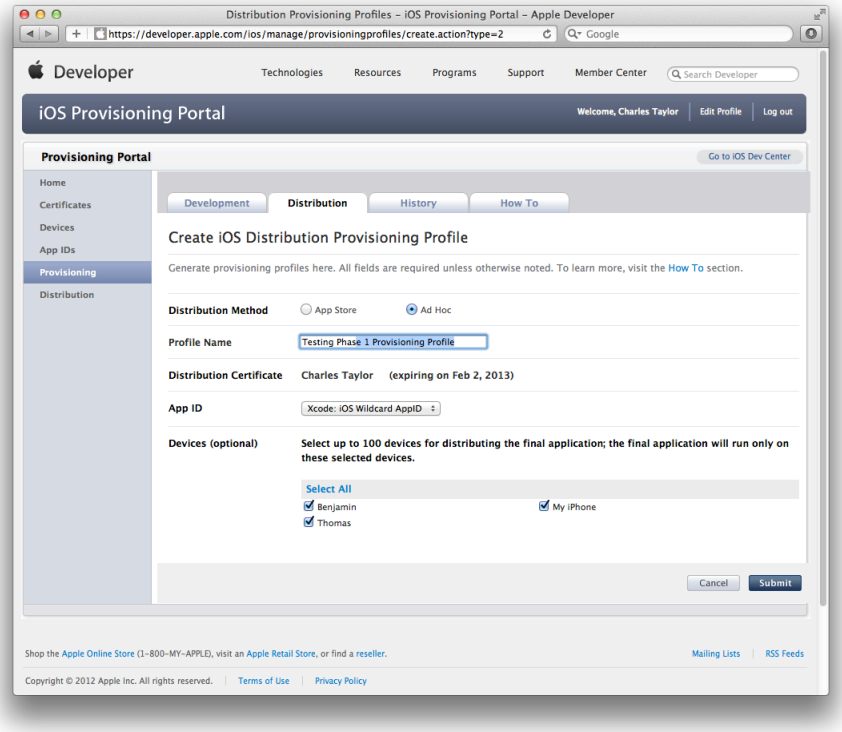
Create an Ad Hoc Provisioning Profile

Using the iOS Provisioning Portal, you create an ad hoc provisioning profile containing your app ID and the device IDs.

To create an ad hoc provisioning profile . . .

1. Log in to the iOS Provisioning Portal from the Member Center.

2. Select Provisioning in the sidebar.
3. Select the Distribution tab.
4. Select New Profile.
5. Select Ad Hoc as the distribution method.



6. Enter a profile name.
7. Confirm that your distribution certificate is displayed.
8. Select Xcode: iOS Wildcard AppID as the app ID.
9. Select the devices you want to be able to run the app.
10. Click Submit.

    When the status of the ad hoc provisioning profile changes from Pending to Active, you can begin using it. You may need to refresh the webpage in iOS Provisioning Portal to see the status change.

Create an iOS App Store Package for Testing

When the app is ready to be tested, use Xcode to create an archive and generate an iOS App Store Package (a file with a `.ipa` filename extension).

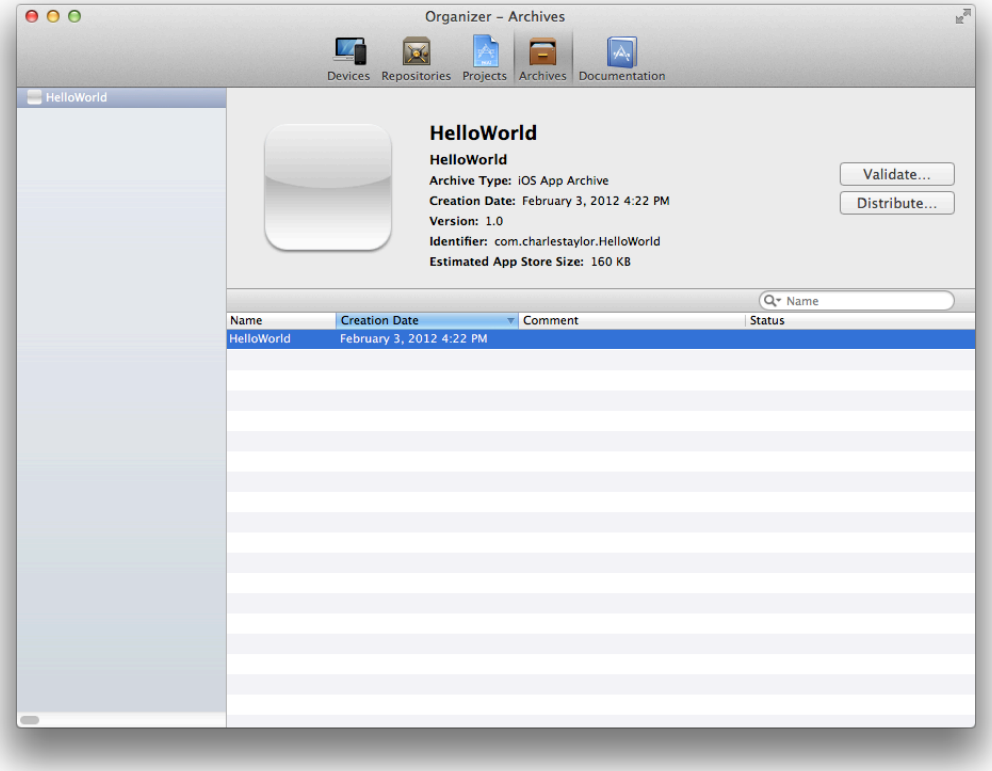
To create an archive . . .

1. Select the Xcode project window.
2. Choose iOS Device from the scheme toolbar menu.

   You cannot create an archive of a simulator build. If an iOS device is connected, the iOS Device menu item appears when you disconnect it.

3. Choose Product > Archive.

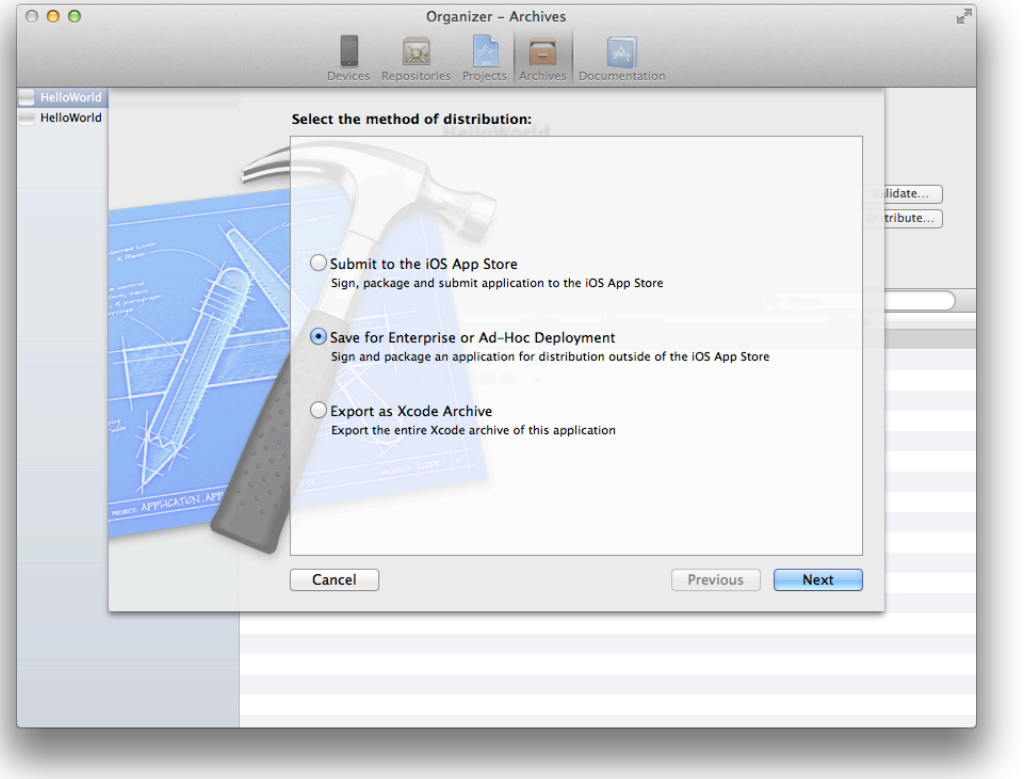
   The Archives organizer appears and displays the new archive.



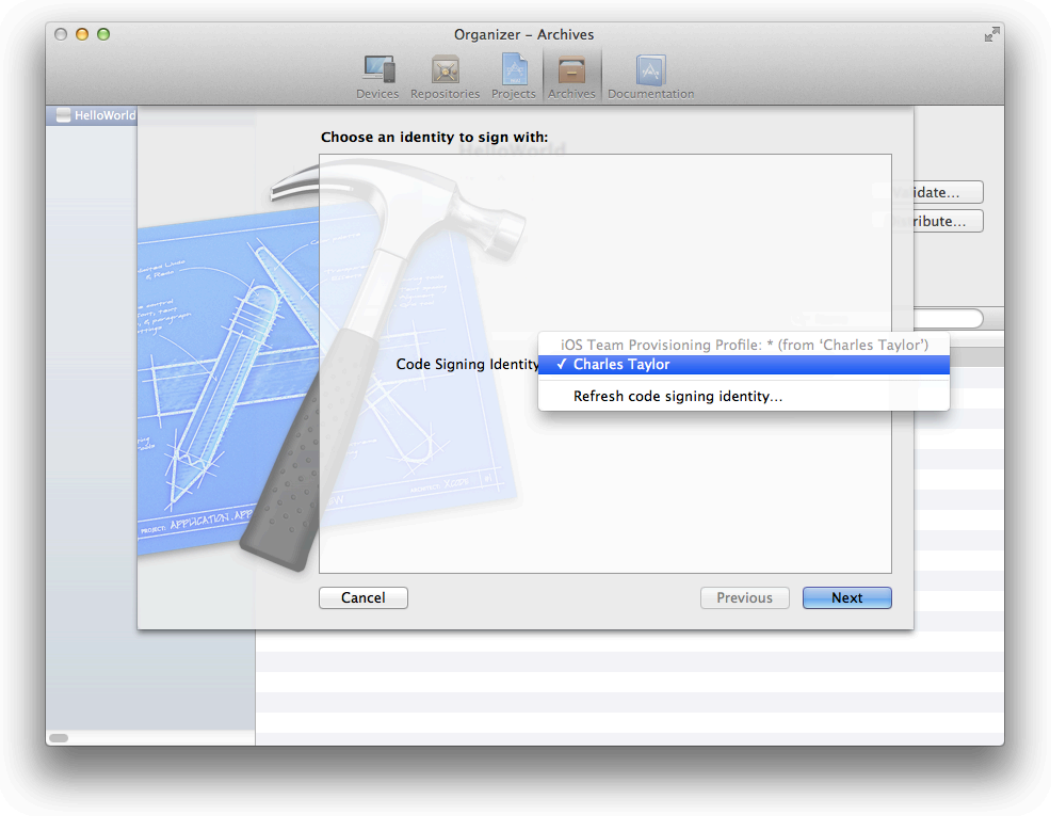
To create an iOS App Store Package for testing on devices . . .

1. In the Archives organizer, select the archive.
2. Click the Distribute button.

3. Select “Save for Enterprise or Ad-Hoc Deployment” and click Next.



4. Choose your development certificate from the Code Signing Identity pop-up menu and click Next.



5. Enter a filename and location for the App Store package and click Save.

   The file will have a `.ipa` extension

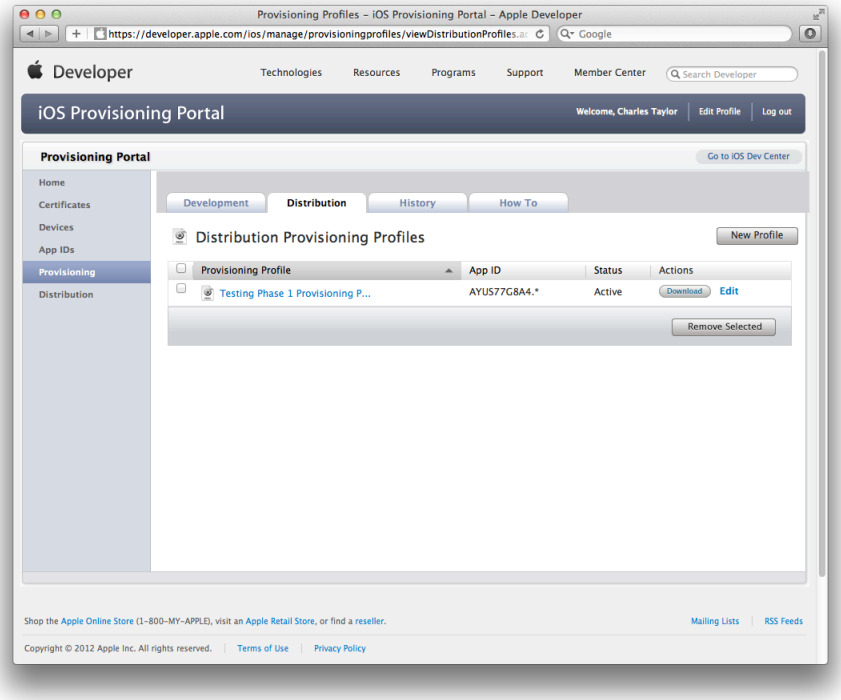
Install the Ad Hoc Provisioning Profile and App on the Device

Finally, download the ad hoc provisioning profile from iOS Provisioning Portal and give it to testers along with the IPA file. Testers will use iTunes to install the provisioning profile and the app on their device. When the app crashes on a device, iOS creates a record of that event. The next time the tester connects the iOS device to iTunes, iTunes downloads those records (known as crash logs) to the tester's Mac. Testers should send these crash logs to you.

To download the ad hoc provisioning profile . . .

1. Log in to the iOS Provisioning Portal from the Member Center.
2. Select Provisioning in the sidebar.

3. Select the Distribution tab.



4. In the Actions column click the Download button for the ad hoc provisioning profile you want to download.

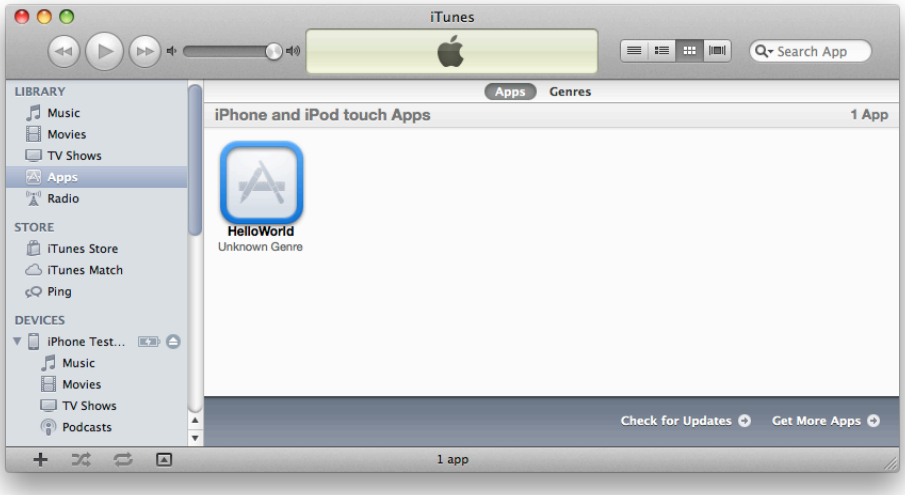
   A file with the provisioning profile name and the `.mobileprovision` extension appears in your Downloads folder.

After you send the application and ad hoc provisioning profile to testers, they perform the following tasks on their devices. Send these instructions to testers.

To install the app on a device . . .

1. Connect the testing device to a Mac running iTunes.
2. In the Finder, drag the provisioning profile (the file with the `.mobileprovision` extension) to the iTunes icon in the Dock.
3. Double-click the app archive (the file with the `.iap` extension).

The app appears in the iTunes app list.



4. In the Devices section, select the device.
5. Select the Apps tab.
6. Click Sync Apps and HelloWorld.
7. Click the Apply button to sync the device.

   This uploads the app and the provisioning profile to the device so that the user can start testing.

Send Crash Reports to Developers

It's difficult to simulate a crash using the HelloWorld app, but when you distribute your own app for testing, it may crash on test devices and you'll want to collect the crash reports. Send these instructions to testers, along with the ad hoc provisioning profile and app package.

To send crash reports to developers . . .

1. Connect the testing device to a Mac running iTunes.

   iTunes downloads the crash reports to your Mac.

2. In the Finder, choose Go > Go to Folder.
3. Enter `~/Library/Logs/CrashReporter/MobileDevice`.
4. Open the folder identified by your device's name.
5. Select the crash logs named after the app you're testing.

6. Choose Finder > Services > Mail > Send File.
7. In the New Message window, enter the developer's address in the To field and appropriate text in the Subject field.
8. Choose Message > Send.
9. (Optional) Delete the crash reports you sent in order to avoid sending duplicate reports later.

Creating Your App Record in iTunes Connect

When an app is sold in the App Store, the store displays a lot of information about the app, including its name, a description, an icon, screenshots, and contact information for your company. To provide that information, you log in to iTunes Connect, create a record for the app, and complete some forms. In this chapter you learn how to go through this process.

> **Note** Normally, you create your iTunes Connect app record late in the development process because there's a time limit from when you create the record to when you must submit your app. However, some Apple technologies, including Game Center and In-App Purchase, require that an iTunes Connect record to be created earlier. For example, with In-App Purchase, you need to create the app record so that you can add the details of the items you want to sell. This content needs to be created before the development process is complete so that you can use it to test the code you added to implement In-App Purchase.

When you create an iTunes Connect app record, an assistant guides you through the process and asks detailed questions about your company and app. For example, you need an official company name that iTunes Store displays for your app, a support email address, and support URL. You also need some screenshots of your app ready for upload. Some of this information is entered once and cannot be changed later. Other information can be changed up until you submit the first version of your app. Always refer to the online help for the latest information and instructions on using iTunes Connect.

Before You Begin

Before you begin, make sure that you have these assets ready to enter into the forms:

- The date when you want to ship your app (you can set the latest date allowed and change it later)
- A brief description of your app that iTunes will display to customers
- An app icon (512 x 512 pixels) ready for upload
- At least one screenshot of your app ready for upload
- An internal version number for your submission
- A bundle ID that you've set to match your app ID

Therefore before you can create the actual iTunes Connect app record, you need to accomplish three main tasks, using Xcode:

1. Capture screenshots.
2. Set the launch image.
3. Set your bundle ID.

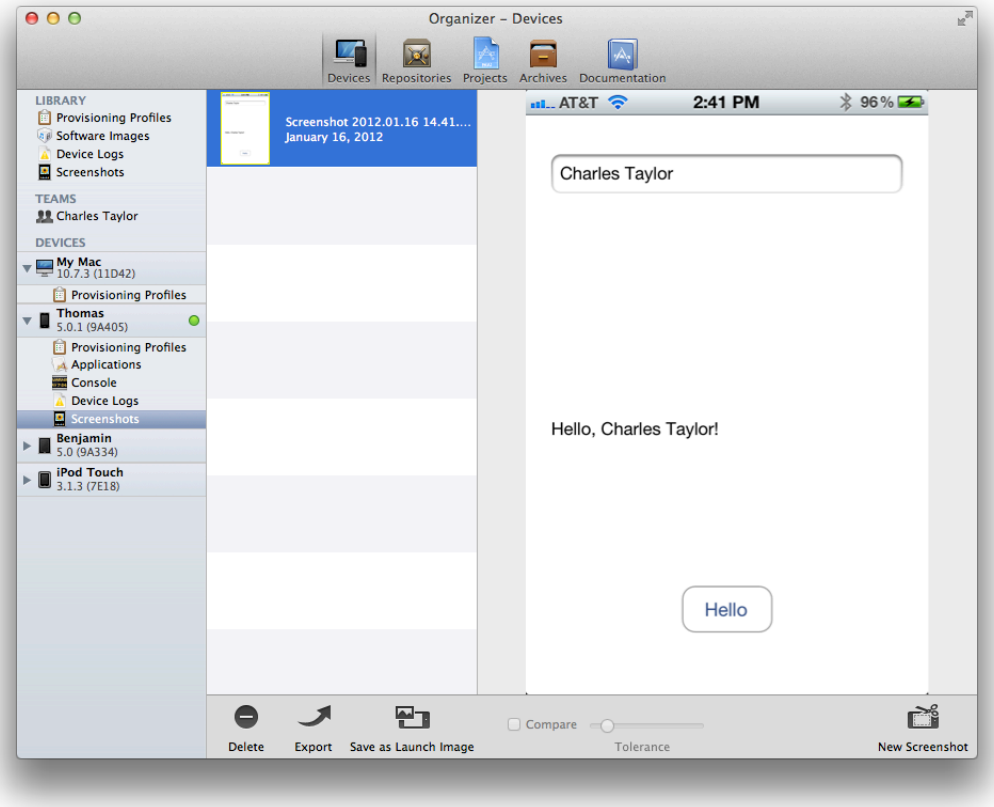
Capture Screenshots

Using Xcode, you can capture screenshots of your app on a device and save the images on your desktop to upload to iTunes Connect later and use as a launch image.

To capture a screenshot on your device . . .

1. Connect the device to your Mac.
2. In Xcode, run your app on the device (as described in "To launch the app on the device" (page 20)).
3. Configure the app on the device the way you want it.
4. In Xcode, open the Devices organizer.
5. In the Devices section, click the disclosure triangle next to the iOS device.
6. Select Screenshots .

7. Click New Screenshot in the lower-right corner.



To save a PNG file of a screenshot, drag the screenshot from Xcode to the desktop. Optionally, move the screenshots to a folder so that you can keep them all in one place.

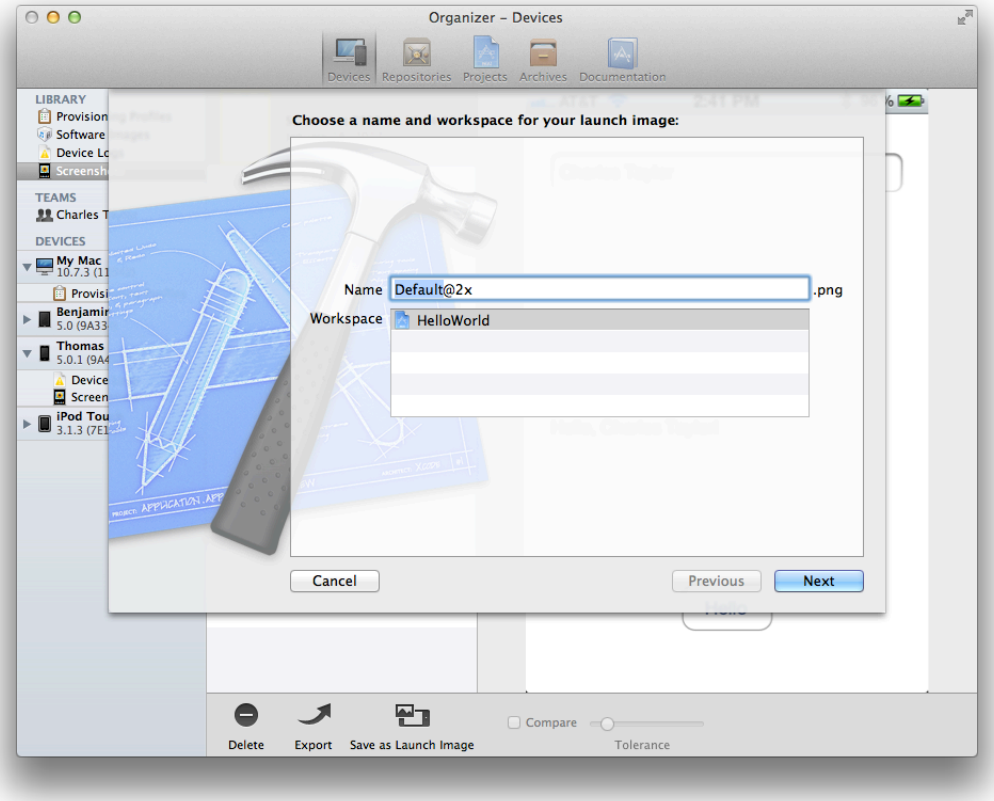
Set the Launch Image

Besides capturing screenshots, you should set a launch image that acts as a placeholder for your application's user interface at launch time. Do this at the same time that you capture screenshots for iTunes Connect.

To set your launch image . . .

1. In the Xcode Devices organizer, select the screenshot in the Screenshots folder of the Library section.
2. Select "Save as Launch Image."

3. Specify the name of the image and the target app, and click Next.



After performing these steps, the corresponding launch image is set in the Summary pane in Xcode. For example, if the screenshot was captured on an iPhone 4, the screenshot now appears in the Summary pane as the Retina Display launch image in iPhone/iPod Deployment Info.

Set the Bundle ID to Match the App ID

The record in iTunes Connect also includes a field for a bundle ID; the value you place in this field must exactly match the bundle ID for the app. The app name and version you enter in iTunes Connect must also match the Xcode project configuration.

You set the bundle ID for your app when you create your Xcode project as described in "Creating an Xcode Project" (page 12). For your convenience, the bundle ID defaults to a reverse-domain name using the company identifier followed by the product name that you entered when you created your Xcode project—for example, `com.charlestaylor.HelloWorld`, where `com.charlestaylor` is the company identifier and `HelloWorld` is the product name. The bundle ID is actually set in the `Info.plist` file of your Xcode project and can be changed to any identifier that follows the RFC 1034 specification.

The bundle ID that you enter in iTunes Connect cannot be changed after the first version of your app is approved or you enabled some specialized technologies, such as iCloud storage or push notifications. Therefore, you should pick the final bundle ID for your app now.

To set your bundle ID in the Info.plist file . . .

1. In the Xcode project navigator, select the project.
2. Select your target in the Targets section, in the second column, to display the project editor.
3. Click the Info tab.
4. Enter the bundle ID in the Value column of the "Bundle identifier" row.

Create the iTunes Connect App Record

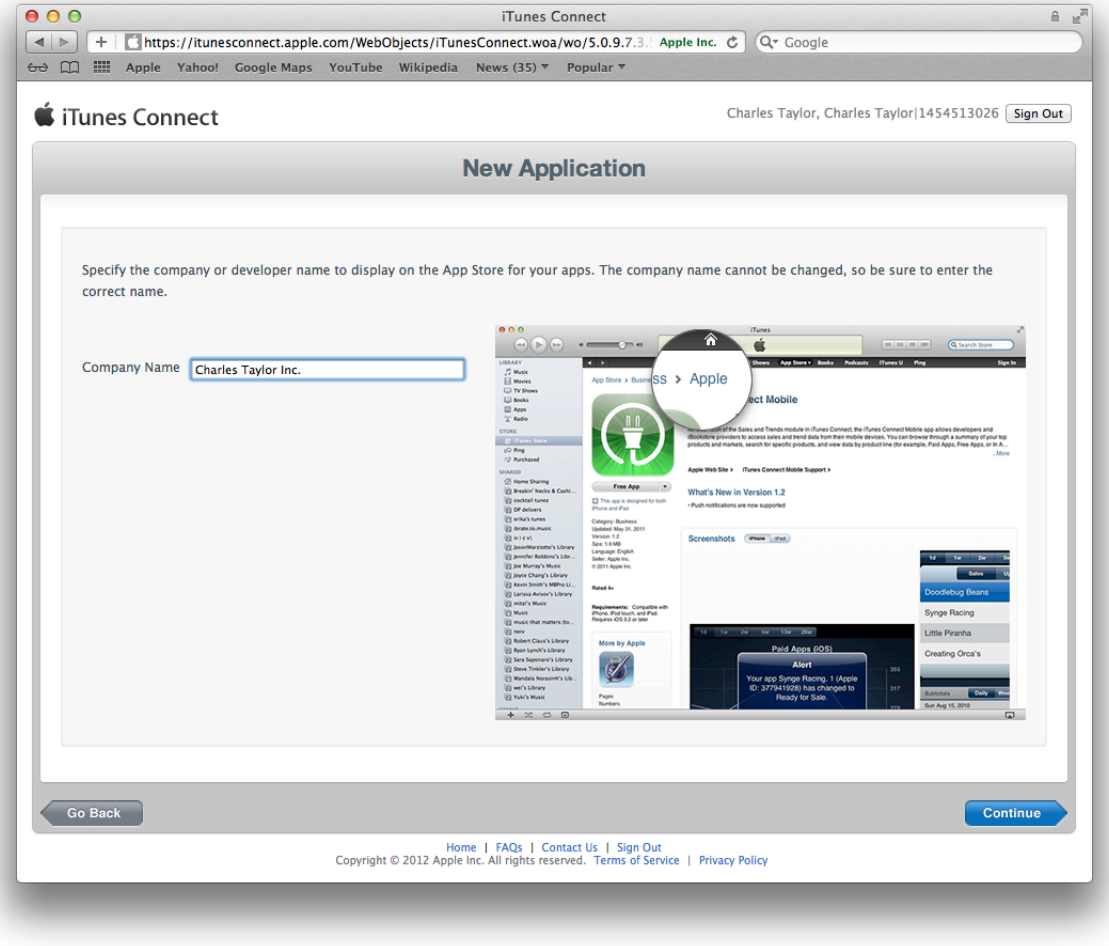
Next, you create the actual iTunes Connect app record and complete some forms. After you finish this process the status of your app is "Prepare for Download." You need to answer additional questions about export compliance before the status changes to "Waiting for Upload." The app record status needs to be at least "Waiting for Upload" to submit your app to the App Store.

Warning The steps to create your iTunes Connect app record and submit your app to the App Store are provided in this tutorial for you to refer to later when you are ready to submit your own app to the App Store. The HelloWorld app and other settings that appear in screenshots are examples of the type of information you need to provide. Do not create an app record for the HelloWorld app and submit it using your account.

To create an iTunes Connect app record . . .

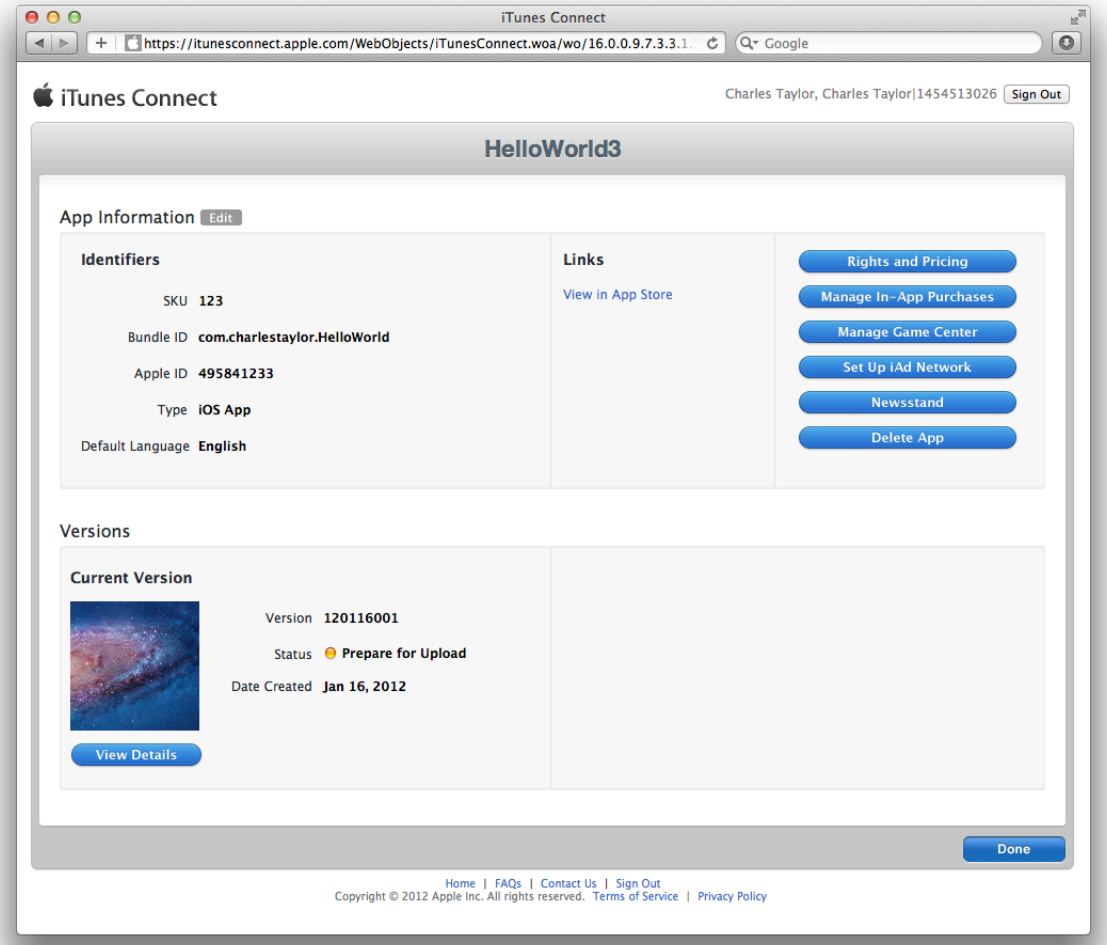
1. Log in to iTunes Connect (described in "To go to iTunes Connect" (page 11)).
2. Select Manage Your Applications.

3. Click Add New App.



4. Complete the forms by inserting your app's information.

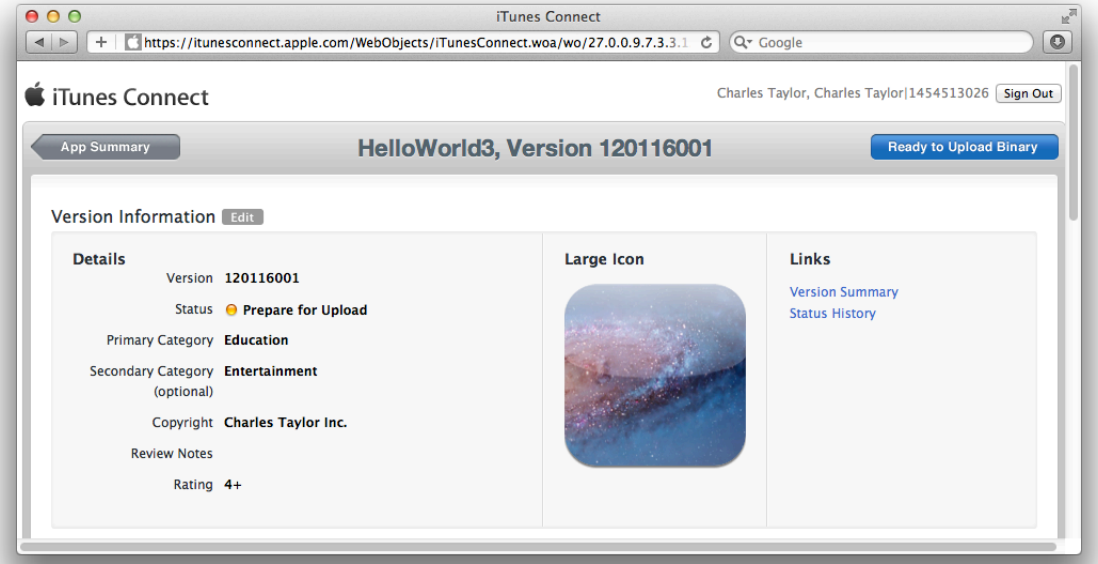
5. Click Done.



To complete the export compliance question . . .

1. In iTunes Connect, select Manage Your Applications.
2. Select your app.

3. Click View Details.



4. Click "Ready to Upload Binary."
5. Answer the Export Compliance question and click Save.
6. Click Continue.

   The status of the app should change to "Waiting for Upload."

Submitting Your App

Submitting your app to the App Store is a multistep process involving several tools. First, create a distribution provisioning profile using iOS Provisioning Portal. Then create an archive, validate it, and submit it to the App Store using Xcode. When your app is approved, set the date the app will be available to customers using iTunes Connect. Finally, don't forget to respond to user issues after you ship your first version.

Before You Begin

Before starting, you should have a distribution certificate you created in "Testing Your App on Many Devices and iOS Versions" (page 22) and the status of your iTunes Connect app record should be "Waiting for Upload" or later, as described in "Creating Your App Record in iTunes Connect" (page 37).

Now, create a distribution provisioning profile using the iOS Provisioning Portal.

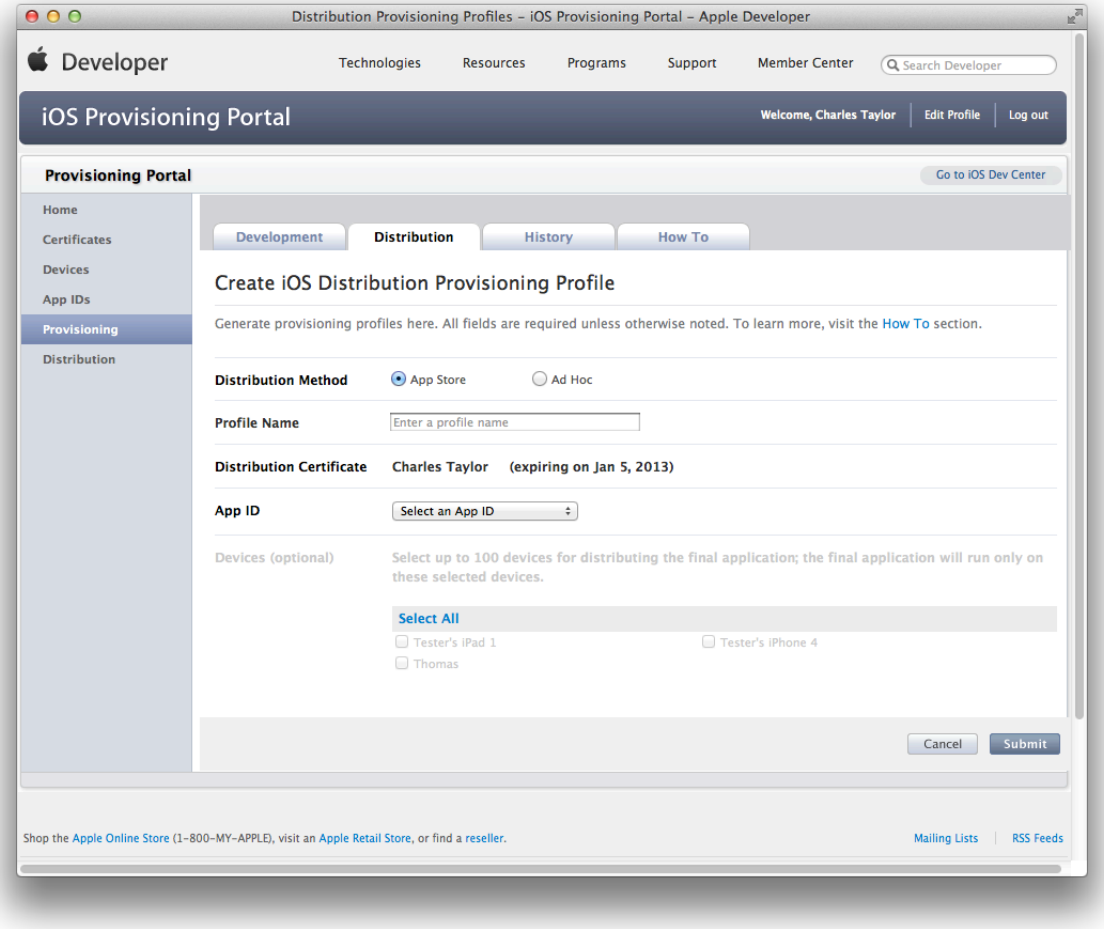
Create a Distribution Provisioning Profile

When the app is ready for publication, you create a distribution provisioning profile by selecting App Store as the method of distribution. The steps are similar to creating an ad hoc provisioning profile for testing except that you select an app ID only. You do not select any signing certificates or device IDs.

To create a distribution provisioning profile . . .

1. Log in to the iOS Provisioning Portal from the Member Center.
2. Select Provisioning in the sidebar.
3. Select the Distribution tab.
4. Click New Profile.

5. Select App Store as the distribution method.



6. Enter a profile name.
7. Confirm that your distribution certificate is displayed.
8. Choose Xcode: iOS Wildcard AppID as the app ID.
9. Click Submit.

   When the status of the ad hoc provisioning profile changes from Pending to Active, you can begin using it. You may need to refresh the webpage in the iOS Provisioning Portal to see the status change. If you are using Safari, select View > Reload Page.

Create and Validate the Archive

When the app is ready for submission, use Xcode to create and validate an archive. It's recommended that you use the version of your app that you built for testing in "Testing Your App on Many Devices and iOS Versions" (page 22) if it exhibited no problems during testing and is the version of the app you want to ship. Otherwise, you may inadvertently submit an untested version of your app.

> **Note** If your testers report crashes or other bugs and you fixed them, it is always wise to have them test a new version of your app. Ask them to verify that the bug is fixed before you submit this new version to the App Store for approval.

However, the archive that you submit must be signed with your distribution certificate contained in the distribution provisioning profile you created in "Create a Distribution Provisioning Profile" (page 45). You download the distribution provisioning profile from iOS Provisioning Portal and import it into Xcode. Then you can sign the archive using your distribution certificate.

Before you can submit the archive to the App Store, it must pass validation tests. The validation process performs an automated check against the app in the archive and against the information you provided in your iTunes Connect record. If problems are found during validation, you need to fix them before continuing. At this stage in the development process, it's good practice to periodically validate your archive to identify these types of problems early.

To download the distribution provisioning profile . . .

1. Log in to the iOS Provisioning Portal from the Member Center.
2. Select Provisioning in the sidebar.
3. Select the Distribution tab.
4. In the Actions column click the Download button for the distribution provisioning profile you want to download.

   A file with the provisioning profile name and the `.mobileprovision` extension appears in your Downloads folder.

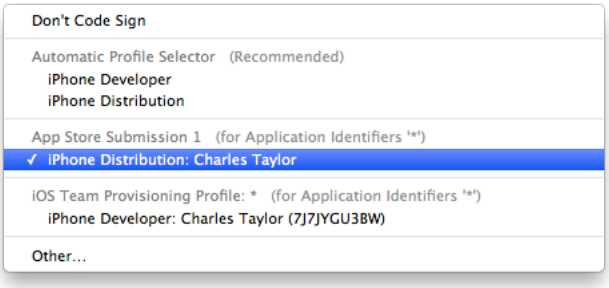
To import the distribution provisioning profile . . .

1. In Xcode, choose Window > Organizer.
2. Click Devices.
3. Select Provisioning Profiles.
4. Drag the provisioning profile with the `.mobileprovision` extension to the Devices organizer.

After Xcode imports the distribution provisioning profile, it should appear in the Devices organizer.

To set the code signing identity to your distribution certificate . . .

1. Select the project.
2. Click Build Settings.
3. Click All.
4. Type `Code Signing` in the search field in the Build Settings pane of the project editor.
5. From the Code Signing Identity pop-up menu, in the distribution provisioning profile section, choose the certificate that begins with "iPhone Distribution:" followed by your name.



To create the archive . . .

1. Select the Xcode project window.
2. Choose iOS Device from the scheme toolbar menu.

> **Note** You cannot create an archive of a simulator build.

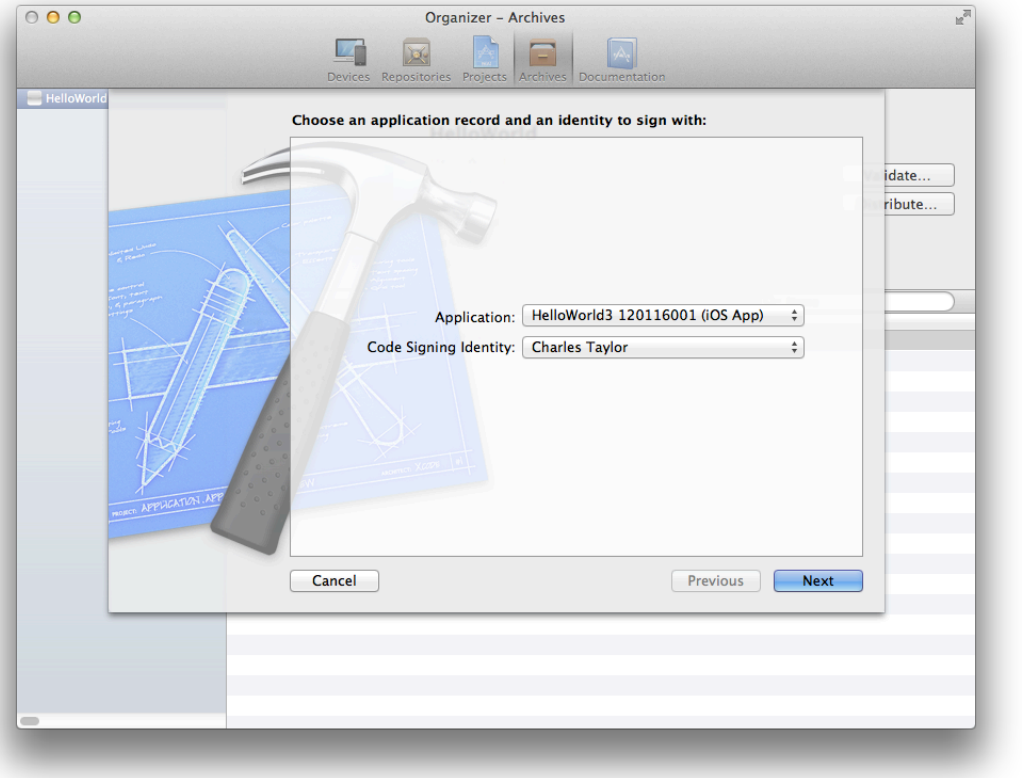
3. Choose Product > Archive.

   The Archives organizer appears and displays the new archive.

To validate the archive . . .

1. In the Archives organizer, select the archive.
2. Click the Validate button.
3. Enter your iTunes Connect credentials and click Next.

4. Select the app you want to share and the appropriate signing identity, and click Next.



5. Review validation issues found, if any, and click Finish.

You need to fix any validation issues, create a new archive, and validate it again. You cannot proceed until the archive passes the validation tests.

Submit and Ship Your App

Submitting your app to the App Store is a multistep process too. First, you transmit the archive to Apple, then Apple approves your app, and then you set a ship date before it actually appears on the App Store.

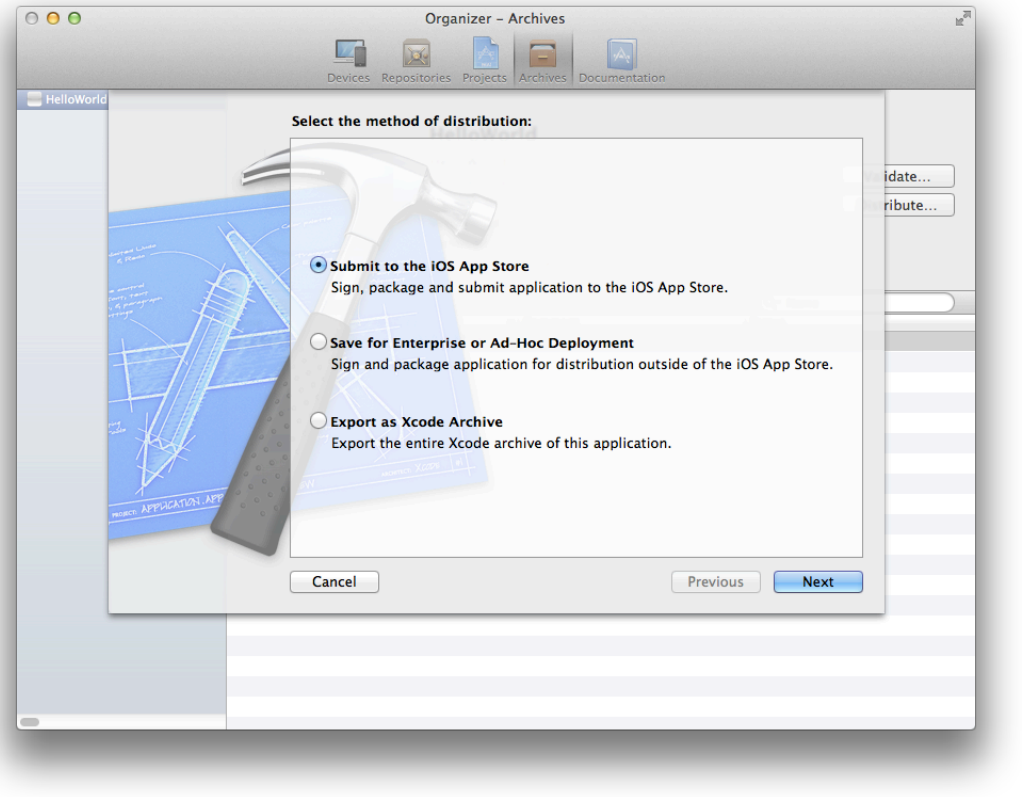
Warning The steps to create your iTunes Connect app record and submit your app to the App Store are provided in this tutorial for you to refer to later when you are ready to submit your own app to the App Store. The HelloWorld app and other settings that appear in screenshots are examples of the type of information you need to provide. Do not create an app record for the HelloWorld app and submit it using your account.

Submit the Archive

Only after passing validation tests, can you submit your app to the App Store. In fact, Xcode validates your archive again during the submission process.

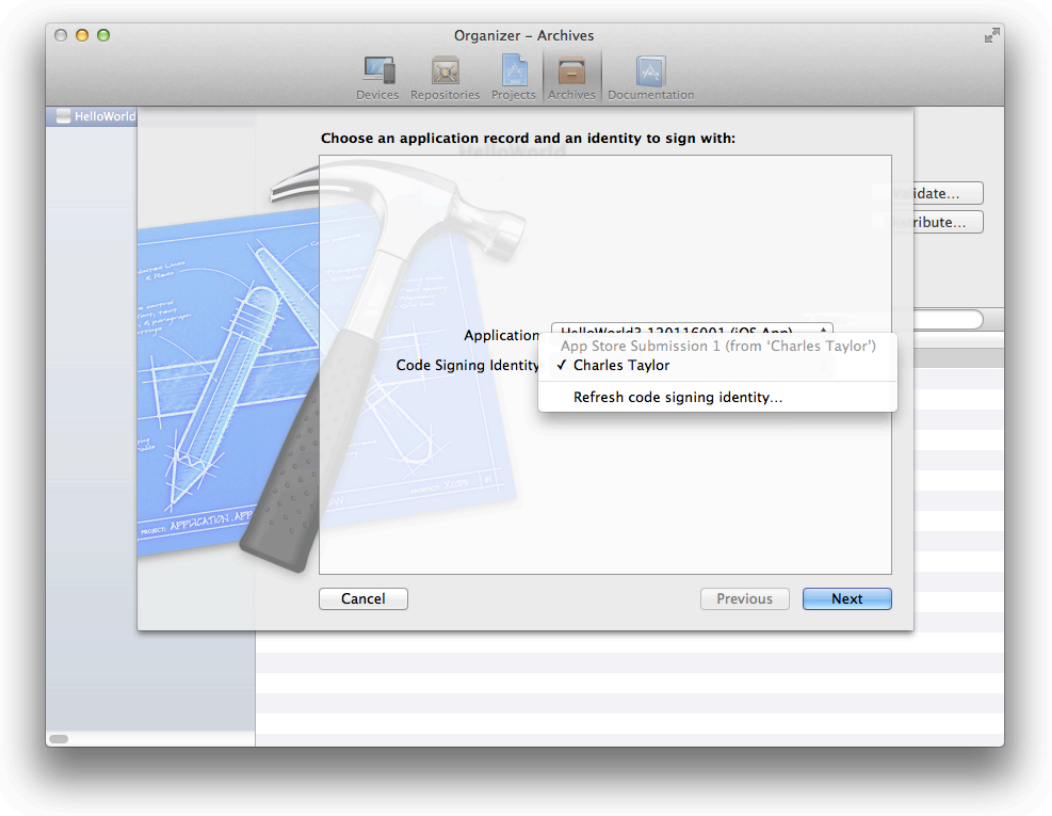
To submit the archive to the App Store . . .

1. In the Archives organizer, select the archive.
2. Click the Distribute button.
3. Select "Submit to the iOS App Store" and click Next.



4. Enter your iTunes Connect credentials and click Next.
5. Select the app you want to share and the appropriate signing identity, and click Next.

Xcode runs the validation tests again. If issues are found, click Cancel and fix them before continuing.



6. Enter a filename and location for the App Store package and click Save.

Xcode transmits the archive to Apple, where it is examined to determine whether it conforms to the app guidelines. If the app is rejected, correct the problems that were brought up during app approval and resubmit it. Before you submit the app, you should read *iOS Human Interface Guidelines* and *App Store Review Guidelines for iOS Apps* to avoid problems.

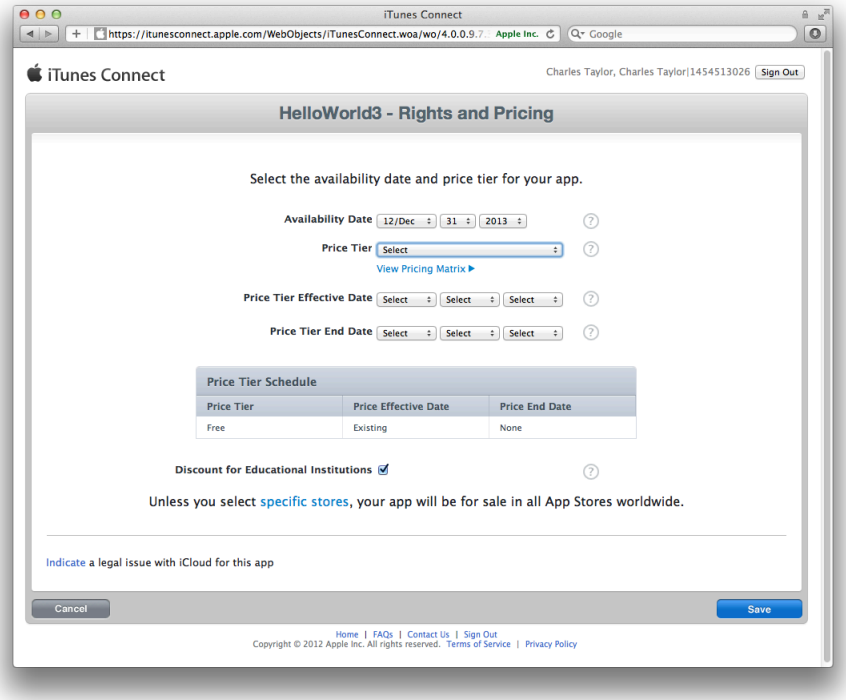
Ship Your App

Use iTunes Connect to set a date when the app is available on the App Store. For example, you can choose a date that immediately releases the app to the App Store after it is approved, or you can set a date for sometime in the future. Using a later availability date allows you to arrange other marketing activities around the launch of your app.

To set the availability date . . .

1. Log in to iTunes Connect (described in "To go to iTunes Connect" (page 11)).
2. Select Manage Your Applications.

3. Select your app in iOS App Recent Activity.
4. Click Rights and Pricing.
5. Choose a date from the Availability Date pop-up menus.



6. Optionally, edit the other fields on this form.
7. Click Save.

Changes you make to Rights and Pricing go live immediately (expect 24 hours for a full refresh of the changes on the App Store).

Respond to User Issues

You can't just submit the app to the App Store and forget about it. Instead, expect to manage the app records and maintain the app throughout its lifetime. Once it's available on the App Store, you need to monitor your app, respond to user issues, and submit updates as needed.

Pay attention to how users perceive the app. Customer ratings and reviews on the App Store can have a big effect on the success of the app; if users run into problems, work quickly to determine the bug and submit a new version of the app through the approval process.

Investigate iTunes Connect data. iTunes Connect provides data to help you determine how successful the app is, including sales and financial reports, customer reviews, and crash logs submitted to Apple by users. Crash logs are particularly important, because they represent significant problems users are seeing in the app. You should make investigating these reports a high priority.

Except for low-memory crash logs, all crash logs contain stack traces for each thread at the time of termination. To view a crash log, you need to open it in the Xcode Organizer. As long as your Mac has the archive corresponding to the version of the app that generated the crash log, Xcode automatically resolves any addresses in the crash log with the actual classes and functions in the app.

Review and Troubleshooting

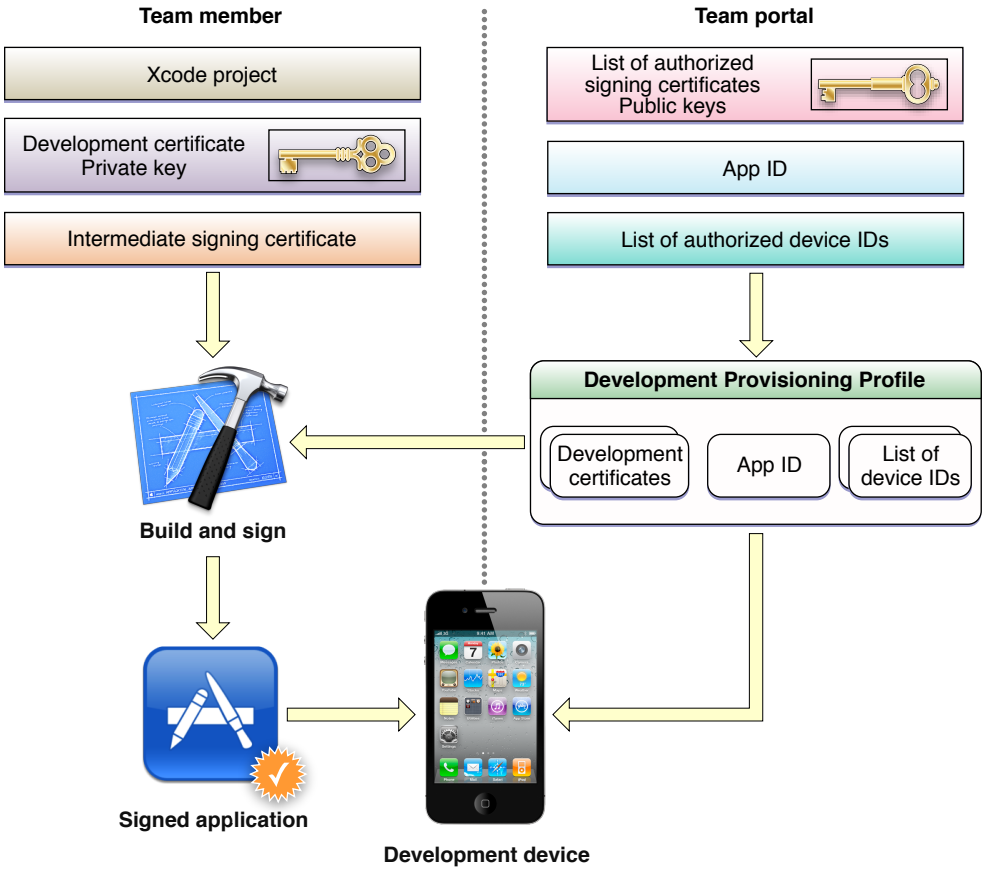
This tutorial has taught you how to provision your devices for development and testing, and how to prepare your app for distribution. However, provisioning and code signing are complicated. All the pieces need to be in the right place for you to run your app on a device and distribute it successfully. If you inadvertently change your app configuration, a key in your keychain, or an iOS Provisioning Portal asset, provisioning may fail. Fortunately, there are some common problems that are easy to fix.

For a complete discussion of troubleshooting, read "iOS Development: Troubleshooting".

Review: How Provisioning Works

A provisioning profile allows an app to run on a device. A provisioning profile resides on the iOS portal and is downloaded by Xcode and eventually installed on a device. Development and ad hoc provisioning profiles are conceptually similar. A development provisioning profile is used by the team to test the app during development and contains a list of development certificates, an app ID, and list of device IDs. An ad hoc provisioning profile allows outside testers to run the app on specific devices and contains a single distribution certificate, an app

ID and list of device IDs. Both development and ad hoc provisioning profile are installed on a device. If the configuration of the signed app matches the provisioning profile installed on the device, the app launches on the device.



When you create a development or distribution certificate on your Mac, the private key is stored in your keychain and the public key is stored in the portal. These certificates are issued by Apple and validated by an intermediate signing certificate, called *Apple Worldwide Developer Relations Certification Authority*, which is added to your System keychain when you install Xcode.

If you use a specialized technology that requires an explicit app ID, you need to register that app ID using iOS Provisioning Portal. The app ID you create needs to match the bundle ID set in the Xcode project. The bundle ID is contained in the signed app that is installed on the device, and it is compared with the app ID in the provisioning profile at launch time. If the bundle ID matches the app ID, the app launches.

New devices that you add to the portal using Xcode are automatically added to the default iOS Team Provisioning Profile but are not automatically added to specialized provisioning profiles. If you use a specialized provisioning profile, you need to update the provisioning profile using iOS Provisioning Portal and refresh it in Xcode to run your app on a new device.

Verify Your Certificates Using Keychain Access

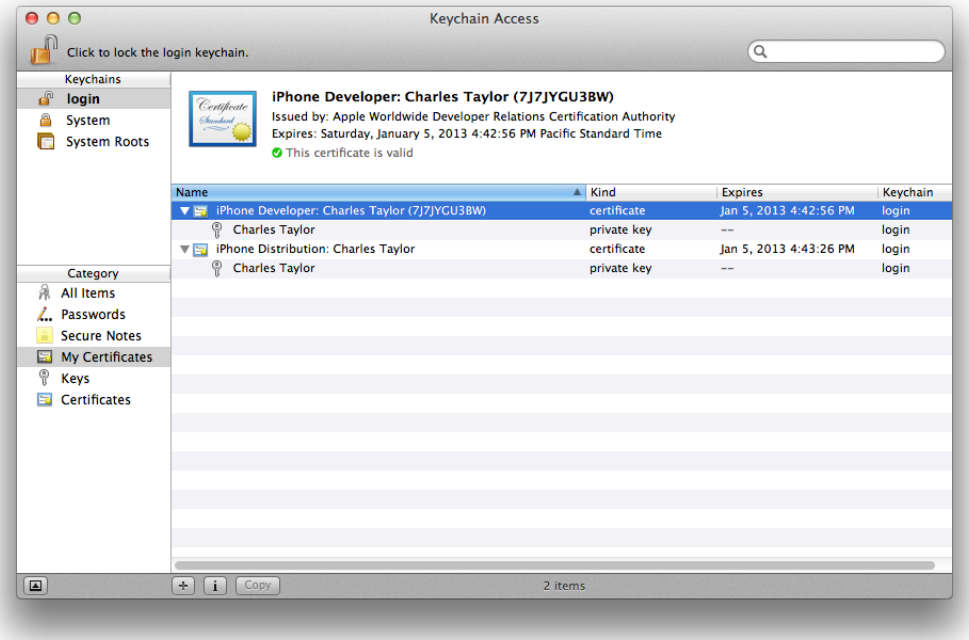
After you request a development or distribution certificate using Xcode and it is approved, the certificate is installed in your login keychain. In Xcode, code signing fails if these certificates are not valid. If you accidentally removed the intermediate signing certificate, you can retrieve it from iOS Provisioning Portal and install it again. First, you should verify the state of your certificates using Keychain Access.

To verify your certificates using Keychain Access . . .

1. Launch Keychain Access, located in the `/Applications/Utilities` folder.
2. In the Category sidebar, select Certificates .
3. In the window's search field, enter `iPhone`.

   Two certificates beginning with the text "iPhone" should appear. The name of the development certificate begins with the text "iPhone Developer:" and the distribution certificate begins with the test "iPhone Distribution:" followed by your name. Each certificate should have a disclosure triangle next to the name.

   You should remove other certificates beginning with "iPhone" that do not have a disclosure triangle. This can occur if the associated private key is accidentally deleted from Keychain Access.



4. Click the disclosure triangle next to the name of the certificates to reveal the private keys.

   The private key is stored in your keychain and the public key is stored in the portal.

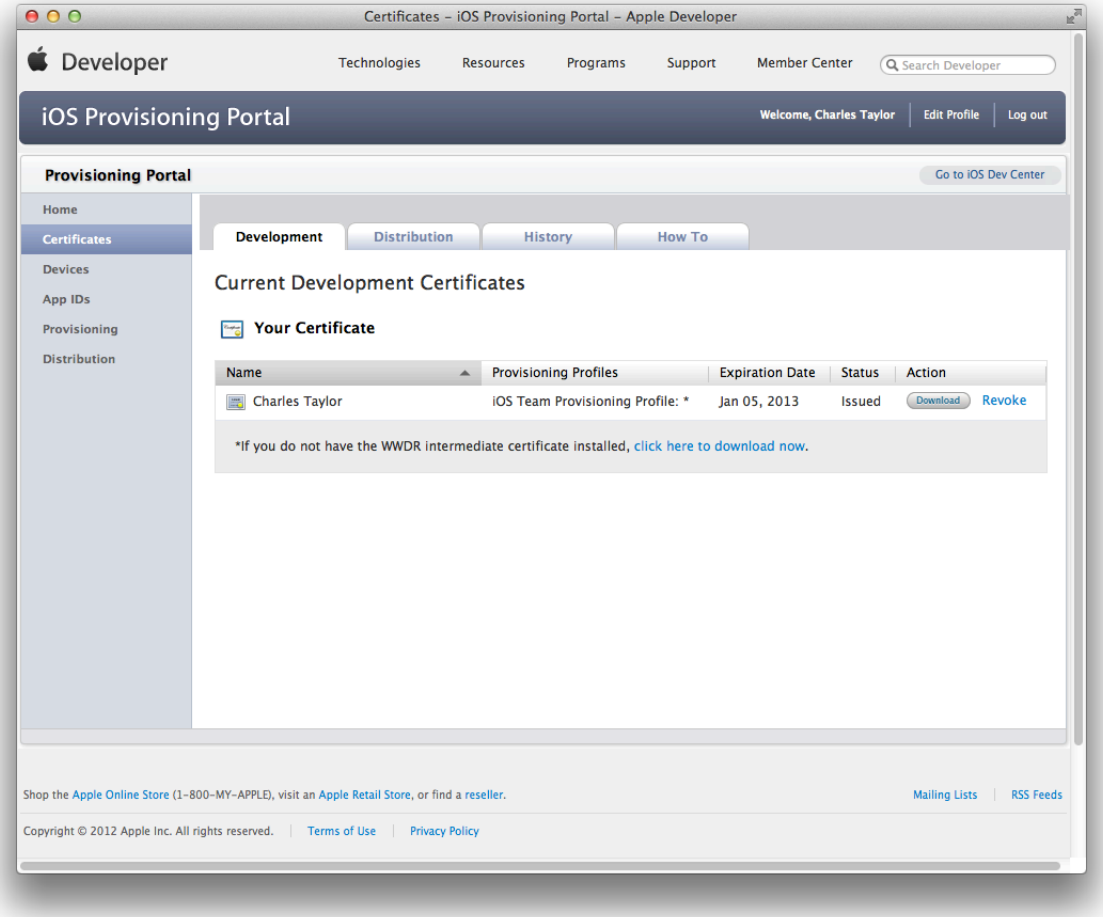
5. Select each certificate separately.

   The pane above should display a green circle containing a checkmark, and the text next to the circle should read “This certificate is valid.”

To install the intermediate certificate . . .

1. Log in to the iOS Provisioning Portal from the Member Center.
2. In the sidebar, click Certificates.
3. Click the text that follows “If you do not have the WWDR intermediate certificate installed.”

   The WWDR certificate is downloaded.



4. Double-click the certificate file to install it in your System keychain.

Export and Import Your Digital Identities

Because a signing certificate public key is stored on the portal and the private key is stored in your login keychain, you can't refresh your provisioning profiles and certificates in Xcode to replace a missing private key. Instead, you should export your digital identities after they are first created as a backup or if you want to develop on another Mac. The exported file is password protect. As with any personal information, you should not share this file and the password with others. Later, you can import your digital identities to repair your keychain or on another Mac you want to use for development.

When you refreshed your provisioning profiles and created your development and distribution certificates for the first time, Xcode displays a prompt asking if you want to save your developer profile. If you did not choose the export option then, you should export your digital assets now. If you move to another development system, you need to import your digital identities on the new system.

To export your code signing certificates and provisioning profiles . . .

1. In the Devices organizer, select a team from the Team section.
2. Click Export.
3. Enter a filename and password.
4. Click Save.

   The file saved has a `.developerprofile` extension.

To import your code signing certificates and provisioning profiles . . .

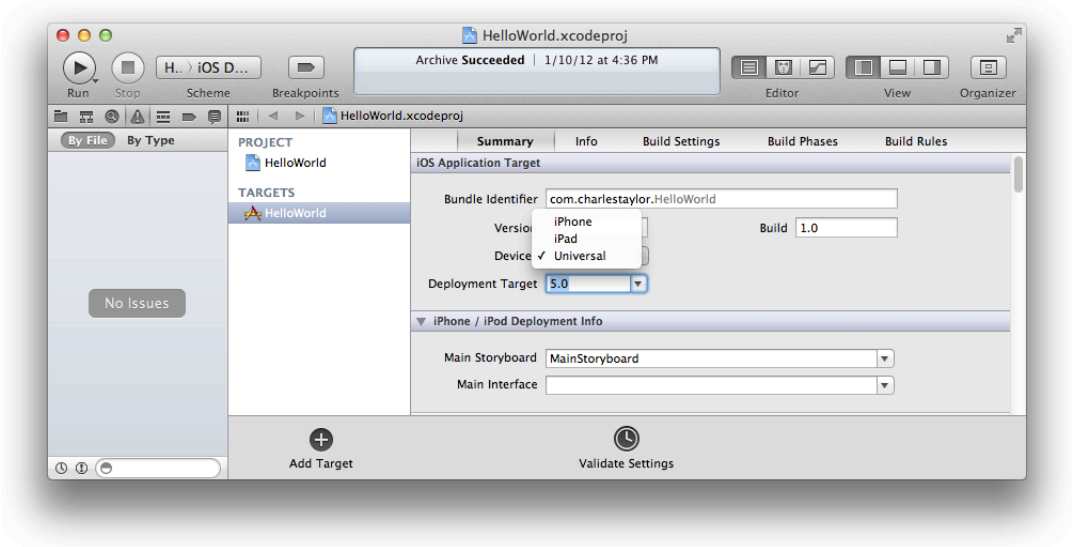
1. In the Devices organizer, select a team from the Team section.

   If the Team section doesn't appear, refresh your provisioning profiles before continuing, as described in "To request your development certificate" (page 16).
2. Click Import.
3. Select the developer profile file you want to import.
4. Enter the password for the file.
5. Click Open.

   The code signing certificates contained in the file are added to Xcode and to your keychain. The provisioning profiles are added to Xcode.

Verify the iOS App Target Settings

When you distribute an ad hoc provisioning profile and installer package to testers, even though testers may successfully install the app on their device, it may not launch. This can occur if the iOS App Target settings do not match the device and installed iOS version. Verify that the iOS App target settings, in the Summary pane in Xcode, are correct. For example, if the Devices pop-up menu is set to iPad, your app won't launch on an iPhone. Choose Universal from this pop-up menu if you want to support all types of devices. If your Deployment Target is set to the most recent iOS version, your app may not run on older devices.



Next Steps

This tutorial assumes that you can develop and test your app on devices using the iOS Wildcard App ID and iOS Team Provisioning Profile. However, if you use any of the specialized technologies—such as iCloud storage, push notifications, or Game Center—you need to use an explicit app ID and specialized ad hoc and distribution provisioning profiles. Furthermore, if are not an individual developer, you may need to add persons to your team and perform other team administration tasks.

Using iCloud Storage

iCloud storage allows you to share the user's data among multiple instances of your app running on different iOS and Mac OS X devices. You can test your iCloud app using the iOS Team Provisioning Profile that Xcode creates for you but first you need to enable and set iCloud storage entitlements using Xcode. To learn more about using iCloud storage, read *Your Third iOS App: iCloud*.

Using Other Technologies

If you use push notifications, In-App Purchase, or Game Center, you need to register an explicit app ID, in some cases, enable these technologies, and create specialized provisioning profiles for development and testing using the iOS Provisioning Portal. Read *Tools Workflow Guide for iOS* to learn how to perform these administrative tasks, and read the following documents to learn more about specific technologies:

- *Local and Push Notification Programming Guide*
- *In-App Purchase Programming Guide*
- *Game Kit Programming Guide*

Managing Your Team

If you enrolled in the iOS Developer Program as a company, you are the primary contact for your development team. If your team members also need signing certificates and provisioning profiles for development and testing, you need to add team members to your account and manage these assets. Read *iOS Team Administration Guide*, and for Xcode tasks read *Tools Workflow Guide for iOS*.

Getting Your App Approved

Apps are approved only if they adhere to the published Apple guidelines. To learn how to design an app for the App Store, read *iOS Human Interface Guidelines* and *App Store Review Guidelines for iOS Apps*.

Marketing Your App and Managing Your Account

There are additional tasks you need to perform to market and maintain your product. These tasks are described in *iTunes Connect Developer Guide*.

Document Revision History

This table describes the changes to *Your First App Store Submission* .

Date	Notes
2012-03-07	Updated for Xcode 4.3.1.
2012-02-16	New tutorial that describes how to provision your devices for testing and submit your app to the App Store.

Apple Inc.
1 Infinite Loop
Cupertino, CA 95014
408-996-1010