Launching Your App on Devices

All iOS, tvOS, and watchOS apps must be code signed and provisioned to launch on a device. Mac apps that use certain app services must be code signed to launch too. *Provisioning* is the process of preparing and configuring an app to launch on devices and use app services. Xcode uses information you provide to create a team provisioning profile for you when you assign your Xcode project to a team or the first time you add capabilities to your app. For example, Xcode automatically creates your development certificate and registers a connected device or your Mac. Xcode uses this information to create a provisioning profile that's ultimately installed on the device. For iOS, tvOS, and watchOS apps, Xcode runs an app on a device if that device is in the provisioning profile. Similarly, a Mac app that uses certain app services launches only if the Mac is in the provisioning profile.

You can launch your app on a device using a free Apple ID account, but the capabilities available to your app depends on the platform and your Apple Developer Program membership, described in Supported Capabilities.

> **Tip:** To avoid issues later when you distribute your app, you should test your app running on actual devices. To create and run tests with Xcode, read *Testing with Xcode*. To learn about the Xcode debugging tools, read *Debugging with Xcode*. To do in-depth analysis and profiling using Instruments, read *Instruments User Guide*.

Launching Your Mac App

To launch your Mac app, click the Run button in the project navigator. If the signing identity is set to None, Xcode launches your app. Otherwise, Xcode registers your Mac and adds it to the team provisioning profile before launching your app. If a prompt appears asking whether `codesign` can sign the app using a key in your keychain, click Always Allow.

Launching Your App on a Device (iOS, tvOS, watchOS)

It takes just a few steps to launch your app on a device if you previously created your code signing identity and team provisioning profile, as described in Creating the Team Provisioning Profile. Otherwise, a series of dialogs and warnings may appear as Xcode resolves the code signing issues in the process of launching your app.

To launch an app on a device

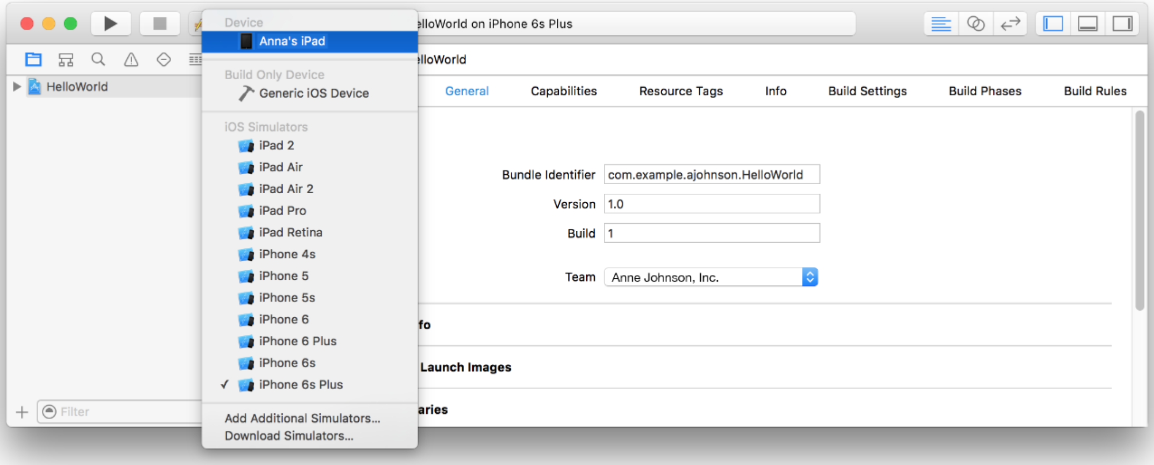
1. Connect the device to your Mac.

   > **watchOS Note:** Connect an iPhone to your Mac that is paired with an Apple Watch.

2. In the project navigator, choose your device from the Scheme toolbar menu.

   > **watchOS Note:** If multiple Apple Watches are paired to an iPhone, the active Apple Watch appears in the Scheme toolbar menu. If you change the active Apple Watch using the iPhone, the Scheme toolbar menu item changes to the active watch. To view all Apple Watches paired to an iPhone, open the Devices window, select the iPhone, and view the watches in the Paired Watches table.

   Xcode assumes you intend to use the selected device for development and automatically registers it for you.



Bundle Identifier com.example.ajohnson.HelloWorld

Version 1.0

Build 1

Team Anne Johnson, Inc.

If your device is disabled in the Scheme toolbar menu because it is ineligible, fix the issue before continuing. Under Ineligible Devices, hover the mouse over the device to read the reason why it's ineligible. For example, if the OS version is lower than the deployment target, upgrade the OS version on the device or choose the version you want to target from the Deployment Target pop-up menu (located in the Deployment Info section). Then select the device from the Scheme toolbar menu.

3. Click the Run button.

   Xcode installs the app on the device before launching the app.

> **tvOS Note:** If a message appears stating that the Apple TV is not activated, connect the Apple TV to a television and perform the user setup steps. Connect the Apple TV to your Mac again and in Xcode, select the Apple TV in the Devices window. If the warning icon and “Apple TV is not activated.” message disappears, you can install and launch your app on the device.

4. If a prompt appears asking whether `codesign` can sign the app using a key in your keychain, click Always Allow.

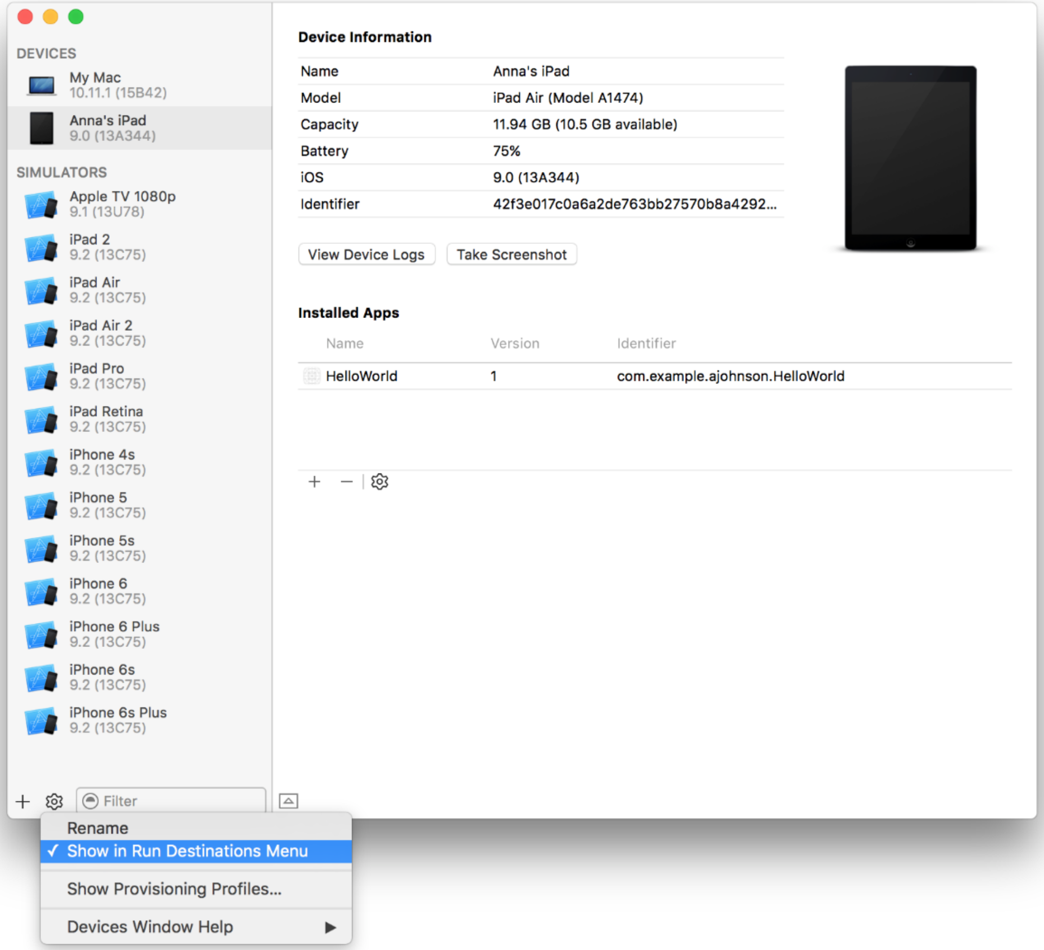
While developing your app, run and test it on all of the devices and OS versions that you intend to support. Because different instruments are available in a simulator, also test your app in a simulator using Instruments and other tools before distributing your app. For details on how to use device simulators to test your app, read *Simulator User Guide*.

Removing a Device from the Scheme Menu (iOS, tvOS, watchOS)

If you want Xcode to ignore a connected device—you don't want Xcode to add it to the team provisioning profile—remove it from the Scheme menu.

To remove a device from the Scheme menu

1. Connect the device to your Mac.
2. In Xcode, choose Window > Devices, and select the device under Devices.
3. In the lower-left corner of the Devices window, click the Action button (the gear icon to the right of the Add button).



Device Information

Name	Anna's iPad
Model	iPad Air (Model A1474)
Capacity	11.94 GB (10.5 GB available)
Battery	75%
iOS	9.0 (13A344)
Identifier	42f3e017c0a6a2de763bb27570b8a4292...

Installed Apps

Name	Version	Identifier
HelloWorld	1	com.example.ajohnson.HelloWorld

4. Deselect “Show in Run Destinations Menu” from the pop-up menu.

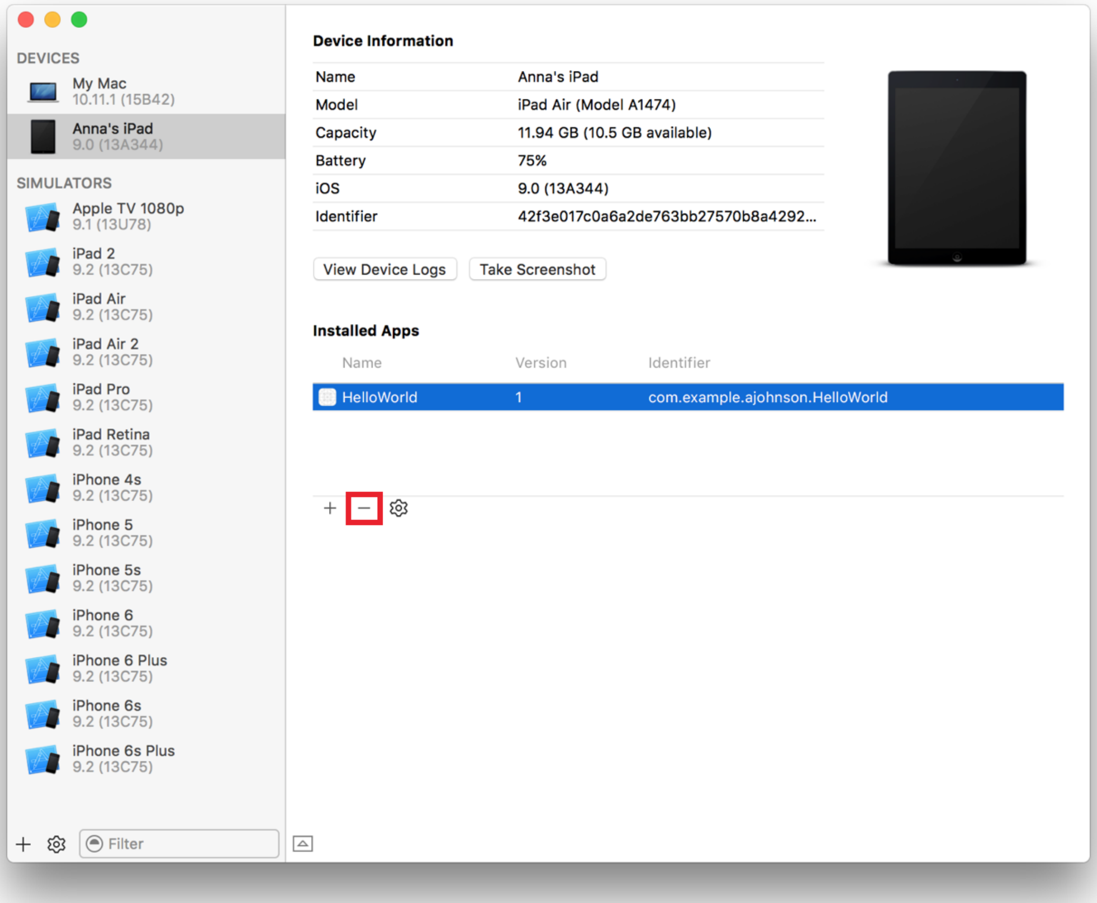
   In the project editor, the device disappears from the Scheme menu.

Removing an App from a Device (iOS, tvOS, watchOS)

When you launch an app on a device, Xcode installs the app on the device. Later, you can remove the app from the device using the Devices window in Xcode.

To remove an app from a device

1. Connect the device to your Mac.
2. Choose Window > Devices, and select the device under Devices.
3. In the Installed Apps section, select the app from the table and click the Delete button (–) in the lower-left corner of the table.



Device Information

Name	Anna's iPad
Model	iPad Air (Model A1474)
Capacity	11.94 GB (10.5 GB available)
Battery	75%
iOS	9.0 (13A344)
Identifier	42f3e017c0a6a2de763bb27570b8a4292...

Installed Apps

Name	Version	Identifier
HelloWorld	1	com.example.ajohnson.HelloWorld

4. In the dialog that appears, click Delete.

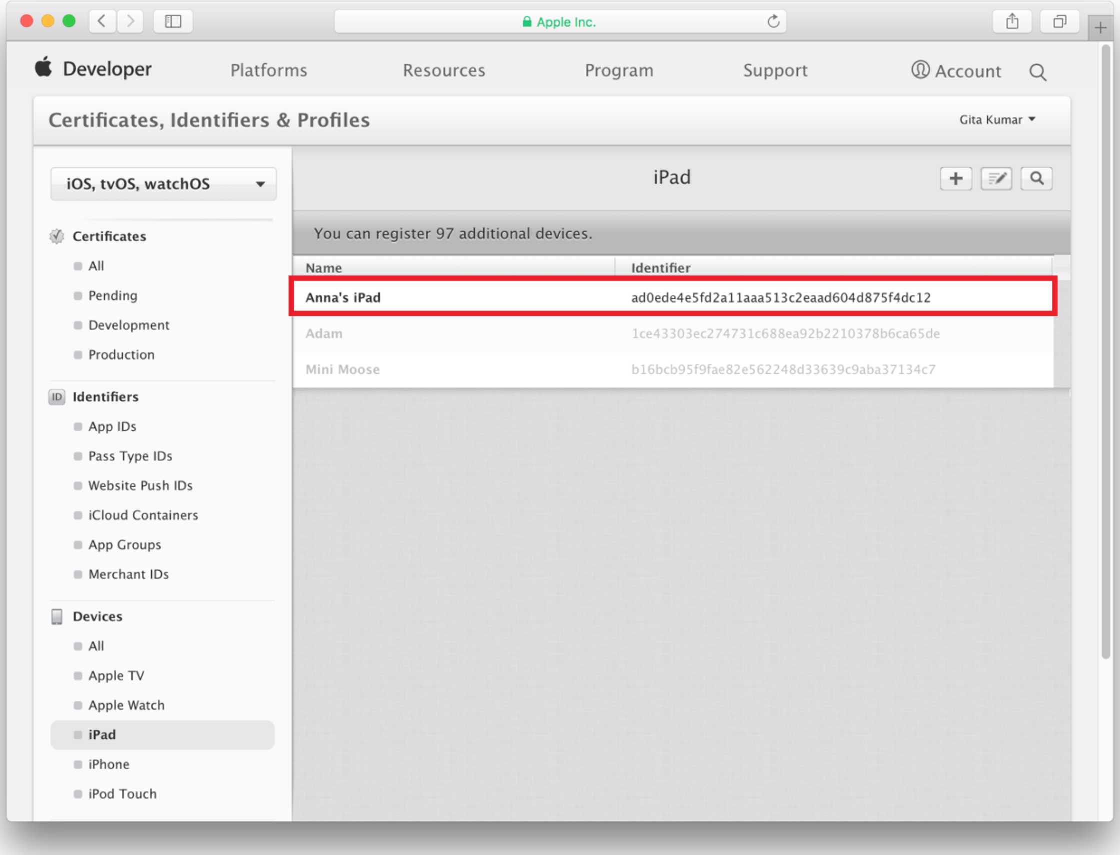
Verifying Your Steps

To learn more about how Xcode provisions your app, examine the team provisioning profile in your developer account. You can verify that the device or Mac was registered and added to the team provisioning profile.

To verify that your device is registered

1. Go to Certificates, Identifiers & Profiles and for Mac apps, choose OS X from the pop-up menu on the left.
2. Under Devices, select All or a device family.

   The device you registered should appear enabled in the list. Enabled devices appear in black text, and disabled devices appear in gray text.



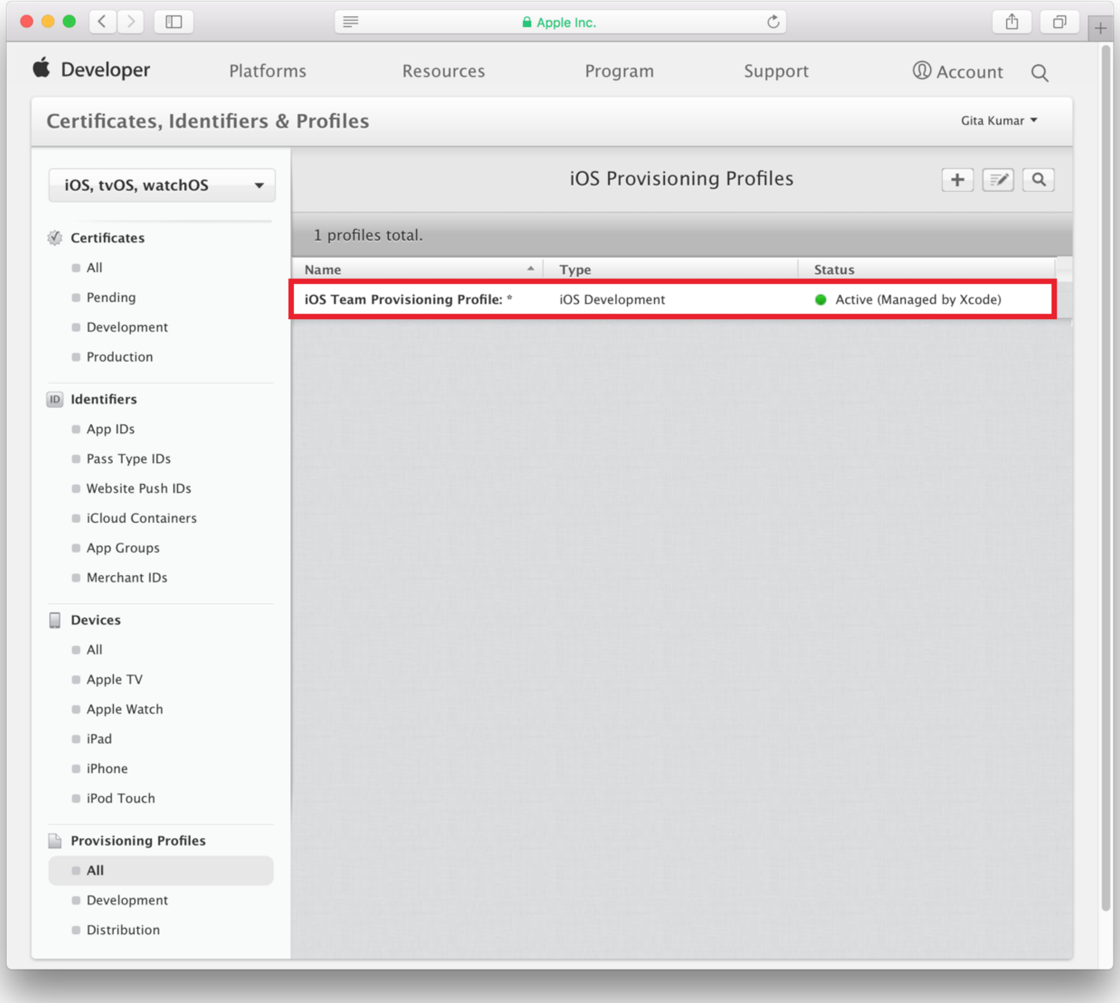
You can register 97 additional devices.

Name	Identifier
Anna's iPad	ad0ede4e5fd2a11aaa513c2eaad604d875f4dc12
Adam	1ce43303ec274731c688ea92b2210378b6ca65de
Mini Moose	b16bcb95f9fae82e562248d33639c9aba37134c7

To verify that your device is added to the team provisioning profile

1. Go to Certificates, Identifiers & Profiles and for Mac apps, choose OS X from the pop-up menu on the left.
2. Under the Provisioning Profiles section, select All.

   The team provisioning profile appears. For an iOS app, the team provisioning profile starts with the text “iOS Team Provisioning Profile.”



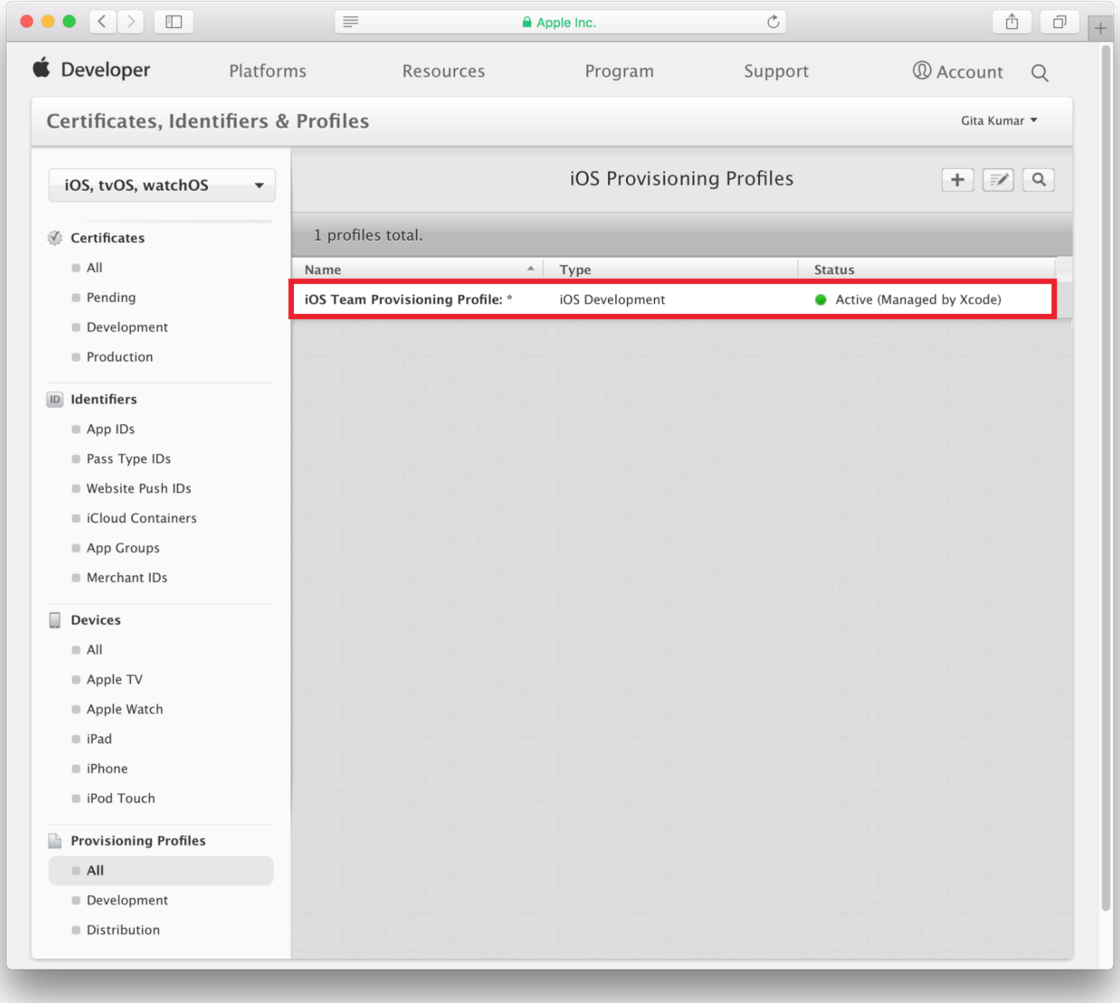
1 profiles total.

Name	Type	Status
iOS Team Provisioning Profile: *	iOS Development	● Active (Managed by Xcode)

> **Note:** You may have multiple team provisioning profiles depending on the capabilities you add and the number of apps.

3. Click the team provisioning profile to view its details.

   The team provisioning profile contains an App ID—for example, either a wildcard App ID (XC Wildcard) or an explicit App ID that matches the bundle ID. The screenshot below shows an iOS provisioning profile.



1 profiles total.

Name	Type	Status
iOS Team Provisioning Profile: *	iOS Development	● Active (Managed by Xcode)

Listed beneath the App ID is the number of development certificates and devices contained in the provisioning profile. The values should equal the total number of development certificates and enabled devices contained in your account. If you're an individual, you should have only one development certificate.

Troubleshooting

There are several reasons why your app may not run on a device.

- If the device appears ineligible in the Scheme toolbar menu, fix the issue before continuing.

  Under Ineligible Devices, hover the mouse over the device to read the reason why it's ineligible.
- If a warning or error message appears below the Team pop-up menu in the General pane of the project editor, follow the steps in Creating the Team Provisioning Profile to fix the issue.
- Read Adding Capabilities to fix similar provisioning errors in the Capabilities pane.

Assuming that you followed and verified the steps in this chapter, check the Xcode project settings. The configuration of your project may not match the configuration of your device.

- To verify that the Deployment Target is less than or equal to the OS software version installed on your device, read Setting the Deployment Target.

Team Provisioning Profiles in Depth

A *team provisioning profile* is a development provisioning profile that Xcode manages for you. A development provisioning profile allows your app to launch on devices and use certain app services during development. For an individual, a team provisioning profile allows all apps signed by you to run on all of your registered devices. For an organization, a team provisioning profile allows *any* app developed by a team to be signed by *any* team member and installed on *any* team device.

The team provisioning profile contains:

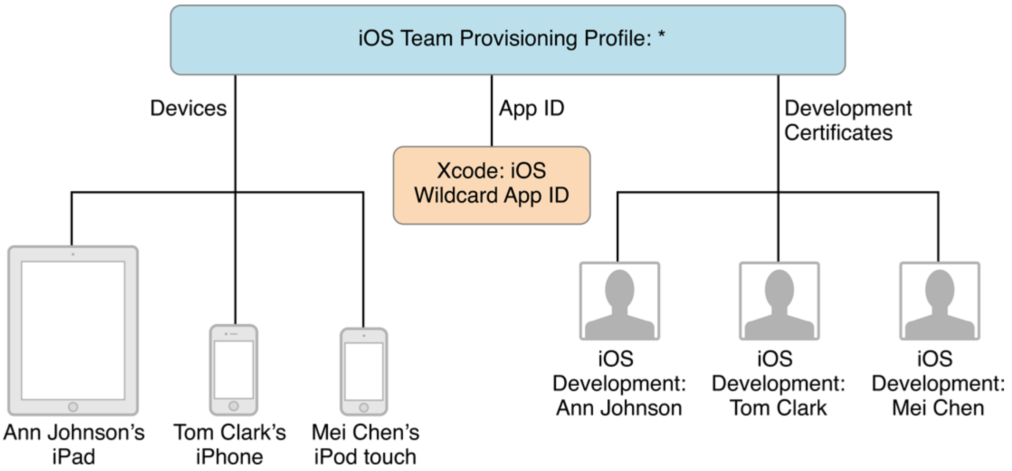
- A wildcard App ID that matches all your team's apps or an explicit App ID that matches a single app
- All devices associated with the team
- All development certificates associated with the team

Xcode creates App IDs and team provisioning profiles as needed depending on the configuration and capabilities of your app. Xcode adds all of the devices and development certificates from all team members to the team provisioning profile. Thereafter, Xcode updates the team provisioning profile whenever you register a device, create a development certificate, or modify the App ID. (Changes you make to your team assets using your developer account don't automatically update team provisioning profiles.)

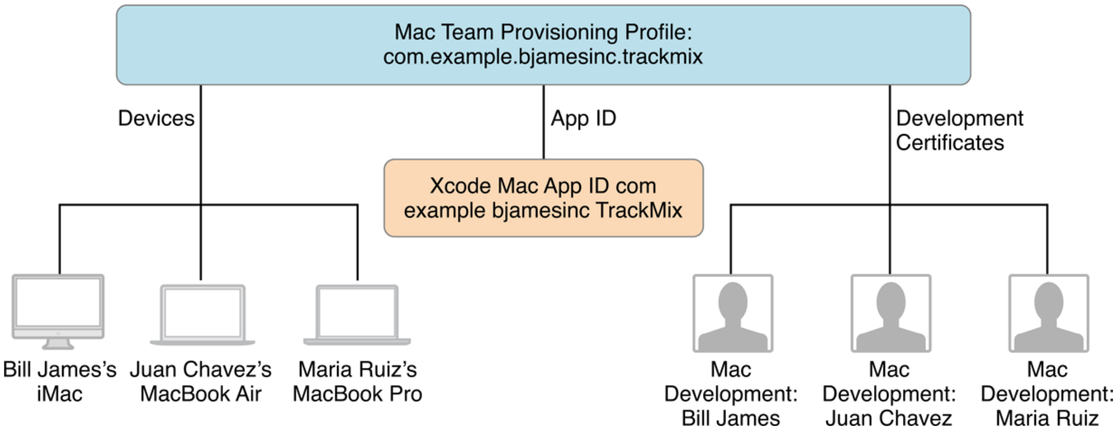
If your app can use a wildcard App ID during development, Xcode creates a team provisioning profile containing a wildcard App ID that it also creates. The team provisioning profile name is `[platform] Team Provisioning Profile: [App ID]` where `platform` is either iOS, Mac, or tvOS, and an asterisk (`*`) indicates a wildcard App ID. For example, the profile name for an iOS app is `iOS Team Provisioning Profile:*`.

You can use a wildcard App ID only with a subset of app services. If you add a capability that requires an explicit App ID—for example, iCloud, Game Center, or In-App Purchase—Xcode creates an explicit App ID and a corresponding team provisioning profile. For example, if the bundle ID for an iOS app is `com.example.ajohnson.HelloWorld`, the corresponding team provisioning profiles name is `iOS Team Provisioning Profile: com.example.ajohnson.HelloWorld`. Because an explicit App ID exactly matches the project's bundle ID, you can register only one explicit App ID per bundle ID. Therefore, if one already exists in your developer account, Xcode uses it in the team provisioning profile instead of creating one for you.

This diagram shows an iOS Team Provisioning Profile using a wildcard App ID for an organization with three team members.


iOS Team Provisioning Profile: *
Devices
App ID
Development Certificates
Xcode: iOS Wildcard App ID
Ann Johnson's iPad
Tom Clark's iPhone
Mei Chen's iPod touch
iOS Development: Ann Johnson
iOS Development: Tom Clark
iOS Development: Mei Chen

This diagram shows a Mac Team Provisioning Profile using an explicit App ID for an organization developing a Mac app.


Mac Team Provisioning Profile: com.example.bjamesinc.trackmix
Devices
App ID
Development Certificates
Xcode Mac App ID com example bjamesinc TrackMix
Bill James's iMac
Juan Chavez's MacBook Air
Maria Ruiz's MacBook Pro
Mac Development: Bill James
Mac Development: Juan Chavez
Mac Development: Maria Ruiz

Recap

In this chapter, you learned how to use the team provisioning profile, which Xcode creates and manages for you, to launch your app on devices. You learned what information Xcode needs to create the team provisioning profile, how to verify that Xcode registered your devices, and how to troubleshoot if Xcode displays errors or warnings.