

# Maintaining Identifiers, Devices, and Profiles

You use your developer account to manage the identifiers, devices, and profiles used to code sign your app, enable it to launch on devices, and use certain app services. Xcode creates and manages these assets for you during development. When you're ready to distribute your app for testing, you use your developer account to manage some of these assets yourself. For example, you can create custom development provisioning profiles, and bulk register devices used for testing. This chapter covers miscellaneous tasks you perform to maintain identifiers, devices, and profiles.

For how to use Website Push IDs, read *Notification Programming Guide for Websites*.

**Important:** A team agent can perform all the tasks in this chapter. If you are a team admin or team member, see Team Privileges for the tasks you can perform.

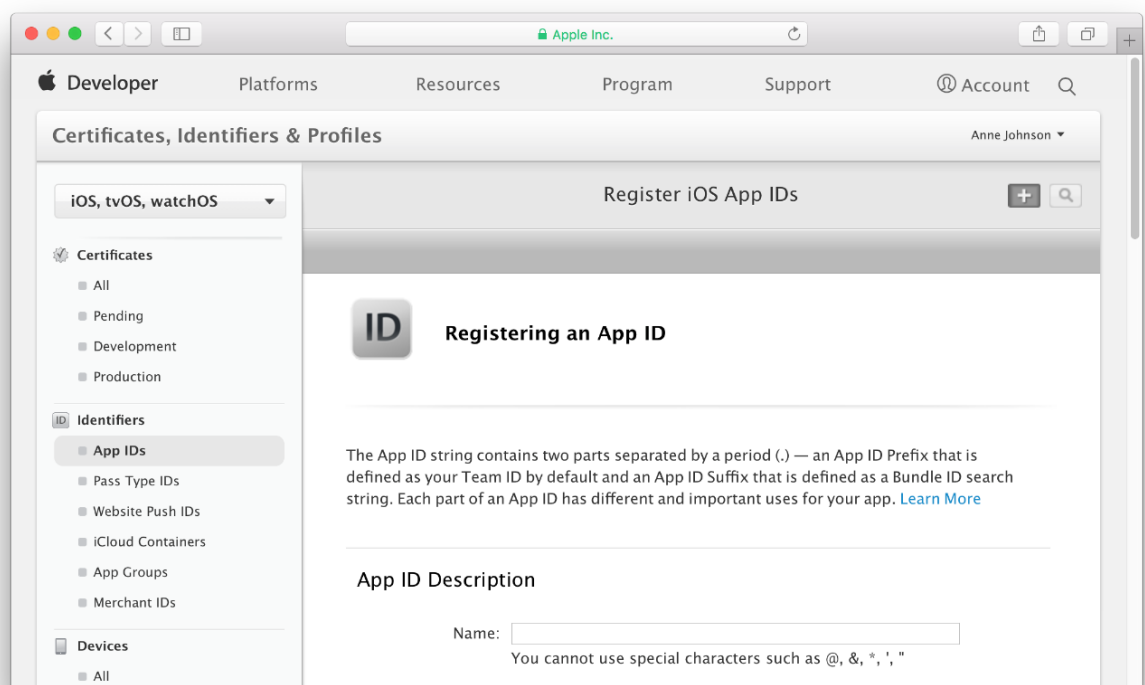
## Registering App IDs

You can create a wildcard [App ID](#) that matches one or more apps or an explicit App ID that exactly matches your bundle ID. The app services enabled for an App ID serve as a whitelist of the services one or more apps may use. What services an app actually uses is configured in the Xcode project. You can enable app services when you create an App ID or modify these settings later. Game Center and In-App Purchase are enabled by default for an explicit App ID.

In most cases, the wildcard and explicit App ID that Xcode creates for you when you add capabilities to your app, as described in [Adding Capabilities](#), are sufficient to develop, test, and distribute your app. There can be only one explicit App ID per app, so if one already exists, Xcode uses it to create team provisioning profiles.

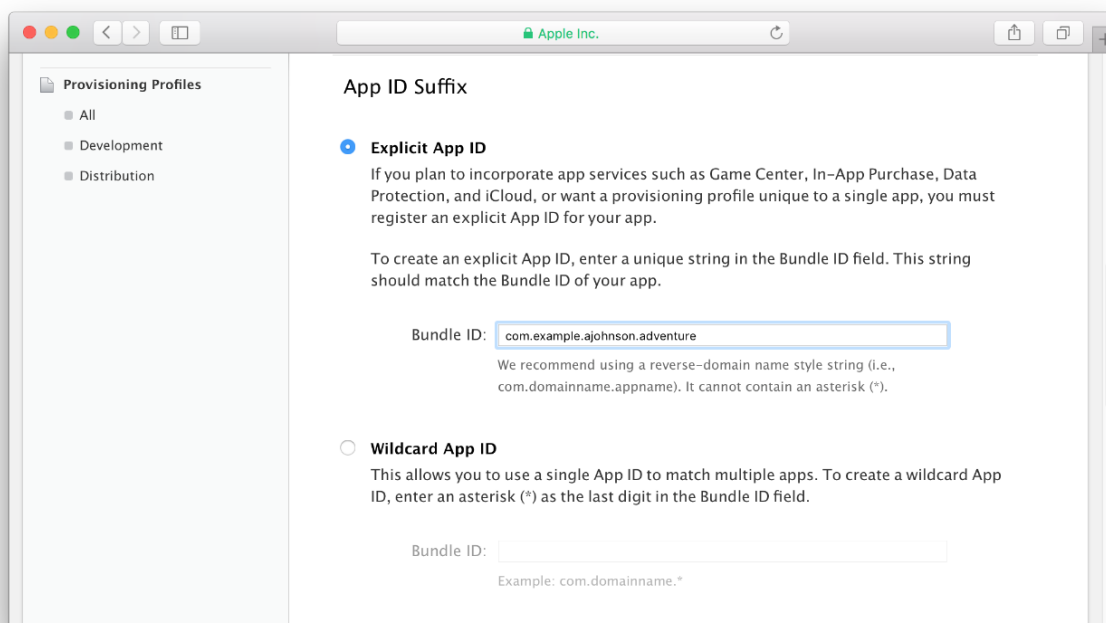
### To register an App ID

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Identifiers, select App IDs.
3. Click the Add button (+) in the upper-right corner.
4. Enter a name or description for the App ID in the Description field.



5. To create an explicit App ID, select Explicit App ID and enter the app's bundle ID in the Bundle ID field.

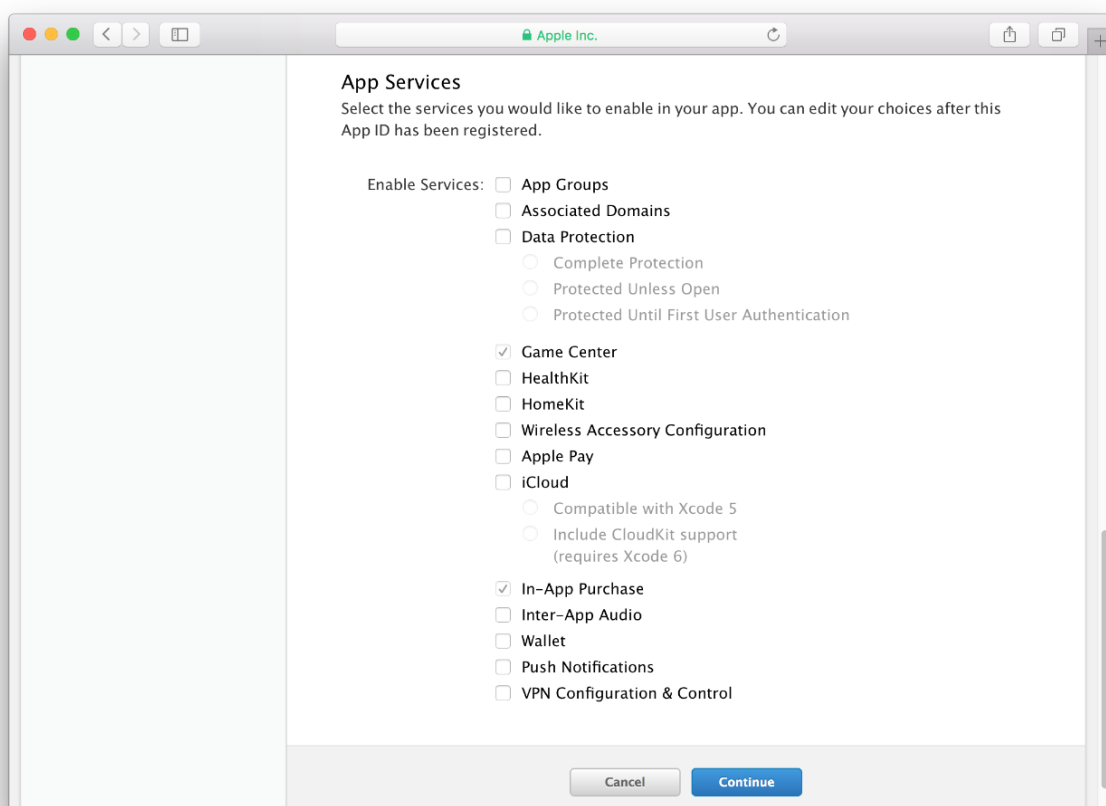
An explicit App ID exactly matches the bundle ID of an app you're building—for example, `com.example.ajohnson.adventure`. An explicit App ID can't contain an asterisk (\*).



If you entered a bundle ID in the target's Summary pane in Xcode, the App ID you enter here should match your bundle ID.

6. To create a wildcard App ID, select Wildcard App ID and enter a bundle ID suffix in the Bundle ID field.
7. Select the corresponding checkboxes to enable the app services you want to use.

A checkbox is disabled if the technology requires an explicit App ID and you're creating a wildcard App ID, or the technology is enabled by default.



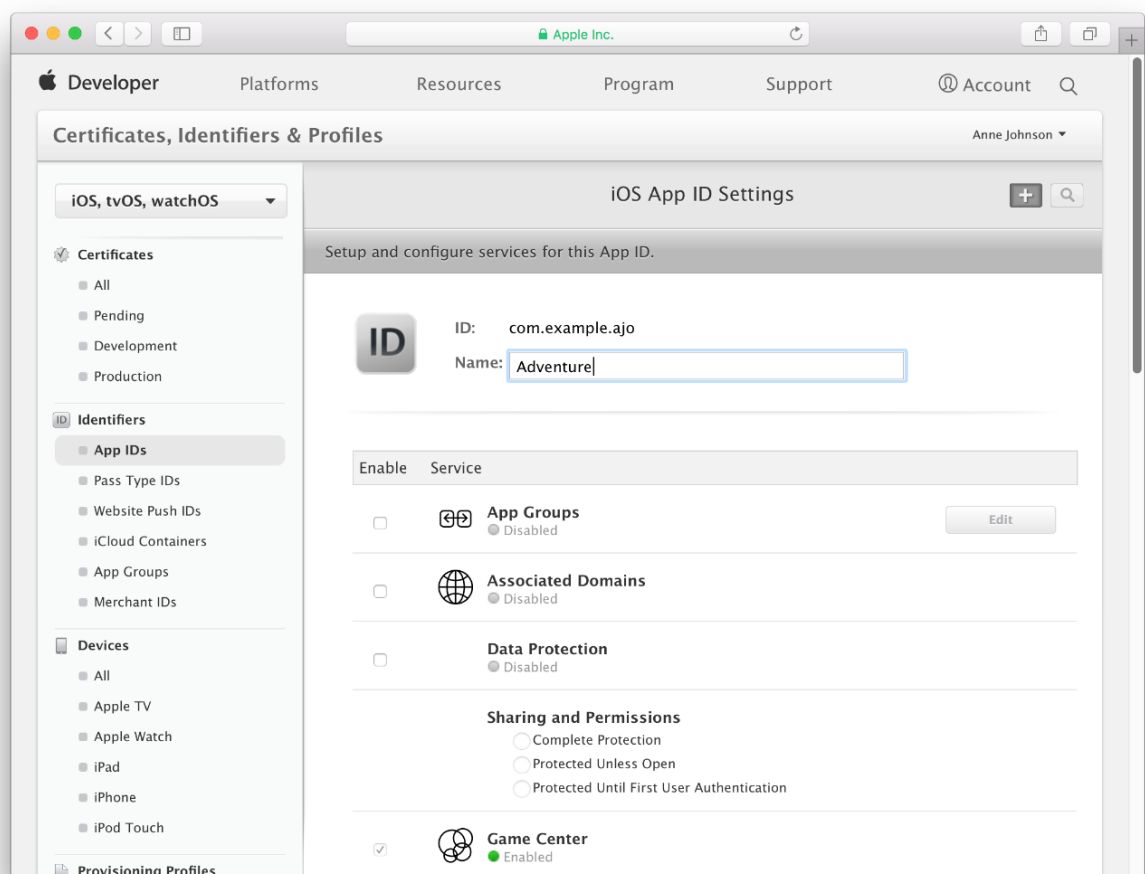
8. Click Continue.
9. Review the registration information, and click Register.
10. Click Done.

## Editing App IDs

You can edit an App ID directly using your developer account.

### To enable app services for an existing App ID

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Identifiers, select App IDs.
3. Select the App ID you want to change, and click Edit.
4. Select the corresponding checkboxes to enable the app services you want to allow.



5. If a warning dialog appears, click OK.  
Later, you'll regenerate the provisioning profiles that use the App ID.
6. Click Done.

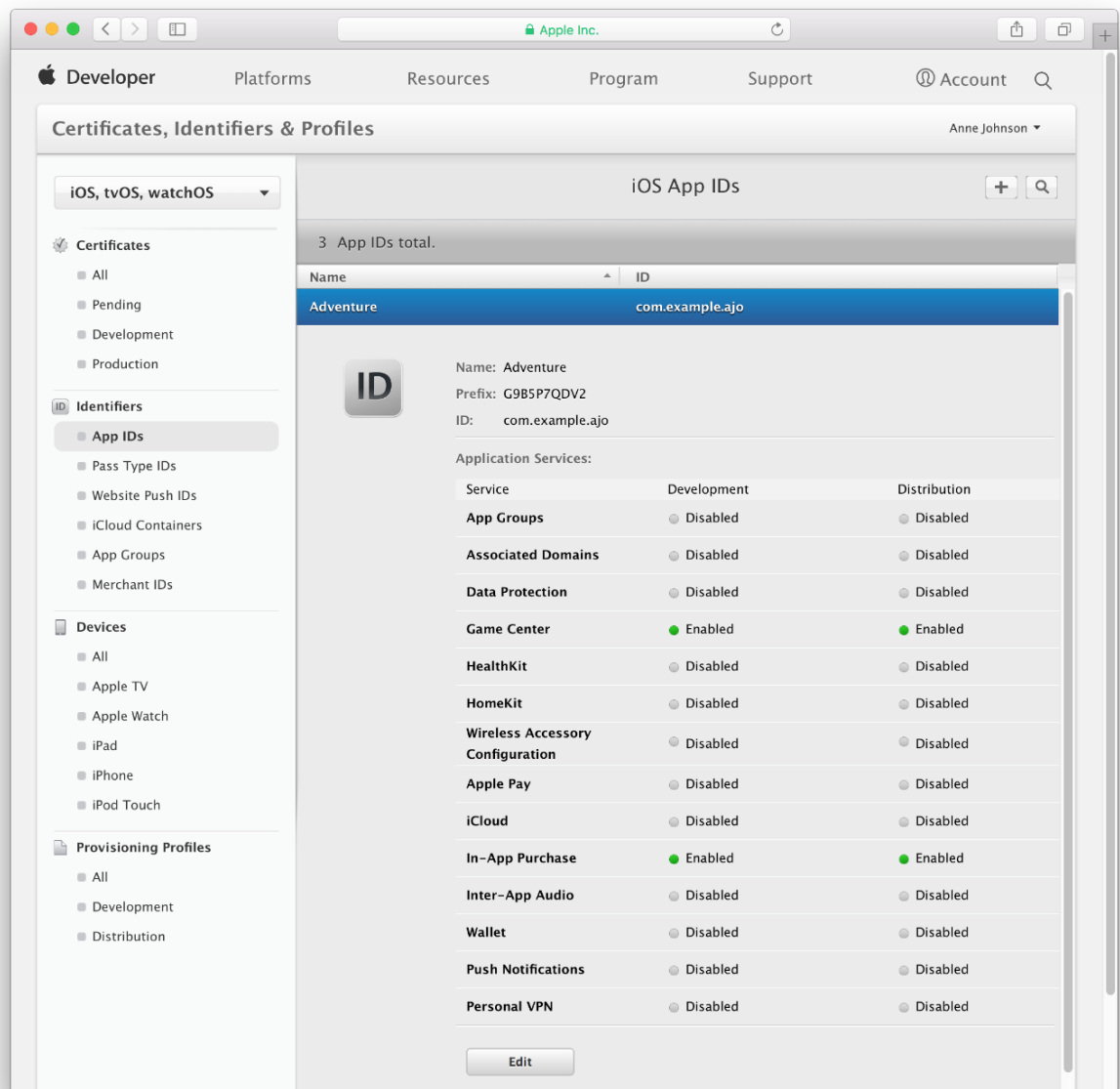
To fully enable app services that appear as configurable in your developer account, read [Adding Capabilities](#).

## Deleting App IDs

You can also remove App IDs when you no longer need them. However, you cannot delete an explicit App ID for an app you uploaded to iTunes Connect.

## To remove an App ID

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Identifiers, select App IDs.
3. Select the App ID you want to delete, and click Edit.



4. Scroll to the bottom of the page, and click Delete.
5. Read the dialog that appears, and click Delete.

Provisioning profiles that contain a deleted App ID become invalid. You can change the App ID in the provisioning profiles, as described in [Editing Provisioning Profiles in Your Developer Account](#), or delete them.

## Locating Your Team ID

Occasionally, you may need to know your Team ID. For example, if you want to receive ownership of an app from another developer, the developer needs your Team ID to initiate the app transfer in iTunes Connect. The Team ID is a unique 10-character string generated by Apple that's assigned to your team. You can find your Team ID using your developer account.

### To locate your Team ID

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Membership in the sidebar.

Your Team ID appears in the Membership Information section under the team name.

# Registering Devices Using Your Developer Account

In your developer account, you can register devices individually as needed or upload a file with information on multiple devices. Each year, you’re allowed to register a fixed number of devices. The maximum number of devices you can register is 100 device per product family per membership year. If you later disable a device, as described in Disabling and Enabling Devices Using Your Developer Account, it doesn’t decrease your count of registered devices.

## Locating Device IDs

You need the name and device ID for each device you want to register. There are several ways to obtain device IDs depending on whether you have Xcode installed.

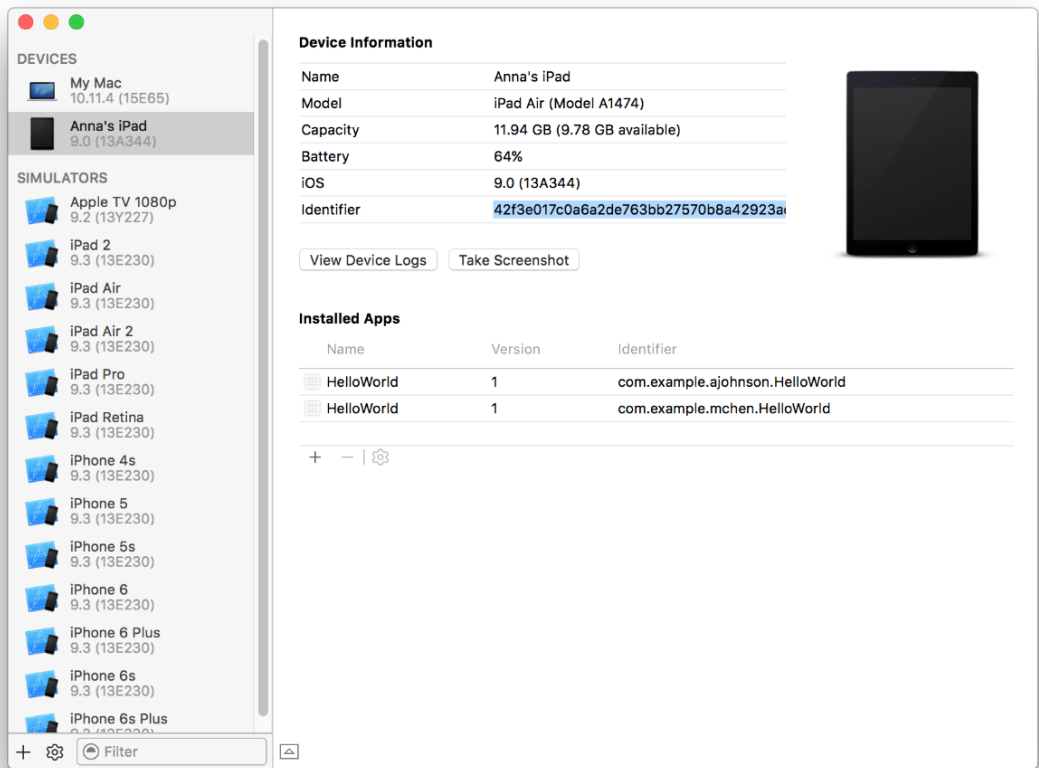
### Locating Device IDs Using Xcode

In Xcode, a team member can select a device in the Devices window to display the device ID.

#### To locate a device ID using Xcode

- 1. Choose Window > Devices.
- 2. For iOS and tvOS apps, connect a device to your Mac. For watchOS apps, connect an iPhone paired an Apple Watch.
- 3. In the Devices window under Devices, select your device (your Mac is a device too).

The device identifier appears in the Identifier field.

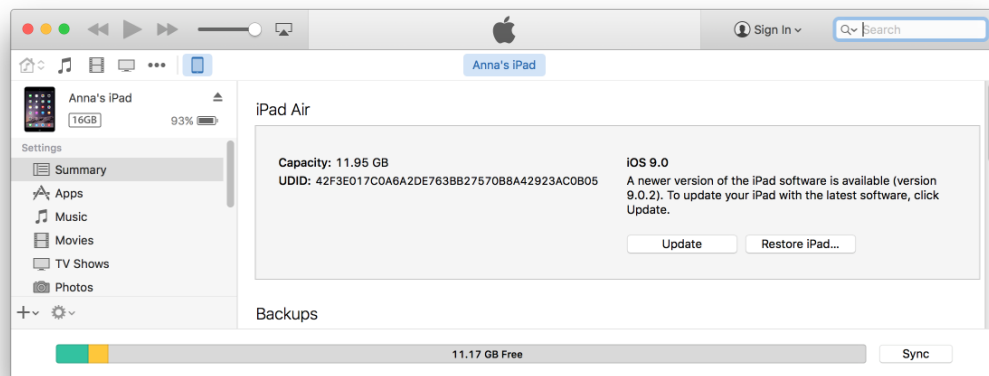


### Locating iOS Device IDs Using iTunes (iOS, tvOS)

For iOS devices, you can also obtain a device ID using iTunes. For example, testers follow these steps to get their device ID using iTunes when they don’t have Xcode installed.

### To get an iOS device ID using iTunes

1. Launch iTunes on your Mac.
2. Connect your device to your Mac.
3. In the upper-right corner, select the device.
4. In the Summary pane, click the Serial Number label (for iOS, it appears under Capacity or Phone Number).  
The label Serial Number changes to UDID and displays the device ID.



5. Copy the device ID by Control-clicking the identifier and choosing Copy from the shortcut menu.
6. Paste the device ID in a document or an email message.

### Locating Device IDs Using System Information (iOS, tvOS, Mac)

You can get a device ID using the System Information app. For example, use this method if you want to register a Mac for testing that isn't used for development.

#### To locate your Mac device ID using System Information

1. Open the System Information app located in the `/Applications/Utilities` folder.
2. Select Hardware in the left column.

The device ID, or hardware UUID, appears near the bottom of the Hardware Overview pane and is in the format `xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`.

Also, use this method if you want to collect device IDs for iOS or Apple TV devices without using Xcode.

#### To locate a connected device ID using System Information

1. Connect the device to your Mac.
2. Open the System Information app located in the `/Applications/Utilities` folder.
3. Select USB under Hardware in the left column.
4. On the right, select the connected device under USB Device Tree.

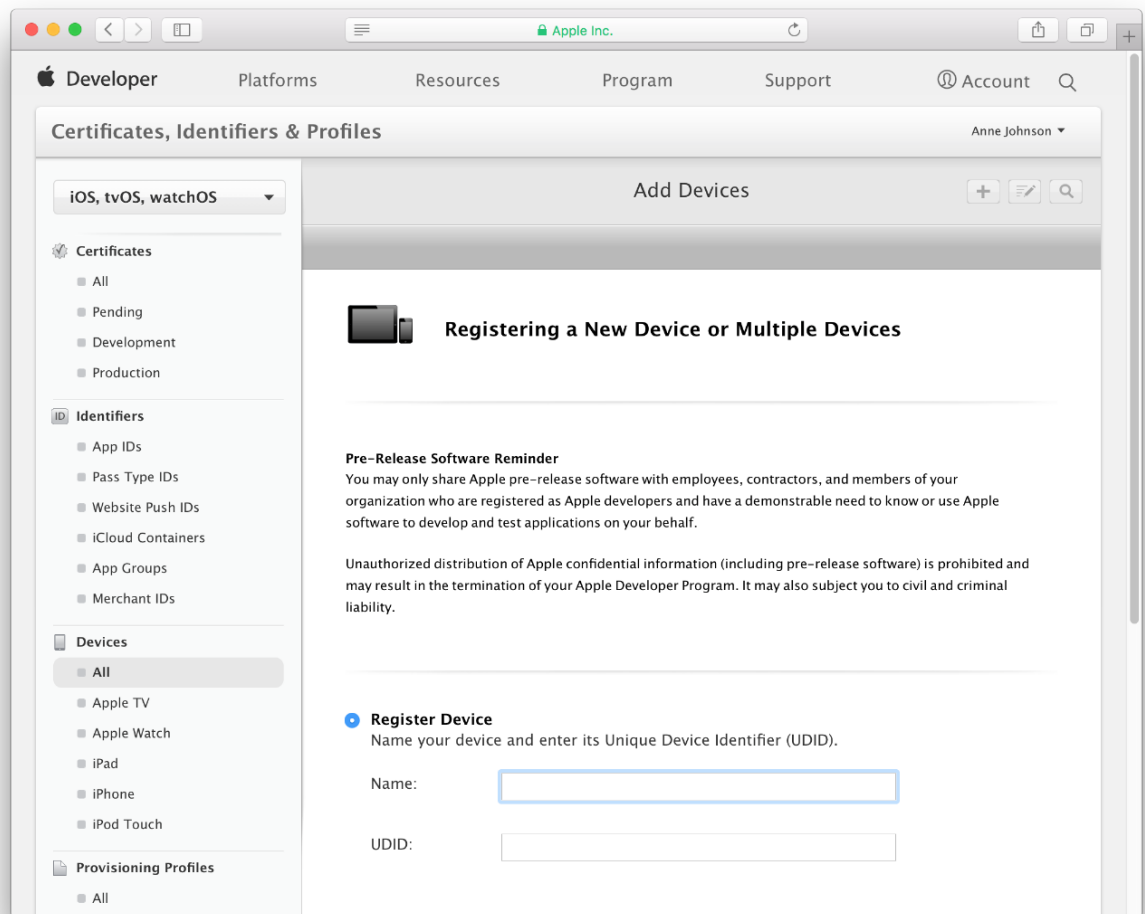
The device ID, or "Serial Number", appears below.

## Registering Individual Devices

To register a device using your developer account, you need to have the device name and device ID.

#### To register a single device

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Devices, select All.
3. Click the Add button (+) in the upper-right corner.
4. Select Register Device.
5. Enter a device name and the device ID (UDID).



6. Scroll to the bottom of the page, and click Continue.

7. Review the registration information, and click Register.

## Registering Multiple Devices

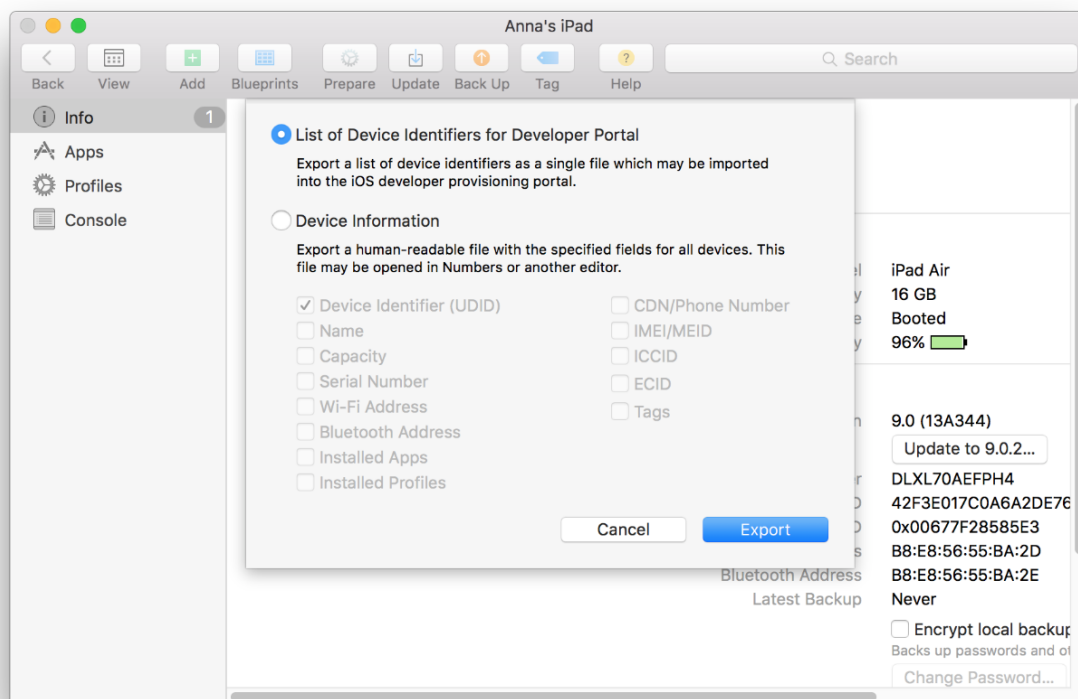
If you have many test devices, you can create a file containing the device names and device IDs and upload the entire file to your developer account. Your developer account supports these two file formats: a property list file with a `.deviceids` file extension and a plain text file. The file format you choose depends on whether you have access to the devices you want to register.

## Creating a Property List Devices File Using Apple Configurator

If you have access to the testing devices, you can use Apple Configurator to create a property list file that contains all your device names and device IDs. Download Apple Configurator from the Mac App Store.

### To create a property list devices file

1. In Apple Configurator, choose Actions > Export > Info.
2. Select “List of Device Identifiers for Developer Portal” and click Export.



3. Choose a file name and location, and click Save.

The file has a `.deviceids` extension.

## Creating a Plain Text Devices File

If you don't have access to the testing devices, you can create a `.txt` file containing the device names and device IDs you collect using another method. In this case, create a tab-delimited file with one device ID and one device name in each row. You can use the first row for your headers, because that row is ignored when parsed.

## Uploading the Devices File

In your developer account, the steps to upload the devices file are the same for both file formats.

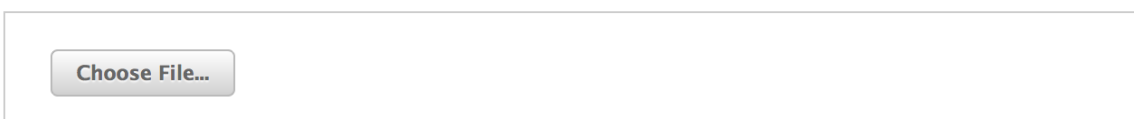
### To register multiple devices

1. In Certificates, Identifiers & Profiles, and for Mac apps, choose OS X from the pop-up menu on the left.
2. Under Devices, select All.
3. Click the Add button (+) in the upper-right corner.
4. Select Register Multiple Devices.
5. Click Choose File.

#### • Register Multiple Devices

Upload a file containing the devices you wish to register. Please note that a maximum of 100 devices can be included in your file and it may take a few minutes to process.

[Download sample files](#)



6. Select the file you want to upload, and click Choose.

Select either the `.deviceids` or `.txt` file you created earlier.



7. Click Continue.
8. Review the registration information, and click Submit.

## Installing Prereleases on Your iOS or Apple TV Device

Before you begin, download the prerelease you want to install on your device from Apple Developer.

### To install a prerelease on your development device

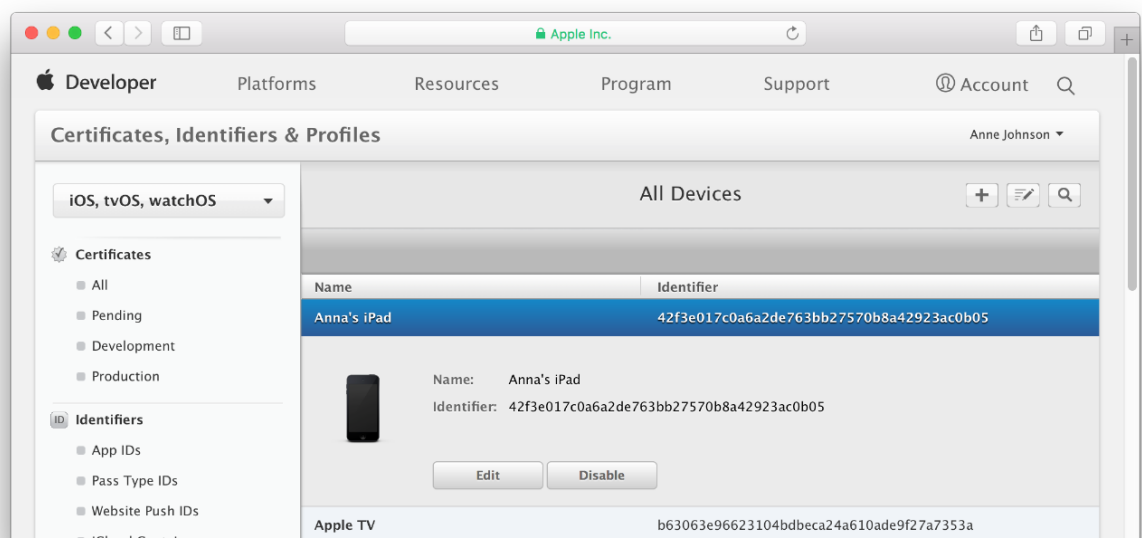
1. Connect the device to your Mac.
2. In the iTunes, select the device in the upper-left corner.  
For Apple TV devices, if the device doesn't appear in iTunes, put it in recovery mode.
3. In the Summary pane, Option-click the Restore iPhone/iPad/iPod or Restore Apple TV button.
4. Select the prerelease software restore image, and click Open to begin installation.
5. When installation is complete, activate the device and restore its contents using iTunes.

## Disabling and Enabling Devices Using Your Developer Account

You can disable and enable a device, but you can't delete it from your developer account. You can disable a device you no longer use for development or testing. However, doing so invalidates provisioning profiles that contain the device and doesn't increase your total count of devices for the year. You can also enable a device that you previously disabled.

### To disable or enable a device

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Devices, select All.
3. Select the device you want to disable or enable.
4. Click either Enable or Disable.



5. In the dialog that appears, click either Enable or Disable again.

To regenerate invalid team provisioning profiles after disabling a device, read [Downloading Provisioning Profiles in Xcode](#). To remove a disabled device from other provisioning profiles you manage, read [Editing Provisioning Profiles in Your Developer Account](#).

You can also enable a device using Xcode. For iOS apps, Xcode automatically registers a connected device that's chosen from the Scheme toolbar menu in the project editor. For Mac apps, Xcode automatically registers the Mac that's running Xcode. If the device or Mac was previously registered but is disabled, Xcode enables it.

You may still run your app on a disabled device if it is in an older version of a provisioning profile that is installed on the device. To remove old versions of your provisioning profiles, read [Verifying and Removing Provisioning Profiles on Devices](#).

## Managing Apps on Devices

In the Devices window, you can manage instances of your app installed on an iOS or tvOS device. Before you begin, connect the device to your Mac, choose **Window > Devices**, and select the device under **Devices**. For iOS and tvOS apps, connect a device to your Mac. For watchOS apps, connect an iPhone paired to an Apple Watch.

### Removing Apps from Devices

To remove all data files associated with an app on a device, delete the app from the device. You might do this to test code that runs only when your app first launches. In Xcode, you can remove only the apps that you created. You can't remove any of the system-installed or third-party apps from the device using Xcode.

#### To remove an app from a device

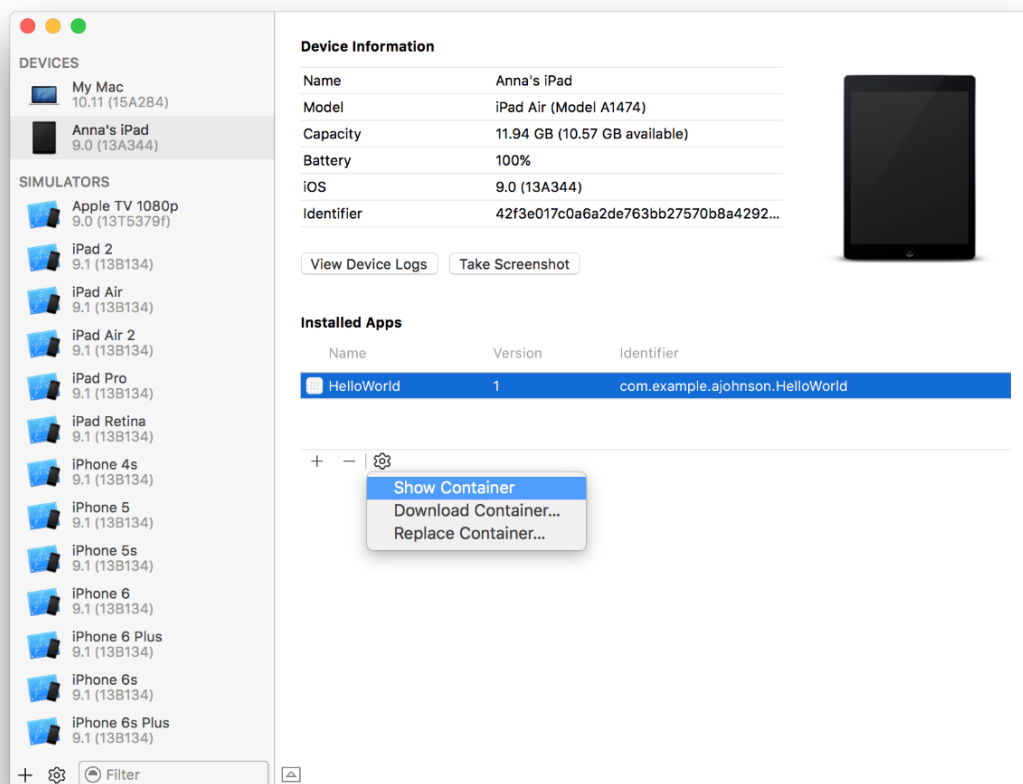
1. In the Installed Apps table, select an app and click the Delete button (–) below the table.
2. In the dialog that appears, click the Delete button.

### Viewing, Downloading, and Replacing App Containers on Devices

You can view the file structure of an app container on a device directly in Xcode but not the contents of the files.

#### To view the contents of an app container

1. Under Installed Apps, select the app from the list.
2. From the Action menu (the gear icon to the right of the Delete button), choose Show Container.



### 3. Click Done.

To examine or modify the files in an app container, download it to your file system. Analyzing the contents of an app's container may provide important information to help you diagnose problems or tune the app.

#### To download the contents of an app container

1. Under Installed Apps, select the app from the list.
2. From the Action pop-up menu, choose Download Container.
3. Enter a filename in the Save As text field, and click Save.

To test the app in a particular state, replace the app container on a device.

#### To replace an app container on a device

1. Under Installed Apps, select the app from the list.
2. From the Action menu, choose Replace Container.
3. Select a container from the File Picker and click Open.

## Creating Provisioning Profiles Using Your Developer Account

You can create both development and distribution provisioning profiles yourself using your developer account.

Because Xcode creates and manages team provisioning profiles for you, you only create a development provisioning profile if you want to restrict development of an app to specific team members and devices. Follow the steps in Creating Development Provisioning Profiles if you want to create your own development provisioning profile.

Xcode creates an iOS ad hoc provisioning profile for you when you export an app archive and select the ad hoc deployment option, as described in Exporting Your App for Testing Outside the Store. To create an ad hoc provisioning profile directly in your developer account, read Creating Ad Hoc Provisioning Profiles (iOS, tvOS, watchOS).

Similarly, Xcode creates a store provisioning profile for you when you submit your app to the store. To create a store provisioning profile directly in your developer account, read Creating Store Provisioning Profiles.

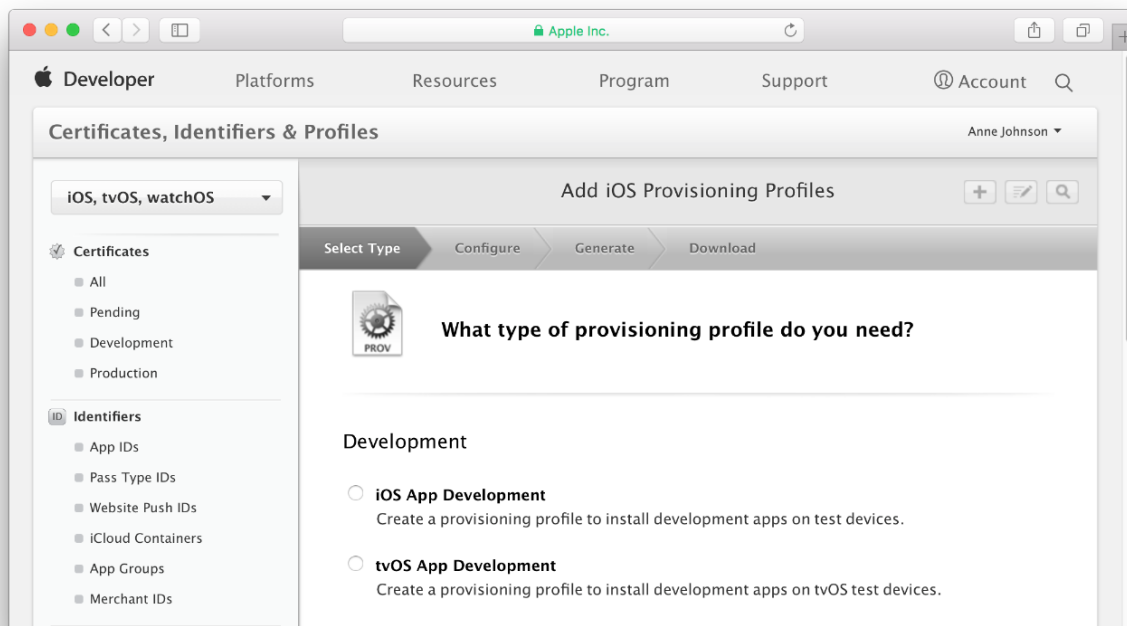
After creating provisioning profiles, download provisioning profiles in Xcode, as described in Downloading Provisioning Profiles in Xcode. The provisioning profiles should appear in the Provisioning Profiles table in the view details dialog in Accounts preferences.

## Creating Development Provisioning Profiles

Before creating a development provisioning profile, verify that you have an [App ID](#), one or more development certificates, and one or more devices. If you want to register your own App ID, read Registering App IDs. (You can also use one of the App IDs that Xcode manages for you.) If you need to create your development certificate, read Creating Signing Identities. If you need to register devices, read Registering Devices Using Your Developer Account.

#### To create a development provisioning profile

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All.
3. Click the Add button (+) in the upper-right corner.
4. Under Development, select the type of provisioning profile you want to create and click Continue.
  - For iOS and watchOS apps, select iOS App Development.
  - For tvOS apps, select tvOS App Development.
  - For Mac apps, select Mac App Development.



5. Select the App ID you want to use for development, and click Continue.
6. Select one or more development certificates, and click Continue.
7. Select one or more devices, and click Continue.
8. Enter a profile name, and click Generate.
9. Click Done.

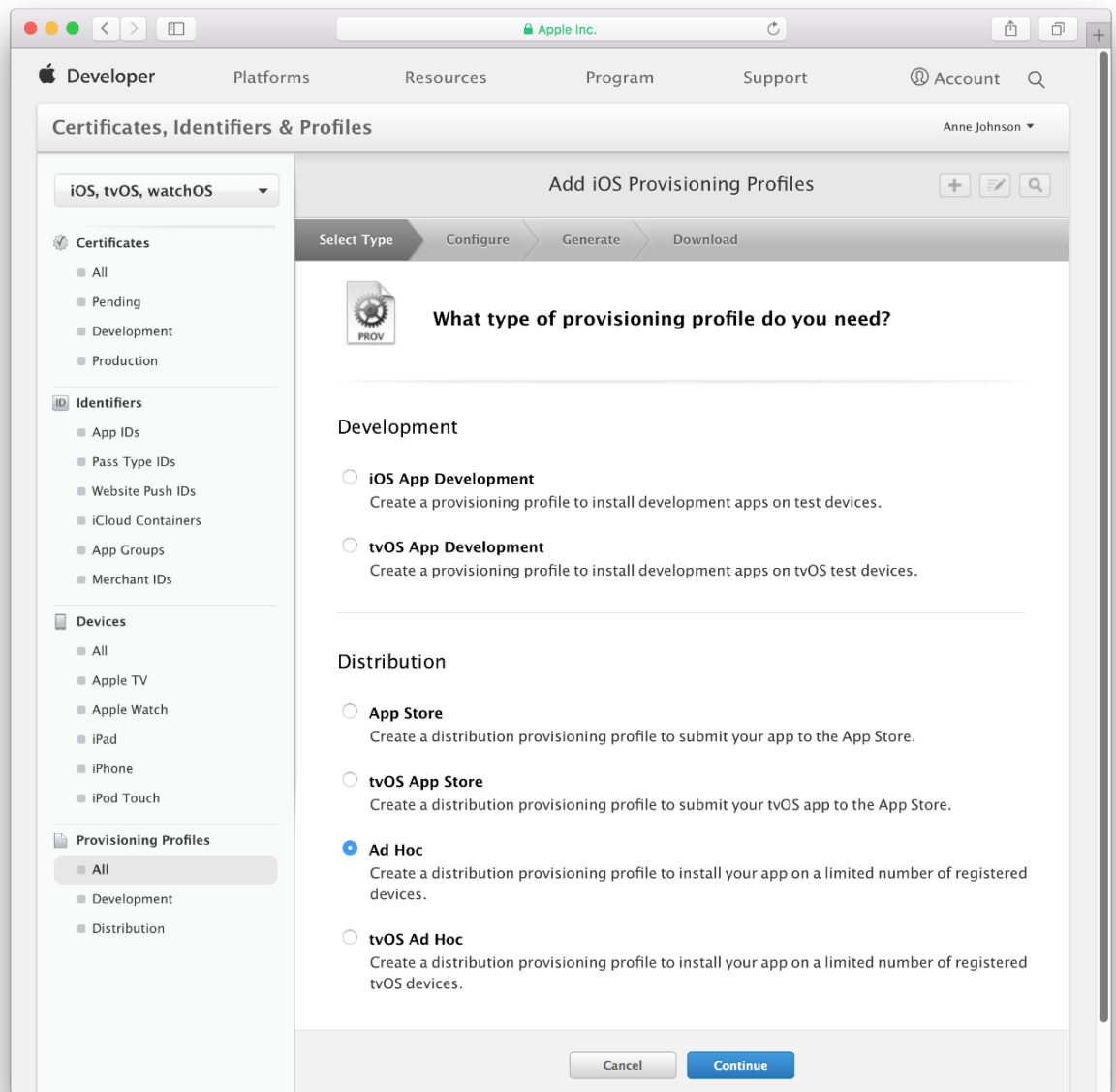
## Creating Ad Hoc Provisioning Profiles (iOS, tvOS, watchOS)

An ad hoc provisioning profile allows testers to run your app on their device without needing Xcode. To create an ad hoc provisioning profile, you select an [App ID](#), a single distribution certificate, and multiple test devices.

**Note:** Alternatively, you can create an ad hoc provisioning profile in Xcode by exporting your app, described in [Exporting Your App for Testing \(iOS, tvOS, watchOS\)](#).

### To create an ad hoc provisioning profile

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All.
3. Click the Add button (+) in the upper-right corner.
4. Select Ad Hoc or tvOS Ad Hoc as the distribution method, and click Continue.



5. Choose the App ID you used for development, which matches your bundle ID, from the App ID pop-up menu, and click Continue.

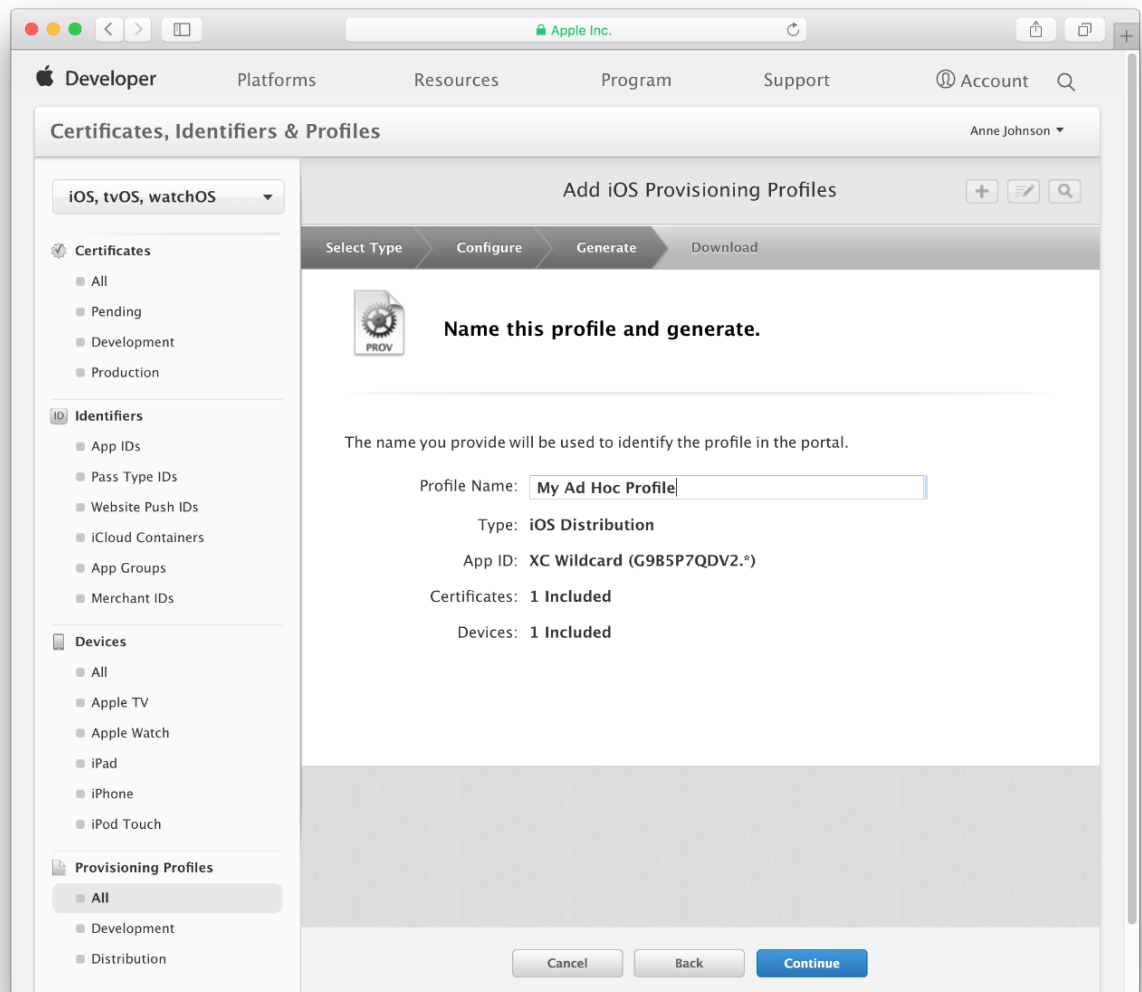
If you used a team provisioning profile during development and the menu contains only the XC Wildcard, select it. If the menu contains another Xcode-managed explicit App ID (it begins with “XC” followed by the bundle ID), select that App ID. If you created your own App ID, select that one.

6. Select the distribution certificate you want to use, and click Continue.

If you don’t have a distribution certificate, create one using Xcode, as described in [Creating Signing Identities](#), before continuing.

7. Select the devices you want to use for testing, and click Continue.

8. Enter a profile name, and click Continue.



Wait while your developer account generates the provisioning profile.

9. At the bottom of the page, Click Done.

In Xcode, download the ad hoc provisioning profile, as described in [Downloading Provisioning Profiles in Xcode](#). The ad hoc provisioning profile should now appear in the Provisioning Profiles table in the view details dialog in Accounts preferences.

## Creating Store Provisioning Profiles

Before uploading your app to the store, you provision it using a store provisioning profile. (For Mac apps that don't enable any app services, you can code sign your app using just a distribution certificate.) You don't select any devices to create a store provisioning profile.

**Note:** Alternatively, you can create a store provisioning profile in Xcode by exporting or uploading your app, described in [Exporting Your App for Testing \(iOS, tvOS, watchOS\)](#), [Exporting Your App for Testing \(Mac\)](#), and [Uploading Your App to iTunes Connect](#).

### To create a store provisioning profile

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All.
3. Click the Add button (+) in the upper-right corner.
4. Select the distribution method, and click Continue.
  - For iOS and watchOS apps, select App Store.
  - For tvOS apps, select tvOS App Store.
  - For Mac apps, select Mac App Store.

5. Choose the App ID you used for development (the App ID that matches your bundle ID) from the App ID pop-up menu, and click Continue.

If you used a team provisioning profile during development and the menu contains only the Xcode Wildcard App ID, select it. If the menu contains an Xcode-managed explicit App ID (it begins with "Xcode" and contains your bundle ID), select that App ID. If you created your own App ID, select that one.

6. Select your distribution certificate, and click Continue.

A store provisioning profile contains a single distribution certificate.

7. Enter a profile name, and click Continue.

Wait while your developer account generates the provisioning profile.

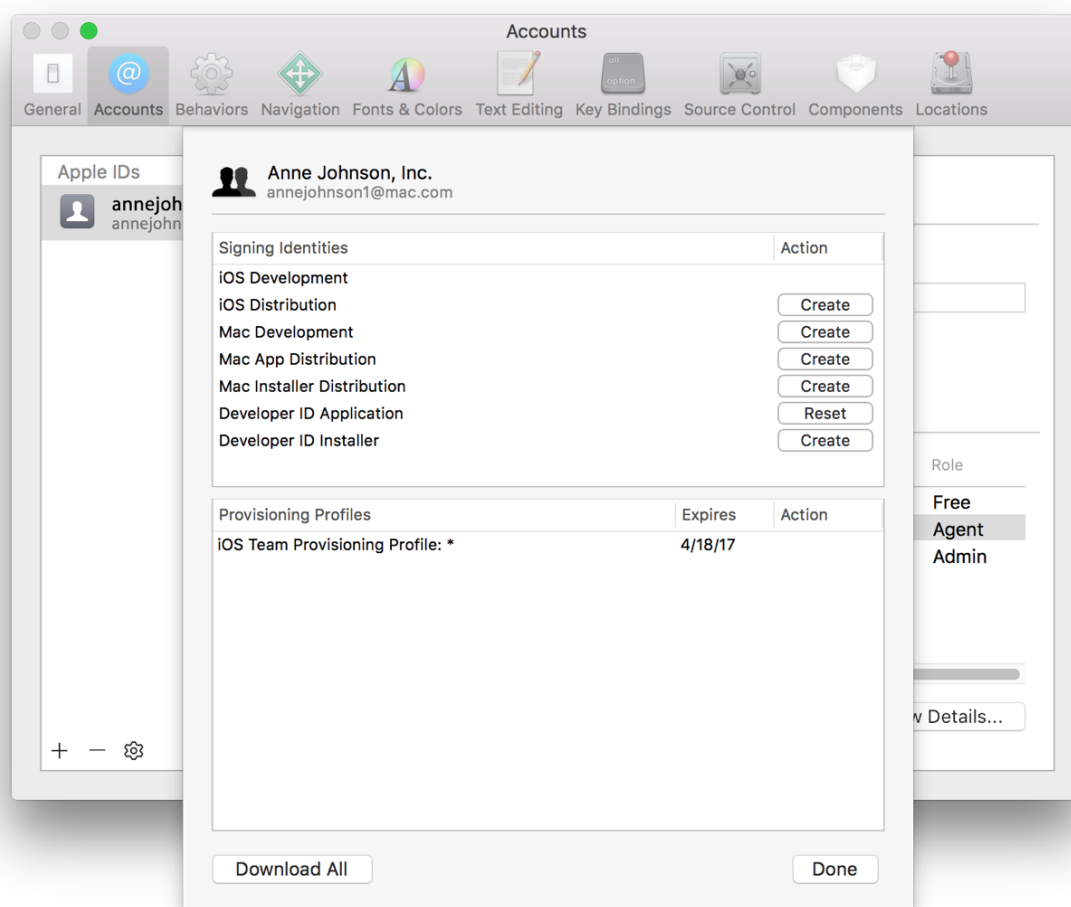
8. At the bottom of the page, Click Done.

## Downloading Provisioning Profiles in Xcode

Changes you make in your developer account aren't automatically reflected in Xcode. For example, if you create or edit a provisioning profile in your developer account, download provisioning profiles in Xcode to get those changes. If you revoke a development certificate, download provisioning profiles to regenerate the team provisioning profiles managed by Xcode.

### To download provisioning profiles in Xcode

1. In the Xcode Preferences window, click Accounts.
2. Select your team, and click View Details.



3. In the dialog that appears, click the "Download all" button in the lower-left corner under the Provisioning Profiles table.

Xcode updates the list of profiles in the Provisioning Profiles table.

# Using Custom Provisioning Profiles

Occasionally, you may want to use your own custom development provisioning profile instead of the team provisioning profile that Xcode manages for you. For example, you might do this if you want to limit development of an app to a subset of developers or if you're testing different app configurations. If so, create a development provisioning profile, as described in [Creating Provisioning Profiles Using Your Developer Account](#), and set your code signing identity build setting to use the new profile.

When you build the app, you code sign it with the signing identity matching the certificate contained in the provisioning profile you want to use. The possible values for the Code Signing Identity build setting pop-up menu are:

- **Don't Code Sign.** Don't sign your app. Choosing this option disables entitlements including sandboxing.
- **Automatic Profile Selector.** Selects an identity that matches your developer or distribution certificate name.
- **Identities without Provisioning Profiles.** Selects a code signing identity that isn't in a provisioning profile.
- **Other.** Selects a specific code signing identity. The code signing identities in your default keychain are listed by the name. Expired or otherwise invalid identities are dimmed and can't be chosen.

A menu item appears in the Code Signing Identity build setting pop-up menu for each provisioning profile to which your development certificate belongs. The default setting is the platform-specific development certificate that appears in the Automatic Profile Selector menu item, which matches your development certificate. For a description of each type of certificate that may appear in this menu, refer to [Table 14-2](#).

Before you begin, decide whether to set the Code Signing Identity build setting at the project or target level. For a single target, you can set this build setting at either the project or target level as long as you're consistent. For multiple targets that use the same code signing identity, set this build setting at the project level. For multiple targets that use different code signing identities, you set this build setting for each individual target. For example, choosing the project level ensures that any helper apps inside of your project are code signed as well as the main app.

Set the Provisioning Profile build setting to your development profile and the Code Signing Identity build setting to your development certificate.

## To set the code signing identity to your development certificate

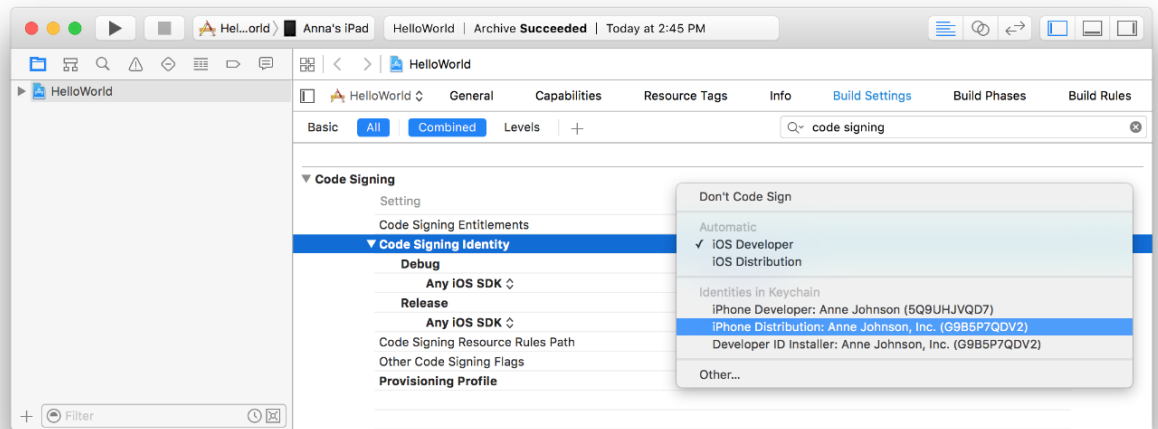
1. In the Xcode project editor, select the target.

**Important:** If you want to sign multiple targets with the same code signing identity, select the project, not a target.

2. Click Build Settings.
3. In the Build Settings pane, click All and type `code signing` in the search field.
4. From the Provisioning Profile pop-up menu, choose your development provisioning profile.  
The default value is Automatic.
5. If necessary, from the Code Signing Identity pop-up menu, choose your development certificate.

For iOS apps, choose the certificate in the provisioning profile menu item that begins with the text "iPhone Developer:" followed by your name. For Mac apps, choose the certificate in the provisioning profile menu item that begins with the text "Mac Developer:" followed by your name.





Your app is code signed the next time you build it. You can build and run your Mac app by simply clicking the Run button. For an iOS app, follow the steps in [Launching Your App on a Device](#) (iOS, tvOS, watchOS) to sign your app and launch it on a device.

To use the team provisioning profile again later, change the Provisioning Profile build setting to None. To learn more about Apple's code signing technology, read *Code Signing Guide*.

## Troubleshooting

If the development provisioning profile doesn't appear in the Provisioning Profile pop-up menu, download provisioning profiles, as described in [Downloading Provisioning Profiles in Xcode](#). Then try to set the Provisioning Profile build settings again.

If a code signing error occurs when you build the app, verify that the Code Signing Identity build setting is correct. Also, check whether the Code Signing Identity build setting is set at the project or target level (target settings override project settings). To troubleshoot the Code Signing Identity build setting, read [Build and Code Signing Issues](#).

## Migrating to Xcode-Managed Provisioning Profiles

Xcode creates and updates code signing identities and provisioning profiles for you. However, if your project was created using Xcode 4 or earlier, you may need to set the Provisioning Profile build setting to Automatic to use this feature.

### To use Xcode-managed provisioning profiles

1. In the Xcode project editor, select the target.
2. Click Build Settings.
3. In the Build Settings pane, click All and type `Code Signing` in the search field.
4. From the Provisioning Profile pop-up menu, choose Automatic.

Xcode sets the highest Code Signing Identity build setting to Don't Code Sign, and the "Any ... SDK" Code Signing Identity build settings to iOS Developer (for iOS apps) and Mac Developer (for Mac apps).

If necessary, set the project-level Provisioning Profile build setting to Automatic. To troubleshoot provisioning profiles, read [Provisioning Issues](#).

## Editing Provisioning Profiles in Your Developer Account

You don't need to re-create provisioning profiles to edit them. You can change the name of a provisioning profile and other properties depending on the type of provisioning profile. For all types of provisioning profiles, you can change the App ID. For an iOS, tvOS, and watchOS app, you can add devices to an ad hoc provisioning

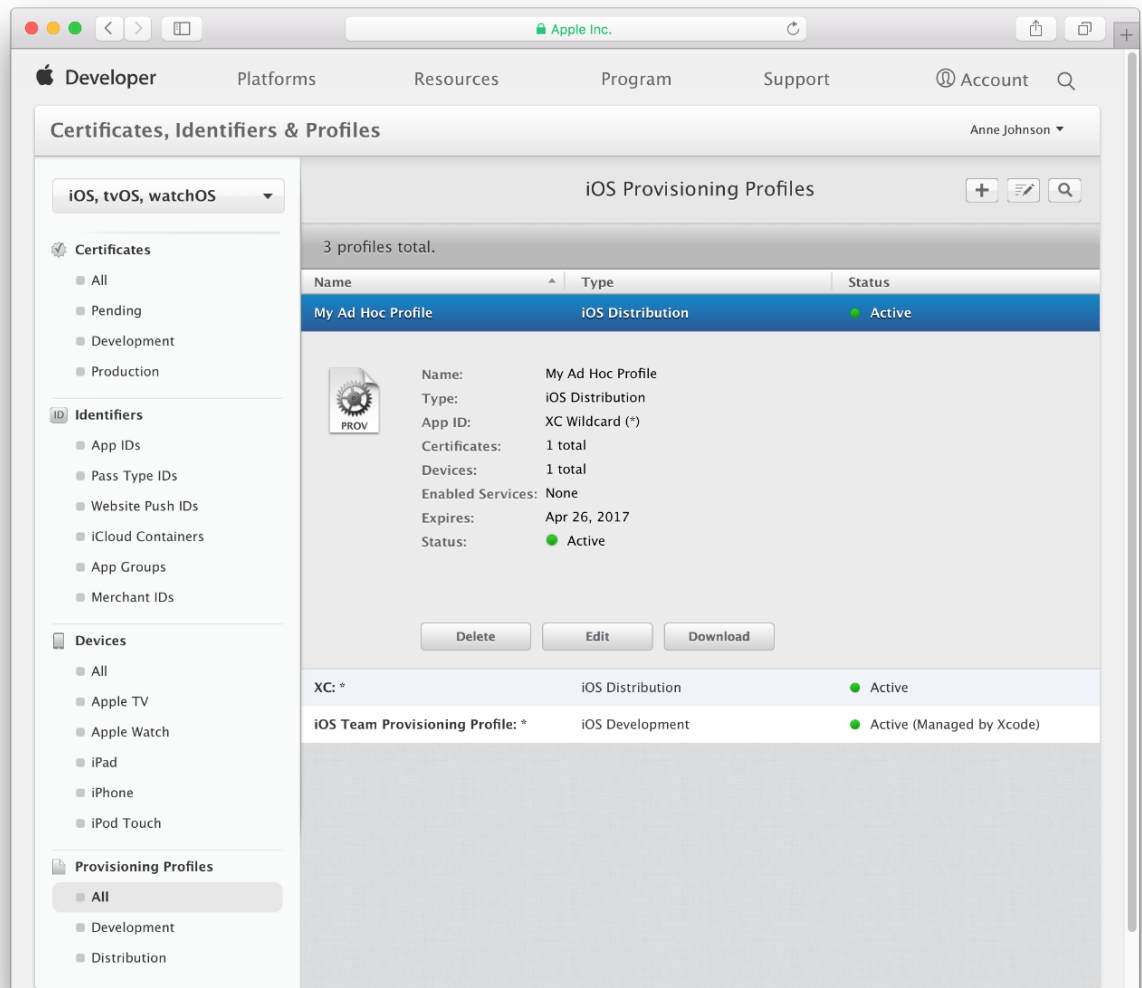
profile. If you change a provisioning profile, remember to replace instances of the provisioning profile that you installed on devices.

If you want to repair a provisioning profile because you re-created a certificate, follow the steps in Re-Creating Certificates and Updating Related Provisioning Profiles.

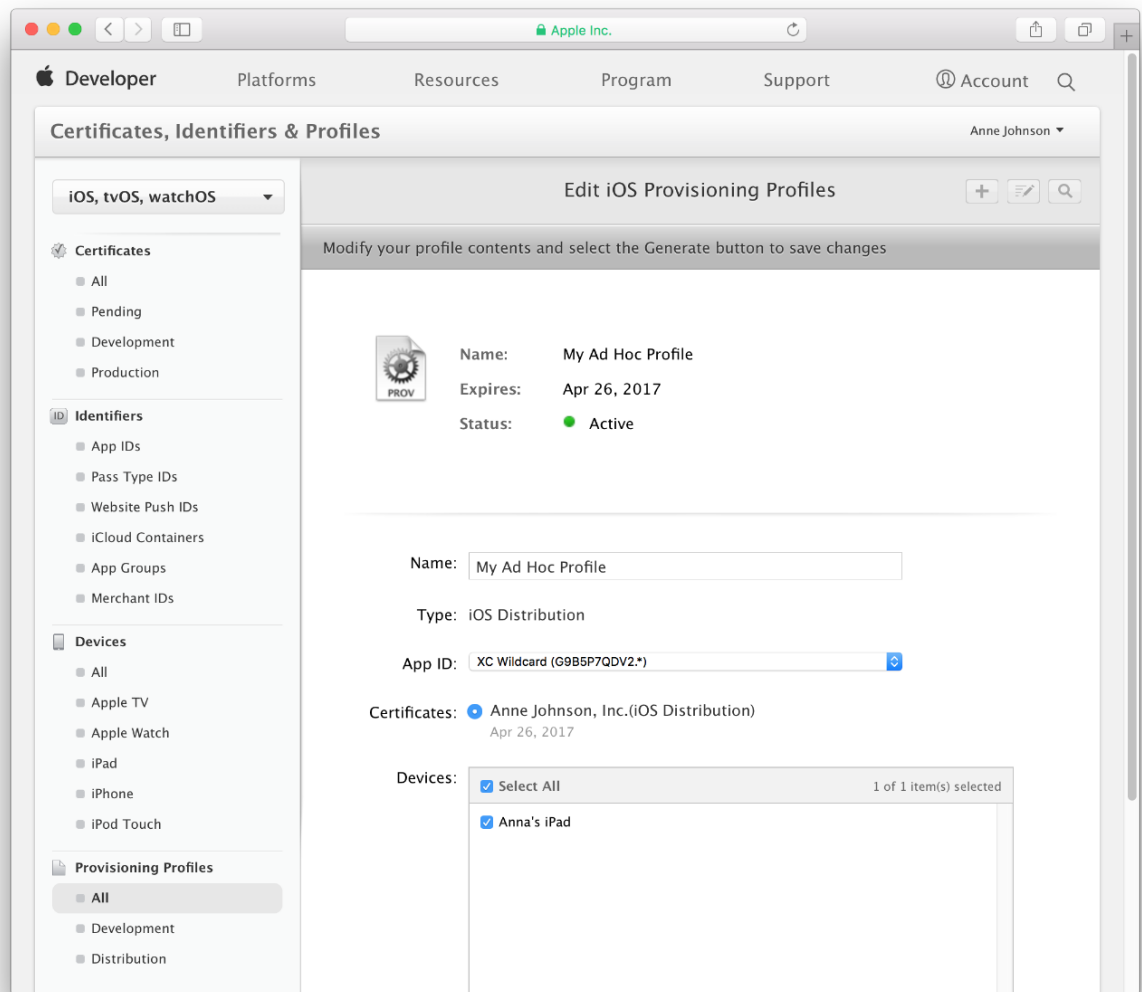
**Note:** You can't modify the team provisioning profile managed by Xcode.

### To edit a provisioning profile

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All, Development, or Distribution.
3. Select the provisioning profile you want to modify, and click Edit.



4. Make your changes to the provisioning profile, such as changing its name, adding certificates, or selecting a different set of devices.



#### 5. Click Generate.

After doing so, download provisioning profiles in Xcode, as described in [Downloading Provisioning Profiles in Xcode](#).

## Renewing Expired Provisioning Profiles

If a provisioning profile expires, the provisioning profile's status displays Expired in your developer account. You renew an expired provisioning profile by editing and re-generating it, as described in [Editing Provisioning Profiles in Your Developer Account](#). You don't need to make any changes to the provisioning profile. Just scroll to the bottom of the Edit iOS Provisioning Profile or the Edit Mac Provisioning Profile page, and click Generate.

If the expired provisioning profile is installed on your device, remove it, as described in [Verifying and Removing Provisioning Profiles on Devices](#). If the provisioning profile is an ad hoc provisioning profile, re-sign and distribute your app using the regenerated provisioning profile, as described in [Exporting Your App for Testing \(iOS, tvOS, watchOS\)](#).

## Verifying and Removing Provisioning Profiles on Devices

Use Xcode to verify if your provisioning profile is installed on a device or to remove expired provisioning profiles from a device.

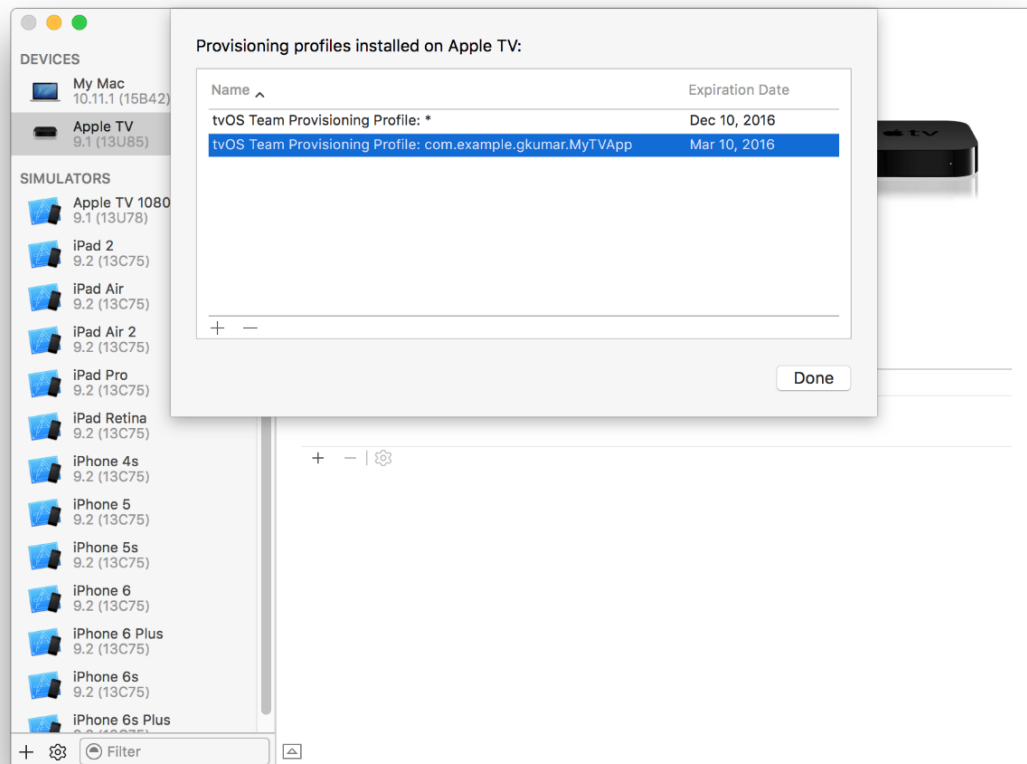
**Note:** You rarely install a provisioning profile yourself because when you launch an app on a device, iOS and OS X automatically install the embedded provisioning profile in the app's bundle on the device.

**To verify or remove a provisioning profile**

1. For iOS and tvOS apps, connect the device to your Mac. For watchOS apps, connect the iPhone paired to an Apple Watch.
2. Choose Window > Devices, and select the device under Devices.
3. At the bottom of the left column, click the Action button (the gear icon to the right of the Add button).
4. Choose Show Provisioning Profiles from the pop-up menu.

The provisioning profiles sheet appears.

5. To remove a provisioning profile, select it from the list and click the Delete button (-) in the lower-left corner.



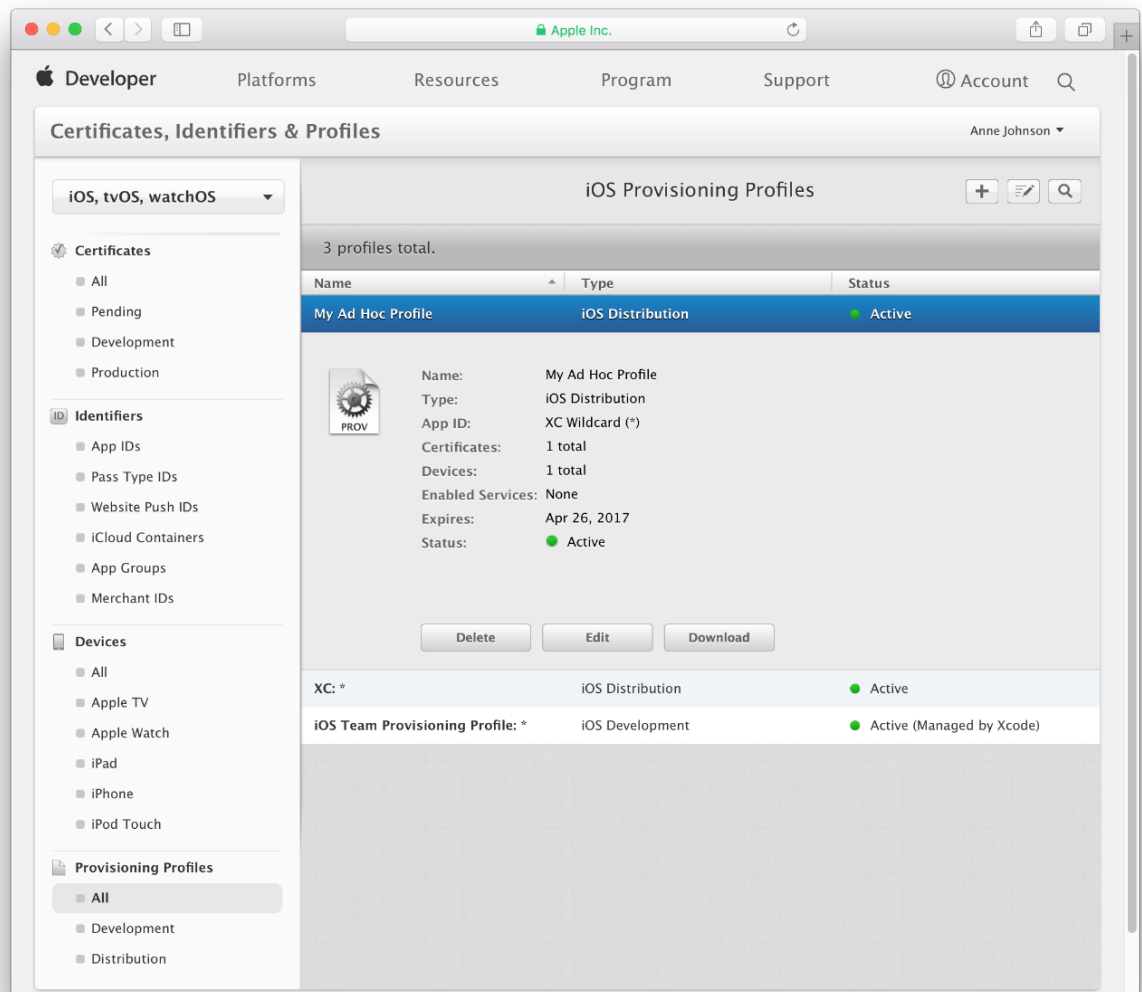
6. Click Done.

## Removing Provisioning Profiles from Your Developer Account

Occasionally, you may need to remove a provisioning profile from your team.

**To remove a provisioning profile from your team**

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All.
3. Select the provisioning profile you want to remove.
4. Click Delete.



5. In the dialog that appears, click Delete.

After doing so, download provisioning profiles in Xcode, as described in [Downloading Provisioning Profiles in Xcode](#), and remove the provisioning profile from your devices, as described in [Verifying and Removing Provisioning Profiles on Devices](#).

## Downloading Provisioning Profiles from Your Developer Account

If necessary, you can download specific provisioning profiles directly from your developer account.

### To download a provisioning profile

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Provisioning Profiles, select All.
3. Select the provisioning profile.
4. Click Download.

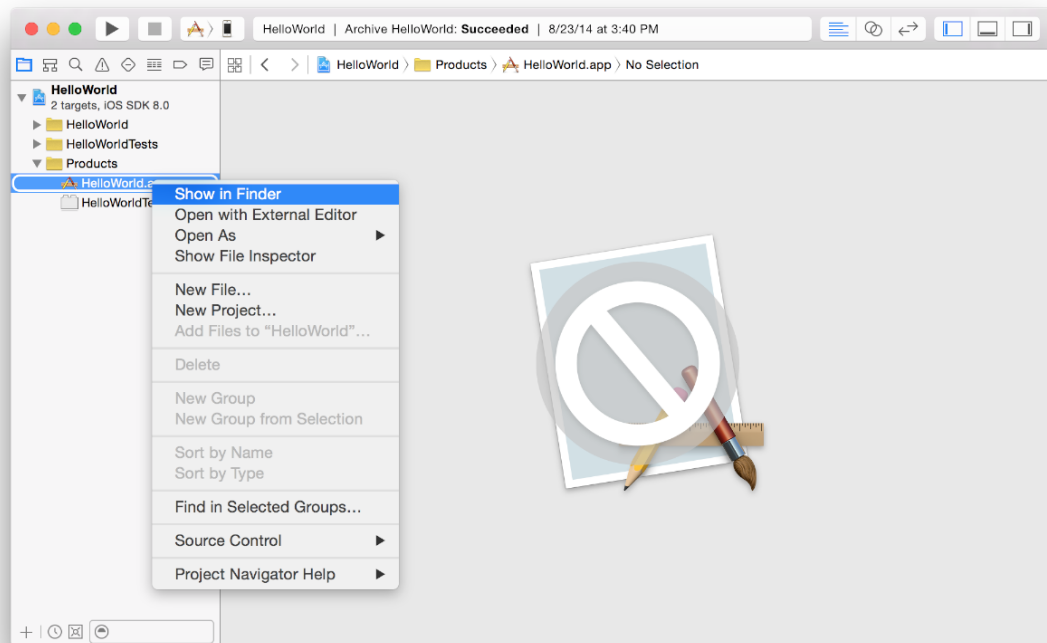
A file with the provisioning profile name and the `.mobileprovision` extension appears in your Downloads folder.

## Verifying the App Binary Entitlements

Some entitlements (for example, App Sandbox entitlements) are set in your Xcode project and others are set in the provisioning profile. You can check if the signed app has the correct entitlements by examining the app's signature and if discrepancies occur, by examining the embedded provisioning profile located in the app binary.

### To verify the entitlements of a signed app

1. In Xcode, select your project in the project navigator.
2. Click the disclosure triangle next to the project to reveal its contents.
3. Click the disclosure triangle next to Products to reveal the app binary.
4. Control-click the app binary file, and choose "Show in Finder" from the shortcut menu to go to the Xcode build location in the Finder.



5. Launch Terminal (located in /Applications/Utilities), and enter this text followed by a space character (do not press Return):

```
codesign -d --entitlements -
```

6. In the Finder, drag the app binary to Terminal.
7. Press Return.

For example, the output for an iOS app that is enabled for iCloud key-value storage contains the `com.apple.developer.ubiquity-kvstore-identifier` entitlement key. The output for a Mac app that is enabled for App Sandbox contains the `com.apple.security.app-sandbox` entitlement key.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>application-identifier</key>
  <string>G9B5P7QDV2.edu.self.HelloWorld</string>
  <key>com.apple.developer.pass-type-identifiers</key>
  <array>
    <string>G9B5P7QDV2.*</string>
  </array>
  <key>com.apple.developer.ubiquity-container-identifiers</key>
  <array>
    <string>G9B5P7QDV2.edu.self.HelloWorld</string>
```

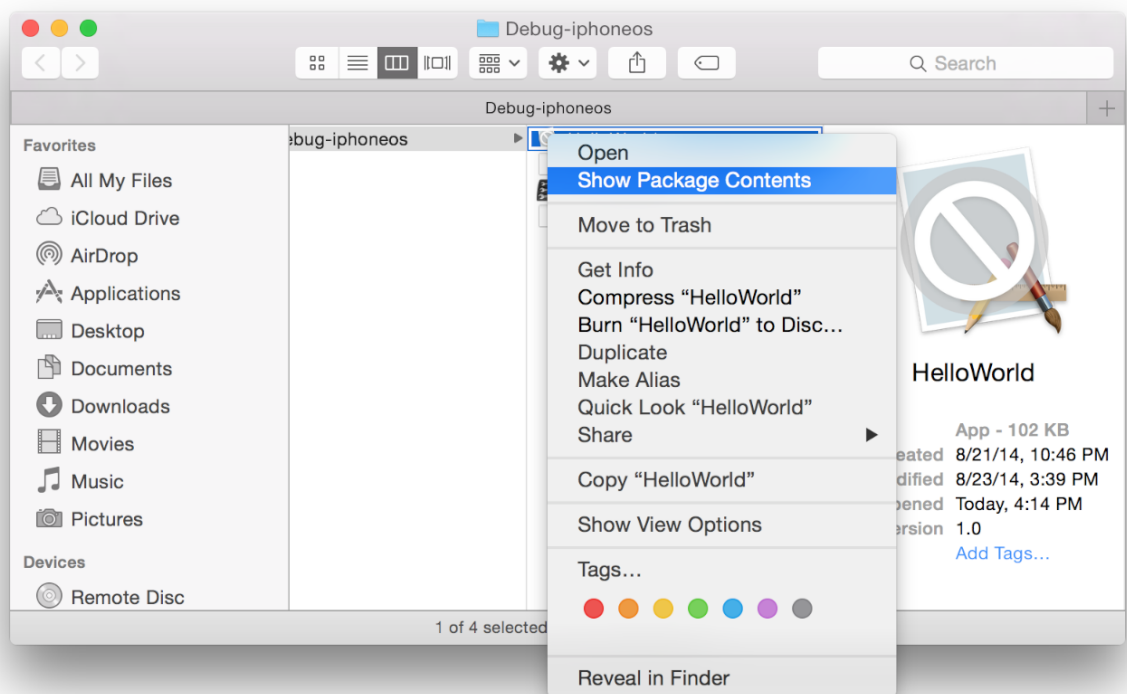
```
</array>
<key>com.apple.developer.ubiquity-kvstore-identifier</key>
<string>G9B5P7QDV2.edu.self.HelloWorld</string>
<key>get-task-allow</key>
<true/>
</dict>
</plist>
```

If the app's entitlements are different than what you have configured, verify that the embedded provisioning profile is correct. First, you need to locate the embedded provisioning profile.

### To locate the embedded provisioning profile in the app binary

1. In Xcode, select your project in the project navigator.
2. Click the disclosure triangle next to the project to reveal the contents.
3. Click the disclosure triangle next to Products to reveal the app binary.
4. Control-click the app binary file, and choose "Show in Finder" from the shortcut menu to go to the Xcode build location in the Finder.
5. In the Finder, Control-click the app binary file, and choose Show Package Contents from the shortcut menu.

For iOS apps, a provisioning profile called `embedded.mobileprovision` appears in the Finder window. For Mac apps, the embedded file is called `embedded.provisionprofile`.

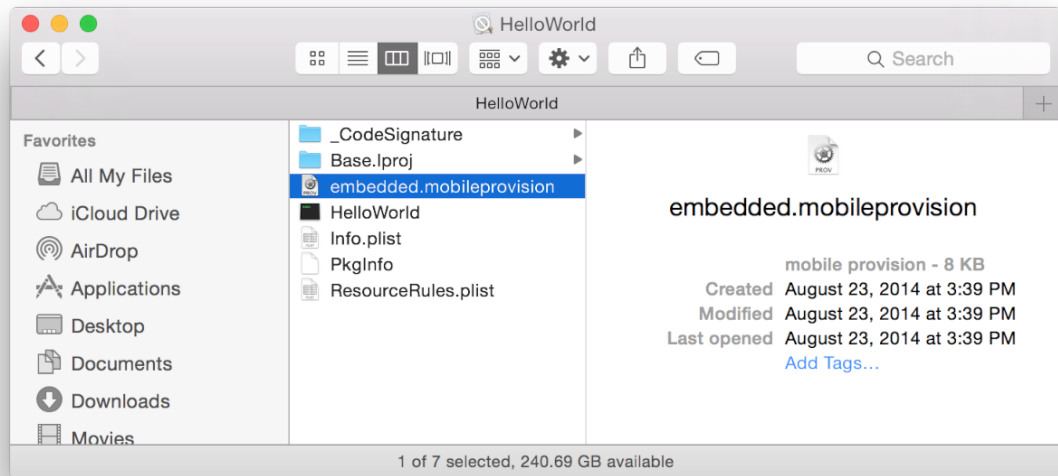


### To verify the entitlements of the embedded provisioning profile

1. Launch Terminal (located in `/Applications/Utilities`), and enter this text (do not press Return):

```
security cms -D -i
```

2. In the Finder, drag the provisioning profile in the app binary to Terminal.



### 3. Press Return.

This command outputs a property list in XML format.

### 4. Locate the `Entitlements` key in the output and verify that the `application-identifier` key has the correct entitlements.

For example, the following listing shows an iOS app that is enabled for data protection, Wallet, and iCloud. iCloud entitlements begin with the text `com.apple.developer.ubiquity`.

```
<key>Entitlements</key>
  <dict>
    <key>application-identifier</key>
    <string>G9B5P7QDV2.*</string>
    <key>com.apple.developer.default-data-protection</key>
    <string>NSFileProtectionComplete</string>
    <key>com.apple.developer.pass-type-identifiers</key>
    <array>
      <string>G9B5P7QDV2.*</string>
    </array>
    <key>com.apple.developer.ubiquity-container-identifiers</key>
    <array>
      <string>G9B5P7QDV2.*</string>
    </array>
    <key>com.apple.developer.ubiquity-kvstore-identifier</key>
    <string>G9B5P7QDV2.*</string>
    <key>get-task-allow</key>
    <true/>
    <key>inter-app-audio</key>
    <true/>
    <key>keychain-access-groups</key>
    <array>
      <string>G9B5P7QDV2.*</string>
    </array>
  </dict>
```

See `codesign` and `security` for more ways to use these commands.



## Recap

In this chapter, you learned how to maintain your certificates and provisioning profiles in a valid state and remove assets that you no longer need. To resolve specific certificate and provisioning profile issues, read [Troubleshooting](#).

---

Copyright © 2016 Apple Inc. All Rights Reserved. [Terms of Use](#) | [Privacy Policy](#) | Updated: 2016-04-29