

# Distributing Apple Developer Enterprise Program Apps

The Apple Developer Enterprise Program allows you to develop proprietary, in-house iOS, and watchOS apps that you can distribute to your employees and outside the store.

If you are not a member of the Apple Developer Enterprise Program, go to [Apple Developer Enterprise Program](#) to join.

**Note:** Members of the Apple Developer Enterprise Program can also create Developer ID certificates to distribute Mac apps, described in [Distributing Apps Outside the Mac App Store](#).

## Developing Apple Developer Enterprise Program Apps

The workflow for developing Apple Developer Enterprise Program apps is similar to the workflow used by any large organization that develops multiple apps for the store. During development, let Xcode manage your assets for you and use your developer account only as needed. Xcode will create your App ID and configure your project correctly to use the app services you enable.

### Build Your Team (Team Agent)

If you are the team agent (the person who joins the Apple Developer Enterprise Program), build your team first by inviting team members and assigning roles. Assign the team admin role to people who help you manage your team. Assign the team member role to persons who develop your app but don't have permission to distribute it. The team agent and team admins share the responsibility of exporting your app for distribution outside of the store. The tasks a team agent can perform are a superset of the tasks that a team admin can perform. Initially, the team agent is the only team member. To invite others, read [Inviting Team Members and Assigning Roles](#).

To learn how to manage your team, read [Managing Your Developer Account Team](#).

### Create Shared Team Provisioning Profiles (Team Admin)

If you are a team admin, perform these steps to enable your development team:

1. Create a team provisioning profile and enable capabilities.

Follow the same steps as an individual developer to create a team provisioning profile and enable capabilities, described in [Configuring Your Xcode Project for Distribution and Adding Capabilities](#). Xcode automatically creates an appropriate App ID and provisioning profile for you. If you want to use APNs, read [Configuring Push Notifications](#) for additional steps. For the capabilities available to an enterprise app, read [Supported Capabilities](#).

2. Optionally, register test devices.

Xcode will automatically register a device connected to a Mac and selected from the Scheme toolbar menu by team members.

**Important:** Each team member should have his or her own Apple ID and development certificate. Don't share signing identities between team members.

Team admins can also remove team members as needed. If you use contractors to develop your apps, invite them to join your team and assign them the team member role. When their contract ends, remove them from your team, described in [Removing Team Members](#).

## Begin Development (Team Member)

If you are a team member, perform these steps to get started:

1. In Xcode, add your Apple ID to the Accounts preferences, described in [Adding Your Apple ID Account in Xcode](#).
2. To create your development certificate and register your device, follow the steps in [Configuring Your Xcode Project for Distribution](#).

Select a team that is a member of the Apple Developer Enterprise Program from the Team pop-up menu. Xcode will regenerate and download team provisioning profiles as needed.

## Testing Apple Developer Enterprise Program Apps

You use an ad hoc provisioning profile, described in [Archiving Your App](#), to export an Apple Developer Enterprise Program app from Xcode for beta testing. Only the team agent and team admins can create an ad hoc provisioning profile for distribution. After you export your app, consider using Xcode Server to distribute it to testers and other team members. See *Xcode Server and Continuous Integration Guide* for more information about using Xcode Server.

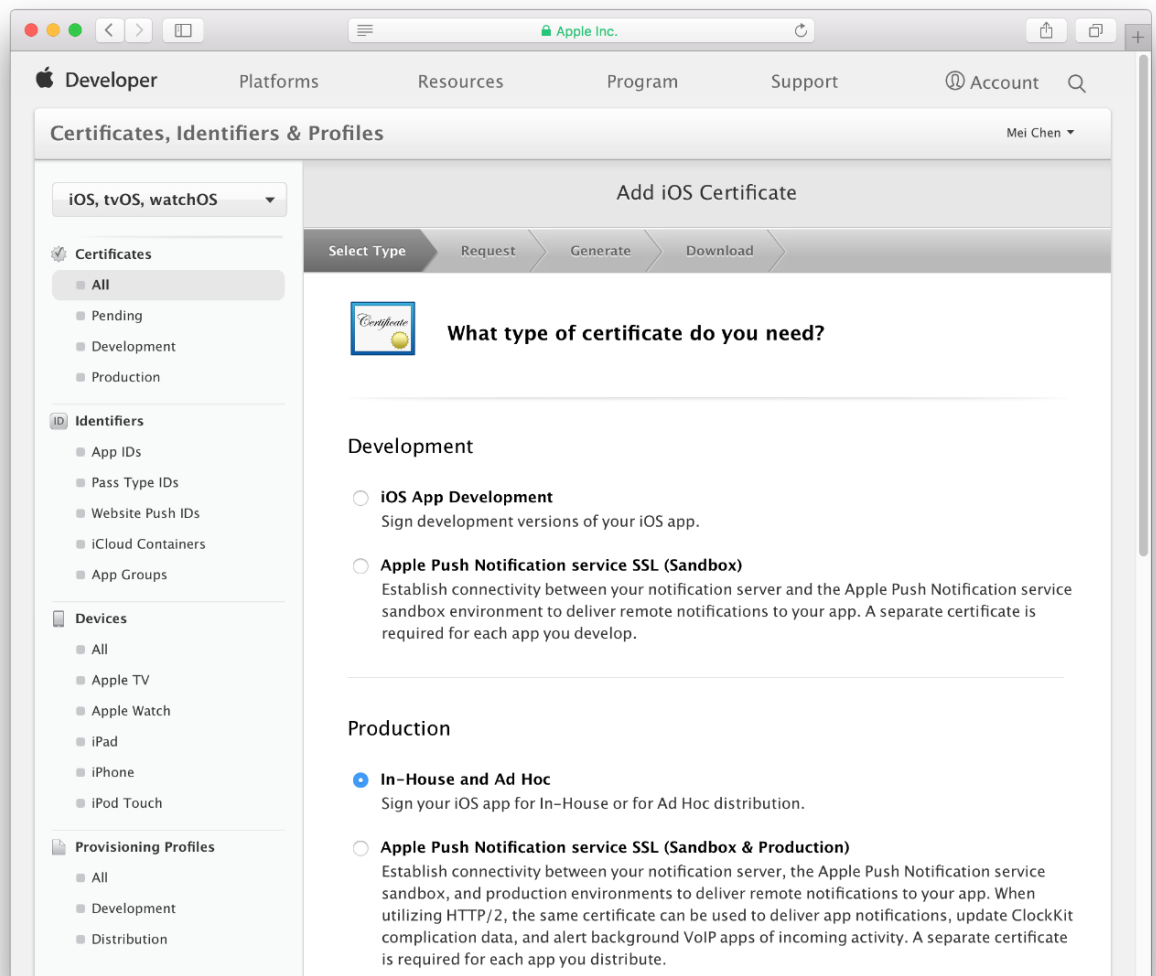
**Important:** As a member of the Apple Developer Enterprise Program, you don't have an iTunes Connect account or the ability to distribute your app for beta testing using iTunes Connect.

## Creating Additional Enterprise Distribution Certificates

As a member of the Apple Developer Enterprise Program, you are allowed to create multiple distribution certificates.

### To create another distribution certificate

1. Sign in to [developer.apple.com/account](https://developer.apple.com/account), and click Certificates, IDs & Profiles.
2. Under Certificates, select All.
3. Click the Add button (+) in the upper-right corner.
4. Under Production, select "In-House and Ad Hoc" and click Continue.



5. Follow the instructions to create a certificate signing request (CSR) using Keychain Access, and click Continue.
6. Click Choose File.
7. Select a CSR file (with a `.certSigningRequest` extension), and click Choose.
8. Click Continue.
9. Click Download.

The certificate file appears in your Downloads folder.

To install the certificate in your keychain, double-click the downloaded certificate file (with a `.cer` extension). The distribution certificate appears in the My Certificates category in Keychain Access.

## Managing Expiring Certificates and Provisioning Profiles

You are responsible for managing your team's certificates and provisioning profiles. Apple Developer Enterprise Program certificates expire after three years and provisioning profiles expire after one year.

Before a distribution certificate expires, create an additional distribution certificate, described in [Creating Additional Enterprise Distribution Certificates](#). You cannot renew an expired certificate. Instead, replace the expired certificate with the new certificate, described in [Replacing Expired Certificates](#).

If a distribution provisioning profile expires, verify that you have a valid distribution certificate and renew the provisioning profile, described in [Renewing Expired Provisioning Profiles](#).

Xcode manages your development certificates and team provisioning profiles for you.

# Exporting Your App In-House

To export your app for distribution to your employees and outside the store:

1. Archive your app.
2. Export the archive as an *iOS App* file (a file with a `.ipa` filename extension).

**Note:** If you don't distribute your app using a Mobile Device Management (MDM) system, users can install your app using the iOS App file but then they will need to manually trust your organization to launch the app, described in [Manually Trusting an Enterprise Developer](#).

## Creating an Archive

Create an archive of your app regardless of the type of distribution method you select. Xcode archives allow you to build your app and store it, along with critical debugging information, in a bundle that's managed by Xcode.

**Important:** Save the archive for every version of an app you distribute. You need the debugging information stored in the archive to decipher crash reports later.

### To create an archive

1. In the Xcode project editor, choose a generic device or your device name from the Scheme toolbar menu.  
You can't create an archive of a simulator build. If a device is connected to your Mac, the device name appears in the Scheme toolbar menu. Otherwise, choose the generic device.
2. Choose Product > Archive.

The Archives organizer appears and displays the new archive.

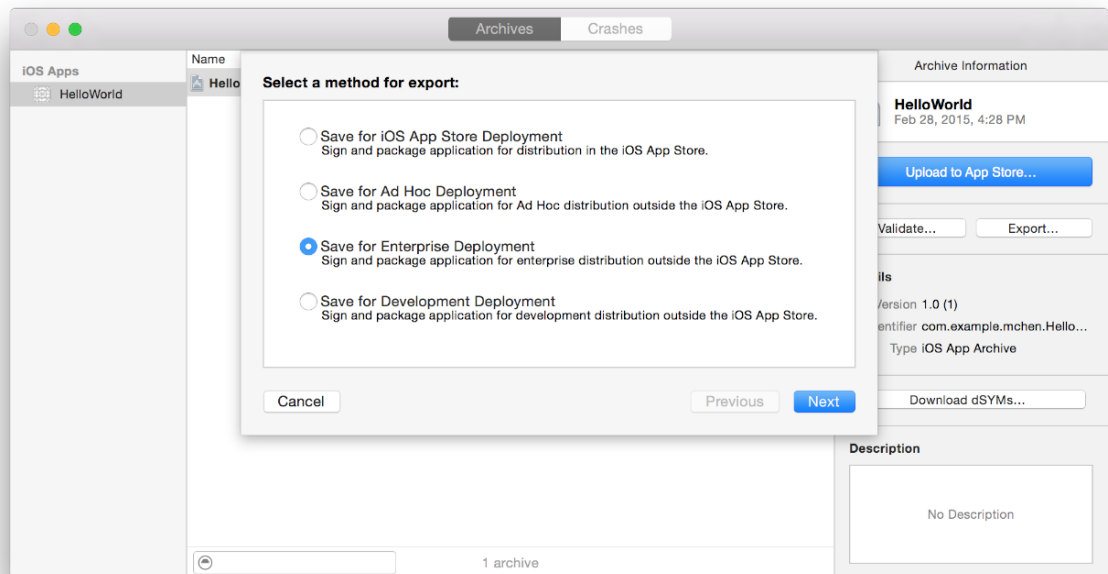
Xcode runs preliminary validation tests on the archive and may display a validate warning in the project editor. For example, if you don't set required app icons, as described in [Setting Individual App Icon and Launch Image Files](#), an `Info.plist` warning message appears. If you see this warning, fix the issue and create the archive again.

## Creating an iOS App File

You create an iOS App file so that users can install your app on their device. You generate an iOS App file (a file with a `.ipa` filename extension) from the archive. Xcode automatically creates the necessary distribution certificates and provisioning profiles for you when you export the app.

### To create an iOS App file for distribution

1. In the Archives organizer, select the archive.
2. Click the Export button, select "Save for Enterprise Deployment," and click Next.

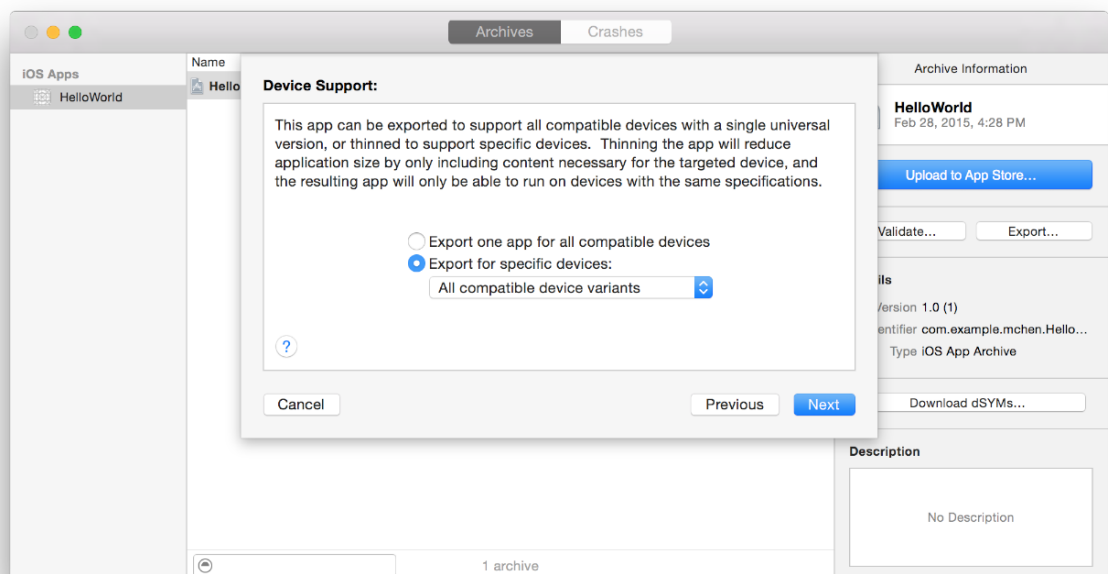


3. In the dialog that appears, choose a team from the pop-up menu, and click Choose.

If necessary, Xcode creates a distribution certificate, provisioning profile, and explicit App ID for you.

4. In the Device Support dialog, choose whether to export the universal app or variants for a specific devices, and click Next.

- If you want to run the app on any supported device, select “Export one app for all compatible devices.”
- If you want to test all device variances, select “Export for specific devices” and choose “All compatible device variants” from the pop-up menu.
- If you want to test a specific device variant, select “Export a thinned app for a specific device” and choose the device family from the pop-up menu.



5. In the dialog that appears, review the app, its entitlements, and provisioning profile.

You can not distribute an enterprise app using a wildcard App ID. The name of the distribution provisioning profile begins with the text `XC:` followed by the App ID. If you are using a wildcard App ID, the name of the distribution provisioning profile is `XC:*`.

The Finder shows the exported that has an `.ipa` extension.

6. Review the build options, and click Next.

If you use on-demand resources, check “Include manifest for over-the-air installation.” The manifest file is an XML plist used by a device to find, download, and install apps from your web server.

7. If you request a manifest file, enter details about your web server in the “Distribution manifest information” dialog that appears, and click Export.

Enter the following information:

- **Name.** The name of the app displayed during download and installation.
- **App URL.** A fully qualified HTTPS URL for the iOS App file.
- **Display Image URL.** A fully qualified HTTPS URL for an app icon that is displayed during download and installation. The image file must be 57 x 57 pixels and in PNG format.
- **Full Size Image URL.** A fully qualified HTTPS URL for a larger image that is displayed in iTunes. The image file must be 512 x 512 pixels and in PNG format.

8. Enter a filename and location for the iOS App file, and click Export.

## Manually Trusting an Enterprise Developer

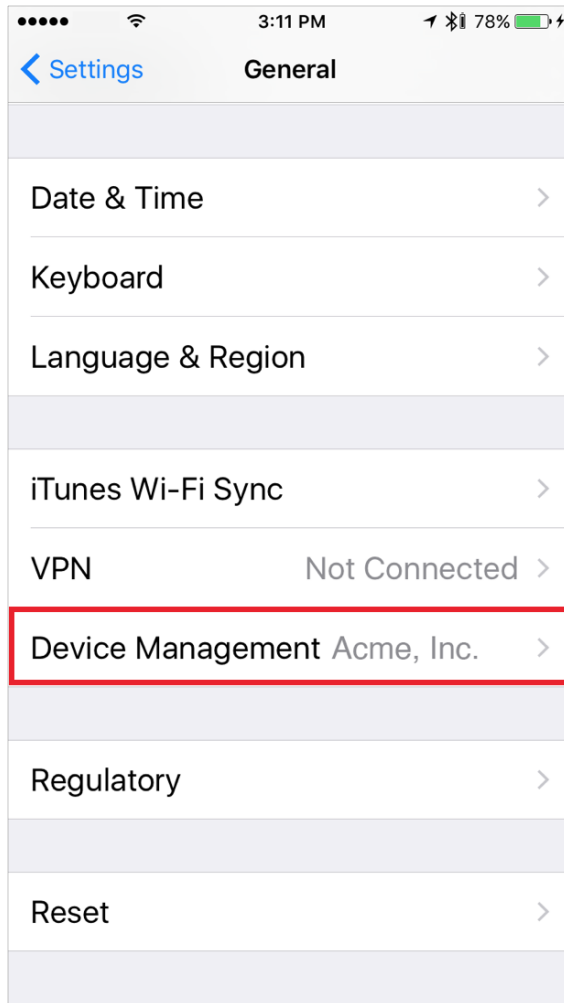
If you send the iOS App File to a user and they manually install it on their device, then they must also manually trust your organization before launching the app. Instruct the user to follow these steps to trust your organization.

**Note:** An Internet connection is required to trust an enterprise developer. If you're behind a firewall, make sure it's configured to allow connections to <https://ppg.apple.com>.

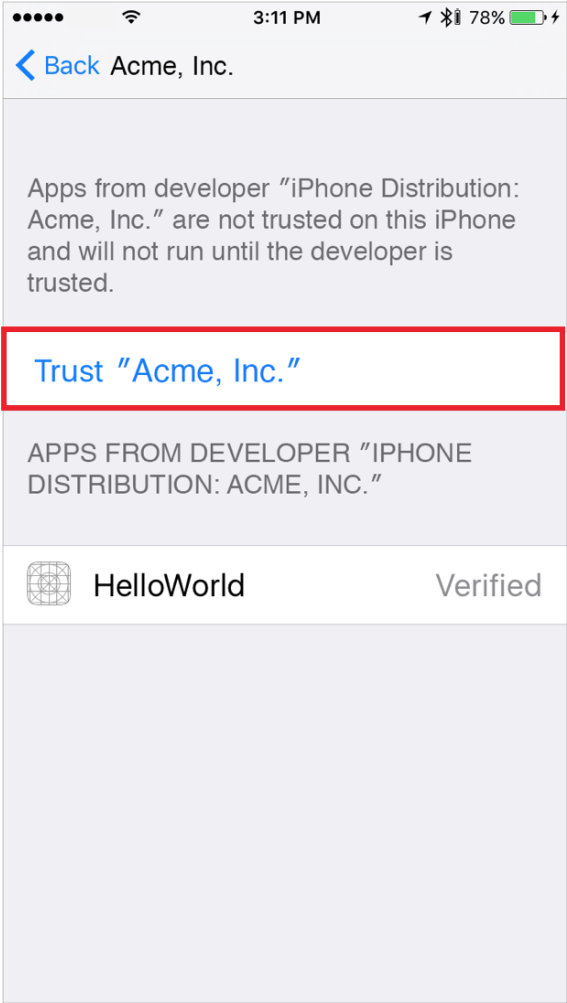
### To trust an enterprise developer

1. On the device, launch the app, and in the “Untrusted Enterprise Developer” dialog that appears, click Cancel.  
This step adds Device Management to Settings.
2. Tap Settings > General > Device Management.

**Note:** If profiles are installed on your device, the setting is called Profiles & Device Management.



3. Under Enterprise App, tap the organization name.
4. Tap Trust “[Organization Name]”.



5. In the dialog that appears, tap Trust.

**Note:** If you are not connected to the Internet, a “Not Verified” message appears. Connect the device to the Internet and tap the Verify App button.

To untrust an enterprise developer, go to Settings > General > Device Management > [Organization Name], and delete all the apps from the organization.

To manually install an enterprise app without needing Xcode installed on your Mac, read [Installing Your App on Test Devices Using iTunes \(iOS, watchOS\)](#) and [Installing Your App on Test Devices Using Apple Configurator 2 \(iOS, watchOS, tvOS\)](#).

# Learn More About Server Tools

Take advantage of Xcode server tools that support large software development teams.

| To learn about  | Read   |
|---|--|
| <b>Continuous integration</b><br>Use Xcode service running on OS X Server to automate building, analyzing, testing, and archiving your app. | <i>Xcode Server and Continuous Integration Guide</i> |
| <b>Automated testing</b><br>Set up tests that can be run by Xcode service.  | <i>Testing with Xcode</i>                            |



|  |   |
|--|---|
| <b>Host source control repositories on servers</b><br>Use Xcode service to connect to remote repositories. | <i>Xcode Server and Continuous Integration Guide</i><br><i>Source Control Management Help</i>   |
| <b>Mobile Device Management</b><br>Use MDM to deploy business and education apps.                          | Go to<br><a href="http://www.apple.com/support/iphone/enterprise/">http://www.apple.com/support/iphone/enterprise/</a><br>and in the left column, click Mobile Device Management. |

## Recap

In this chapter, you learned a variation of the development and distribution steps for Apple Developer Enterprise Program members. You learned how to build your team, create shared team assets, export your app for testing, and later, export your app for distribution outside the store.