In this guide

Legal notice .......................................................... 3
Introduction .......................................................... 4
Certificates required for building apps ......................... 5
Enroll in Apple’s iOS Developer Program ..................... 6
  Provide bank and tax information to Apple (if you are offering paid content) ......................... 6
Read the App Store Review Guidelines for iOS apps ........ 9
Download and install Xcode on your Mac .................... 10
Create certificates .................................................... 11
  Who creates the certificates? .................................. 11
  Overview .......................................................... 11
Create a Development Certificate Signing Request .......... 12
Generate a Development Certificate ............................ 16
Create a p12 Development Certificate ........................ 18
Create a Distribution Certificate Signing Request .......... 20
Generate a Distribution Certificate .............................. 23
Create a p12 Distribution Certificate ........................ 25
Test your certificates ............................................... 27
Create an App ID ..................................................... 28
Register your test iPads/iPhones with Apple ................. 32
Set up Apple push notifications (optional) .................... 36
  Create a Development Push Certificate Signing Request ... 36
  Generate a Development Push Certificate .................. 40
Create a p12 Development Push Certificate ................. 42
Create a Production Push Certificate Signing Request ..... 44
Generate a Production Push Certificate ....................... 47
Create a p12 Production Push Certificate ................... 49
Create mobileprovision files ...................................... 51
  Create a development mobileprovision file ................. 51
  Create a distribution mobileprovision file ................ 55
Create required assets ............................................. 59
Create an iTunes Connect record for the app ............... 62
Create In-App Purchases for retail collections ............ 69
Set up subscription apps .......................................... 73
  Generate or view a shared secret ............................ 74
  Create an In-App Purchase for a retail subscription .... 75
Build and sign the development app ........................... 78
Test the development app ......................................... 79
  Test an app with free content ............................... 79
  Test an app with retail content ............................. 80
Create and sign the distribution app ........................... 82
Submit your app to the App Store .............................. 83
Wait for approval from Apple ................................... 85
Update an app ....................................................... 85
iOS publishing troubleshooting ................................. 86
Legal notice

THE CONTENTS OF THIS GUIDE is Adobe Confidential and subject to the Terms of Use, is furnished under license and may be used or copied only in accordance with the terms of such license. No part of this guide may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, recording, or otherwise, without the prior written permission of Adobe Systems Incorporated.

Please note that the content in this guide is protected under copyright law even if it is not distributed with software that includes an end user license agreement.

The content of this guide is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Adobe Systems Incorporated. Adobe Systems Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in the informational content contained in this guide.

Please remember that existing artwork or images that you may want to include in your project may be protected under copyright law. The unauthorized incorporation of such material into your new work could be a violation of the rights of the copyright owner. Please be sure to obtain any permission required from the copyright owner.
Introduction

IF YOU ARE READING THIS GUIDE, you are in the process of creating an iOS app with Adobe Experience Manager Mobile (AEM Mobile). A different publishing guide is available for Digital Publishing Suite (DPS).

This process of submitting an app to the App Store can be completed by anyone willing to read directions, pay attention to detail, and follow step-by-step instructions. Most of the steps do not require an understanding of Adobe InDesign. For this reason, you may choose to enlist the help of a co-worker or someone else in your company to help you with this process.

Don't wait until the last minute to read this guide. You can expect to spend two or three hours completing all these steps. You may wish to break this into a couple of sessions. Once you have completed the steps, you will need to wait for Apple to approve or reject your app. As of this writing, this process is typically taking about seven days.

Apple governs the app submission process. Many of the steps in this guide describe processes that will take place on various Apple websites. You must use a Macintosh computer to create the certificates required for the submission and for the final upload of the app to the App Store.

INFORMATION with a gray background indicates that the task must be performed on an Apple website. For more information about these tasks, refer to Apple documentation at bit.ly/N9sNck and bit.ly/N9fBUV or contact Apple.
Certificates required for building apps

FOR EACH APP YOU CREATE, Apple requires various certificates and files. The primary purpose of this guide is to help you create these required files.

- Developer P12 certificate (and password)
- Distribution P12 certificate (and password)
- Developer mobileprovision file
- Distribution mobileprovision file
- Developer push P12 certificate and password (required only if push is enabled)
- Production push P12 certificate and password (required only if push is enabled)

After you use the Apple Developer site to create these certificates and mobileprovision files, you use them for building apps, along with image files and other app configuration details discussed later in this guide.
Enroll in Apple’s iOS Developer Program

YOU OR YOUR ORGANIZATION MUST BE ENROLLED in Apple’s iOS Developer Program to submit your app to the App Store. You need to keep your enrollment current and paid for as long as you want your app to appear on the App Store.

If you have not already signed up for the iOS Developer Program, visit bit.ly/Rjz90j to get started. For purposes of creating apps, you can enroll as either an individual or a company.

Someone else in your company may already be enrolled in the iOS Developer Program. You may be able to ask your IT department to create the required App ID, certificates, and mobileprovision files. If so, this will make the process much easier for you.

After you enroll in the iOS Developer Program and submit your payment, Apple will send you an email message that prompts you to activate your account.

Provide bank and tax information to Apple (if you are offering paid content)

If you wish to make your app or any of your app’s content available for sale through Apple, you must agree to the iOS Paid Applications agreement. If your app and content is going to be free, you can skip to page 9.

1. Go to itunesconnect.apple.com. You will need to sign in with your Apple ID.
2. Click on Contracts, Tax, and Banking.

ADOBE TIP

As you move through the steps in this guide you will create an Adobe ID and password, Apple ID and password, certificate passwords, and a unique App ID for each app that you create. Don’t get all these IDs and passwords confused! Create a system to help you keep all this straight.
3. Click the Request button next to the iOS Paid Applications contract.

![Contracts, Tax, and Banking](image)

4. If acceptable, accept the terms of the agreement, and click the Submit button.

![Review Agreement](image)
5. The iOS Paid Applications contract will appear in the Contracts in Process section. Click on the Set Up buttons below Contact Info, Bank Info, and Tax Info, and provide the information requested.

6. Once you've provided all the requested information, you will need to wait for Apple to approve the contract. When the contract is approved, it will move to the Contracts in Effect section.

For more information, see the "Managing Contracts, Taxes, and Banking" section of Apple's iTunes Connect Developer Guide at bit.ly/NtQgqX.
Read the **App Store Review Guidelines** for iOS apps

APPLE REVIEWS ALL APPS that are submitted to the App Store to be sure that the apps meet the App Store Review Guidelines. Apps that do not meet these guidelines will be rejected by Apple. With this in mind, it is good to review these guidelines before you get too far into the process of developing your app. This will pay off in the end by ensuring that your app passes Apple's review process.

Download and install Xcode on your Mac

XCODE IS APPLE’S development environment” for creating Mac, iPhone, and iPad apps. Xcode is required to sign apps. You don't need to learn Xcode or any programming. The latest version of Xcode can be downloaded from bit.ly/RjArsh. To sign apps, Xcode 8.1 is required.

Xcode is an approximately 1.5 gigabyte download. Depending on your Internet connection speed, this download may take awhile. Be prepared to do this step well ahead of time so that you can afford to wait.

If you install Xcode, the Application Loader utility and a WWDR intermediate certificate are installed automatically.

One more note about Xcode. If you refer to Apple developer documentation when building apps, keep in mind that the app building process performs certain tasks that other developers use Xcode to perform.
Create certificates

THE APP BUILDER will generate two different apps—a development app and a distribution app. The development app is used for testing; the distribution app is what you’ll submit to Apple once you’ve tested and approved the development app. Apple requires both apps to be signed by a valid “certificate” before they can run on the iPad. In this section, you will learn how to create two required p12 certificates.

Note: You can use the same p12 certificates for multiple apps. They are not tied to the App ID. However, each app you create requires a separate App ID and separate mobileprovision files (described later).

Who creates the certificates?
The steps in this guide assume you will create the required certificates. If you work for an organization, someone else in your company might be responsible for creating the required certificates. If so, make the necessary arrangements to use those certificates for building your app.

Overview
The process of creating a p12 certificate file might seem complicated, but it’s not that difficult if you follow the steps carefully.

Creating the certificates involves a back-and-forth process between the Apple Developer site and the Keychain Access utility on your computer. You use the Keychain Access utility to create a certificate signing request (a different one for each certificate), and then submit the signing request in the Apple Developer site. Then you download the certificate (.cer) file and open it in the Keychain Access utility. Once the certificate is available in the keychain, you can sign the app. You can also export the p12 file and create a password if someone else needs to sign the app. You will complete this process for the development (testing) app and for the distribution (submission) app.
Create a Development Certificate Signing Request

1. On the Apple iOS Developer site, go to “Certificates, Identifiers & Profiles” at bit.ly/OVbhu5. Sign in with your Apple ID.

2. Click Certificates on the left side of the window.

3. Click Development on the left side of the window under Certificates.

   Selecting the options on the left side displays the certificates of each category. At this stage, you shouldn't have any development certificates. You just need one development certificate for testing all your iOS apps.

4. Click the Plus icon in the upper right corner of the window to start creating a certificate.
5. In the “What type of certificate do you need?” section, select iOS App Development. Click Continue.
As indicated on the Apple Developer site, you’ll now use the Keychain Access utility to create a Certificate Signing Request (CSR).

6. Open the Keychain Access utility, found in the Utilities folder in your Applications folder.

7. Click on the My Certificates category on the left.

8. Click on an empty space in the window on the right to ensure that no existing keys or certificates are selected.


During this process, you are going to be creating several files that you will need to keep track of. It is helpful to create a new folder in an easy-to-access location to store all of these “bits and pieces” that you will need later in the process.
10. For User Email Address, specify a valid email address.

11. For Common Name, enter the name of your company or division. This certificate can be reused for multiple apps that you might create, so you don't need to enter the app name here.

12. Leave the CA Email Address field blank.

13. Select Saved to Disk.

14. Click the Continue button.

15. Specify the name of the certificate signing request and where the file will be saved, and click Save. You will use this file to create a development certificate in the next section.

16. When the certificate request is created, click Done.
Generate a Development Certificate

Now that you've created the certificate signing request file, you need to submit it to the Apple Developer site to create the required development certificate.

1. In the Apple Developer site, click Continue to advance to the “Generate your certificate” screen. Click Choose File, and double-click the certificate request file you just saved in the previous steps.

2. Click Generate.
Your development certificate is generated.

3. Click Download, and then download the certificate to a known location.

   This development certificate is called ios_development.cer. This file is required to create your p12 certificate in the Keychain Access utility. You’ll do that next.
Add the Certificate to the Keychain

1. In the Finder, locate the ios_development.cer file you created, and double-click it. This launches the Keychain Access utility and installs the certificate.

2. To see if the certificate was installed correctly, click on the Keys category on the left, and then click the right-pointing arrow to open the private key. You should see the certificate beneath the private key. If double-clicking the .cer file does not install the certificate, launch Keychain Access, choose File > Import Items, and specify the .cer file. If that doesn't work, exit and re-start the Keychain Access utility and try again.

Once the certificate is added to your keychain, you can now use it for signing apps.

Export a p12 Development Certificate

Exporting a .p12 certificate is necessary only if you want to send the certificate to someone else so that they can sign apps.

1. With Keys selected on the left, Control-click or right-click the private key associated with your certificate, and choose Export “[name].”

Adobe Tip

If you forget the p12 certificate password you create here, there is no way to retrieve it—you would need to create your p12 certificates all over again. So don’t forget this password! You can use the same password for the distribution and development p12 certificates.
**Important**: Control-click the private key (highlighted in red below), not the certificate. (You can also select both the private key and certificate before exporting.)

2. Save your key in the Personal Information Exchange (.p12) file format. Specify a name that distinguishes it from the p12 file you’ll create for the distribution certificate. For example, we’re naming our development certificate Pluralist Design Development.p12. Save the certificate into a known location where you won’t lose track of it. Do not use any characters other than a–z, 0–9, hyphens, underscores, and spaces in the filename.

3. When prompted, specify a password for your p12 certificate, and click OK.
Remember this password. Do not use a blank password.

4. Specify the Administrator password for your computer, and click Allow.

Keep your development p12 certificate (and password information) in a safe place. Next, you will follow similar steps to create the distribution p12 certificate.

Create a Distribution Certificate Signing Request
You've created the development certificate for your testing apps. Now you need to create the distribution certificate for your production apps.

1. If the Keychain Access utility is still running on your computer, choose Keychain Access > Quit Keychain Access to quit the program (leaving the Keychain Access utility running while doing steps 2–4 can cause problems).

2. On the Apple Developer site, click Production (under Certificates) on the left side of the window. Again, you need only one distribution certificate.
3. Click the Plus icon in the upper right corner of the window to start creating a certificate.

5. Reopen the Keychain Access utility, found in the Utilities folder in your Applications folder.

6. Click on the My Certificates category on the left.

7. Click on an empty space in the window on the right to ensure that no existing keys or certificates are selected.


9. For User Email Address, specify a valid email address.
10. For Common Name, enter the name of your company or division. This certificate can be reused for multiple apps that you might create, so you don't need to enter the app name here.

11. Leave the CA Email Address field blank.

12. Select Saved To Disk.

13. Click the Continue button.

14. Specify the name of the certificate signing request and where the file will be saved, and click Save.

15. When the certificate request is created, click Done.

**Generate a Distribution Certificate**

Now that you've created the certificate signing request file, you need to submit it to the Apple Developer site to create the required distribution certificate.

1. In the Apple Developer site, click Continue to advance to the "Generate your certificate" screen. Click Choose File, and double-click the certificate request file you just saved in the previous steps.
2. Click Generate.

Your distribution certificate is added to the Apple Developer site.

3. Click Download, and then download the certificate to a known location.

This distribution certificate is called ios_distribution.cer. This file is required to create your p12 certificate.
Add the Distribution Certificate to the Keychain

1. In the Finder, locate the ios_distribution.cer file you created, and double-click it to launch Keychain Access and install the certificate.

2. To see if the certificate was installed, check the Keychain Access utility. If double-clicking the .cer file does not install the certificate as shown below, use File > Import Items.

![Keychain Access screenshot showing installed certificate](image1)

Export the p12 Distribution Certificate

Once the certificate is added to your keychain, you can use the certificate to sign AEM Mobile apps. However, if you want to use a different computer to sign apps, export the .p12 certificate file.

1. Control-click or right-click the private key associated with your certificate, and choose Export “[name].” Important: Control-click the private key (highlighted in red below), not the certificate.

![Keychain Access screenshot showing exported certificate](image2)
2. Save your key in the Personal Information Exchange (.p12) file format. Specify a name that distinguishes it from the p12 file you created for the development certificate. Do not use any characters other than a–z, 0–9, hyphens, underscores, and spaces in the filename.

![Save As: Pluralist Design Distribution.p12](image)

3. When prompted, specify a password for your p12 certificate, and click OK.

Remember this password. Do not use a blank password. You will need this password later when building your app. (You can use the same password for the distribution and development certificates.)

4. Specify the Administrator password for your computer, and click Allow.

You now have both certificates that you need to create apps for the App Store. Store these files in a safe place. You can now delete the certificate signing request file and the ios_distribution.cer file, as they are no longer needed.

![Pluralist Design Development.p12](image)
![Pluralist Design Distribution.p12](image)
Test your certificates

IT IS A GOOD IDEA to test your certificates at this point, to ensure that they are valid.

1. Open the Keychain Access utility, found in the Utilities folder in your Applications folder.

2. Select the Keys category on the left.  

3. Click the right-pointing arrow next to each private key to reveal the certificate within. The certificate name must begin with the words iPhone Developer or iPhone Distribution. If you don't see these words, you will need to recreate your certificates.

4. Click on each certificate, and view the contents of the pane above the certificate. The phrase “This certificate is valid” must appear in this area when you select the Developer certificate and the Distribution certificate. If you don't see this phrase, you will need to recreate your certificates.

Don't worry that the certificates are called iPhone Developer and iPhone Distribution, even though you are developing an app for the iPad or for both the iPad and iPhone. This is normal.
Create an **App ID**

YOU WILL NEED TO CREATE an App ID for your app. This is a unique identifier required by Apple to identify an app. You will use Apple's Developer site to create an App ID.

A different App ID is required for each app that you create.

1. On the Apple iOS Developer site, go to “Certificates, Identifiers & Profiles.” Sign in with your Apple ID.
2. Click Identifiers, and make sure that App IDs is selected.
3. Click the Plus icon in the upper right corner of the window to create an App ID.
4. Type a description, such as the name of your app.
5. For App Services, leave Game Center and In-App Purchase selected (Apple enables these by default), and do not select iCloud or Passbook. Do not select Data Protection. Select Push Notifications if you want to enable push notifications. Selecting this option is required for Newsstand apps (not recommended).

6. For App ID Prefix (previously called "Bundle Seed ID"), leave the option set at Generate New (if this is the first App ID you've generated) or Use Team ID (for additional App IDs). In other words, don't do anything here unless you really know what you're doing.
7. Specify a Bundle ID. The Bundle ID (or Bundle Identifier) is usually specified in a form called reverse domain name or com.domainname.applicationname. In other words, if you work for Pluralist Design, your website is www.pluralistdesign.com, and your application is called “Pluralist Lookbook,” your App ID would be com.pluralistdesign.pluralistlookbook.

8. Click the Continue button.

9. If the settings look correct, click the Submit button, and then click Done.

The App ID will appear in the list. This same App ID will appear in various screens in later steps in this process. For example, when you build the mobileprovision files for your app, you’ll specify this App ID, and the App ID settings will be included in the mobileprovision files.
Register your test iPads/iPhones with Apple

When you build your app, you will have an opportunity to preview the actual app on one or more iPads and iPhones before submitting the app to the App Store. This preview will include your app icon, app name, and the fully functional app. To preview the app, you will need to provide Apple with a list of UDIDs (sort of like serial numbers) of each iOS device on which you will want to preview your app.

Here is one way to discover the UDID of your iPad:
1. Attach your iPad or iPhone to your Macintosh with a USB cable.
2. Launch iTunes.
3. Select your device in the upper right area of iTunes.
4. In the Summary section, locate the serial number of your iPad.

Adobe Tip
If you have several UDIDs to record, it might be easier to use one of the many free UDID apps available on the App Store. Search the App Store for “UDID” to locate these. Typically, these apps will extract the UDID from your iPad and allow it to be emailed in text form to the address you specify.
5. Click the serial number, and it will change to the UDID.

6. Press Command+C to copy the UDID, and then paste the UDID into a text file or an InDesign file for safekeeping. (You don't select the UDID before copying—just press Command+C and the UDID will be copied to the clipboard).

7. Repeat for each iPad, iPhone, or iPod that you will use to preview your app.

8. Go to the Apple iOS Developer site at bit.ly/OVbhu5. You will need to sign in with your Apple ID.

9. Click Devices on the left side of the window.

10. Click the Plus icon in the upper right corner of the window to add a device.
11. Enter a name and a UDID for one of the devices on which you will want to test your app. This is the information you gathered in step 6, above. To enter the information for another device, click the plus icon. When you have entered a UDID for each device that you need, click the Continue button.
You should now have a list of each device on which you will want to test your app displayed in the Devices section of the Provisioning Portal.
Set up Apple push notifications (optional)

If you enable the Apple Push Notification Service (APN) for your viewer app using the Adobe service, you can indicate to your customers when a new collection is available, and you can send text notifications.

The app icon displays a red badge that indicates the number of new or updated issues.

The Apple process for creating .p12 push certificates is similar to the process for creating developer and distribution .p12 certificates.

Create a Development Push Certificate Signing Request

Apple provides a couple of different ways to start creating a push certificate. You can either create a certificate through the Certificates tab or by changing the settings of the App ID you created. Both options do the same thing. We’ll go through the Certificates tab.

1. Go to the Apple iOS Developer site at bit.ly/OVbhu5. You will need to sign in with your Apple ID.
2. Click Certificates on the left side of the window.
3. Click the Plus icon in the upper right corner of the window to start creating a certificate.
4. In the “What type of certificate do you need?” section, select Apple Push Notification Service SSL (Sandbox). Click Continue.
5. Specify the App ID you created. Make sure that you select the appropriate App ID. Then click Continue.

![Add iOS Certificate](image)

As indicated on the Apple Developer site, you'll now use the Keychain Access utility to create a Certificate Signing Request (CSR).

![Add iOS Certificate](image)

6. Reopen the Keychain Access utility, found in the Utilities folder in your Applications folder.

7. Click the My Certificates category on the left.
8. Click on an empty space in the window on the right to ensure that no existing keys or certificates are selected.

![Keychain Access window with selected options highlighted]


![Keychain Access Certificate Assistant menu with requested options highlighted]

10. For User Email Address, specify a valid email address.

11. For Common Name, enter the name of your app.

12. Leave the CA Email Address field blank.
13. Select Saved To Disk.

14. Specify the name of the certificate signing request and where the file will be saved, and click Save.

15. When the certificate is created, click Done.

Generate a Development Push Certificate

Now that you’ve created the certificate signing request file, you need to submit it to the Apple Developer site to create the required development certificate.

1. In the Apple Developer site, click Continue to advance to the “Generate your certificate” screen.
   
   Click Choose File, and double-click the certificate request file you just saved in the previous steps.
2. Click Generate.

Your distribution certificate is added to the Apple Developer site.
3. Click Done. Click the certificate you just created to view its settings.

![Step-by-Step](image)

4. Click Download, and then download the certificate to a known location.

   This APN development certificate is called "aps_developer.cer." Next, you will create the .p12 certificate for this certificate that you can specify in the On-Demand Services Portal.

Create a p12 Development Push Certificate

1. In the Finder, locate the ios_developer.cer file you created, and double-click it.

2. Double-clicking the certificate launches Keychain Access and installs the certificate. To see if the certificate was installed, click on the Keys category on the left, and then click the right-pointing arrow to open the private key. You should see the certificate added to the private key. If double-clicking the .cer file does not install the certificate as shown below, then launch Keychain Access, and choose File > Import Items to specify the .cer file.

3. Control-click or right-click the private key associated with your certificate, and choose Export “[name].” Important: Control-click the private key (highlighted in red below), not the certificate.
4. Save your key in the Personal Information Exchange (.p12) file format. Specify a name that distinguishes it from the p12 file you created for the development certificate. Do not use any characters other than a–z, 0–9, hyphens, underscores, and spaces in the filename.

5. When prompted, specify a password for your p12 certificate, and click OK.

   Remember this password. Do not use a blank password. You will need this password later when building your app. (You can use the same password that you used for other certificates.)

6. Specify the Administrator password for your computer, and click Allow.

7. Exit the Keychain Access utility.

You now have one of the two p12 push certificates that you need to create iOS apps with push notification enabled. Store this file in a safe place. You can now delete the certificate signing request file and the ios_developer.cer file, as they are no longer needed.
Create a Production Push Certificate Signing Request

Now that you've created the Developer push certificate for the test app, the next step is to create the Production push certificate for the distribution app using essentially the same process.

1. In the Apple iOS Developer site (bit.ly/OVbhu5), click Certificates on the left side of the window.
2. Click the Plus icon in the upper right corner of the window to start creating a certificate.
3. In the “What type of certificate do you need?” section, scroll down under Production and select Apple Push Notification Service SSL (Production). Click Continue.
4. Specify the App ID you created. Be careful to select the right App ID. Then click Continue.

As indicated on the Apple Developer site, you’ll now use the Keychain Access utility to create a Certificate Signing Request (CSR).
5. Reopen the Keychain Access utility, found in the Utilities folder in your Applications folder.
6. Click the My Certificates category on the left.
7. Click on an empty space in the window on the right to ensure that no existing keys or certificates are selected.


9. For User Email Address, specify a valid email address.
10. For Common Name, enter the name of your app.
11. Leave the CA Email Address field blank.
12. Select Saved To Disk.

![certificate assistant screenshot]

13. Specify the name of the certificate signing request and where the file will be saved, and click Save.

14. When the certificate is created, click Done.

**Generate a Production Push Certificate**

Now that you've created the certificate signing request file, you need to submit it to the Apple Developer site to create the required production certificate.

1. In the Apple Developer site, click Continue to advance to the "Generate your certificate" screen.
   - Click Choose File, and double-click the certificate request file you just saved in the previous steps.
2. Click Generate.

Your distribution certificate is added to the Apple Developer site.
3. Click Done. Click the certificate you just created to view its settings.

![Certificate Settings](image)

4. Click Download, and then download the certificate to a known location.

   This APN production certificate is called “aps_production.cer.” Next, you will create the .p12 certificate for this certificate that you can specify in the On-Demand Services Portal.

**Create a p12 Production Push Certificate**

1. In the Finder, locate the ios_production.cer file you created, and double-click it.

2. Double-clicking the certificate launches Keychain Access and installs the certificate. To see if the certificate was installed, click on the Keys category on the left, and then click the right-pointing arrow to open the private key. You should see the certificate added to the private key. If double-clicking the .cer file does not install the certificate as shown below, then launch Keychain Access, and choose File > Import Items to specify the .cer file.

3. Control-click or right-click the private key associated with your certificate, and choose Export “[name].” Important: Control-click the private key (highlighted in red below), not the certificate.
4. Save your key in the Personal Information Exchange (.p12) file format. Specify a name that distinguishes it from the p12 file you created for the push development certificate. Do not use any characters other than a–z, 0–9, hyphens, underscores, and spaces in the filename.

![Image of P12 certificate save dialog]

5. When prompted, specify a password for your p12 certificate, and click OK.

   Remember this password. Do not use a blank password. You will need this password later when specifying your push certificate information in the On-Demand Services Portal. (You can use the same password that you used for other certificates.)

6. Specify the Administrator password for your computer, and click Allow.

You now have both of the p12 push certificates that you need to create iOS apps with push notification enabled. Store these files in a safe place. You can now delete the certificate signing request file and the ios_production.cer file, as they are no longer needed.

When you build your app, you indicate whether push notifications are enabled for that app, but you do not specify the push certificates and passwords. Instead, you use the Push Notifications Service. To do this, sign in to the On-Demand Services Portal (https://aemmobile.adobe.com), click Notifications, and specify the certificates. You use the Push Notifications Service to schedule or send push notifications, send text notifications, and replace expiring push certificates.
Create mobileprovision files

THE MOBILEPROVISION FILES determine which iPads or iPhones can install and use your app. You will create two mobileprovision files: a development mobileprovision file and a distribution mobileprovision file. The development mobileprovision file includes a list of iPad IDs called UDIDs. The distribution mobileprovision file does not include any UDIDs, because anyone who downloads your app from the App Store can use it. Unlike the p12 certificates, which can be reused for multiple apps, the mobileprovision files must be recreated for each new app you design.

The mobileprovision file is tied to the App ID. If you edit the App ID—for example, if you enable push notifications—you must re-create the mobileprovision files to include these changes.

Create a development mobileprovision file

1. In the Apple iOS Developer site (bit.ly/OVbhu5), click Provisioning Profiles on the left side of the window.  
2. Click the Development tab.  
3. Click the Plus icon in the upper right corner of the window.
4. Select iOS App Development, and click Continue.

5. Specify the App ID, and click Continue.
6. Select your iOS development certificate.

7. Specify the devices you want to provision. The development app you create for testing will work only on devices included in the development mobileprovision profile.
8. Specify a name. Include “Development” in the name to distinguish it from the distribution file, but don’t include any special characters (such as asterisks) in the name. You can rename these files after generating them. Then click the Generate button.

9. Click Download, and then download the development mobileprovision file to a known location.

You have now created the development mobileprovision file required to build the development and distribution apps. Store this file in a safe place. Next, you will create the distribution mobileprovision file.
Create a distribution mobileprovision file

To create a distribution mobileprovision file, you do not specify any UDIDs in the Devices section. You simply create the mobileprovision file.

1. In the Apple iOS Developer site (bit.ly/OVbhu5), click Provisioning Profiles on the left side of the window.

2. Click the Distribution tab.

3. Click the Plus icon in the upper right corner of the window.
4. Select App Store under Distribution, and click Continue.

Note that you can also select the Ad Hoc option, which allows you to create a distribution app for testing purposes. Creating an Ad Hoc mobileprovision file is especially useful for testing push notifications in a distribution app. You can create both an App Store mobileprovision file for submitting to the store and an Ad Hoc mobileprovision file for testing apps. These instructions cover only the App Store mobileprovision file.
5. Specify the App ID, and click Continue.

![Screen showing the selection of an App ID]

6. Select your iOS distribution certificate.
7. Specify a name. Include “Distribution” in the name to distinguish it from the development file, but don’t include any special characters (such as asterisks) in the name. You can rename these files after generating them. Then click the Generate button.

8. Click the Download button, and save the .mobileprovision file to your computer.

You have now created the two mobileprovision files required to build the development and distribution apps. Keep these files in a safe place.
Create required assets

When you build your app and submit it to Apple, you will need to include a variety of keywords, descriptive text, icons at various sizes, email addresses, copyright information, etc. It is best to gather all of these items well in advance of your app submission date, since some of them may require input from others in your company.

These assets—along with the .p12 certificates and mobileprovision files—are required when you set up your project, build the app, or complete the app submission process in iTunes Connect.

You can read more about these items in the iTunes Connect Developer Guide, at bit.ly/RMGpiP.

Type the following items in a text file stored somewhere you’ll remember, so that you can copy and paste the pieces out of the text file later when needed:

- **App Name**
  The title that will appear under the app icon on the iPad. Limit to 12 characters if you want it to be completely visible. Depending on the characters (an “i” being narrower than a “w”), you may be able to fit 13 or 14 characters. You specify the App Name when building the app.

- **Keywords for the App Store**
  The number of keywords is unlimited, but there is a total character limit of 100 characters. Enter keywords separated by commas, without a space after the comma. The comma characters count towards the 100-character limit. You specify keywords in iTunes Connect.

- **Description for the App Store**
  This can be anywhere from 10–4,000 characters. You specify keywords in iTunes Connect.

- **Primary and Secondary categories for the App Store**
  The secondary category is optional. Apple occasionally adds new categories. Go to the App Store on your iPad, and tap “Categories” to see the categories you can choose from. You specify categories in iTunes Connect.

ADOBE TIP
To test if your app title will display correctly on the iPad without abbreviation, open Safari on the iPad. Visit any web page, and tap the button at the top of the screen. Tap “Add to Home Screen,” and then enter the text you want for your app title. If the text appears correctly on your home screen, it will display correctly in your app.
Copyright
This should be in the form “2015 Acme Inc.” Apple adds the copyright symbol automatically. You specify the Copyright in iTunes Connect.

Support Email
This is an email address of someone Apple can contact if there are problems with your app. This address is not seen by the public. You specify the contact email in iTunes Connect.

Support URL
The URL of a support website for users having questions about the app. This link will be visible on the App Store. You specify the contact email in iTunes Connect.

Marketing URL (optional)
A website for users to get more information about the app. This link will be visible on the App Store. You specify this URL in iTunes Connect.

SKU number
A unique alphanumeric identifier for this app. You specify the SKU number in iTunes Connect.

Adobe Tip
You can use a Photoshop template to automatically create all the required sizes of application icons from a single Illustrator or Photoshop Smart Object. For details, search for “AEM Mobile asset generator.”
Store the following files with clearly labeled filenames in a separate folder:

- **Application icons of various sizes:**
  For iOS apps, create a 180x180 pixel PNG file. You specify the app icon when building the app. Also create a 1024x1024 version of the app icon that you specify in iTunes Connect. One way to produce these is to create a 1024x1024 pixel PNG file, open it in Photoshop, and use Save for Web to create the smaller PNG files. Do not round the corners of the icon; Apple rounds the corners automatically.

- **Screen shots for the App Store**
  Take a screen shot of the best screens from your app. To take a screen shot on the iPad, just navigate to the screen you want to capture, and press the power and home buttons simultaneously. The screen capture will be saved into the "Camera Roll" album of your Photos app. Save these images as PNG or JPEG files at the sizes of the supported device sizes. To determine the required sizes of screen shots, check iTunes Connect when you create the iTunes app record.

- **Splash screen images**
  When your app is first started on the device, a splash screen appears for approximately three seconds. Create 1536x2048, 2048x1536, 640x960, 640x1136, 1242 x 2208, and 2208 x 1242-pixel PNG images. You specify these images when building the app.

- **Certificates and mobileprovision files**
  Use the same .p12 developer and distribution certificates for all your company apps. Create a set of mobileprovision files and .p12 push certificates (if push is enabled) for each separate app. See the previous sections in this document for details.

- **Product IDs for non-free collections or subscription durations**
  When you publish a collection, each collections requires a Product ID. You can use the same Product ID for different collections. If you're selling a collection through iTunes, you need to use iTunes Connect to create an in-app purchase Product ID and make sure that the same Product ID is associated with the collection. Doing so ties the collection to a store purchase. Free collections do not require in-app purchases to be created. See “Create In-App Purchases for retail collections” section on page 69.

  For iTunes subscription apps, you need to use iTunes Connect to create one or more in-app purchases that identify subscription durations. You specify these Product IDs in the project using the Products & Subscriptions section of the Portal. See “Set up subscription apps” section on page 73.

Use your web browser to search for “writing an App Store description” to locate dozens and dozens of sites that will help you write a concise, compelling App Store description.
Create an **iTunes Connect record** for the app

BEFORE YOU CAN SUBMIT YOUR APP TO APPLE FOR DISTRIBUTION, you create a record of your app in iTunes Connect. This record includes the information that appears on the store for the app, the app languages you want, and the information required to manage the app. Once you set up the iTunes Connect record for the app, you can create in-app purchases and submit the distribution app.

The iTunes Connect Developer Guide at [bit.ly/N9fBUV](http://bit.ly/N9fBUV) explains how to submit files for approval. However, the Developer Guide is written primarily for developers, so it can be somewhat intimidating. We created this guide to provide specific information about submitting apps created with Adobe Digital Publishing Suite.

Apple revises the iTunes Connect interface on occasion. The steps in this guide might be out of date on occasion.

2. Sign in with your Apple ID (not your Adobe ID).
3. Click My Apps.

![Image of iTunes Connect interface showing My Apps](image-url)
4. Click the plus icon and choose New iOS App.

In the screens that follow, you will be prompted to fill in details about your app and various icons and assets. You should have already gathered up all these items, as listed in the “Create required assets” section on page 59.

**App Name:** Enter the full name of your app. This is the “long name” that may be up to 35 characters in length.

**Primary Language:** Choose the language that you will be using to enter your app details for display in the App Store.

**Bundle ID:** Choose the App ID you created earlier from the list.

**Version:** Enter a version for this app. Apple wants you to use the same version number here that you use when building the app, but it isn’t required.
SKU Number: Enter a unique alphanumeric identifier for this app.

5. Click Create.

6. Under Versions, specify the information as follows:

   **App Preview Video and Screenshots:** Click each device type and specify the screenshots that will appear in the App Store. Include one to five 768x1024- or 1536x2048-pixel (portrait) or 1024x768- or 2048x1536-pixel (landscape) screenshots of your app. Include one to five screen shots in the
required sizes if your app supports iPhone. For details, click the Info button next to this option, and then Command-click the Learn More link to open the help document in a different tab.

Name: Enter the name of the app as it will appear in the App Store.

Description: Enter the description for the App Store. If your app includes auto-renewable or free subscriptions, provide this information, and include a link to your privacy policy URL.

Keywords: Enter the keywords for the App Store, separated by commas.

Support URL: Enter the URL of a web page that offers information and support for this app.

Marketing URL: Enter the URL of a web page that offers marketing information for this app. This URL will be visible in the App Store.

Privacy Policy URL: Enter the URL that links to your organization's privacy policy. Privacy policies are required for apps that are Made for Kids, offer auto-renewable in-app purchase or free subscrip-
tions, or as otherwise required by law. Privacy policies are also recommended for apps that collect user data.

App Icon: Choose a 1024x1024-pixel 8-bit PNG image.

Version: Enter using a format such as 1.0, unless this is a revision to an existing app.

Primary Category: Choose the category you want the app to be displayed in on the App Store.

Secondary Category: If desired, choose a secondary category.

Rating: Click Edit, select rating information, and click Done.

Copyright: Enter the copyright information in the form of “2013 Acme Inc.” Apple adds the © symbol automatically when the app is published on the App Store.

Routing App Coverage File: This option is not required. To learn more, see Apple documentation.

Build: Shortly after you use Application Loader to submit the distribution .zip file, a plus sign should appear in the Build section. This is usually the final step in submitting the app. For more information, see the section about submitting apps later in this document.

Newsstand: If you are creating a Newsstand app (not recommended), click the icon to enable Newsstand, and specify the requested information, including cover art and Newsstand categories.

Contact Information: Enter the name, phone number, and email address of a person that Apple can contact if there are problems with the app.
Demo Account: Enter the User Name and Password of a testing account so that Apple can verify in-app purchases or direct entitlement. Specify additional accounts in the Notes field.

Notes: Specify additional information for your app to help the Apple representative during the review process.

Version Release: Indicate whether you want the app to be made available on approval or if you want to delay the release until you’re ready.

7. Click Save to save the version information. If required information or assets are missing, Apple will display error messages. Resolve these issues and click Save again.

Availability Date: Leave this date as is, unless you wish to enter a future date to delay the release of your app until after a particular date.

Price Tier: Select Free or whichever price tier that you desire. Click on View Pricing Matrix to see more pricing details.

Discount For Educational Institutions: Select this option if you want to offer a discount to educational institutions.


Specific Stores: Select this option if you want to limit the app to be available only in certain countries.

8. Click Save.

9. If your app includes in-app purchases for collections or subscriptions, click the In-App Purchases tab and specify the appropriate information. For details, see “Create In-App Purchases for retail collections” section on page 69 and “Set up subscription apps” section on page 73.

ADOBE TIP

One reason to limit the availability of your app to certain countries is because Apple only collects sales tax in some countries. If you want Apple to take care of all the sales tax hassle, limit your app to sell only in the countries in which Apple collects sales tax.
Create **In-App Purchases** for retail collections

IF YOU WANT TO SELL COLLECTIONS IN YOUR APP, you need to use iTunes Connect to create in-app purchases for retail (non-free) collections. You can use the same Product ID for multiple collections. For example, you can create a “Spring” Product ID that entitles the user to the March, April, and May collections.

You need to submit each In-App Purchase to Apple for approval. (Apple does not require Product IDs for free collections to be approved.) For each Product ID, it’s important to specify the same Product ID in the AEM Mobile project that you create when setting up the In-App Purchase in iTunes Connect.

***Note:*** Apple reviews the contents of the In-App Purchase issue at its sole discretion. Many publishers submit multiple In-App Purchases without publishing the corresponding content, and Apple has approved. In rare cases, Apple might require you to publish content before approval.

2. On the iTunes Connect main page, click My Apps.
3. If you have already created the app record in iTunes Connect, click it. If you have not yet created the app record, see “Create an iTunes Connect record for the app” section on page 62, and then return to this section.

4. Click In-App Purchases.
   If the Manage In-App Purchases option does not appear, you have not signed up for a retail account by adding your bank info and tax info.

5. Click Create New to create a new in-app purchase.

6. Click the “Non-Consumable” Select button.

7. Specify the pricing and language of your issue.
   - Under Details, specify a Reference Name and Product ID. This Product ID must be identical to the Product ID you use in project settings. Use the "com.company.app.idname" format, such as "com.sportsmag.kayaking.2013january." For best results, use all lowercase letters—avoid extended characters.
   - Add a language.
   - Under Pricing and Availability, select “Cleared for Sale” so that you can test the app. Choose the price for your issue.
• Under Screenshot For Review, specify an image when ready to submit your in-app purchase. Apple will not review the in-app purchase until you include the screenshot.

8. Click Save.

If you did not specify a screen shot, your status is set to Waiting For Screenshot. If you specify the screen shot, you can select the in-app purchase and submit it for review.
9. In the AEM Mobile project, use the Products & Subscriptions section to create a Product ID and associate it with one or more collections. Edit the Product ID to turn off “Free Product.” (See adobe.ly/1MvVMK8 for details.)

If your app has not been approved, your first In-App Purchases must be submitted with a new app version. Select them from the In-App Purchases section of the Version Details page in iTunes Connect, and then click Ready to Upload Binary.

You can submit multiple in-app purchases to Apple for approval. Getting in-app purchases approved in advance helps avoid delays in publishing your content.
Set up subscription apps

IF YOU SET UP A SUBSCRIPTION APP, you give users access to content in your viewer for a specified period of time. Use the Products & Subscriptions section of the Portal to create Subscription IDs that match the Product IDs specified in iTunes Connect.

With AEM Mobile, you can offer two types of subscriptions: Standard or All Access.

**Standard** Standard subscriptions entitle subscribers to all content in which the availability date falls between the start date and end date of their subscription duration. If users stop subscribing, they still have access to that downloaded content.

**All Access** All Access subscriptions entitle subscribers to all content in the app while they are subscribers, regardless of availability date. If users stop subscribing, they are no longer entitled to any non-free content. An All Access subscription is similar to a movie or music subscription service.

If you create a subscription app with retail (paid) content, you must set up In-App Purchases for collections you want to sell, and you must set up an In-App Purchase for the different subscription duration options (3 month, 6 month, 1 year, and so forth). (See [adobe.ly/1MvVMK8](http://adobe.ly/1MvVMK8) for details.)

For each duration, you specify a Product ID. A subscription Product ID should be in the "com.company.app.duration" format. For each subscription Product ID, it's important to specify the same Product ID (also called a Subscription ID) in the AEM Mobile project that you create when setting up the In-App Purchase in iTunes Connect.

Additional information about subscriptions:

- Your customers have the choice of purchasing individual collections or purchasing a subscription. When customers first subscribe, they're entitled to the Product ID—free or retail—with the most recent Availability Date.

- For Standard subscriptions, the Availability Date setting in the subscription Product ID determines which content subscribers are entitled to. For best results, edit the Availability Date settings of Product IDs using a consistent date, such as the first day of each month.
Generate or view a shared secret

Use the In App Purchase process on iTunes Connect to set up Apple subscriptions. When you create a subscription, you generate or view a shared secret, which is a unique code that lets the app communicate with in-app purchase receipts.


2. If you have not already done so, click “Contracts, Tax, and Banking” and specify the necessary information.

3. Click My Apps.

4. In the My Apps page, click the application in which you want to set up subscriptions

5. Click In-App Purchases.

6. Click “View or Generate a Shared Secret.” (If the shared secret appears, do NOT generate a new one. Use the same shared secret for all your apps.)

7. If a Shared Secret section does not appear on the In-App Purchases page, it means you have not filled out the necessary banking information or you have not accepted the terms of agreement. Click Go Back, and then click the “Contracts, Tax, and Banking” option.
8. Copy your shared secret code. In the On-Demand Portal, edit project settings and use the “Store” tab to specify your shared secret.

![Edit Project DPS Overlays](image)

Again, once you generate a shared secret, do not re-generate it. Use the same shared secret code for all your apps.

**Create an In-App Purchase for a retail subscription**

Apple calls a retail subscription an “Auto-Renewable Subscription.”

1. Go to iTunes Connect > My Apps, and click your app.
2. Click In-App Purchases.
3. Click Create New.
4. In the Create New In App Purchase page, select “Auto-Renewable Subscriptions.”
5. Follow the prompts to fill out the rest of the subscription information.

You can create multiple durations. For each duration, specify a Product ID. We recommend that you use the “com.company.app.duration” format, such as “com.sportspub.kayaker.1year.” You specify the subscription Product IDs in the Products & Subscriptions section of the On-Demand Portal.
In iTunes Connect, you can associate only one duration (such as 3 Months) with a Product ID within the same family. If you want to use the same duration to offer both a Standard and All Access subscriptions, create a second subscription family in iTunes Connect.

6. In the Products & Subscriptions section of the On-Demand Portal, create or edit the Subscription ID. (See [adobe.ly/1MvVMK8](http://adobe.ly/1MvVMK8) for details.)

   For best results, use a consistent Availability Date for your Standard subscription Product IDs, such as the first of every month. The Availability Date determines whether the associated collection is part of a Standard subscription duration.
If you have not already done so, use the in-app purchase method in iTunes Connect to assign a price and Product ID to each duration. Customers can then pay for individual collections or for a subscription.
Build and sign the development app

AFTER YOU CREATE the necessary files and certificates, you're ready to create your app using the Apps section of the On-Demand Portal.

See adobe.ly/1D6U4fi for details on creating the app.

See adobe.ly/1JqKGEG for details on signing the app.
Test the development app

THE NEXT STEP is to test the development app to make sure that everything works as designed. In order to complete the steps below, the iPad on which you wish to test the development app must be one that you regularly sync with iTunes on your computer.

Test an app with free content

1. Open iTunes, and drag the signed developer ipa file into your iTunes Library.

2. Connect your iPad, and sync the device.

3. The app should appear on your iPad. Test the app to ensure that it works as desired.

4. If the app works as designed, continue on in this guide to submit the distribution app to the App Store.
Test an app with retail content

While testing the app, you can create a test user that lets you download retail content using the development app without actually paying for the issue. This sandbox testing process is explained in more detail in the “Managing Your In App Purchases” section of the iTunes Connect Developer Guide.

You cannot test retail content in a preflight app.

1. Create Product IDs with “Free Content” deselected and associate them with collections.

   This product ID must be identical to the product ID you specify when submitting the In App Purchase.

2. Using iTunes Connect, create a test user.

   A test user lets you test the process of purchasing an issue without actually having to purchase the issue. To create a test user, go to the main page of iTunes Connect, and click Users and Roles. Click Sandbox Tester. Click the plus sign, and then specify information. Use a valid email address that you can use to verify. Remember the email address and password. You’ll need this information to “purchase” your issue in the sandbox environment.

3. On your iPad, sign out of your account. Tap Settings, tap Store, tap your Apple ID, and tap Sign Out.

   Sign out to make sure that you don’t actually use your real account to purchase the issue.

   Do not sign in using your test user—remain signed out. Otherwise, your test user will treated like a normal user, and you’ll be prompted for your credit card information.
4. Open iTunes, and drag the .ipa file into your iTunes Library.

5. Connect your iPad or iPhone, and sync the device.

   The app should appear on your iPad.

6. When notified of a new issue, follow the prompts, and sign in using your test user account information. “Purchase” the collection or subscription and view retail content.

   If you're prompted for credit card information, exit the viewer app, and sign out of your account.
   Start the viewer again, do not sign in, and then download the issue using the test account.

7. If the app works as designed, continue on in this guide to submit the distribution app to the App Store.
Create and sign the distribution app

When you have determined that your development app works like you want it to on your iOS devices, it is time to sign the Distribution app. This is the .ipa file that you will upload to Apple using Application Loader.

Using the AEM Mobile Signing Tool, sign your downloaded app. Make sure that you use your distribution certificate and mobileprovision file to sign the app.

For details about signing apps, see adobe.ly/1JqKGEG.
Submit your app to the App Store

NOW THAT YOUR APP IS BUILT and tested, the final step is to submit the app to the App Store. To do this, you will use the iTunes Connect Web portal to enter various details about your app, and then upload the Distribution app to Apple for approval.

The iTunes Connect Developer Guide at bit.ly/N9fBUV explains how to submit files for approval. However, the Developer Guide is written primarily for developers, so it can be somewhat intimidating. We created this guide to provide specific information about submitting apps created with Adobe Digital Publishing Suite.

2. Make sure that you have used iTunes Connect to create a record of your app. If you are submitting a new version of the app, make sure that you create a new version.
3. If you have not already downloaded the Application Loader app, click the link to Application Loader in the Build section. (If you downloaded Xcode, Application Loader was also downloaded. However, make sure that you have the most recent version of Application Loader installed.)
4. Start Application Loader. Click Deliver Your App, and follow the prompts to upload the Distribution app (.ipa file) that was created and signed using the distribution certificate.
5. In iTunes Connect, click on My Apps, and click your app.
6. Scroll down to the Build section, and click the plus icon. Note that it might take a few minutes for the binary file submitted with Application Loader to register with iTunes Connect.
7. Specify the application you submitted.

8. Click Save.

9. Answer the encryption questions. Experience Manager Mobile does not encrypt apps.

10. Click Submit for Review.

If you enabled secure content in your app, indicate that your app includes encryption and follow the Apple guidelines. For details, see http://adobe.ly/1hiafVc.

Once you successfully upload your app to Apple, the status of the app in iTunes Connect should change to “Waiting for Review.”

If you are using an older version of Application Loader, you might get an error message such as “the dwarfdump binary must exist” or “com.apple.transporter.util.StreamUtil” when you upload the .zip file. You can solve this problem by downloading and installing the most recent version of Application Loader. Apple requires Mac OS X 10.7 or later to use Application Loader.
Wait for approval from Apple

Apple will notify you via email when your app is accepted or rejected. You can use iTunes Connect to track the status. For details, see the Apple Developer Guide.

If you notice an error in your app after you have submitted it, but before it is approved, you can submit a new binary while your app is still pending. Go the My Apps section of iTunes Connect, select the app, and then click Reject Binary. Then submit a new version of the .zip file in Application Loader, wait a few minutes, and select that file in the Build section of the app details in iTunes Connect.

Update an app

You may want to revise or update an Experience Manager Mobile app that you previously submitted to the App Store. After you build and test the new or edited development app, download and sign a distribution version of the app.

In iTunes Connect, click My Apps, click the app you want to update, and then click Add Version. Use Application Loader to upload the .ipa file, and then submit the version in iTunes Connect.
iOS publishing troubleshooting

Cannot load or view the developer viewer on your iPad or iPhone
Make sure that you included the device's UDID in the developer mobileprovision file. Only provisioned iPads or iPhones can view developer viewer content.

Blank password not accepted
When prompted for a password while creating your p12 certificates, do not use a blank password. Type a real password.

Version number mismatch
After you submit your app, you might get a warning that the version numbers do not match. Apple is letting you know that the version number you specified in iTunes Connect does not match the auto-generated version number in your app. You can ignore this warning. Apple will not reject the app for this reason.

Error: Identity of iPhone developer doesn’t match any identity in any profile
This error occurs when files are generated under different profile names. For best results, make sure that the same person uses the same profile to create the certificates on the same computer.

“the dwarfdump binary must exist” or “com.apple.transporter.util.StreamUtil" error when you upload the .zip file
This error can occur when you are using an older version of the Application Loader. You can solve this problem by downloading and installing the most recent version of the Application Loader.

The “Subscribe” button does not work in development app
Make sure that you have specified the shared secret information in project settings. Also make sure that you have specified the correct In-App Purchase Product ID information in the On-Demand Portal.
Certificates expired or set to expire
If your development and distribution certificates expire, users can continue to use your app. However, you must edit or revoke and rebuild your certificates before you update your app. When you revoke a certificate, follow the same steps described in this guide to create a new certificate.

An expired push production certificate can affect only push itself, not the ability to use the app. When you sign in to the Push Notifications Service, you’ll be notified if your push certificate is expired or is set to expire within 30 days. Use the Apple Developer site to build new push certificate files, and then use the Push Notifications Service to specify the new .p12.

Certificate is invalid due to missing private keys
Certificates might be invalid because the corresponding private key is not in your keychain. Try to restore your missing private keys from a developer profile backup, as described in “Exporting and Importing Certificates and Provisioning Profiles.” If you cannot retrieve your private keys from another Mac, refer to “Re-Creating Certificates and Updating Related Provisioning Profiles” to re-create all your certificates.
You can perform these steps for one or more invalid certificates.

Your certificates have expired
You cannot renew expired certificates. Remove (revoke) and create new ones.

If your Developer ID certificates expire, users can still download and run apps that were signed with these certificates. However, you will need new certificates to update and create new apps.

You’re Missing Signing Certificates
Your signing certificates may be missing from your keychain because you never requested them or because you moved to a Mac on which you haven’t developed apps before.

If you never requested your certificates (there are none in your keychain). Follow the steps to create certificates as described in this guide.

If you moved to a new Mac, export your certificates as a developer profile file on the Mac you first requested the certificates from, and then import them on your new Mac. See “Exporting and Importing Certificates and Provisioning Profiles” in the Apple App Distribution Guide.

If you no longer have access to the other Mac or user account and you did not keep a backup of your certificates, see “Re-Creating Certificates and Updating Related Provisioning Profiles” in the Apple App
Distribution Guide.