

# iOS

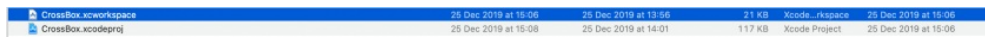
## Building Requirements

To build an iOS app:

- macOS latest version
- Xcode latest version
- [Apple Developer Account](#)
- Cocoapods

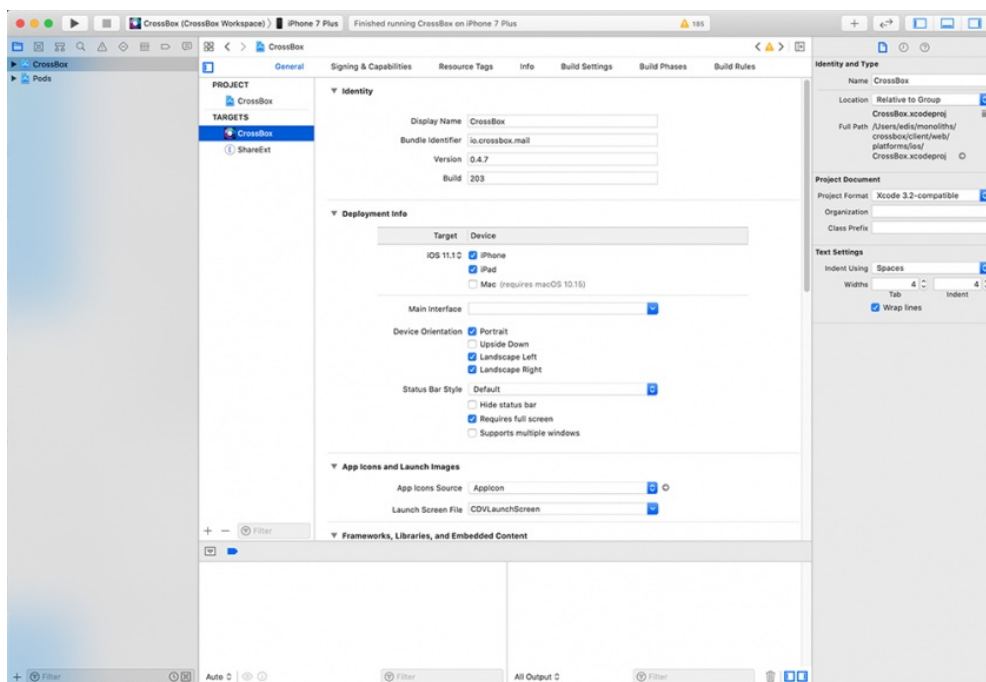
## Open the project

Make sure you open the **.xcworkspace** and **not** .xcodeproj

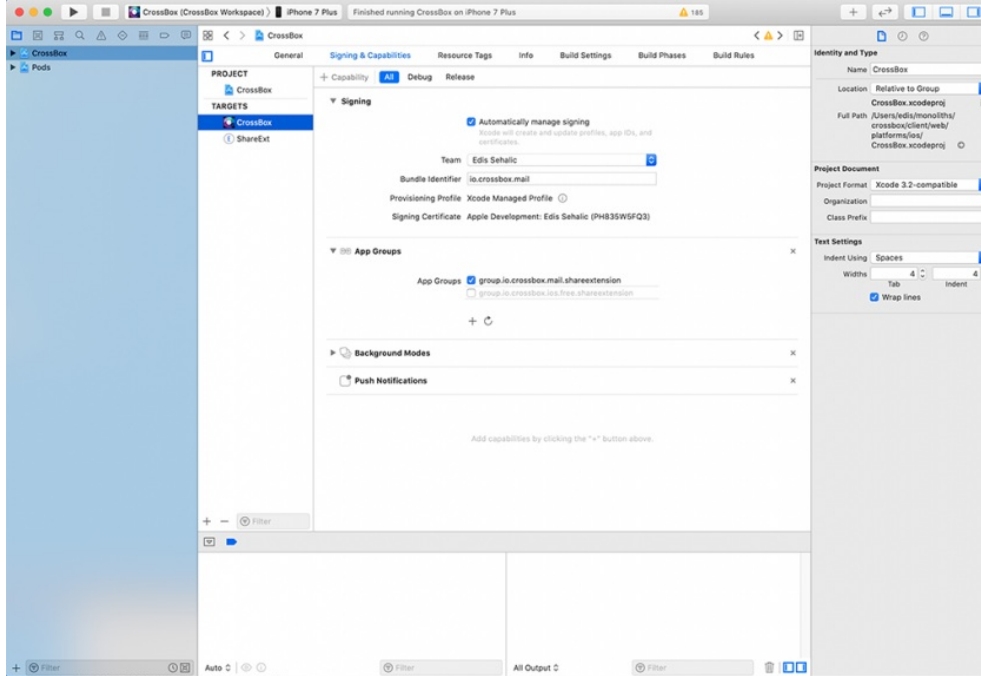


## Certificates and capabilities

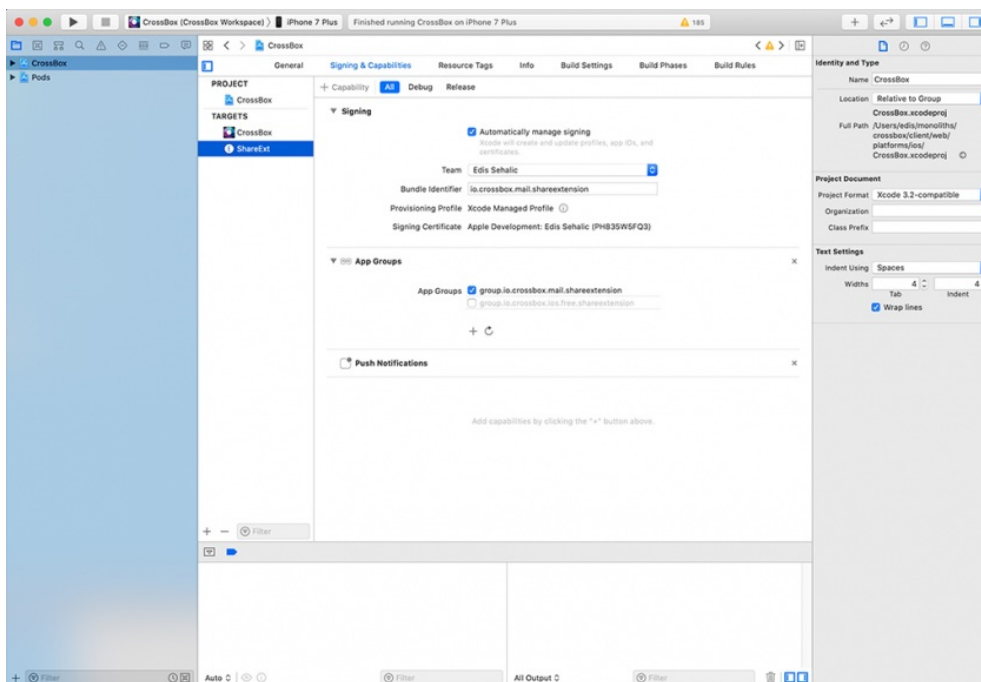
- Click on the First Project in the left side nav
- Go to **"Signing and Capabilities"**
- Add Capability **"Push notifications"**
- Add Capability **"App Groups"**
- **Turn off** automatic signing for both Debug and Release
- Then **turn on** automatic signing for both Debug and Release and select your developer account
- Add a group with the name **group.my.identifier.shareextension** (replace my.identifier with com.yourdomain.app):



- Click on the First Project in the left side nav
- Go to **"Signing and Capabilities"**:



- Add Capability "**App Groups**"
- **Turn off** automatic signing for both Debug and Release
- Then **turn on** automatic signing for both Debug and Release and select your developer account
- Add a group with the name **group.my.identifier.shareextension** (replace my.identifier with com.yourdomain.app):



# Push Notifications Certificates

- Go to <https://developer.apple.com/> and go to **Identifiers**

## Certificates, Identifiers & Profiles

Identifiers	NAME	IDENTIFIER	PLATFORM
	XC io crossbox mail	io.crossbox.mail	iOS
	XC io crossbox mail shareextension	io.crossbox.mail.shareextension	iOS

- Select your app identifier and find **Push notifications** on the list:

## Edit your App ID Configuration

Remove Save

<input type="checkbox"/>		HealthKit	
<input type="checkbox"/>		HomeKit	
<input type="checkbox"/>		Hotspot	
<input type="checkbox"/>		iCloud	<input type="checkbox"/> Include CloudKit support (requires Xcode 6) <input type="checkbox"/> Compatible with Xcode 5 <span>Configure</span>
<input checked="" type="checkbox"/>		In-App Purchase	
<input type="checkbox"/>		Inter-App Audio	
<input type="checkbox"/>		Low Latency HLS	
<input type="checkbox"/>		Mac (Supported only on: macOS)	<span>Configure</span>
<input type="checkbox"/>		Multipath	
<input type="checkbox"/>		Network Extensions	
<input type="checkbox"/>		NFC Tag Reading	
<input type="checkbox"/>		Personal VPN	
<input checked="" type="checkbox"/>		Push Notifications	<span>Edit</span> Certificates (2)
<input type="checkbox"/>		Sign in with Apple	<span>Configure</span>

- Create both certificates and **download them**:

## Apple Push Notification service SSL Certificates

To configure push notifications for this App ID, a Client SSL Certificate that allows your notification server to connect to the Apple Push Notification Service is required. Each App ID requires its own Client SSL Certificate. Manage and generate your certificates below.

### Development SSL Certificate

Name: Apple Development iOS Push Services: io.crossbox.mail  
Type: APNs Development iOS  
Expires: 2020/12/17

Revoke Download

Create an additional certificate to use for this App ID.

Create Certificate

### Production SSL Certificate

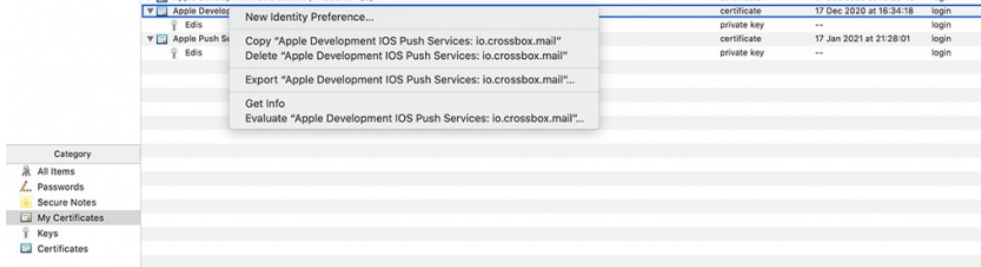
Name: Apple Push Services: io.crossbox.mail  
Type: Apple Push Services  
Expires: 2021/01/17

Revoke Download

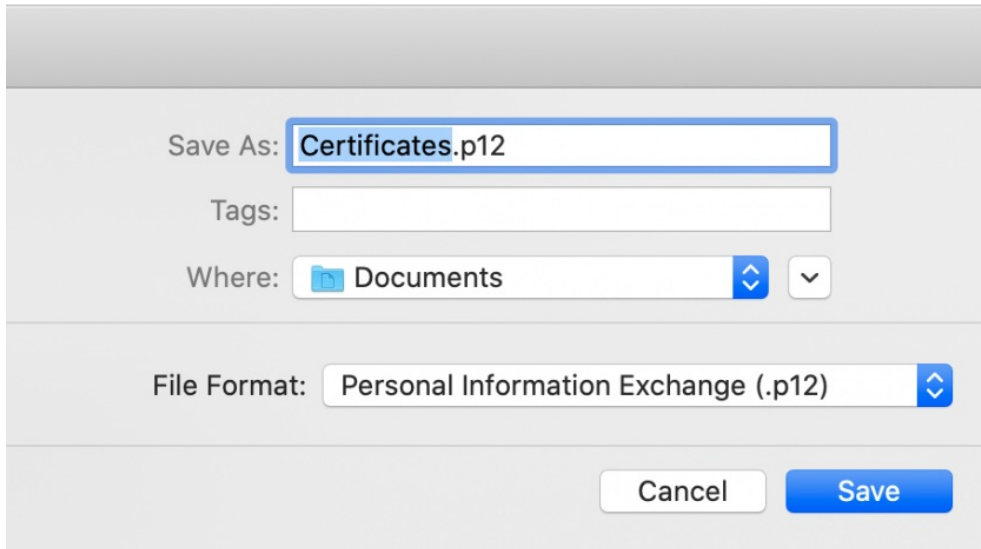
Create an additional certificate to use for this App ID.

Create Certificate

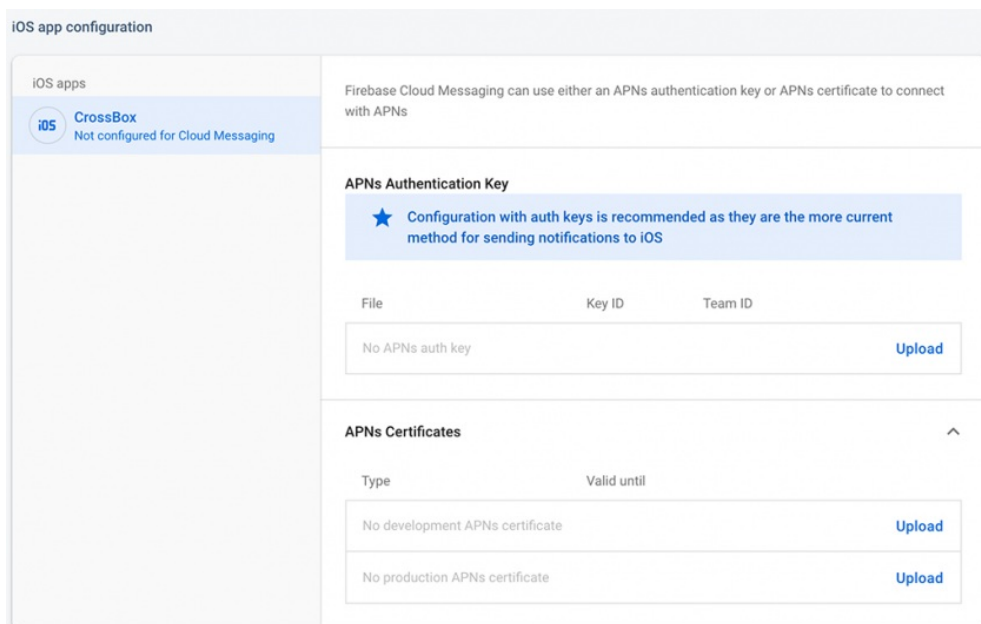
- Double click on the downloaded certificates so they are loaded in the Keychain Access
- Open **Keychain Access**
- Go to **My Certificates**
- Find both certificates:



- Click **Export** and choose **.p12** as file format:



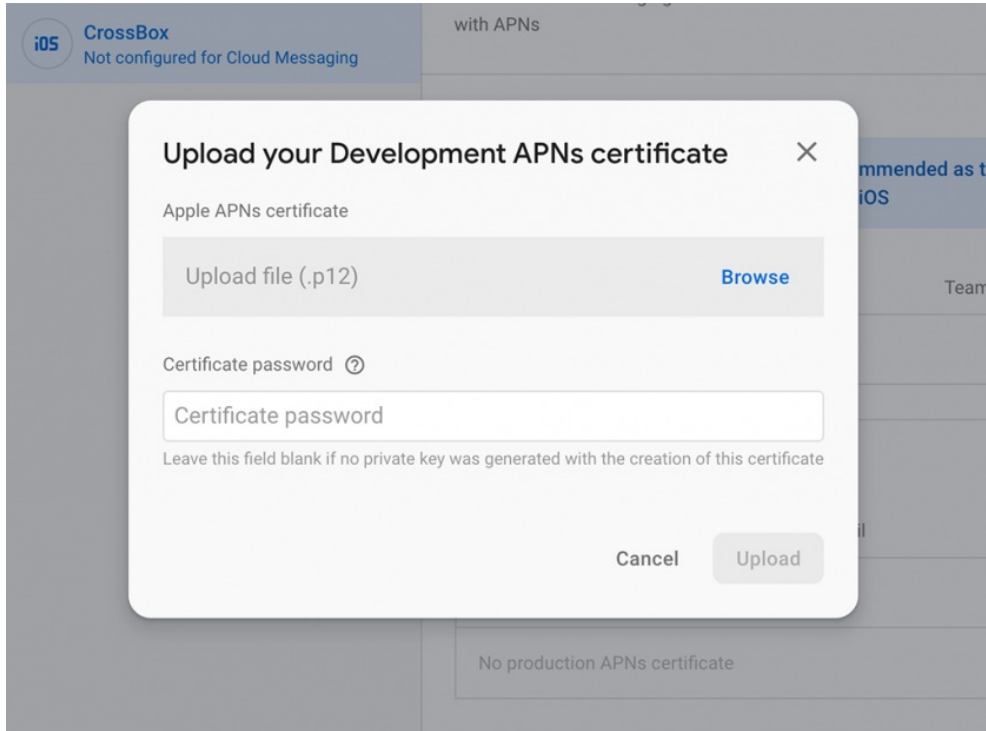
- Make sure you know which certificate is for Development and Production
- Go to Firebase console and go to the project you previously created.
- Open **Cloud Settings**:



- **Upload Production and Development certificates** accordingly:

## APNs Certificates

Type	Valid until
No development APNs certificate	<a href="#">Upload</a>
No production APNs certificate	<a href="#">Upload</a>



## Submission and Testing

- Run the project on the simulator and try logging in
- Use the app and test out if the project has been properly configured
- Open Terminal.app and navigate to the project folder with "cd" command
- Run the command:

```
sh prepare-production.sh
```

**This script prepares the needed libraries for production.** You will no longer be able to run the app in the simulator/test device unless you run:

```
sh prepare-test.sh
```

to revert back libraries.

- Go to <https://appstoreconnect.apple.com/> and create a new app:



My Apps



App Analytics



Sales and Trends



Users and Access



Resources and  
Help

## New App

Platforms ?

iOS  tvOS

Name ?

Primary Language ?

Bundle ID ?

Register a new bundle ID in [Certificates, Identifiers & Profiles](#).

SKU ?

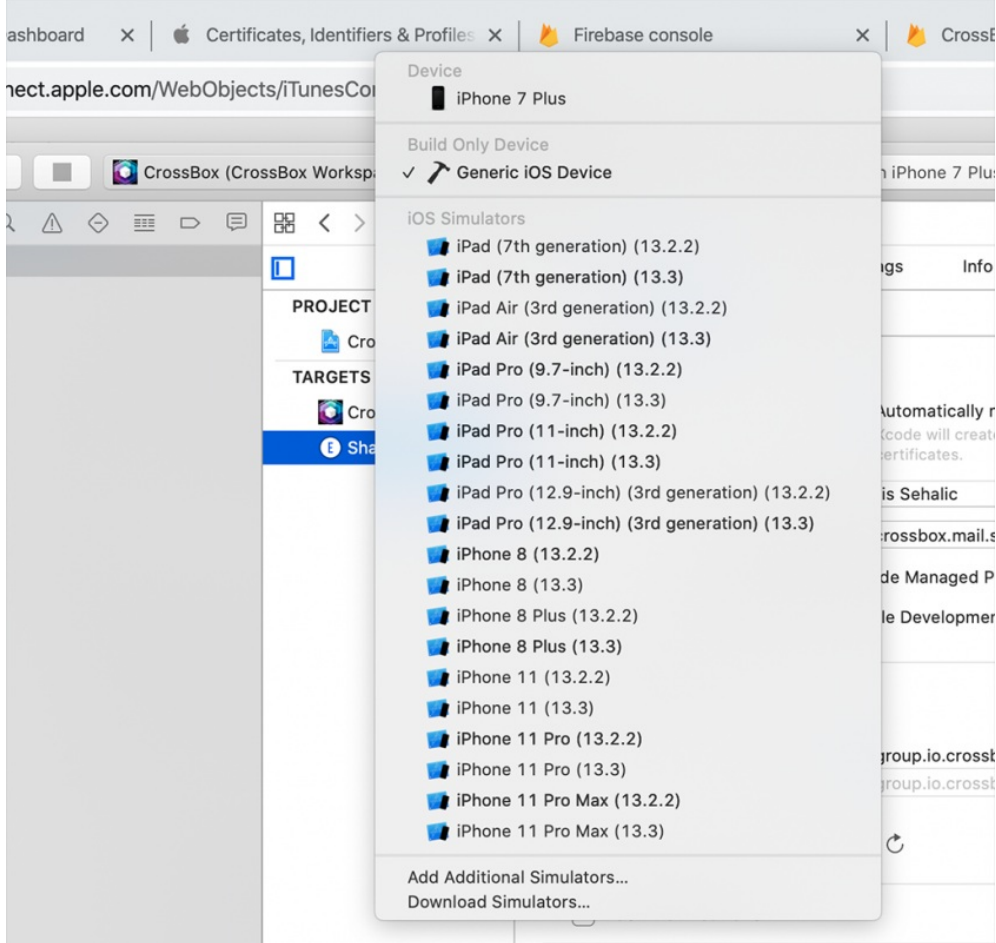
User Access ?

Limited Access  Full Access

Cancel

Create

- Go to XCode and select target build '**Generic iOS Device**'



## Distribution

- Make sure you have **distribution** certificates for your account (you can check if you have them here: <https://developer.apple.com/account/resources/certificates/list>)

- If the list is empty, create one:



- Pick **"Apple Distribution"**

- Apple Distribution  
Sign your apps for submission to the App Store or for Ad Hoc distribution. For use with Xcode 11 or later.  
Maximum number of certificates generated

- Then, go to <https://developer.apple.com/account/resources/profiles/list>

Click + and add the profile for your identifiers your\_app\_id.shareextension and your\_app\_id for the app store:

- Distribution**
- Ad Hoc  
Create a distribution provisioning profile to install your app on a limited number of registered devices.
  - tvOS Ad Hoc  
Create a distribution provisioning profile to install your app on a limited number of registered tvOS devices.
  - App Store  
Create a distribution provisioning profile to submit your app to the App Store.
  - tvOS App Store  
Create a distribution provisioning profile to submit your tvOS app to the App Store.
  - Mac App Store  
Create a distribution provisioning profile to submit your app to the Mac App Store.
  - Developer ID  
Create a Developer ID provisioning profile to use Apple services with your Developer ID signed applications.

- Name your profile so you differentiate it easily by adding a suffix "distribution"  
- Click continue to finish the wizard

- Make sure you have profiles for both identifiers

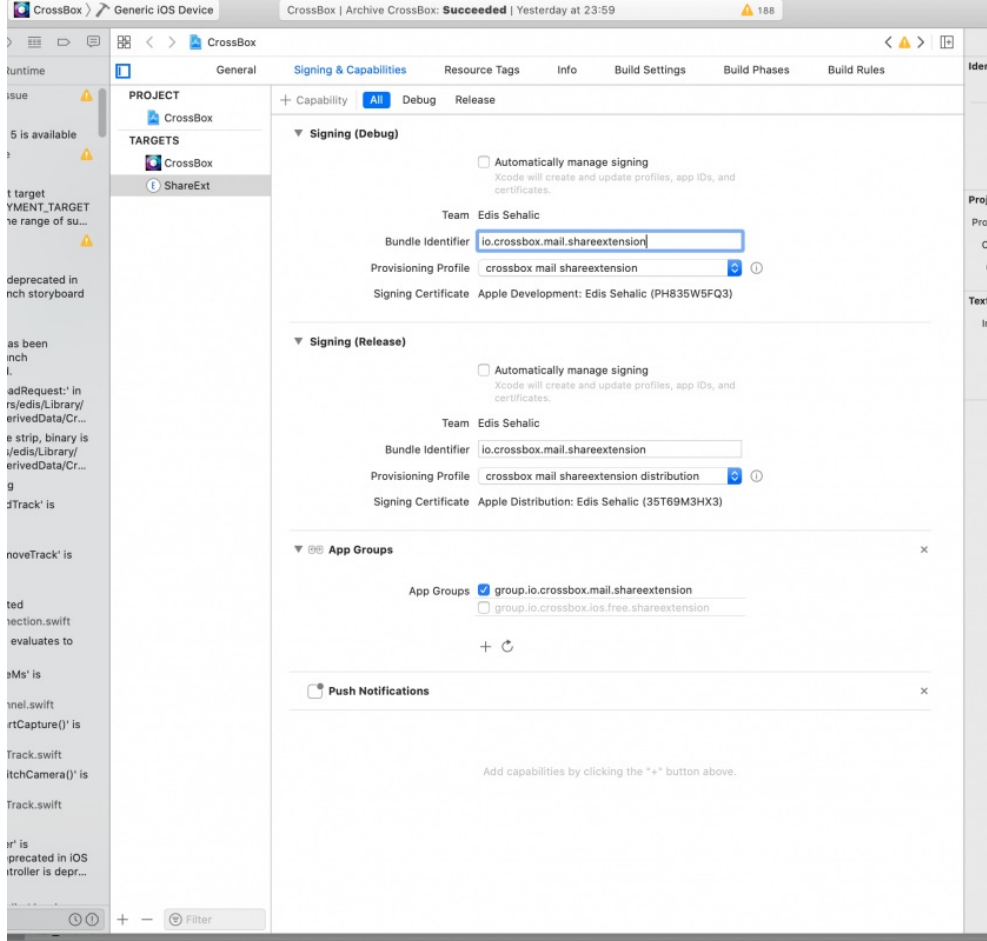
Once you have this list of profiles you can proceed with setting up the signing settings

NAME	PLATFORM	TYPE	EXPIRATION
crossbox mail	IOS	Development	Invalid
crossbox mail distribution	IOS	App Store	2020/12/26
crossbox mail shareextension	IOS	Development	Invalid
crossbox mail shareextension distribution	IOS	App Store	2020/12/26

You will have to configure your **Debug** and **Release** profiles to match. They should look like this:

The screenshot shows the Xcode 'Signing & Capabilities' settings for a project named 'CrossBox'. The interface is divided into several sections:

- General:** Contains tabs for 'Capability', 'All', 'Debug', and 'Release'.
- PROJECT:** Lists 'CrossBox' and 'ShareExt'.
- TARGETS:** Lists 'CrossBox' and 'ShareExt'.
- Signing (Debug):**
  - Automatically manage signing
  - Team: Edis Sehalic
  - Bundle Identifier: io.crossbox.mail
  - Provisioning Profile: crossbox mail
  - Signing Certificate: Apple Development: Edis Sehalic (PH835W5FQ3)
- Signing (Release):**
  - Automatically manage signing
  - Team: Edis Sehalic
  - Bundle Identifier: io.crossbox.mail
  - Provisioning Profile: crossbox mail distribution
  - Signing Certificate: Apple Distribution: Edis Sehalic (35T69M3HX3)
- App Groups:**
  - group.io.crossbox.mail.shareextension
  - group.io.crossbox.ios.free.shareextension
- Background Modes:**
  - Audio, AirPlay, and Picture in Picture
  - Location updates
  - Voice over IP
  - External accessory communication
  - Uses Bluetooth LE accessories
  - Acts as a Bluetooth LE accessory
  - Background fetch
  - Remote notifications
  - Background processing
- Push Notifications:**

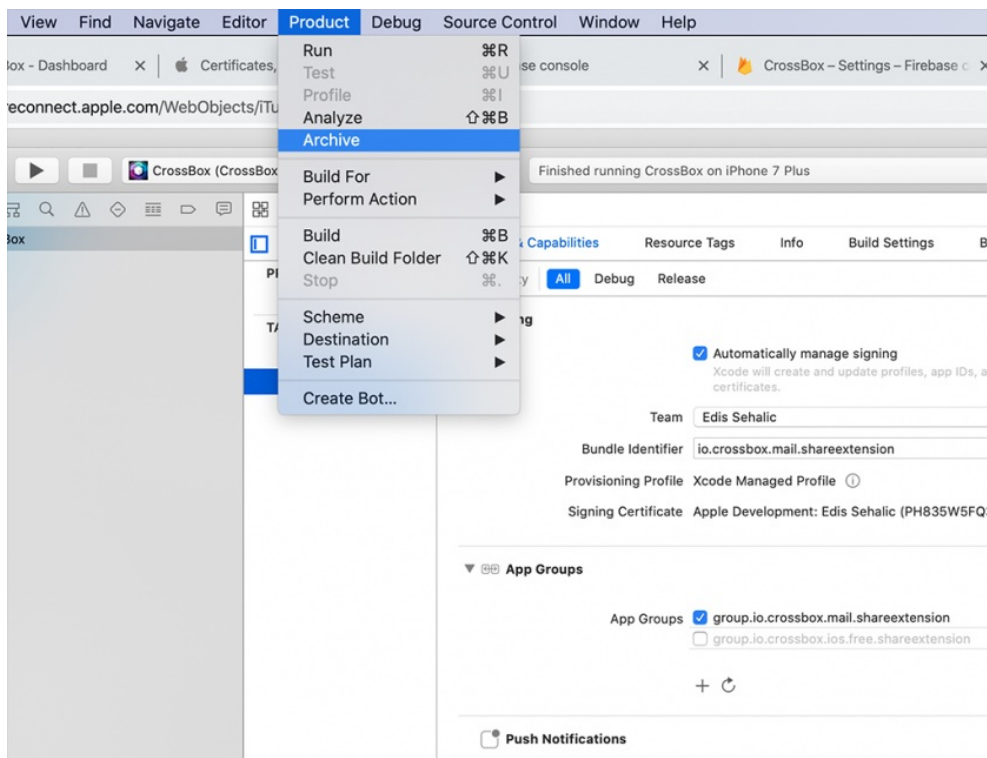


Notice the Signing certificate for Debug and Release: Debug(Apple**Development**)  
Release(Apple **Distribution**)

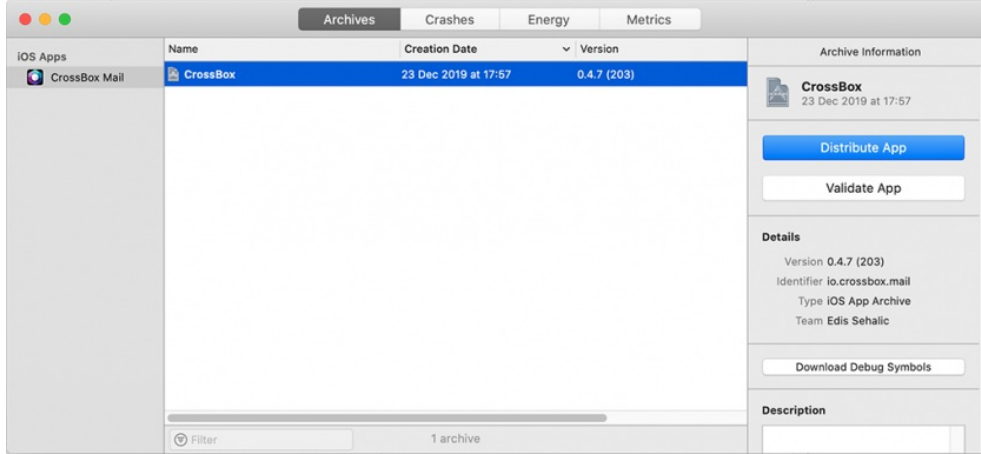
**Both targets must have proper certificates configured**

XCode will complain if the certificates are not properly configured. If you need help with that, you can contact us via Tickets or live Chat.

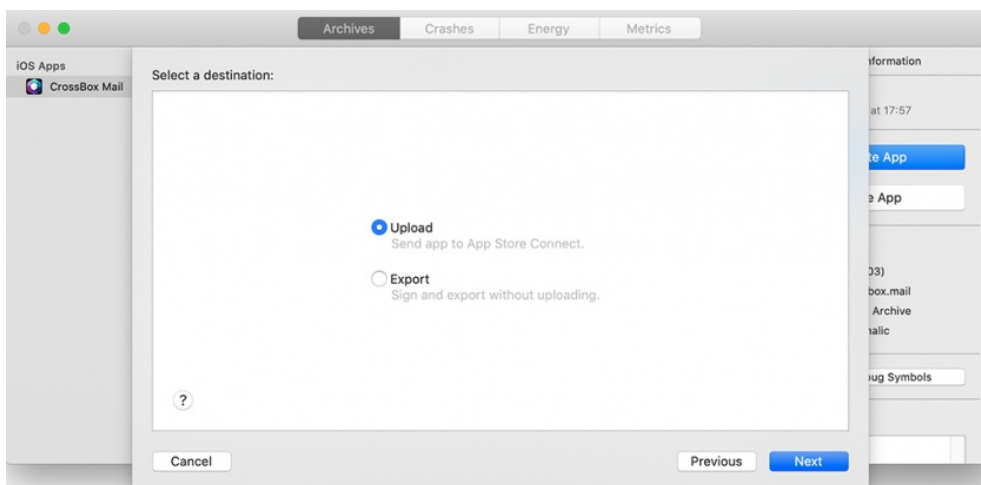
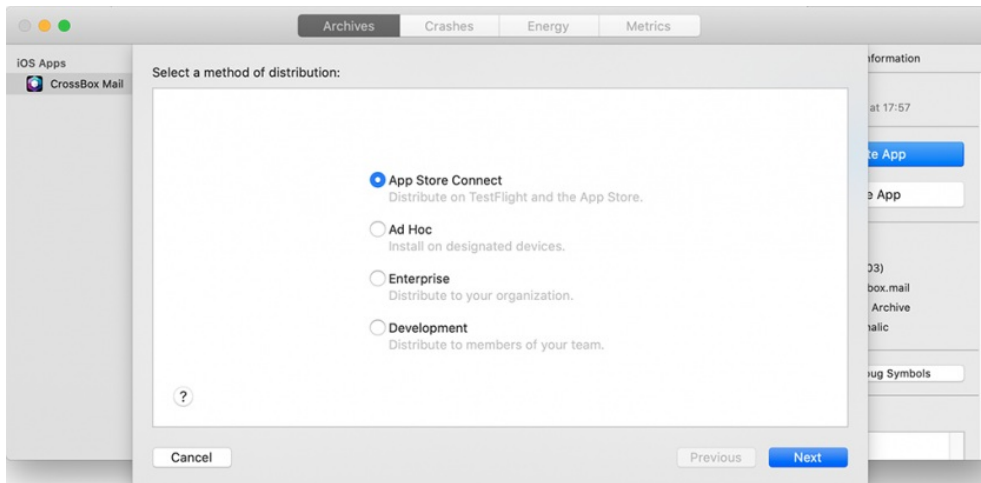
Then go to **Product > Archive**



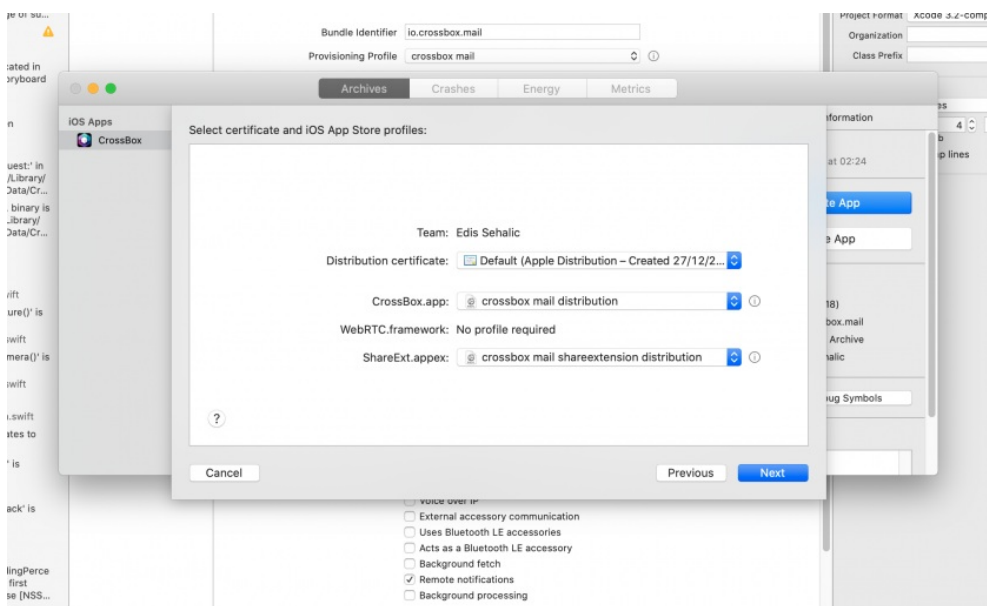
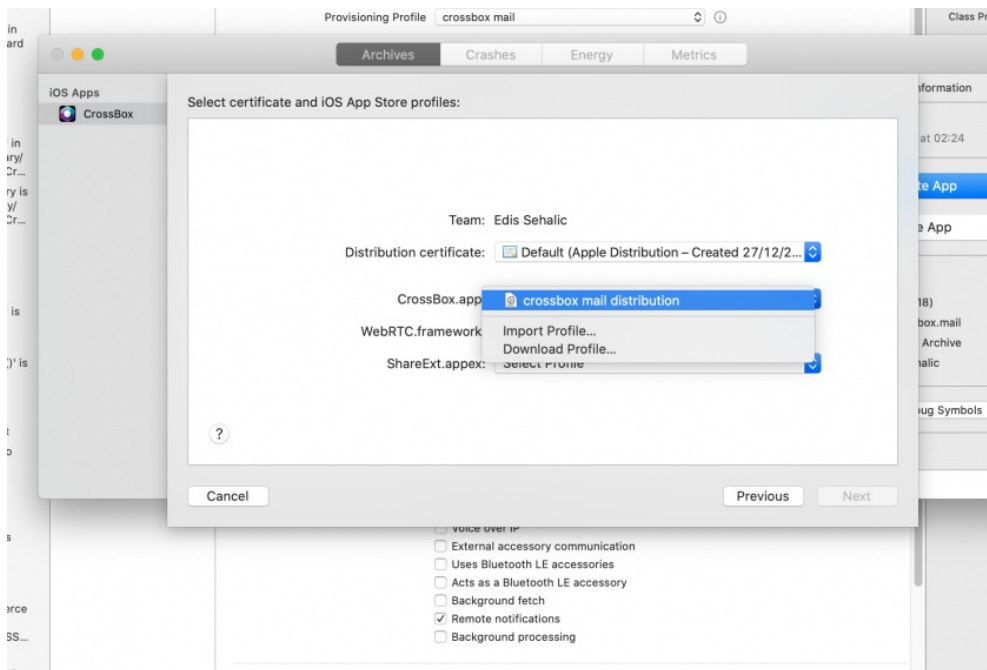
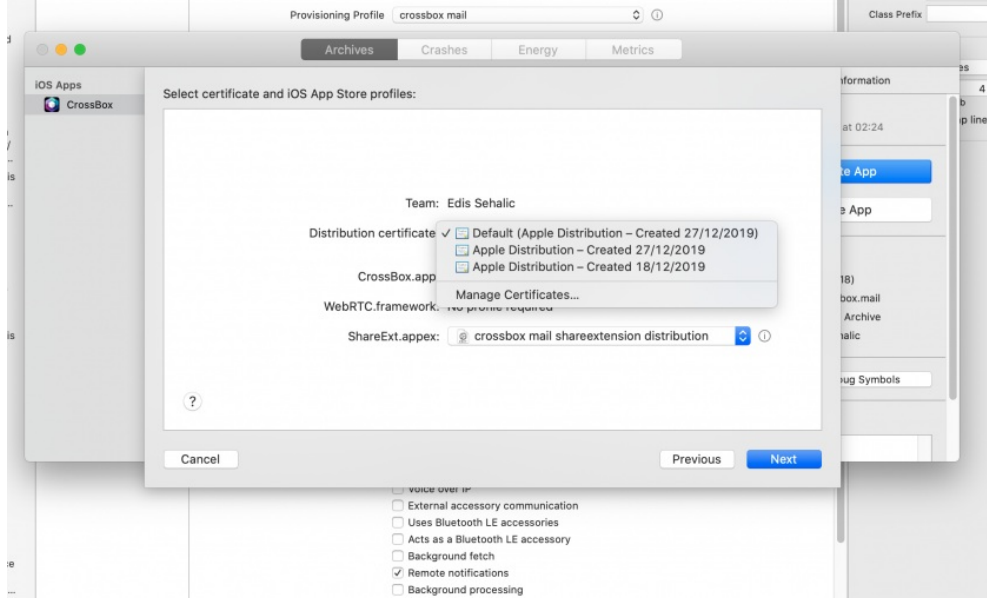
- Click **Distribute App**



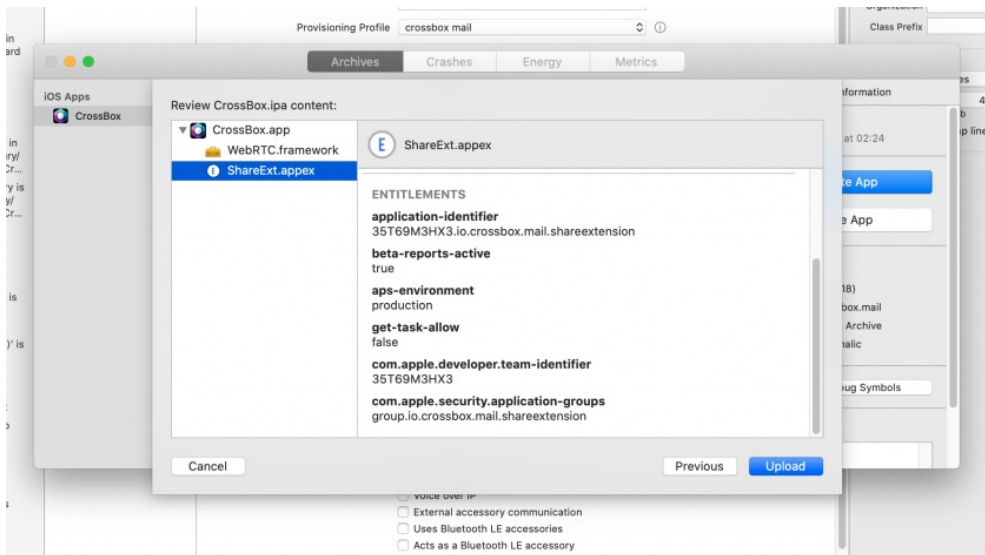
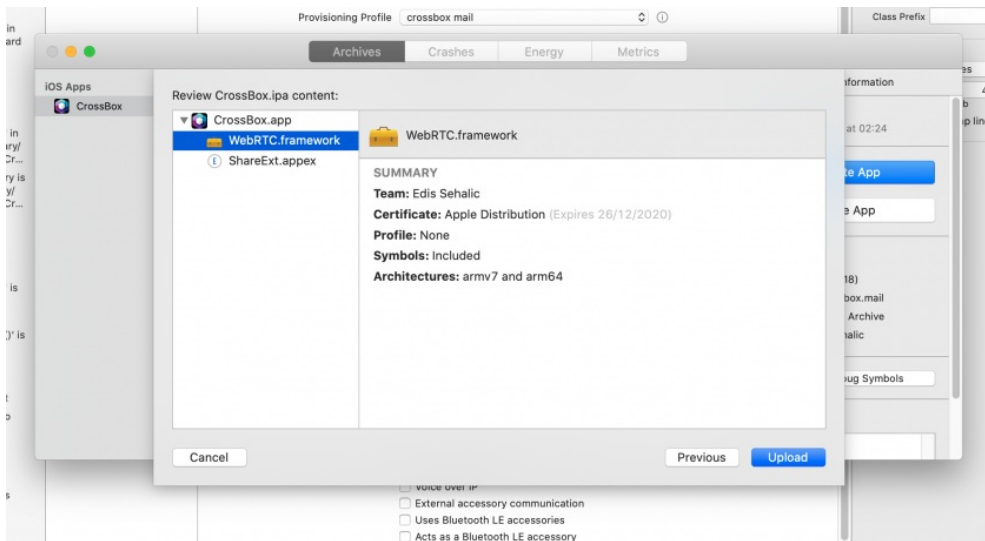
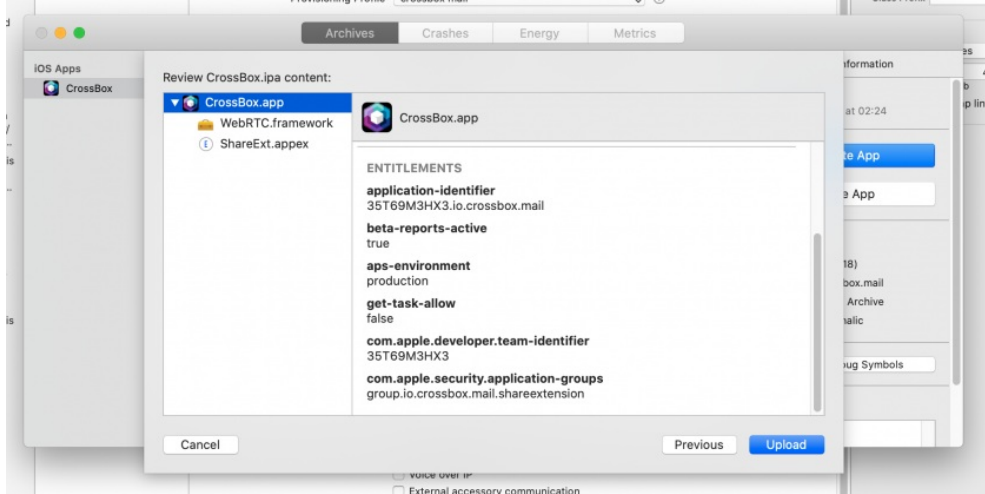
- The app will be validated and uploaded to <https://appstoreconnect.apple.com/>



After this step, you will be asked to confirm your certificates again, please select the ones you set up initially:



After this step, you will see the final step



- After the upload, the processing may take up to an hour.
- You will get an e-mail after the build has been processed by Apple.
- After the submitted app has been processed you can start filling up the information and prepare for submission. Apple will require you to provide a test account which they'll use to review the app

App Store Features **TestFlight** Activity

BUILDS

**iOS**

FEEDBACK

Crashes

Screenshots

TESTERS & GROUPS ?

All Testers (2)

App Store Connect Users

[Add External Testers](#)

APP INFORMATION


Test Information ⚠️

About TestFlight Data ?


## iOS Builds

The following builds are available to test. [Learn more about build status and metrics.](#)

Version 0.4.8

Build	App Store Connect Users ?	External Testers ?
 0.4.8	● Testing Expires in 85 days	● Ready to Submit Expires in 85 days

Version 0.4.7

Build	App Store Connect Users ?	External Testers ?
 0.4.7	● Testing Expires in 85 days	● Ready to Submit Expires in 85 days

**NOTE:** There's a known issue with TestFlight that it might say that processing is not complete, in order to fix the problem logout and login again. Sometimes it may take between 30 minutes to a few hours to complete.

- Apple may ask you to answer some questions about the App.

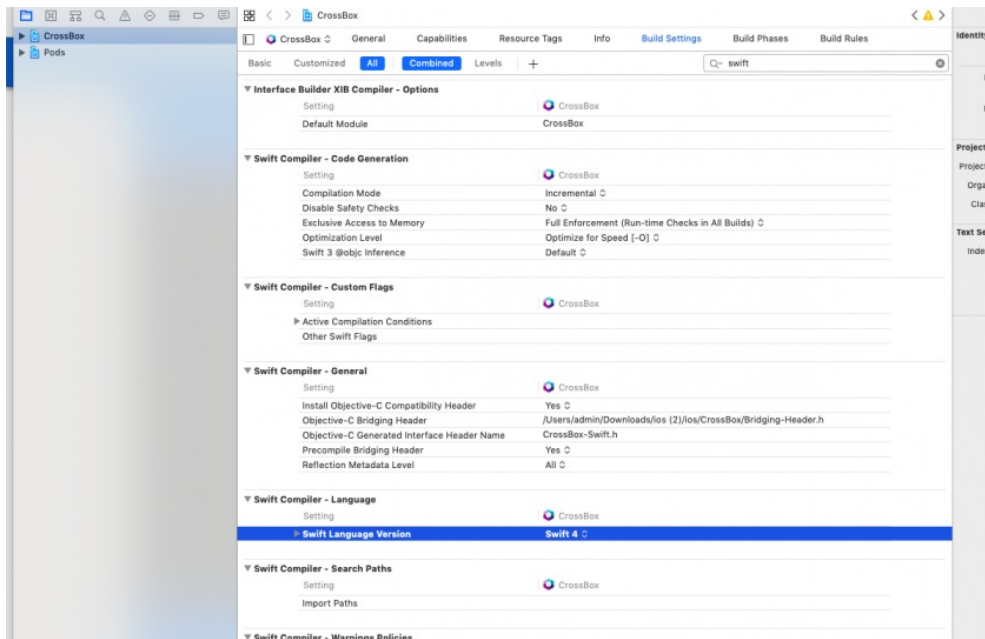
We prepared the answers for you. Replace placeholders APP\_NAME, COUNTRY\_REGION, SERVICE\_PROVIDER\_SIGNUP\_LINK, and SERVICE\_PROVIDER\_WEBSITE with your own.

- What is the purpose of the app?  
APP\_NAME is the All-in-one Communication Suite designed for Small and Medium Enterprises in COUNTRY\_REGION(optional) for their modern-day communication needs.
- Who is your target audience?  
Small and Medium Enterprises and individuals
- Is this app built for a specific company and its internal use? If yes, which company?  
No
- Or is the app built for internal use for your company?  
No
- How do users obtain an account?  
By signing up to SERVICE\_PROVIDER\_SIGNUP\_LINK and purchasing the hosting service for their domain. After signup, they can create an account via admin interface.
- If users create an account to use your app, are there fees involved?  
User pays for the service monthly or yearly via the SERVICE\_PROVIDER\_WEBSITE
- Does your app access any paid content or services?  
The user only pays for the service monthly or yearly via the SERVICE\_PROVIDER\_WEBSITE
- What are the paid content or services, and what are the costs?  
Costs are N to N \$ as listed on the website SERVICE\_PROVIDER\_WEBSITE
- Do individual customers pay for the content or services?  
Yes, they can purchase the service via SERVICE\_PROVIDER\_WEBSITE as there is no requirement to be a business.
- If no, does a company or organization pay for the content or services?  
N/A

## Known issues on IOS

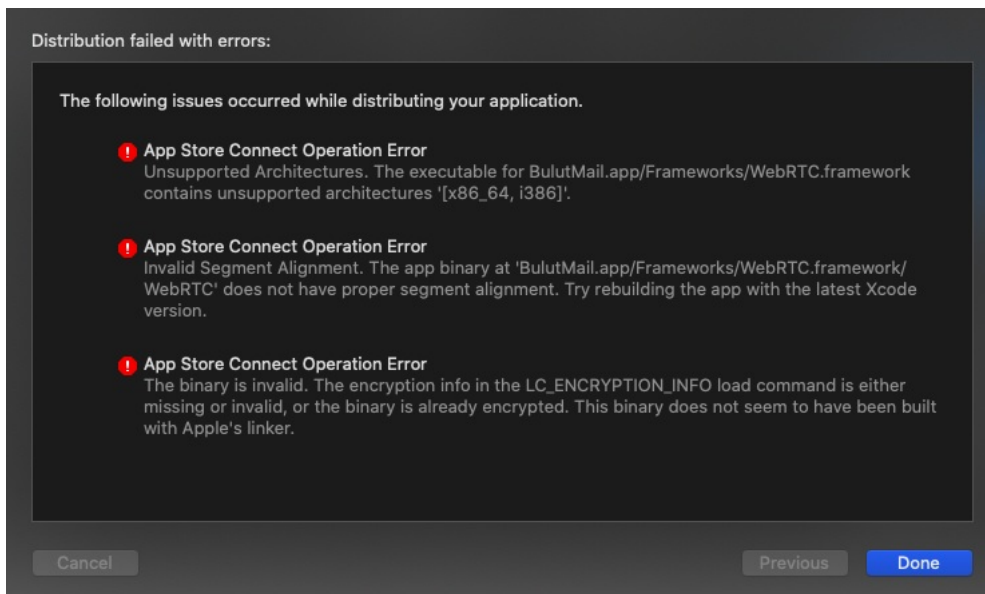
If the build process in Xcode fails, you can try to fix it by:

1. Installing cocoa pods on your machine - <https://cocoapods.org/>
2. Correcting the build settings for the Swift version - make sure you select Swift Version 4



If you get any of the following errors:

- Unsupported Architectures. The executable contains unsupported architectures '[x86\_64, i386]
- Invalid Segment Alignment. The app binary does not have proper segment alignment
- The binary is invalid. The encryption info in the LC\_ENCRYPTION\_INFO load command is either missing or invalid, or the binary is already encrypted



These errors happen if you forgot to run `sh prepare-production.sh` before starting the submission to the Apple Store as explained [here](#).

- Revision #13
- Created 6 years ago by [Docs Admin](#)
- Updated 6 years ago by [Docs Admin](#)