


How to run and Release Mobile

Debugging the iOS app:


1. Set efcMobileApp.iOS as the startup project
2. Pair the Mac by clicking this icon:
 and or type unknown
3. Make sure you are in Debug mode, not Release mode
4. Click the down arrow on the start options, and pick the device you want to simulate - Remote Device will not work
5. Happy testing!

Releasing the iOS app:

1. In the efcMobileApp.iOS/Info.plist file, scroll down to Bundle version and bundle versions string (short) then add 1 to each value.
2. Set efcMobileApp.iOS as the startup project
3. Pair the Mac (see step 2 of Debugging the iOS app)
4. Make sure you are in Release mode, not Debug mode
5. Right click on efcMobileApp.iOS and click Archive...
6. Wait for the Archive to finish - if there's a signing error:
 - a. Right click on efcMobileApp.iOS and click Properties.
 - b. Once there, click the iOS Bundle Signing tab and make sure the Signing Identity is: *iPhone Distribution: eFileCabinet, Inc.* and the Provisioning Profile is: *eFileCabinet Online AppStore* and try the Archive again.
 - c. If there's still an issue, the signing certificate may be out of date. Please get with dev management to get a new signing certificate.
7. Once the Archive has finished, use VNC Viewer to connect to the Mac Mini (if you need credentials, ask dev management)
8. Open XCode
9. On the top toolbar, click Windows and then click Organizers
10. You should see Archives as a tab on the left, and in there you should see the archive you just created in Visual Studio
11. Select the archive and click distribute app
12. Follow the wizard, leaving all the defaults except the one asking about sending reports to Apple and XCode - uncheck that box
13. Please step through next steps with a member of the Dev management.
14. Login into <https://developer.apple.com/> using login from Last pass: Mobile Dev Apple Account. (may need to be logged into the dev mac to get the authentication code on

login)

15. Go to the App Store Connect section and choose Apps:


 and or type unknown

16. Then go to the TestFlight tab at the top. You will now see the most recent version displayed. It will take a while to process.

17. Once done processing it will ask to complete some compliance information. Click on manage and then answer the questions as shown below (I had two different sets of questions I had to ask so both are listed below):

 and  and or type unknown

 and or type unknown

 and or type unknown

 and or type unknown

18. Once you hit start internal testing make sure your apple ID is added to the internal testings users list and download the testflight app on your apple device. Once app is up and running log in and accept the invite to the app.

19. You are not good to test the app.

Debugging the Android app:

1. Set efcMobileApp.Droid as the startup project
2. Make sure you are in Debug mode, not Release mode
3. Run the emulator you have (you may need to install a new device)
4. Make sure you have enabled Hyper-V and Windows Hypervisor Platform in windows features, and that you have enabled Virtualization (or SVM on AMD) in your bios (If you don't android emulation probably won't actually work). You will want to restart your computer when you do this. Also see [this](#) doc for further guidance.

 and or type unknown

5. Wait for a long time - the Android emulator is very slow on startup
6. Alternative: If you have an Android phone, plug it in and on step 3 select your phone on the run dropdown. That way debugging should be a lot faster.

Releasing the Android app:

1. Release can not be done in Visual Studio 2022, Please use Visual Studio 2019
2. Make sure you've followed the steps found [here](#) to make sure JDK 11 is installed and will work with VS 2019.
3. Set efcMobileApp.Droid as the startup project
4. Make sure you are in Release mode, not Debug mode
5. Right click on efcMobileApp.Droid and click Archive...
6. Once the Archive has completed, click Open Distribution (you can also click the 'Distribute' button, and walk through the steps, at the end just make sure you upload to the lowest level track)

7. If at any point you are asked for signing credentials, they are found in the Android Signature Key Info file in the root of the source code.
 8. If at any point you are asked for a keystore file, import the keystore file found in the root of the source code.
 9. Give the resulting apk file to dev management to upload to the play store.
-

Revision #10

Created 18 July 2022 23:47:55 by Bryce Holloway

Updated 9 February 2023 20:20:49 by Quinn Godfrey