

# How to publish an iOS App from Microsoft AppCenter to Apple App Store Connect

**Author** : Tobias Hofmann

**Date** : June 26, 2019

In this blog I will detail how you use [Microsoft's AppCenter](#) to [build](#) an iOS app und publish it directly to iTunes Connect. This allows you to decouple the building, testing and distribution process from the developers. The developer only has to push the app to the repository (I am using Azure DevOps) and AppCenter takes care of the rest.

The steps to do so are:

1. Create app project
2. Configure build
3. Add signing certificates
4. Configure distribution to iTunes Connect

## Create App Project

Open [AppCenter](#) and create a new project.

You can add AppCenter features to you app, but it's optional. I already have a running app that I just want to build and distribute. Next step is to configure the build.

## Build

Select the repository where the source code is hosted. I use Azure DevOps (free tier). Unfortunately, GitLab is not listed and in the free tier I am using it is not possible to add self-hosted git repositories.

AppCenter will connect to Azure DevOps via SSO and list the available projects.

This adds the repository to the build configuration. You'll see the branches and last commit message.

To configure the build, click on the configuration option for the branch. The option will only appear when you hover with your mouse over the branch.

AppCenter will scan the project and find the available XCode settings.

## It's full of stars!

*Where documentation meets reality*

---

You can configure the XCode version to be used for the build. This is very useful when you are using external libraries that do not work with newer XCode versions. For instance, the Fiori libraries included in my project were not released for 10.2.1 and the newer Swift version that comes with it. Therefore, the build exited with an error. Until SAP released an updated version of Fiori for iOS, I had to use XCode 10.2.

AppCenter offers options to automatically increase the build number, or run your XCTests.

## Sign build

To be able to send the app to iTunes, you must sign the build using your certificate and provisioning profile. I wrote two blogs on how to get these:

- [How to get your iOS App provisioning profile](#)
- [How to download your iOS distribution certificate](#)

When you have these available, you can start configuring the app signing. You upload the files and provide needed credentials for your private key.

## Distribute

Next step is to define where you want to distribute the app to. You can send it to the official App Store, App Store Connect Users for your TestFlight beta testers, or to an internal Company Portal.

I am going to distribute the app to App Store Connect for TestFlight. Select App Store Connect. If you do not have yet an account linked to Apple, you can do this here.

AppCenter is connecting to App Store Connect and retrieves a list of apps. I only have one app available, making the selection easier. It also means that you have to create the app first in App Store Connect. AppCenter is not able to create the app definition for you.

## It's full of stars!

*Where documentation meets reality*

---

Select the app and click on Assign.

In case 2FA is enabled for your Apple ID, you will have to provide an app-specific password. [I wrote a blog on how to create an app specific password.](#)

## It's full of stars!

*Where documentation meets reality*

---

After informing the app-specific password, you get back to the previous screen. Click again on assign.

Now AppCenter is configured to connect to Apple Connect. Back at the Distribute builds section, you can select App Store Connect Users.

## Result

You can now click on save or already start your first build.

## **Run build and distribute to App Store Connect**

After the project is created and the build configured, you can start a build. AppCenter will find an available build agent, clone the repository, build, test, sign and distribute the app.

### **AppCenter**

Waiting for a free build agent

### **Build starting**



## **Distribute**

After the build is done, the app is send to Apple Connect and processed there. Apple will check if the build is OK. This will take some time. The status of the build is Processing.

## **App Store Connect**

When processing is done, you get an email form Apple.

## **It's full of stars!**

*Where documentation meets reality*

---

The status of the app in AppCenter and App Store Connect changes and you can distribute the app to your beta testers via TestFlight.