Overview

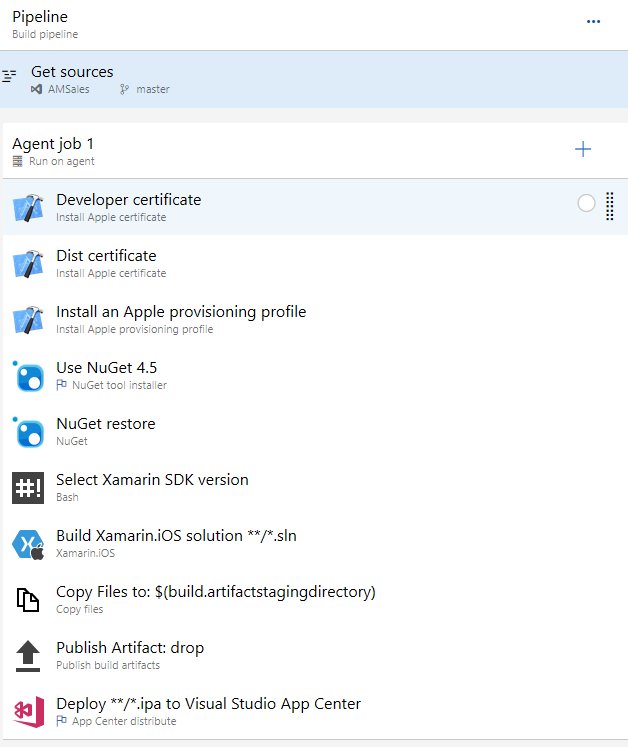
Getting Xamarin iOS apps to build correctly in Azure DevOps can be very tricky. In this guide we will walk through the different steps and where to get the data.

# Create Build Pipe

|  |  |
| --- | --- |
| After you have your project, start a new build pipe and select Xamrin.iOS |  |

# Working Build Pipe Example

Below is an example of a working build pipe which builds, signs and deploys to app center for testing



# Build Pipe Breakdown

|  |  |
| --- | --- |
|  | At the time of this document, choose **macOS-10.13**. Other versions will break the build and linkage. |
|  | This will be the cert you get from your Mac. You have to build and extract from the local KeyStore (on the mac). Remember to set the PR or use a variable like this: $(P12password)  <https://help.apple.com/xcode/mac/current/#/dev154b28f09> |
|  | This is needed to distribute your ipa (app). Remember to set the PR or use a variable like this: $(P12password)  <https://support.magplus.com/hc/en-us/articles/203808748-iOS-Creating-a-Distribution-Certificate-and-p12-File> |
|  | This is obtained from the <https://developer.apple.com/> site. You have to specify your package identifier and what users/UUIDs then download the provisioning profile. |
|  | Leave as default |
|  | At this time 5\_16\_0 seems to work best:  sudo $AGENT\_HOMEDIRECTORY/scripts/select-xamarin-sdk.sh 5\_16\_0 |
|  | There are many tricky steps here. Make sure to specify the signing identifies and deselect “build for iOS Simulator”. Working setting below. |
|  | Leave defaults |
|  | This is pretty straight forward accept for the “slug”. There are no direct errors just will not publish. The slug is not the full URL, but the account and name of app: mysite/myapp |