

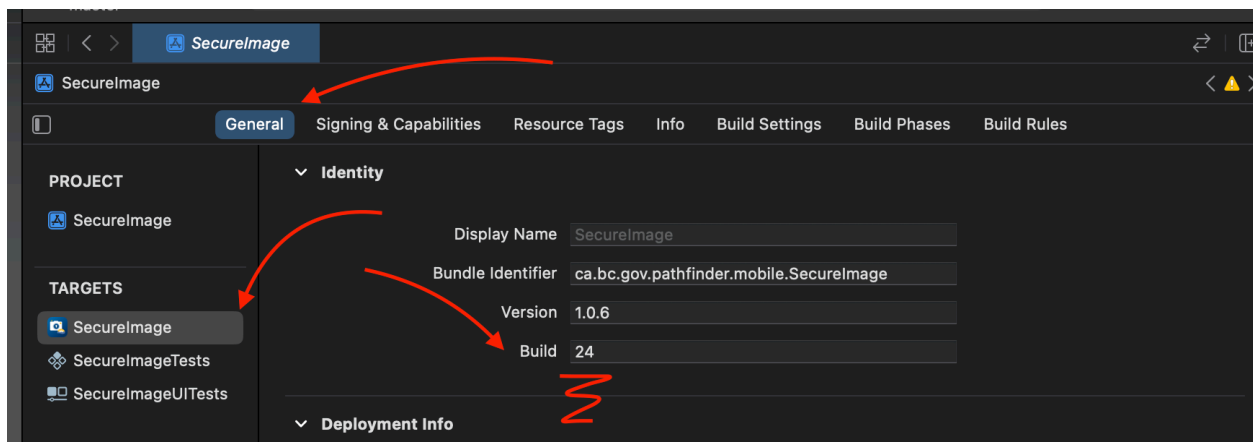
This documentation contains the following sections:

- **Project Pipeline documentation**
 - Describes how the pipeline has been set up.
- **Github Action Workflow Diagram**
 - A diagram that demonstrates the steps in the actions workflow
- **Creating an IPA with Terminal**
 - Instructions for running creating a build locally using the same scripts used by the pipeline
- **Creating an IPA with Xcode**
 - Instructions for creating a build through Xcode.
- **Suggested maintenance**
 - Some problems to be solved

Note:

Remember to update the build number of the project before creating a PR.

If you forget, the build won't upload to the MDM since the same build number already exists.



Project Pipeline documentation

This project's CI pipeline has been implemented using [Github Actions](#).

As described in the documentation page above, workflows are placed in `.github/workflows`. We have created 2 more folders under `.github/` for organizing files related to our workflow:

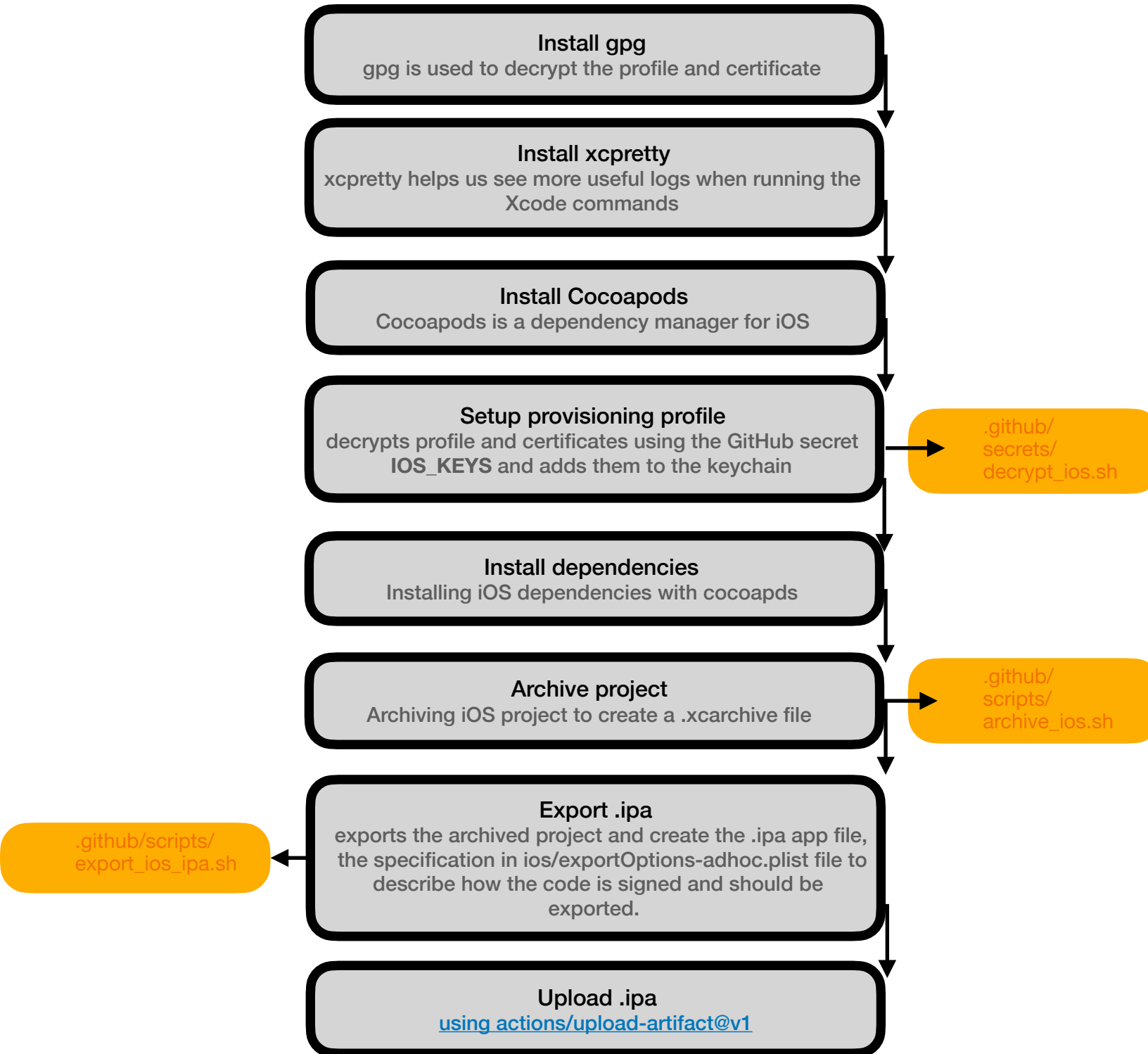
- `.github/scripts`
 - Contains 2 scripts for archiving and exporting our iOS project and generating an ipa file
 - `.github/scripts/archive_ios.sh`
 - `.github/scripts/export_ios_ipa.sh`
- `.github/secrets`
 - Contains an encrypted distribution certificate, an encrypted provisioning profile and a script file to decrypt the certificate and profile and adds them to the keychain to allow for signing our code while creating the ipa file.
 - `AppleDistribution.p12.gpg`
 - `SecureImage__In_House.mobileprovision.gpg`
 - `decrypt_ios.sh`
 - `AppleDistribution.p12` and `SecureImage__In_House.mobileprovision` have been generated and supplied by an administrator of the Govt of British Columbia Apple Account.
 - The 2 encrypted files have been encrypted using `gpg` and with the following commands:
 - `gpg --symmetric --cipher-algo AES256 AppleDistribution.p12`
 - `gpg --symmetric --cipher-algo AES256 SecureImage__In_House.mobileprovision`
 - The encryption password has been added to Github secrets and called **IOS_KEYS**
 - *If these files need to be updated, encrypt them using the commands above (and rename them if necessary to match the above names) and add the password used to GitHub secrets with the name **IOS_KEYS***
 - `.github/secrets/decrypt_ios.sh` decrypts the distribution certificate and provisioning profile and adds them to the keychain for use during the creation of the ipa build file.

We have selected a macOS runner since the iOS project can be only built on the macOS operating system. [For more information on GitHub-hosted runners visit this page](#). According to the documentation page above, the latest available macOS runner is macOS 11, which is used for our iOS workflow.

The workflow is triggered when code is pushed to the master branch and files under the folder `iOS/**/*` have been changed.

Github Action Workflow

Code is pushed to master and filed in iOS/ have been changed



Creating an IPA with Terminal

2 Pre-Requisites:

1) Install Dependencies

install the dependency manager: `sudo gem install cocoapods`

execute **pod install** from the iOS/ folder to install the dependencies since they are git ignored from the repository

2) Install Provisioning profile and certificate in the zip file

Extract the contents of the file, and double click the 2 files to install them

You can obtain the password for the p12 certificate from the admin of the Apple account.

If you don't have access to these 2 files, please contact the admin of the Apple account.

Archive and export:

Archive: [.github/scripts/archive_ios.sh](https://github.com/secure-image-app/scripts/archive_ios.sh)

```
xcodebuild -workspace $PWD/ios/SecureImage.xcworkspace -scheme SecureImage -sdk
iphoneos -configuration AppStoreDistribution archive -archivePath $PWD/build/
SecureImage.xcarchive clean archive
```

You should see:

```
SecureImage.app (in target 'SecureImage' from project 'SecureImage')
  cd /Users/amir/Developer/secure-image-app/ios
  /usr/bin/touch -c /Users/amir/Library/Developer/Xcode/DerivedData/SecureImage-bsoiadmilarxqegbgowpl
paezawg/Build/Intermediates.noindex/ArchiveIntermediates/SecureImage/InstallationBuildProductsLocation/
Applications/SecureImage.app

** ARCHIVE SUCCEEDED **
```

Export: [.github/scripts/export_ios_ipa.sh](https://github.com/secure-image-app/scripts/export_ios_ipa.sh)

```
xcodebuild clean -exportArchive -archivePath $PWD/build/SecureImage.xcarchive -exportPath
$PWD/build -allowProvisioningUpdates -exportOptionsPlist ios/exportOptions-adhoc.plist
xcpretty
```

You should see:

```
ExtensionToBundleIdentifierMap.watchOS of plug-in com.apple.dt.IDEWatchSupportCore
2022-03-25 05:57:09.130 xcodebuild[55534:778431] [MT] IDEDistribution: -[IDEDistributionLogging _create
LoggingBundleAtPath:]: Created bundle at path "/var/folders/79/0_24hs1539d0k7_8xffrtq64000gn/T/SecureI
mage_2022-03-25_05-57-09.130.xcdistributionlogs".
Exported SecureImage to: /Users/amir/Developer/secure-image-app/build
** EXPORT SUCCEEDED **
```

In the Root folder of the project, you will see a new folder called Build.

You can find the final ipa in:

[secure-image-app/build/Apps/SecureImage.ipa](#)

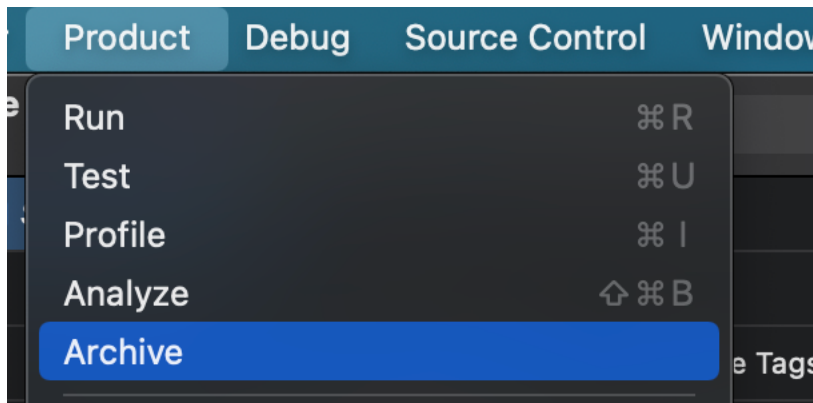
Creating an IPA with XCode

1. Setup and open project

Install Dependencies

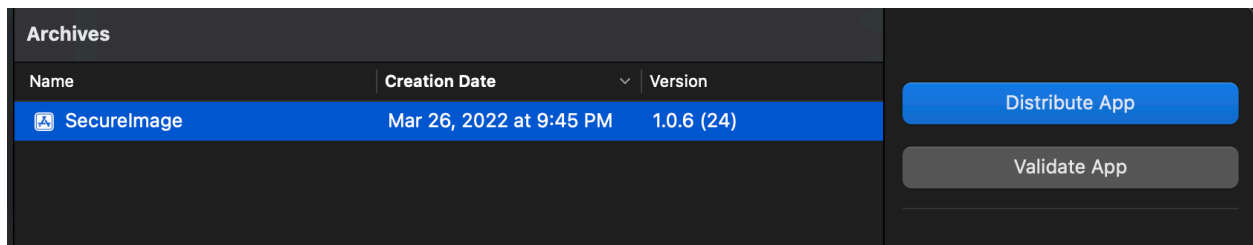
- install the dependency manager: `sudo gem install cocoapods`
- execute `pod install` from the `iOS/` folder to install the dependencies since they are git ignored from the repository
- **Install Provisioning profile and certificate in the zip file**
 - Extract the contents of the file, and double click the 2 files to install them
 - You can obtain the password for the p12 certificate from the admin of the Apple account. If you don't have access to these 2 files, please contact the admin of the Apple account.
- **Open Xcode project**
open `iOS/SecureImage.xcworkspace`

2. Archive project

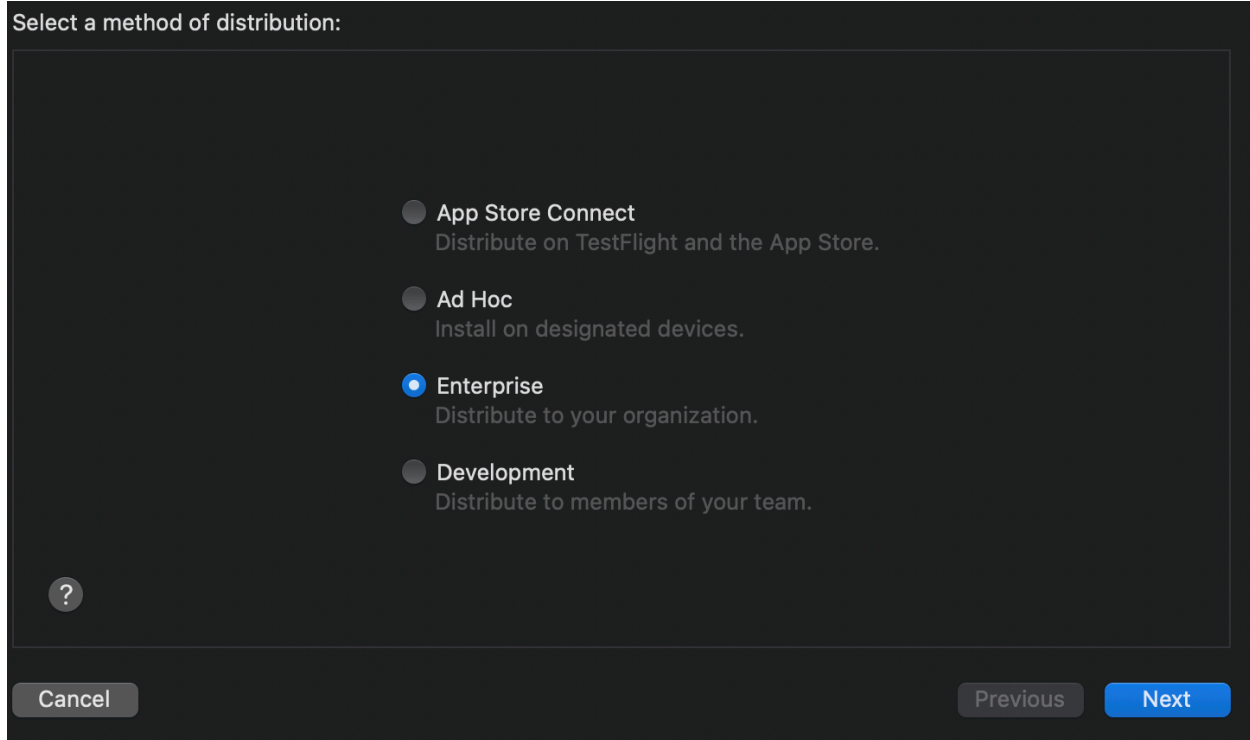


3. Export

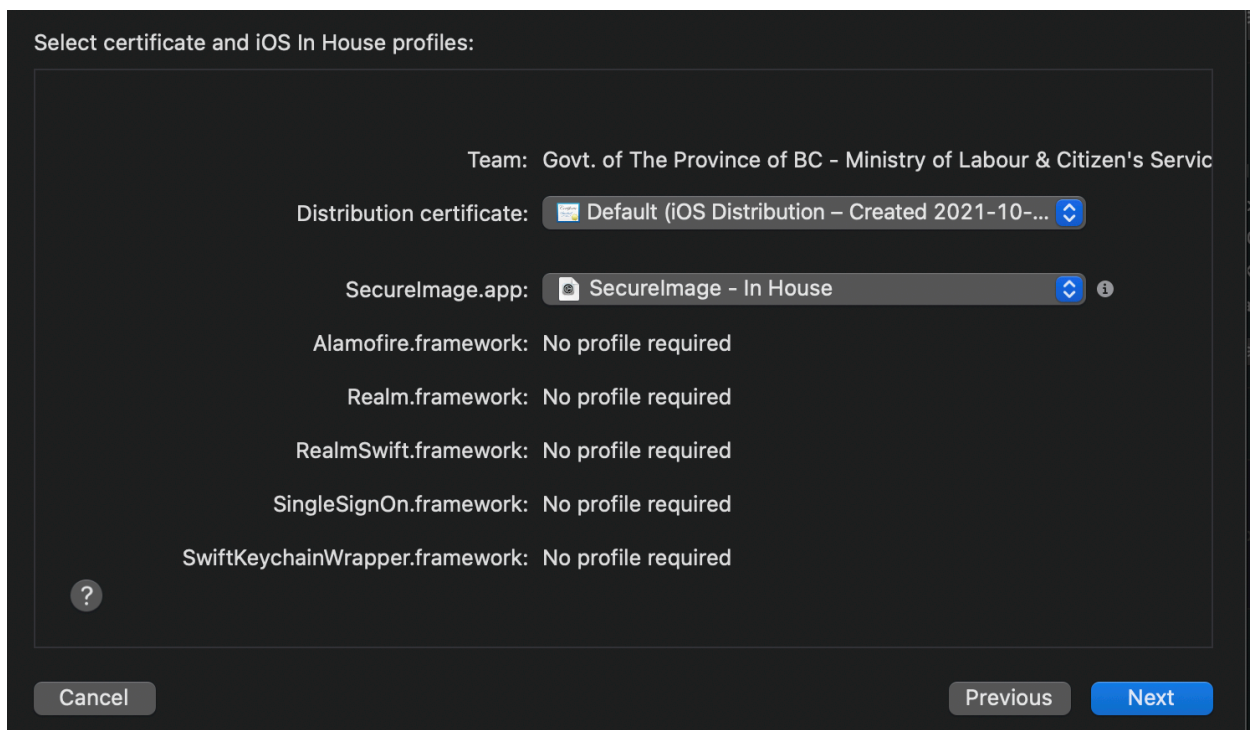
- Select archived project and then click Distribute App



- For distribution method, select **Enterprise**



- Select distribution certificate and provisioning profile



- Finally, select the folder you would like you export to.
when the folder is created, the file with the extension .ipa is the build file.

Suggested maintenance

- Update project to use Swift version 5.
 - The project currently uses an older version of swift
- Update the [SingleSignOn](#) dependency used by the app to use an updated version of its dependencies.
- Update project dependencies
- UI/ Front end code should be refactored to be more modular for added clarity
 - view controller logic, especially the Albums view controller file is very large and should be refactored.
- Consider not using a mono repo
 - currently, we cannot commit the dependencies with for ios in the repository because we would exceed the maximum size limit for GitHub projects. Installing the dependencies each time during the pipeline build is a slow process.