iOS SDK Distribution guide

Creating fat framework

This document describes old-school but still very popular way of distributing frameworks in form of fat frameworks, i.e. frameworks containing several architectures within their binary. To create fat framework follow the steps bellow:

1. Clone project folders from repository

   ```
   git clone <path-to-repo>
   ```

2. Go to project folder and enter to SDK subfolder:

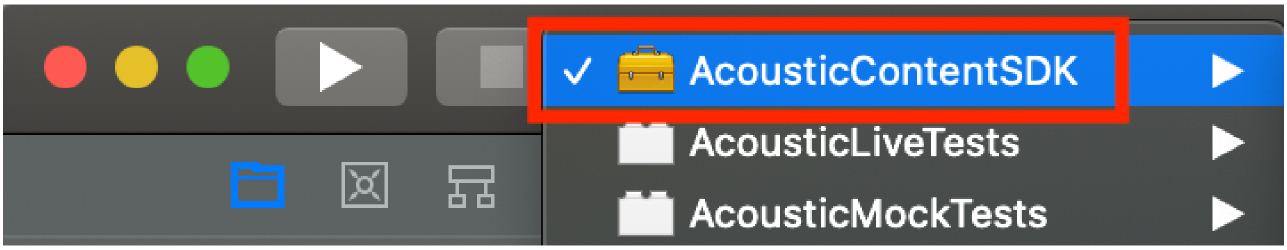
   ```
   cd AcousticContentSDK
   ```

3. Open `AcousticContentSDK.xcproject` in Xcode:

   ```
   open AcousticContentSDK.xcodeproj
   ```

   > or just find it in Finder and double-click on `AcousticContentSDK.xcodeproj` file

4. Select `AcousticContentSDK` schema:



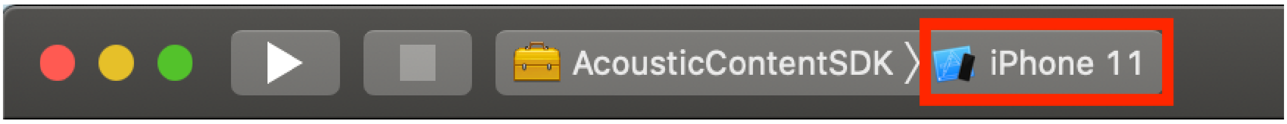
5. Add build script to Archive post-action:

   1. Go to `Edit Scheme` -> `Archive` -> `Post-actions`
   2. Tap `+` and `New Run Script Action`
   3. Set `Provide build settings from` to active non-testing schema,. ex: `AcousticContentSDK`
   4. Paste following script:

   ```
   UNIVERSAL_OUTPUTFOLDER=${BUILD_DIR}/${CONFIGURATION}-universal
   ```

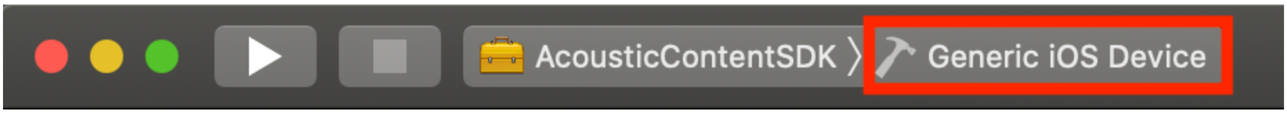
```
# Create output directory

mkdir -p "${UNIVERSAL_OUTPUTFOLDER}"

# Build project for simulator and device architecture

xcodebuild -target "${PROJECT_NAME}" -configuration ${CONFIGURATION} -sdk iphonesimulator ONLY_ACTIVE_ARCH=NO BUILD_DIR="${BUILD_DIR}" BUILD_ROOT="${BUILD_ROOT}" clean build

xcodebuild -target "${PROJECT_NAME}" ONLY_ACTIVE_ARCH=NO -configuration ${CONFIGURATION} -sdk iphoneos BUILD_DIR="${BUILD_DIR}" BUILD_ROOT="${BUILD_ROOT}" clean build

# Copy the framework folders from `iphoneos` build to the universal folder

cp -R "${BUILD_DIR}/${CONFIGURATION}-iphoneos/${PROJECT_NAME}.framework" "${UNIVERSAL_OUTPUTFOLDER}/"

# Try to copy Swift modules from `iphonesimulator` build to the framework in universal folder

BUILD_PRODUCTS="${SYMROOT}/../../../../Products"

cp -R "${BUILD_PRODUCTS}/Debug-iphonesimulator/${PROJECT_NAME}.framework/Modules/${PROJECT_NAME}.swiftmodule/." "${UNIVERSAL_OUTPUTFOLDER}/${PROJECT_NAME}.framework/Modules/${PROJECT_NAME}.swiftmodule"

# Create universal binary file using `lipo` and place the combined executable into the framework in universal folder

lipo -create -output "${UNIVERSAL_OUTPUTFOLDER}/${PROJECT_NAME}.framework/${PROJECT_NAME}" "${BUILD_PRODUCTS}/Debug-iphonesimulator/${PROJECT_NAME}.framework/${PROJECT_NAME}" "${BUILD_DIR}/${CONFIGURATION}-iphoneos/${PROJECT_NAME}.framework/${PROJECT_NAME}"

# Copy the framework to the project's directory

cp -R "${UNIVERSAL_OUTPUTFOLDER}/${PROJECT_NAME}.framework" "${PROJECT_DIR}"
```

6. Select any iPhone Simulator from the list:

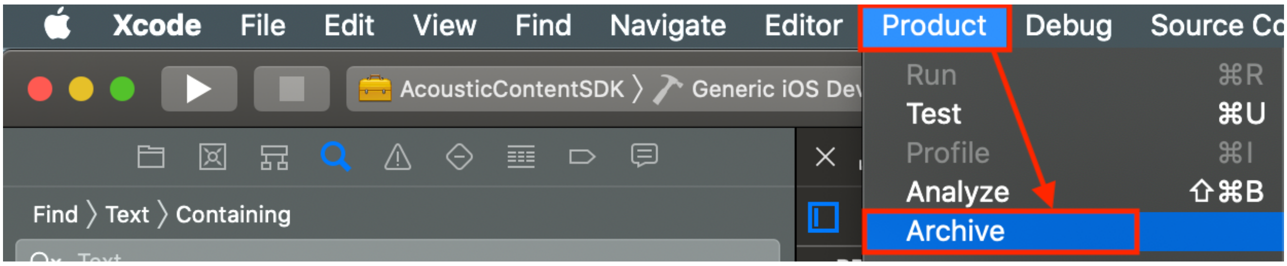


7. Build project for Simulator -> `Cmd+B`

8. Select any Generic iOS Device:



9. Run `Archive` action from Xcode Menu:


Xcode File Edit View Find Navigate Editor Product Debug Source Co
AcousticContentSDK 〉 Generic iOS Dev
Find 〉 Text 〉 Containing
Run ⌘R
Test ⌘U
Profile ⌘I
Analyze ⇧⌘B
Archive

10. Wait when archiving project complete. Ignore Xcode Archives window which just appeared.

11. Verify if fat framework has built successfully return to terminal and run:

```
lipo -info AcousticContentSDK.framework/AcousticContentSDK
```

Correct `output` should include at least two architectures `x86_64` and `arm64`:

```
Architectures in the fat file:
AcousticContentSDK.framework/AcousticContentSDK are: x86_64 arm64
```