Static Library

Table of content(TOC)

Static Library

Chapter 2.0

How to create static library without resources or .a lib

Step 1. File-> New Project-> (framework and library) -> Static Library

Universal target required

Step 2. Select Static Library Target-> File -> New Target-> Cross-platform->Aggregate

Step 3-> Generate Universal .a file. Add script path in universal target

Universal target-> Build Phase->Run Script

Link Script

Path:-
chmod u+x "${SRCROOT}/ScriptFolderName/FileName.sh"
"${SRCROOT}/ScriptFolderName/FileName.sh"

In Last page added script. Take from there

Step 4-> Sharing with application developer

Project>Selected Universal target-> Build->

Navigate to project folder path->Universal-> you library will in this folder.

How use in Application.

Step 1. Copy .a and target.swiftmodule folder and app in sample application

Step 2. Add .a file in bundle. Don't add target.swiftmodule in bundle. Just in keep in folder.

Step 3. Add .a file

-> Target-> framework, libraries, embedded content

Step 4. Added library path and import path

$(PROJECT_DIR)/yourapp/libFolder

Static Library

Chapter 3.0

How to create static library with resources like core data and storyboard or .a lib

Step 1. File-> New Project-> (framework and library) -> Static Library

Universal target required

Step 2. File->New target->MacOS->Bundle->LibNameResources

Step 3. Select target->LibNameResources-> change macOS to iOS

Step 4. Select lib target->Build and Phase->Add LibNameResources Dependency.

Step 5. Select Lib target create Core dataModel
-> Library folder-> New-> Data Model

Step 6. Select target->LibNameResources-Compile sources->Add CoreDataModel

Step 7. If you want create any file resource like image, storyboard etc.

Step 8. Select Static Library Target-> File -> New Target-> Cross-platform->Aggregate

Step 9-> Generate Universal .a file. Add script path in universal target

Universal target-> Build Phase->Run Script

Path:-
chmod u+x "${SRCROOT}/yourscriptfolder/Release.sh"
"${SRCROOT}/yourscriptfolder/Release.sh"

In Last page added script. Take from there

Sharing with application developer

->Selected Universal target-> Build->

Navigate to project folder path->Universal-> you library will in this folder.

Static Library

How use in Application.

Step 1. Copy.a and target.swiftmodule folder and app in sample application

Step 2. Add .a file in bundle. Don't add target.swiftmodule in bundle. Just in keep in folder.

Step 3. Add .a file

-> Target-> framework, libraries, embedded content

Step 4. Added library search path and import path search path.

$(PROJECT_DIR)/yourapp/libFolder

Step 5. Add Library Resource bundle

Frequent ask questions (FAQ)

Question 1. Some time when you add library in sample application and import in class. If will not show for import.

Answers:-

Could be the reason:-

a. Library and Sample Application target mismatch. Means library target iOS 10 and application target is iOS 13 library will not show for import.

b. Missed step 3. Forgot to add .a file in -> Target-> framework, libraries, embedded content

c. Or forgot to copy paste swiftmodule in application lib folder

d. Missing library search path or header search path or import path

Question 2. When you are using objective - C class in library or framework. is it required to add -Objc in Other linker flag
Answer:-
Yes, When we are using objective classes in framework or library. And Library and framework in Objective C Project. Otherwise code will not compile in Objective C. If swift framework or library using in swift project no need to add.

Static Library

Question 3. Sometime Core data bundle not init. Not able to find model.

Answers:-

```
    guard let resourcePath =
Bundle.main.path(forResource:”ResourceTargetName”, ofType: "bundle"),
            let bundle = Bundle(path: resourcePath) else {

                fatalError("Failed to locate data model schema file.")
        }

DataManager.setUp(withDataModelName:”storename”, bundle: bundle,
persistentStoreName:”storename”)
```

Script start for swift static lib

```
# Type a script or drag a script file from
yo######################
# Options
######################

REVEAL_ARCHIVE_IN_FINDER=false

FRAMEWORK_NAME="${PROJECT_NAME}"
#STEP 1 Getting framwork name same as lib name.
echo 👍 1 Building Lib  ${PROJECT_NAME} for Universal

SIMULATOR_LIBRARY_PATH="${BUILD_DIR}/${CONFIGURATION}-
iphonesimulator"

echo 👍 2 Simulator Lib path:- ${SIMULATOR_LIBRARY_PATH}

DEVICE_LIBRARY_PATH="${BUILD_DIR}/${CONFIGURATION}-iphoneos"

echo 👍 3 Device Lib path:- ${DEVICE_LIBRARY_PATH}

Universal_Dir_Name="Universal"

Universal_Directory_Path=${SRCROOT}

Destination_Path="${Universal_Directory_Path}/$
{Universal_Dir_Name}"
```

```
echo 👍 4 Uinversal Lib path:- ${Destination_Path}
```

Static Library

```
echo 👍 Start Building Library
# Build Frameworks

xcodebuild -target "${PROJECT_NAME}" -scheme "${PROJECT_NAME}"
-sdk iphonesimulator -configuration ${CONFIGURATION}
OBJROOT="${OBJROOT}/DependentBuilds"

xcodebuild -target "${PROJECT_NAME}" -scheme "${PROJECT_NAME}"
-sdk iphoneos -configuration ${CONFIGURATION}  OBJROOT="$
{OBJROOT}/DependentBuilds"

# Create directory for universal
######################
echo 👍 End Building Library

rm -rf "${Destination_Path}"
echo 👍 6 Deleted directory from path ifs exist:- $
{Destination_Path}.


echo 👍 5 Remove Old Universal Lib from path before creating
New one.

#mkdir "${UNIVERSAL_LIBRARY_DIR}"

if [ ! -d "${Destination_Path}" ]; then
mkdir "${Destination_Path}"
echo 👍 6 Created Universal Directory at path:- $
{Destination_Path}.
# Control will enter here if $DIRECTORY doesn't exist.
fi

######################
# Make an universal binary
######################

echo 👍 7 Belew commands will merge Simulator and Device
Library which we can use debug on device and Simulator

lipo "${SIMULATOR_LIBRARY_PATH}/lib${FRAMEWORK_NAME}.a" "$
{DEVICE_LIBRARY_PATH}/lib${FRAMEWORK_NAME}.a" -create -output
"${Destination_Path}/lib${FRAMEWORK_NAME}.a"
```

```
echo 👍 8 Copy devices swiftmodule files at Destination_Path
```

Static Library

```
cp -r "${DEVICE_LIBRARY_PATH}/${FRAMEWORK_NAME}.swiftmodule"

"${Destination_Path}/${FRAMEWORK_NAME}.swiftmodule"

echo 👍 8 Merge Simulator swiftmodule files to in device
swiftmodule files. which is required  generic FAT lib

cp -r "${SIMULATOR_LIBRARY_PATH}/$
{FRAMEWORK_NAME}.swiftmodule/" "${Destination_Path}/$
{FRAMEWORK_NAME}.swiftmodule/"

echo ⛳✅ 9 Created Universal Lib at path:- $
{Destination_Path}. This is Universal lib, Copy .swiftmodule
and .a file and add in sample project

#ur workspace to insert its path.
exit 0
```

Script End

Chapter 3.1

How to use static lib (.a) file in framework

Step 1. Copy paste static lib and lib.swiftmodule in framework project path.

Step 2. Add in bundle lib file in framework bundle.

Step 3. Set import path
Project-> Build setting -> import path->Add Lib folder path.

$(inherited)
$(PROJECT_DIR)/libfoldername.

Now you can import and use lib in framework

Static Library

How to use static lib (.a) file in project

Step 1. Copy paste static lib and lib.swiftmodule in project path.

Step 2. Add in bundle lib file in bundle.

Step 3. Set import path
Project-> Build setting -> import path->Add Lib folder path.

$(inherited)
$(PROJECT_DIR)/LibFoldername.

Now you can import and use lib in project

Swift Framework

Chapter 4.0

Create Swift Framework

How to create swift framework and user in application.

Step 1. File -> New -> Project->Framework -> Framework name

Step 2. Select Min iOS Version support

Step 3. Create class, and resources

Step 4. Build for Generic Device or Simulator

Step 5. Product and show in finder. In this path you will see framework build target.

We have separate framework for device and simulator.

How to create FAT Framework, which is useable for simulator, device and App Store.

Step 1. Create one new framework

Xcode->File->New Project->Framework Name

Step 2. Create Universal Target

Xcode-> File-Target->Cross platform->Aggregate-> FrameworkNameUniversal

Step 3. Project-> FrameworkNameUniversal-> Build Phase-> Tap on + -> Add Run Script

Step 4. Link Run Script

chmod u+x "${SRCROOT}/ScriptFolderName/ScriptFileName.sh"
"${SRCROOT}/ScriptFolderName/ScriptFolderName.sh"

Step 5. Select Universal Target and build for generic device

Step 6. Find framework in working directory (Universal)

Swift Framework

How to use separate framework for device and simulator in application

Step 1. If we want to use only in simulator pick simulator frame work or use only in device pick device framework.

Step 2.
For simulator only use in simulator.
Project-> Products-> Show in finder-> Debug-iphonesSimulator->
For Device only useable in device
Project-> Products-> Show in finder-> Debug-iphones->

Step 3. Copy and paste at project path.

Step 4. Build Phases-> Embed Libraries-> tab plus-.Add framework.

Chapter 4.1

How to use Single framework for device and simulator in application

Step 1. Pick Fat framework from Universal target

Step 2. Copy and paste at project path.

Step 3. Build Phases-> Embed Libraries-> tab plus-.Add framework.

Framework script start

```
######################
# Options
######################

REVEAL_ARCHIVE_IN_FINDER=false

FRAMEWORK_NAME="${PROJECT_NAME}"
#STEP 1 Getting framwork name same as target name .
echo 👍 1 Building framework  ${PROJECT_NAME} for Universal

SIMULATOR_LIBRARY_PATH="${BUILD_DIR}/${CONFIGURATION}-
iphonesimulator/${FRAMEWORK_NAME}.framework"

DEVICE_LIBRARY_PATH="${BUILD_DIR}/${CONFIGURATION}-iphoneos/$
{FRAMEWORK_NAME}.framework"
```

```
UNIVERSAL_LIBRARY_DIR="Universal"

Destination_Path="${UNIVERSAL_LIBRARY_DIR}/$
{FRAMEWORK_NAME}.framework"


######################
# Build Frameworks
######################

# Build Frameworks

xcodebuild -target "${PROJECT_NAME}" -scheme "${PROJECT_NAME}"
-sdk iphonesimulator -configuration ${CONFIGURATION}
OBJROOT="${OBJROOT}/DependentBuilds"

xcodebuild -target "${PROJECT_NAME}" -scheme "${PROJECT_NAME}"
-sdk iphoneos -configuration ${CONFIGURATION}  OBJROOT="$
{OBJROOT}/DependentBuilds"

######################
# Create directory for universal
######################

rm -rf "${UNIVERSAL_LIBRARY_DIR}"

mkdir "${UNIVERSAL_LIBRARY_DIR}"

mkdir "${FRAMEWORK}"


######################
# Copy files Framework
######################

cp -r "${DEVICE_LIBRARY_PATH}/." "${Destination_Path}"


######################
# Make an universal binary
######################

lipo "${SIMULATOR_LIBRARY_PATH}/${FRAMEWORK_NAME}" "$
{DEVICE_LIBRARY_PATH}/${FRAMEWORK_NAME}" -create -output "$
{Destination_Path}/${FRAMEWORK_NAME}"

echo 👍 8 Belew commands will merge Simulator and Device
Library swift module and create fat library, which we can use
```

```
debug on device and Simulator

cp -r "${SIMULATOR_LIBRARY_PATH}/Modules/$
{FRAMEWORK_NAME}.swiftmodule/" "${Destination_Path}/Modules/$
{FRAMEWORK_NAME}.swiftmodule/"

echo ⛳✅ 9 Created Universal Lib at path:- $
{Destination_Path}.

exit 0
```
Framework script end

Frequent ask questions(FAQ)

Question 1. If build for simulator or Generic Device. Framework will useable for both.
Answer: No, It only be available for build for. If build for simulator it will available for simulator only and give error for generic device. when use project and run project in device. Same case with device. If build with generic device available only for device.
Framework is not working for simulator and device. Because both use separated architecture

If we want use same single framework for both device and simulator and App Store. We have to create separated FAT framework. We have to follow process to create FAT framework.

Question 2. How to create single framework for device, App Store and simulator?

Answer:- We have to create fat framework.

Question 3. Some time we see framework build for these and these Architecture not for this architecture.

Answer:-
1. There is chance you may build framework device and using in simulator or build for simulator and using for device.

2. Or created FAT framework, but .swiftModule having only device Architectures or only simulator Architectures
By using script we can merge device and simulator . swiftModule and replace framework module. We can do this in fat script .

Question 3. Access swift framework on Objective C project. It may sometime error.

Swift Framework

Answer:- Don’t forgot to Other linker flag -Objc

Chapter 6.0
Help

1. Check .a file build architectures

Step 1. Open terminal
Step 2. Navigate to .a file path
Step 3. Run command

lipo -info filename.a

2. Check framework build architectures

Step 1. Open terminal
Step 2. Navigate to framework file path
Step 3. Run command

lipo -info frameworkname.framework/frameworkname

3. How to remove unsupported Architecture from framework

Step 1. Open terminal
Step 2. Navigate to framework file path
Step 3. Run command

lipo -remove architucturename frameworkname.framework -o frameworkname.framework

4. Merge device and simulator .swiftModule for .a file. If facing issue architecture issue in compiling.

Add below line script after creating framework.

echo 8 Copy devices swiftmodule files at Destination_Path Static Library

cp -r "${DEVICE_LIBRARY_PATH}/${FRAMEWORK_NAME}.swiftmodule" "${Destination_Path}/${FRAMEWORK_NAME}.swiftmodule/"

echo 9 Merge Device and Simulator swift module

cp -r "${SIMULATOR_LIBRARY_PATH}/${FRAMEWORK_NAME}.swiftmodule/" "${Destination_Path}/${FRAMEWORK_NAME}.swiftmodule/"

Swift Framework

4. Merge device and simulator .swiftModule for framework. If facing issue architecture issue in compiling.

echo 👍 8 Belew commands will merge Simulator and Device Library swift module and create fat library, which we can use debug on device and Simulator

```
cp -r "${SIMULATOR_LIBRARY_PATH}/Modules/$
{FRAMEWORK_NAME}.swiftmodule/" "${Destination_Path}/Modules/$
{FRAMEWORK_NAME}.swiftmodule/"
```