

How to build examples using Xcode before 10

- [Preparing examples for building](#)
- [Building and launching examples using Xcode](#)

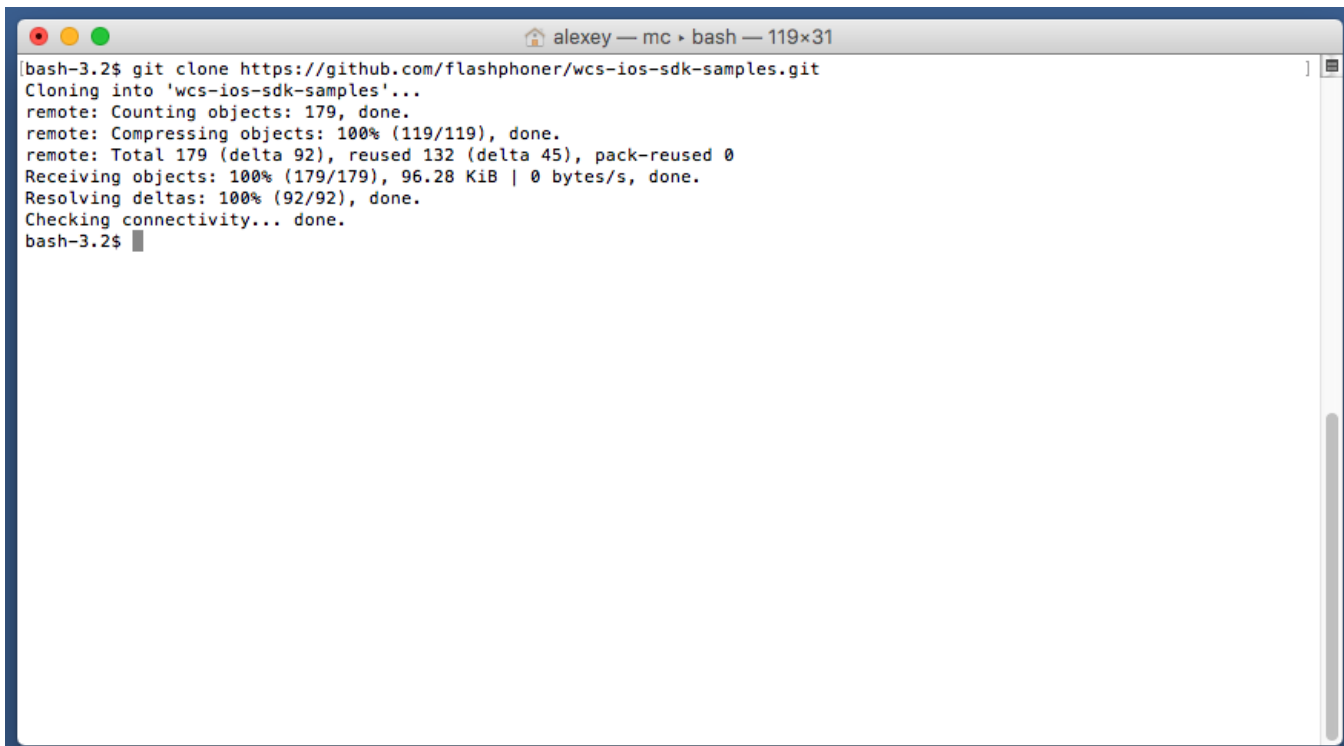
Preparing examples for building

1. Install Cocoapods to build dependencies.

```
sudo gem install cocoapods
```

2. Download the source code of the examples for Mac.

```
git clone https://github.com/flashphoner/wcs-ios-sdk-samples.git
```

A screenshot of a macOS terminal window. The title bar shows 'alexey — mc ▸ bash — 119x31'. The terminal text shows the execution of 'git clone https://github.com/flashphoner/wcs-ios-sdk-samples.git'. The output indicates successful cloning of 179 objects, with a total size of 96.28 KiB. The prompt 'bash-3.2\$' is visible at the end of the last line.

```
alexey — mc ▸ bash — 119x31
bash-3.2$ git clone https://github.com/flashphoner/wcs-ios-sdk-samples.git
Cloning into 'wcs-ios-sdk-samples'...
remote: Counting objects: 179, done.
remote: Compressing objects: 100% (119/119), done.
remote: Total 179 (delta 92), reused 132 (delta 45), pack-reused 0
Receiving objects: 100% (179/179), 96.28 KiB | 0 bytes/s, done.
Resolving deltas: 100% (92/92), done.
Checking connectivity... done.
bash-3.2$
```

3. Download the iOS SDK

```
wget http://flashphoner.com/downloads/builds/flashphoner_client/wcs-ios-sdk/WCS-iOS-SDK-2.2.2.tar.gz
```

```
alexey — mc • bash — 119x31
[bash-3.2$ wget http://flashphoner.com/downloads/builds/flashphoner_client/wcs-ios-sdk/WCS-iOS-SDK-2.2.2.tar.gz
--2016-11-18 21:06:20-- http://flashphoner.com/downloads/builds/flashphoner_client/wcs-ios-sdk/WCS-iOS-SDK-2.2.2.tar.g
z
Resolving flashphoner.com (flashphoner.com)... 216.224.182.33
Connecting to flashphoner.com (flashphoner.com)|216.224.182.33|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11505812 (11M) [application/x-gzip]
Saving to: <<WCS-iOS-SDK-2.2.2.tar.gz>>

WCS-iOS-SDK-2.2.2.tar.gz      100%[=====] 10.97M  405KB/s   in 19s

2016-11-18 21:06:40 (597 KB/s) - <<WCS-iOS-SDK-2.2.2.tar.gz>> saved [11505812/11505812]

bash-3.2$
```

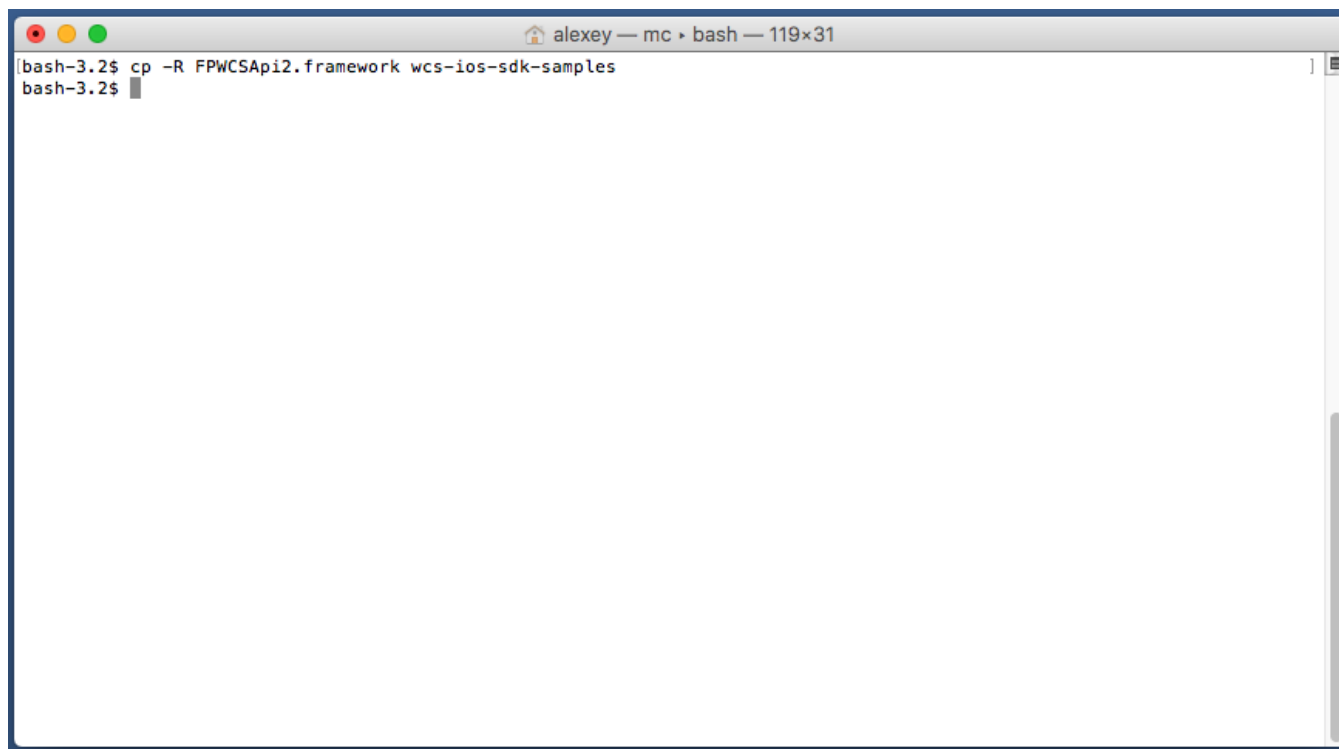
4. After unpacking, the iOS SDK is a framework in the FPWCsApi2.framework folder

```
tar -xvzf WCS-iOS-SDK-2.2.2.tar.gz
```

```
alexey — mc • bash — 119x31
[bash-3.2$ tar -xvzf WCS-iOS-SDK-2.2.2.tar.gz
x FPWCsApi2.framework/
x FPWCsApi2.framework/FPWCsApi2
x FPWCsApi2.framework/Headers/
x FPWCsApi2.framework/Info.plist
x FPWCsApi2.framework/Version.txt
x FPWCsApi2.framework/Headers/FPWCsApi2.h
x FPWCsApi2.framework/Headers/FPWCsApi2Model.h
x FPWCsApi2.framework/Headers/FPWCsApi2Session.h
x FPWCsApi2.framework/Headers/FPWCsApi2Stream.h
x FPWCsApi2.framework/Headers/RTCAudioSource.h
x FPWCsApi2.framework/Headers/RTCAudioTrack.h
x FPWCsApi2.framework/Headers/RTCAVFoundationVideoSource.h
x FPWCsApi2.framework/Headers/RTCDataChannel.h
x FPWCsApi2.framework/Headers/RTCEAGLVideoView.h
x FPWCsApi2.framework/Headers/RTCFileLogger.h
x FPWCsApi2.framework/Headers/RTCI420Frame.h
x FPWCsApi2.framework/Headers/RTCIceCandidate.h
x FPWCsApi2.framework/Headers/RTCIceServer.h
x FPWCsApi2.framework/Headers/RTCLogging.h
x FPWCsApi2.framework/Headers/RTCMediaConstraints.h
x FPWCsApi2.framework/Headers/RTCMediaSource.h
x FPWCsApi2.framework/Headers/RTCMediaStream.h
x FPWCsApi2.framework/Headers/RTCMediaStreamTrack.h
x FPWCsApi2.framework/Headers/RTCNSGLVideoView.h
x FPWCsApi2.framework/Headers/RTCOpenGLVideoRenderer.h
x FPWCsApi2.framework/Headers/RTCPair.h
x FPWCsApi2.framework/Headers/RTCPeerConnection.h
x FPWCsApi2.framework/Headers/RTCPeerConnectionDelegate.h
```

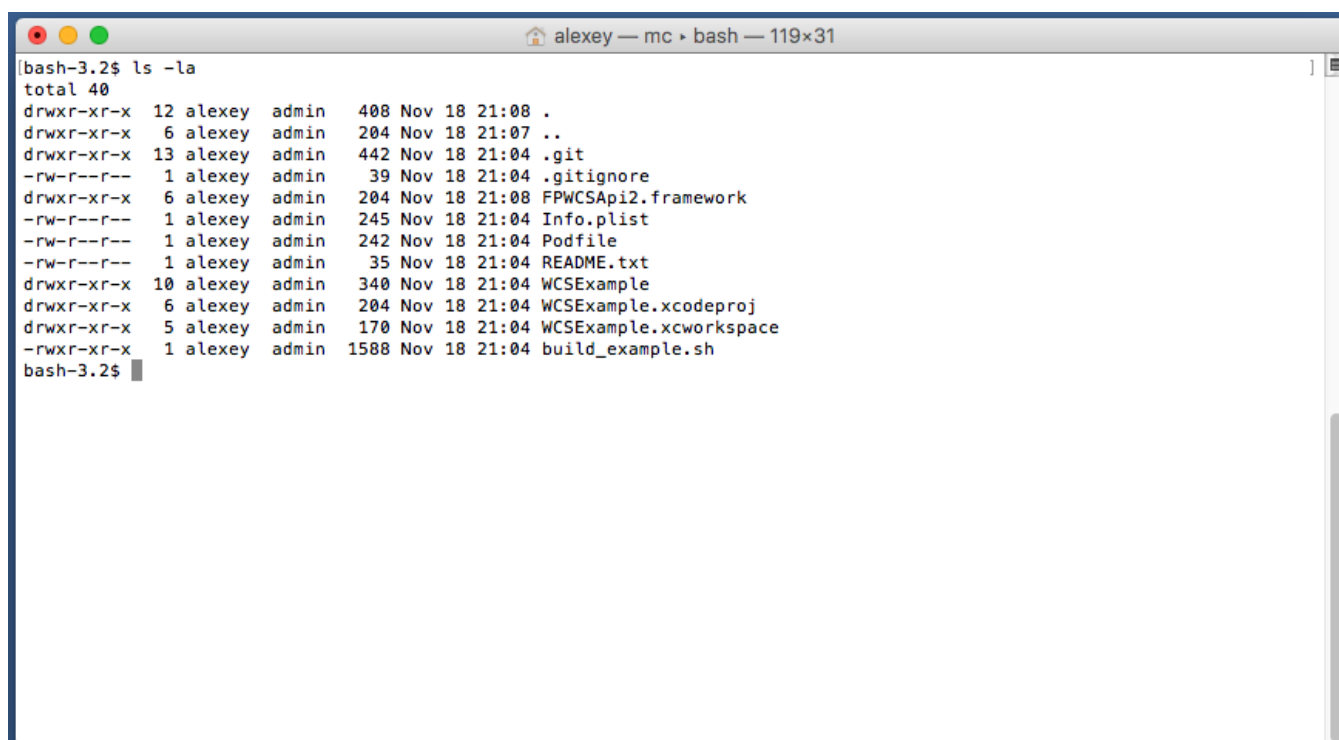
5. Copy the unpacked iOS SDK (framework) to the sample folder

```
cp -R FPWCSEApi2.framework wcs-ios-sdk-samples
```



A terminal window titled 'alexey — mc • bash — 119x31'. The prompt is '[bash-3.2\$]'. The command 'cp -R FPWCSEApi2.framework wcs-ios-sdk-samples' has been entered and executed. The prompt is now '[bash-3.2\$]' again, indicating the command was successful.

6. As a result, we have got the sample folder with the framework (iOS SDK). Now we can start building.

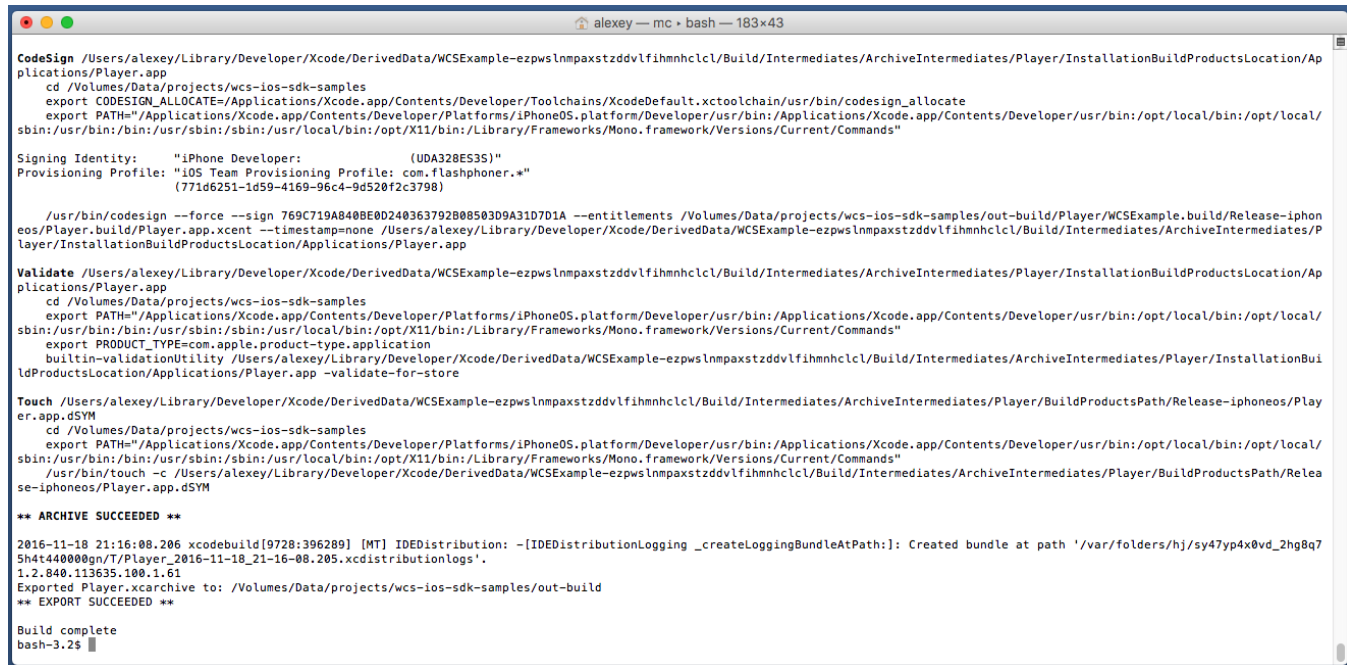


A terminal window titled 'alexey — mc • bash — 119x31'. The prompt is '[bash-3.2\$]'. The command 'ls -la' has been entered and executed. The output shows the contents of the current directory with permissions, owner, group, size, date, and filename.

```
[bash-3.2$] ls -la
total 40
drwxr-xr-x 12 alexey admin 408 Nov 18 21:08 .
drwxr-xr-x  6 alexey admin 204 Nov 18 21:07 ..
drwxr-xr-x 13 alexey admin 442 Nov 18 21:04 .git
-rw-r--r--  1 alexey admin  39 Nov 18 21:04 .gitignore
drwxr-xr-x  6 alexey admin 204 Nov 18 21:08 FPWCSEApi2.framework
-rw-r--r--  1 alexey admin 245 Nov 18 21:04 Info.plist
-rw-r--r--  1 alexey admin 242 Nov 18 21:04 Podfile
-rw-r--r--  1 alexey admin  35 Nov 18 21:04 README.txt
drwxr-xr-x 10 alexey admin 340 Nov 18 21:04 WCSEExample
drwxr-xr-x  6 alexey admin 204 Nov 18 21:04 WCSEExample.xcodeproj
drwxr-xr-x  5 alexey admin 170 Nov 18 21:04 WCSEExample.xcworkspace
-rwxr-xr-x  1 alexey admin 1588 Nov 18 21:04 build_example.sh
bash-3.2$
```

7. Run the build script and wait while all dependencies and examples are built

```
./build_example.sh
```

A terminal window titled 'alexey — mc — bash — 183x43' showing the execution of the build script. The script performs several steps: CodeSign, Signing Identity, Provisioning Profile, Validate, Touch, and Archive. It includes paths to various system and project files. The output shows the successful completion of the build and archive process.

```
CodeSign /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/Player/InstallationBuildProductsLocation/Applications/Player.app
  cd /Volumes/Data/projects/wcs-ios-sdk-samples
  export CODESIGN_ALLOCATE=/Applications/Xcode.app/Contents/Developer/Toolchains/XcodeDefault.xctoolchain/usr/bin/codesign_allocate
  export PATH="/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin:/Applications/Xcode.app/Contents/Developer/usr/bin:/opt/local/bin:/opt/local/sbin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin:/Library/Frameworks/Mono.framework/Versions/Current/Commands"

Signing Identity:      "iPhone Developer:      (UDA328E535)"
Provisioning Profile:  "iOS Team Provisioning Profile: com.flashphoner.*"
                      (771d6251-1d59-4169-96c4-9d520f2c3798)

  /usr/bin/codesign --force --sign 769C719A8408E0D240363792B00503D9A31D7D1A --entitlements /Volumes/Data/projects/wcs-ios-sdk-samples/out-build/Player/WCSEExample.build/Release-iphon
eos/Player.build/Player.app.xcent --timestamp=none /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/P
layer/InstallationBuildProductsLocation/Applications/Player.app

Validate /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/Player/InstallationBuildProductsLocation/App
lications/Player.app
  cd /Volumes/Data/projects/wcs-ios-sdk-samples
  export PATH="/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin:/Applications/Xcode.app/Contents/Developer/usr/bin:/opt/local/bin:/opt/local/
sbin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin:/Library/Frameworks/Mono.framework/Versions/Current/Commands"
  export PRODUCT_TYPE=com.apple.product-type.application
  builtin-validationUtility /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/Player/InstallationBui
ldProductsLocation/Applications/Player.app --validate-for-store

Touch /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/Player/BuildProductsPath/Release-iphon
eos/Player.app.dSYM
  cd /Volumes/Data/projects/wcs-ios-sdk-samples
  export PATH="/Applications/Xcode.app/Contents/Developer/Platforms/iPhoneOS.platform/Developer/usr/bin:/Applications/Xcode.app/Contents/Developer/usr/bin:/opt/local/bin:/opt/local/
sbin:/usr/bin:/bin:/usr/sbin:/sbin:/usr/local/bin:/opt/X11/bin:/Library/Frameworks/Mono.framework/Versions/Current/Commands"
  /usr/bin/touch -c /Users/alexey/Library/Developer/Xcode/DerivedData/WCSEExample-ezpwslmpaxstzddvlfihmhclcl/Build/Intermediates/ArchiveIntermediates/Player/BuildProductsPath/Relea
se-iphon
eos/Player.app.dSYM

** ARCHIVE SUCCEEDED **

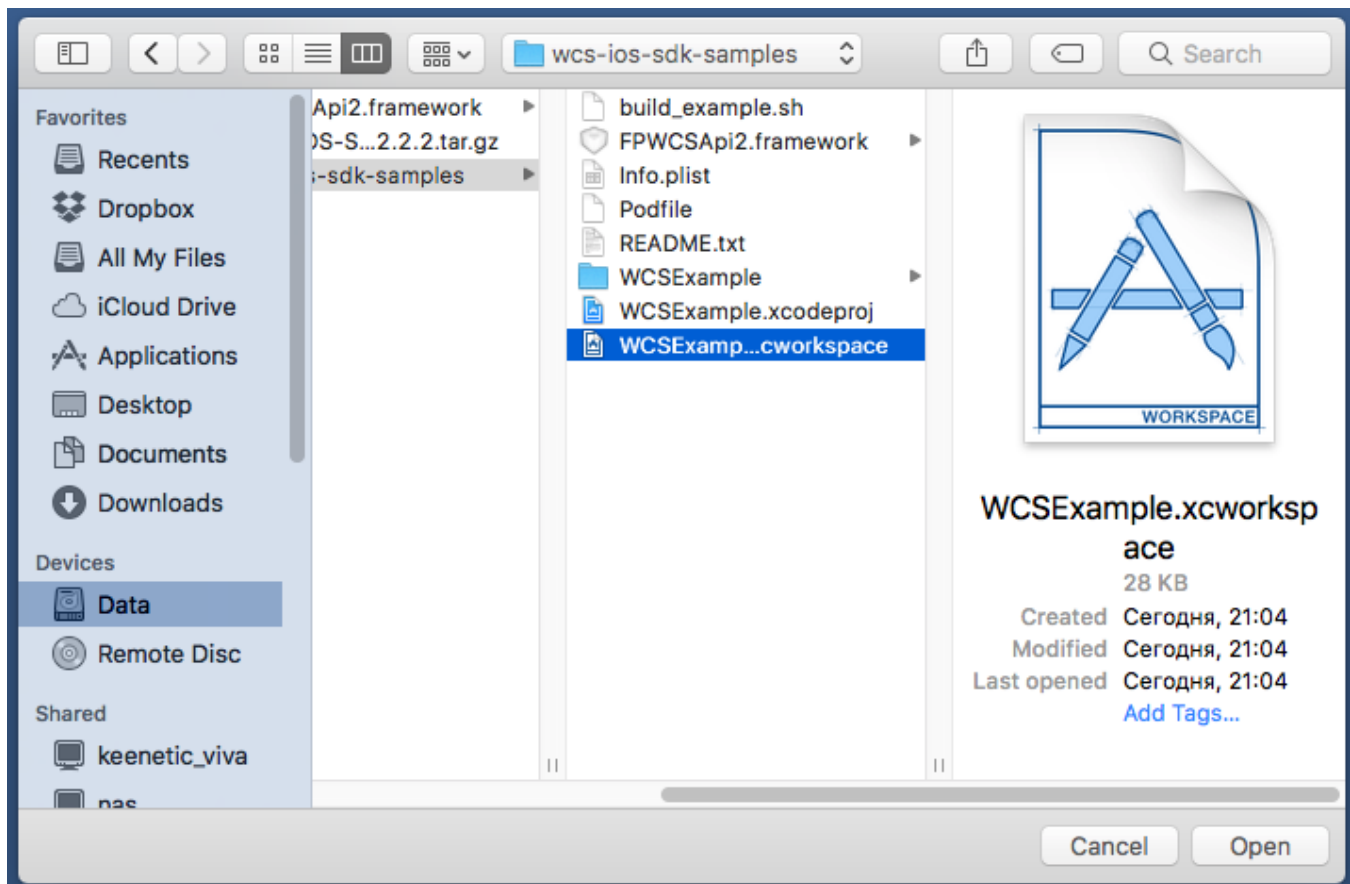
2016-11-18 21:16:08.206 xcodebuild[9728:396289] [MT] IDEDistribution: -[IDEDistributionLogging _createLoggingBundleAtPath:]: Created bundle at path '/var/folders/hj/sy47yp4x8vd_2hg8q7
5h4t440000gn/T/Player_2016-11-18_21-16-08.205.xcdistributionlogs'.
1.2.840.113635.100.1.61
Exported Player.xcarchive to: /Volumes/Data/projects/wcs-ios-sdk-samples/out-build
** EXPORT SUCCEEDED **

Build complete
bash-3.2$
```

Building and launching examples using Xcode

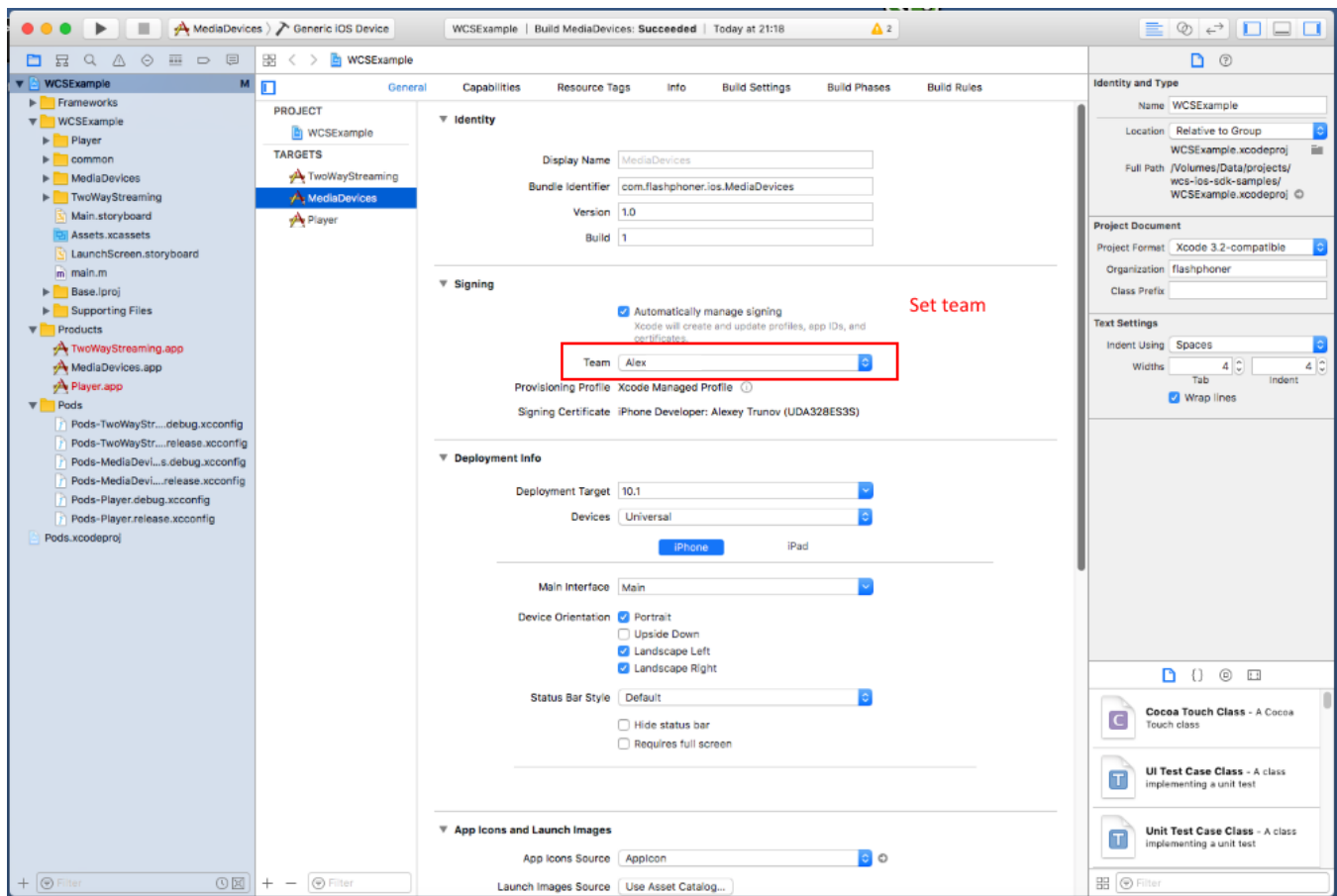
8. Now, as soon as all dependencies are ready (thanks, cocoapods), open workspace in Xcode.

Important! You should open the workspace, not the project file. Otherwise, the build may be broken.

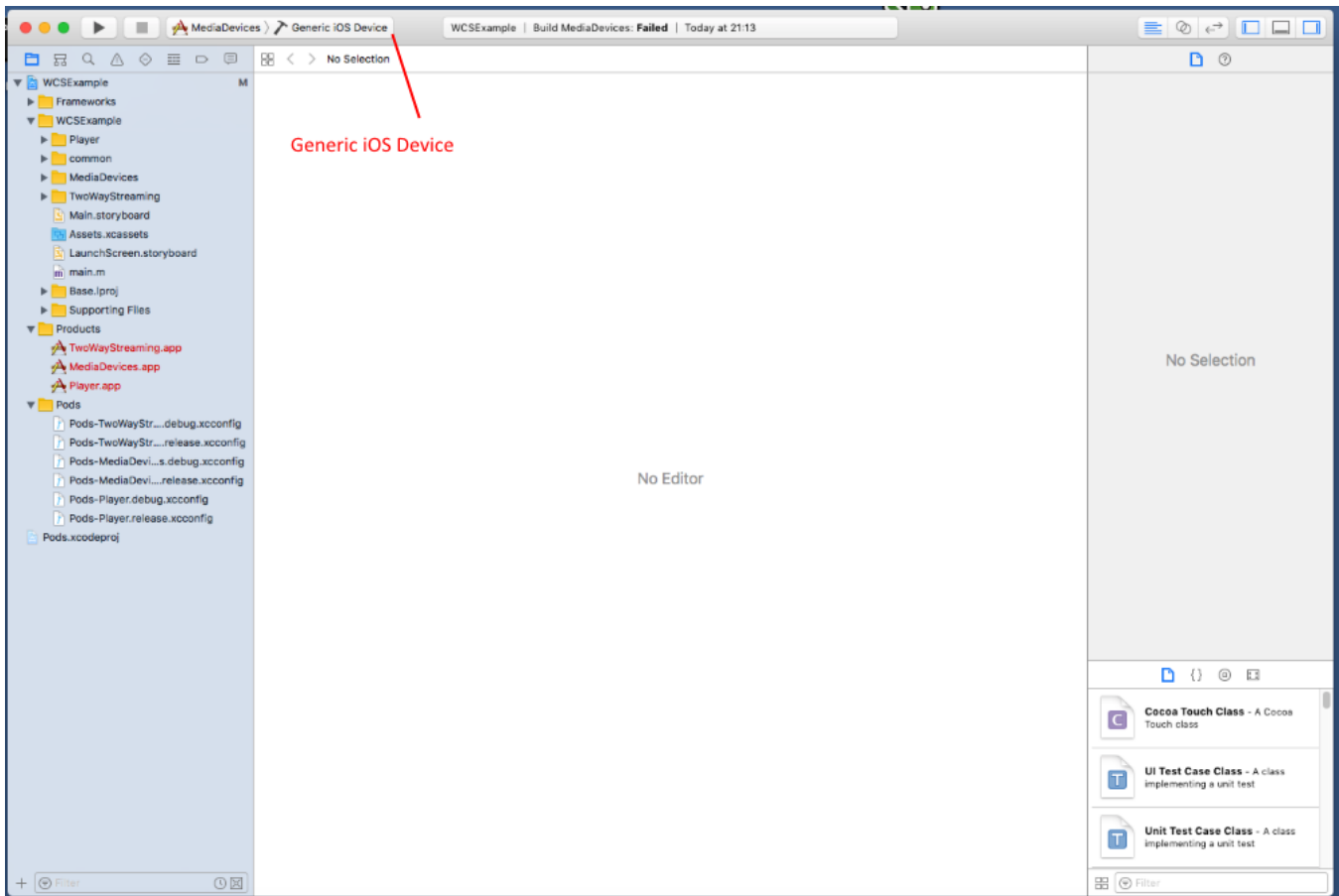


9. If you have 'Setting team' problems on step 7, try specifying the team in Xcode in the settings of each of the compiled projects.

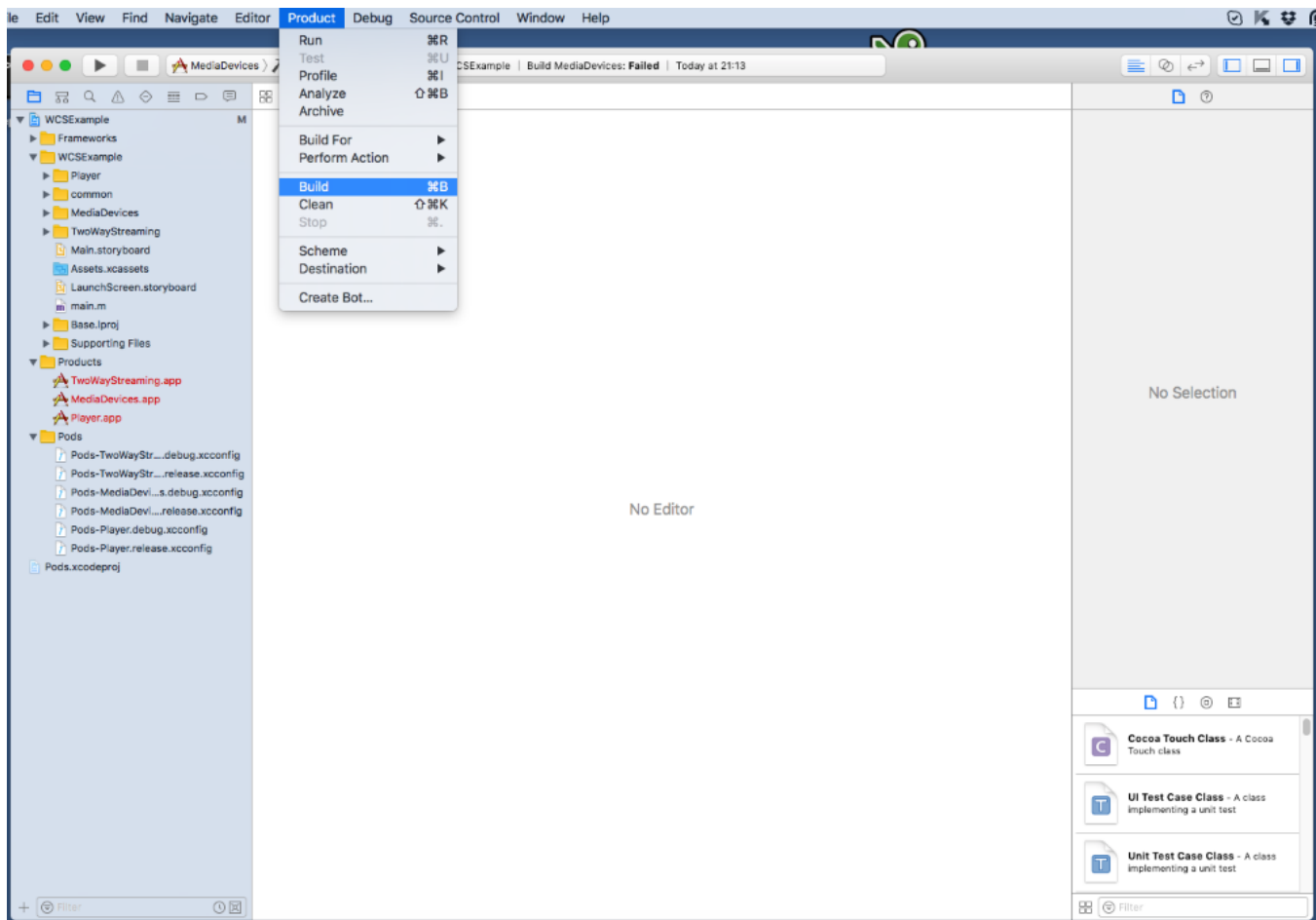
To do this, click WCSEExample in the left menu. After specifying the Team, build the examples again using the script as shown on the step 6.



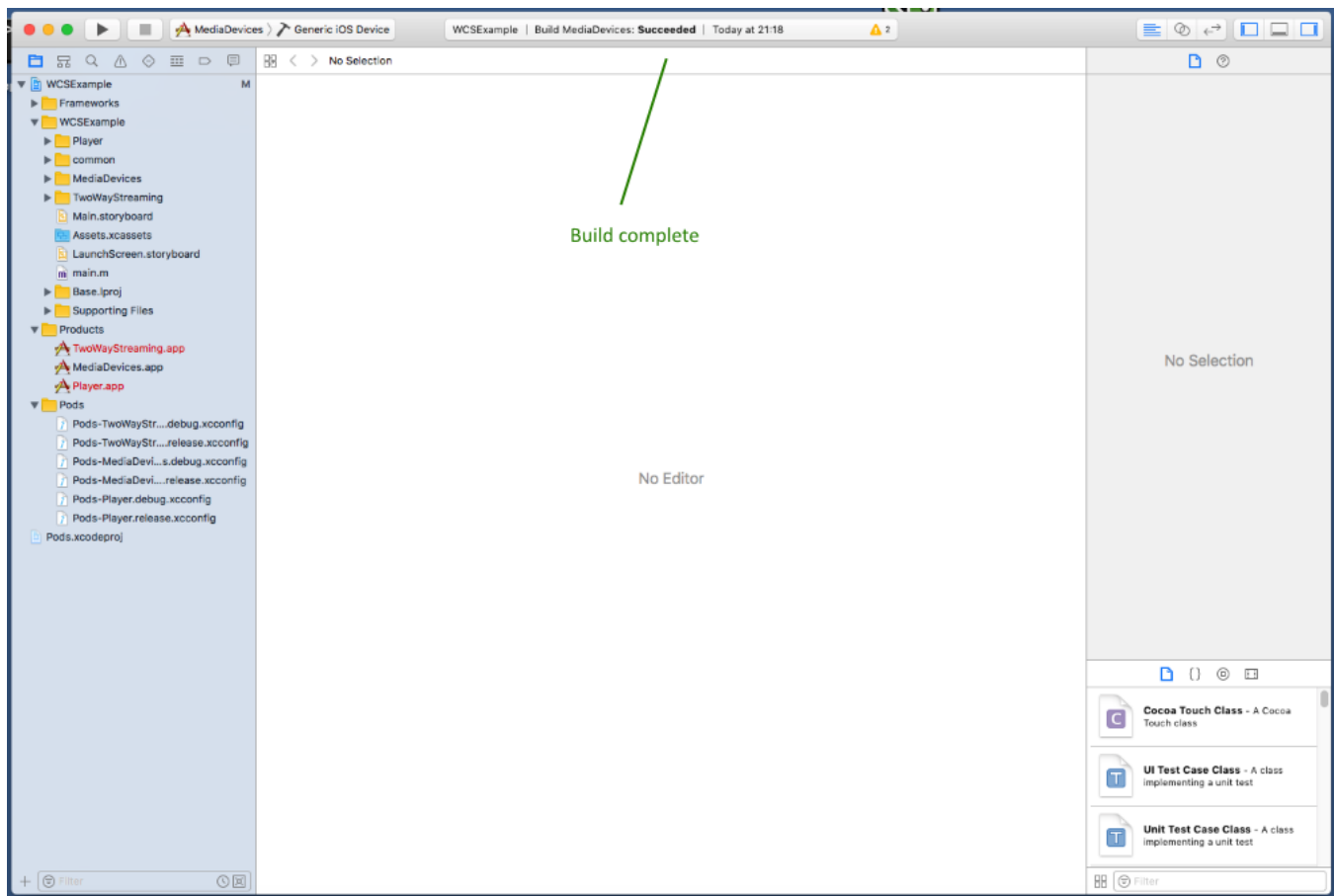
10. Now, we build the Media Devices example in Xcode. To do this, select Generic iOS Device in the build targets.



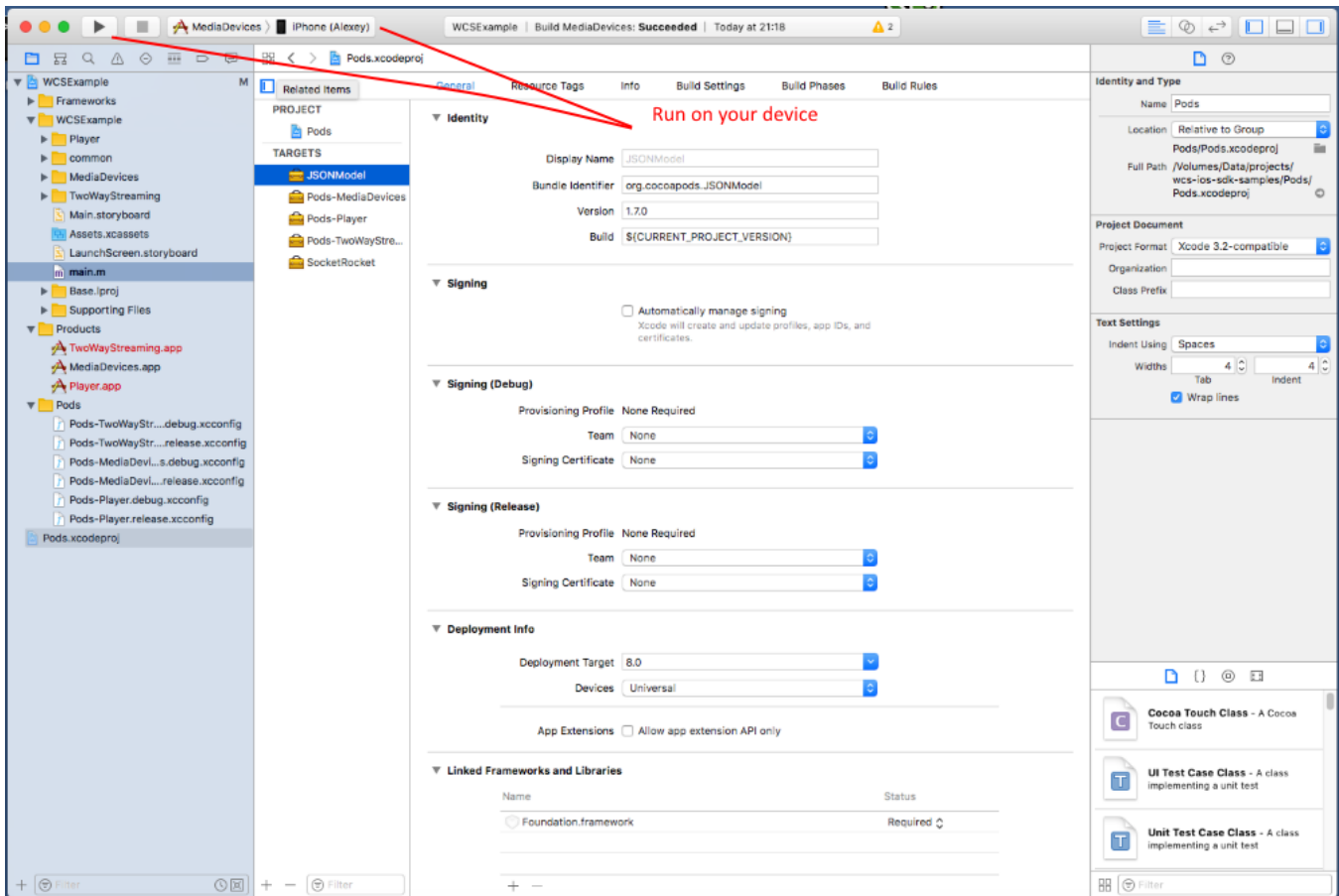
11. Run the build from the Product – Build menu



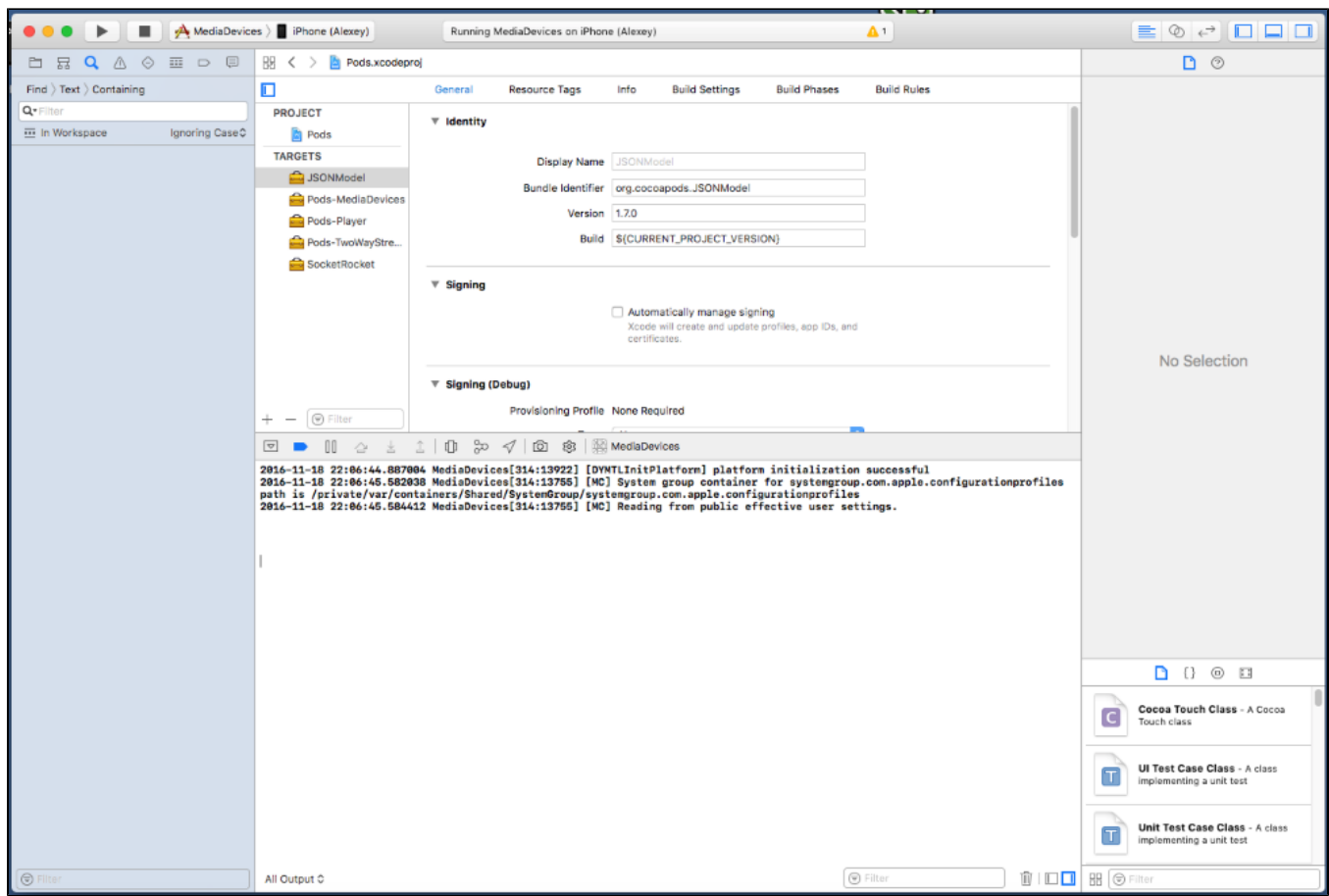
12. If the build successfully completes, you should see the Build MediaDevices message:Succeeded



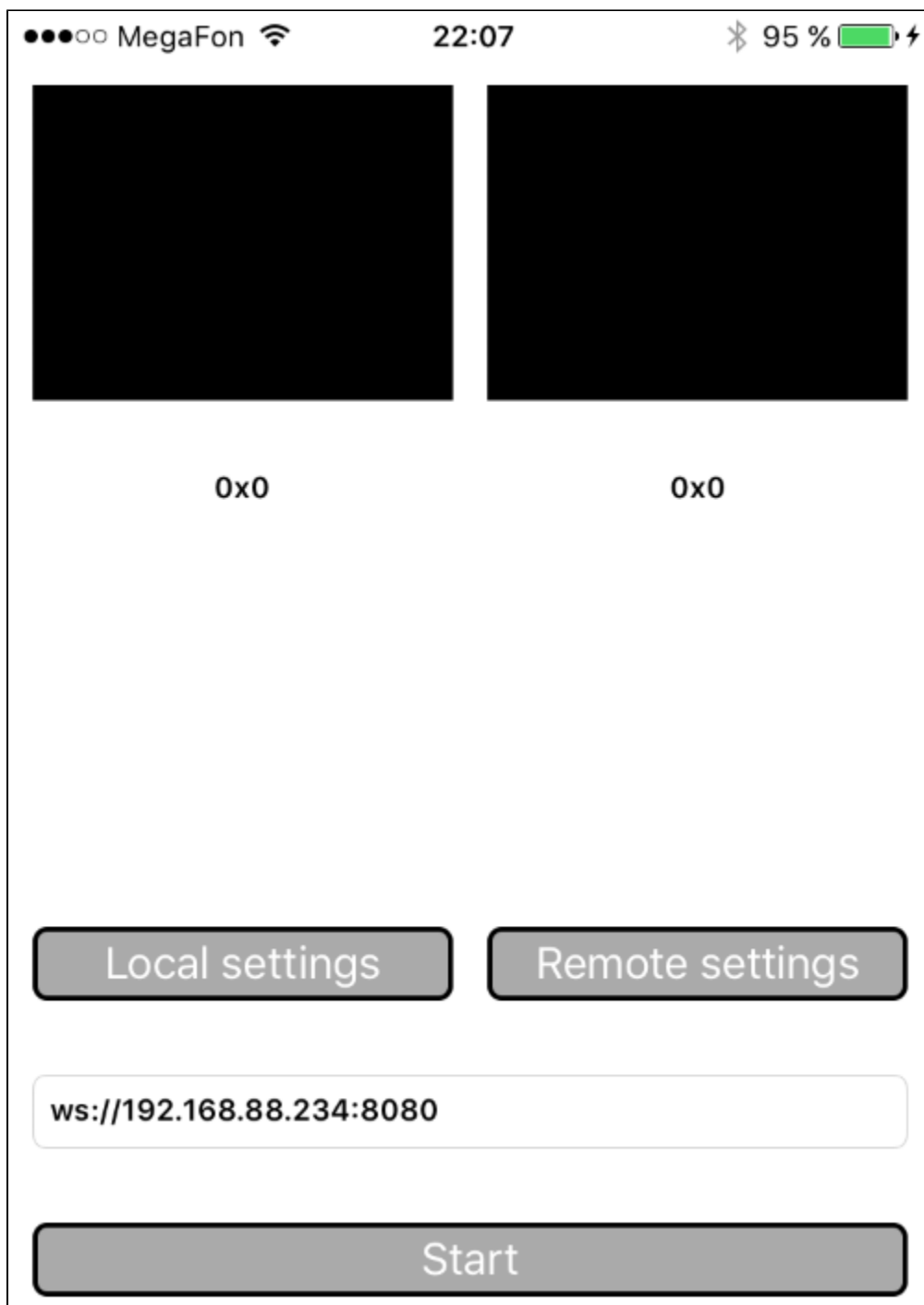
13. Connect your iPhone or iPad via the USB and select it in the targets to run the Media Devices example



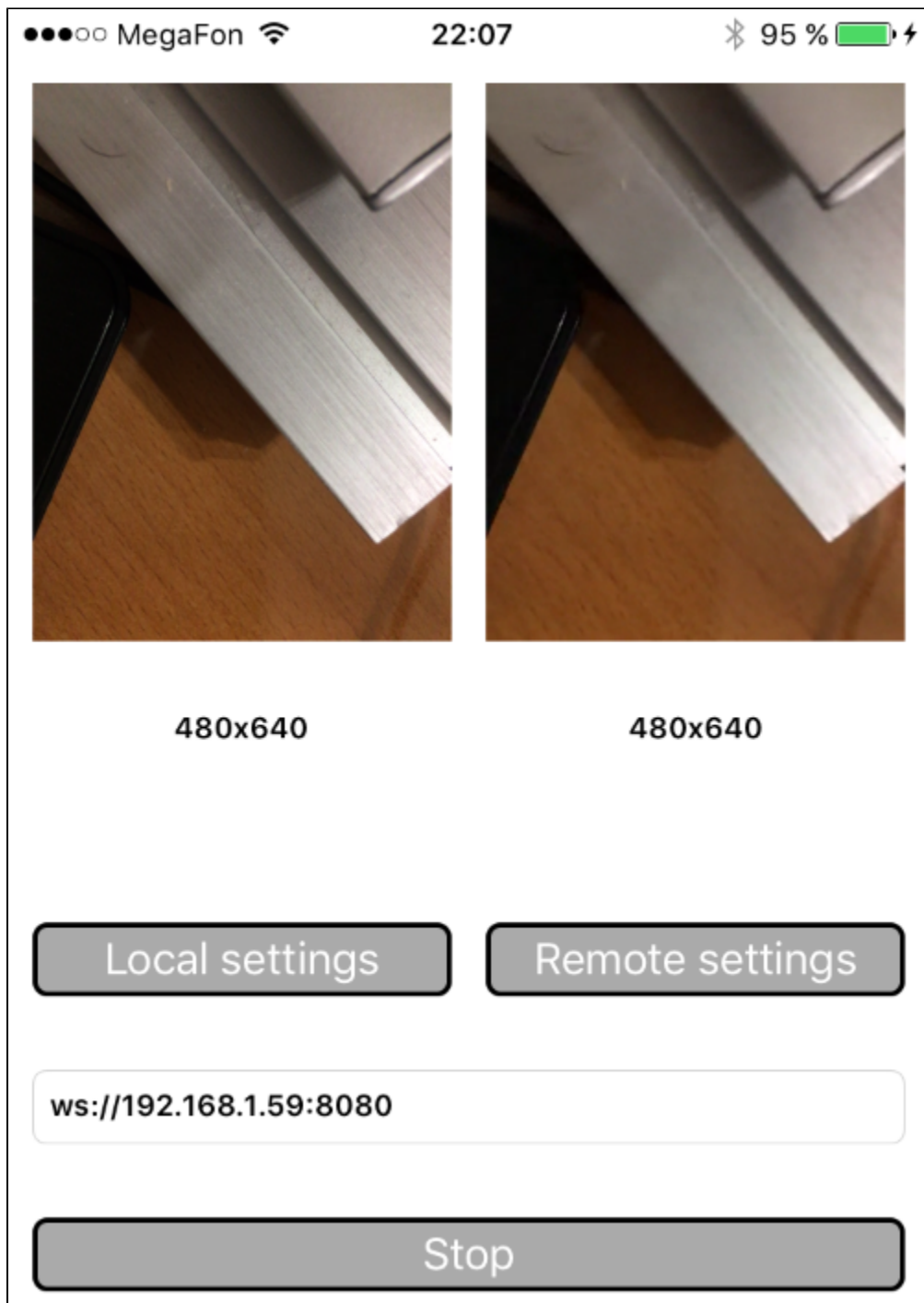
14. After successful run, the debug information is displayed in the lower part. This means, the Media Devices example has been correctly installed to iPhone or iPad and is running.



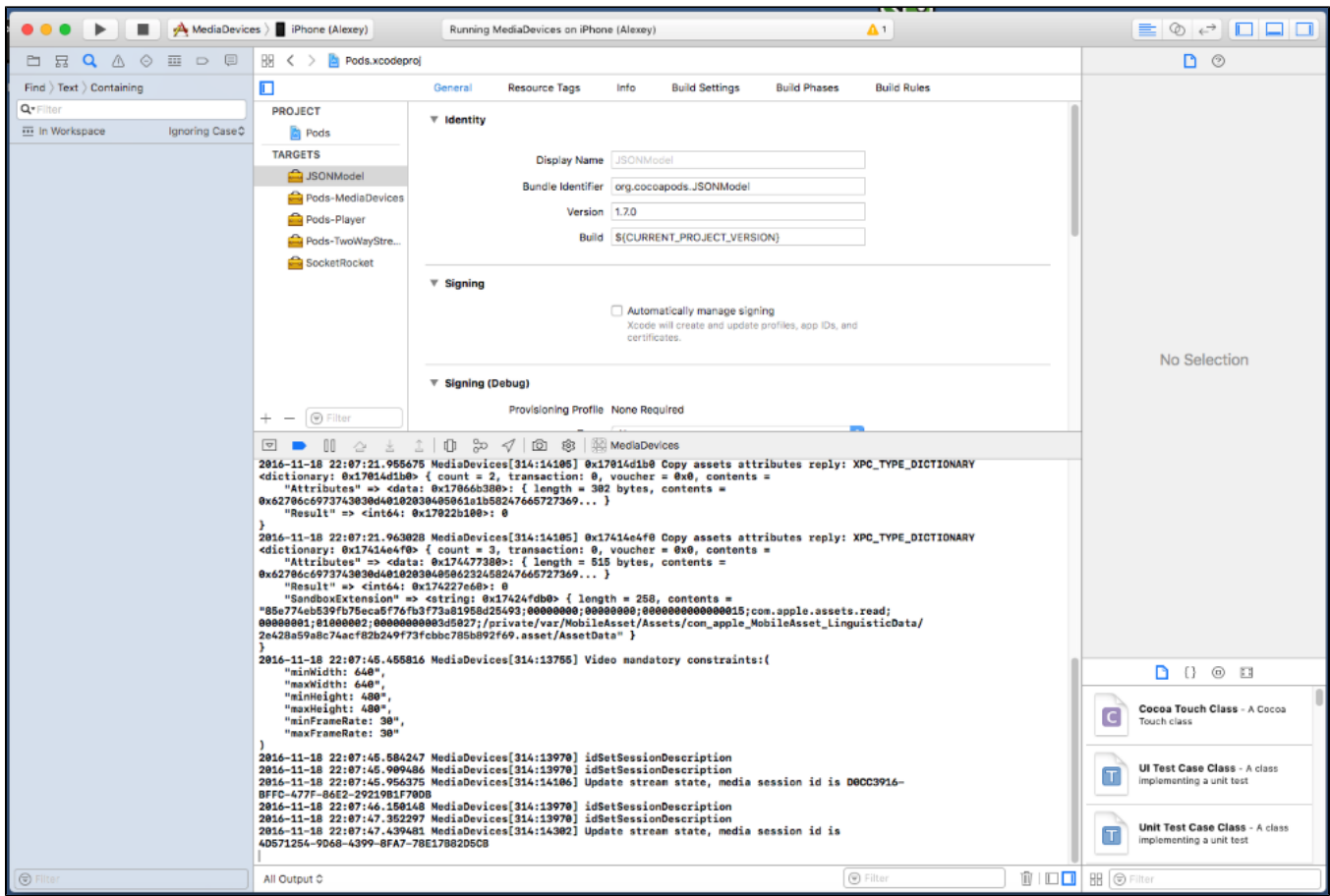
15. On iPhone, you should see the interface of the application you can start testing using the WCS server



16. Connect to the server and send a video stream from the web camera to the iPhone.



17. In Xcode logs we can see the debug information.



So, we have built Media Devices on Mac OS Sierra from the source code using the iOS SDK (FPWCSApi2.framework) and executed this example on iPhone 6 working under iOS 10.1.1. The example demonstrated successful streaming of a video through Web Call Server 5.