

写在前面：

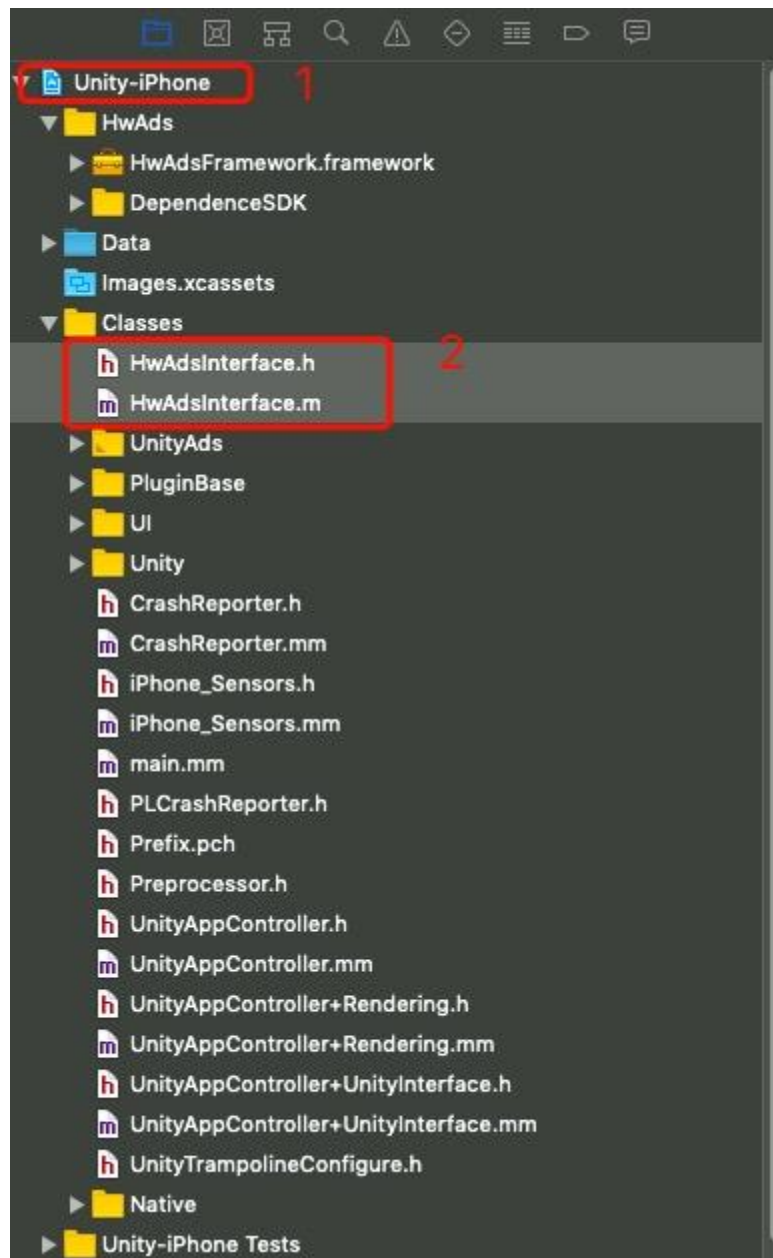
本文档中有 3 个值，与项目相关，需要【找对接人员要】：

1. 初始化接口，服务端的参数
2. Admob 的 app id
3. Facebook 的 app id

1 iOS SDK 接入

对于 unity，首先需要将 unity 接口中的 3 个类文件放到游戏中，与游戏进行关联，再导出 xcode 工程，在 xcode 工程中做相应的修改。

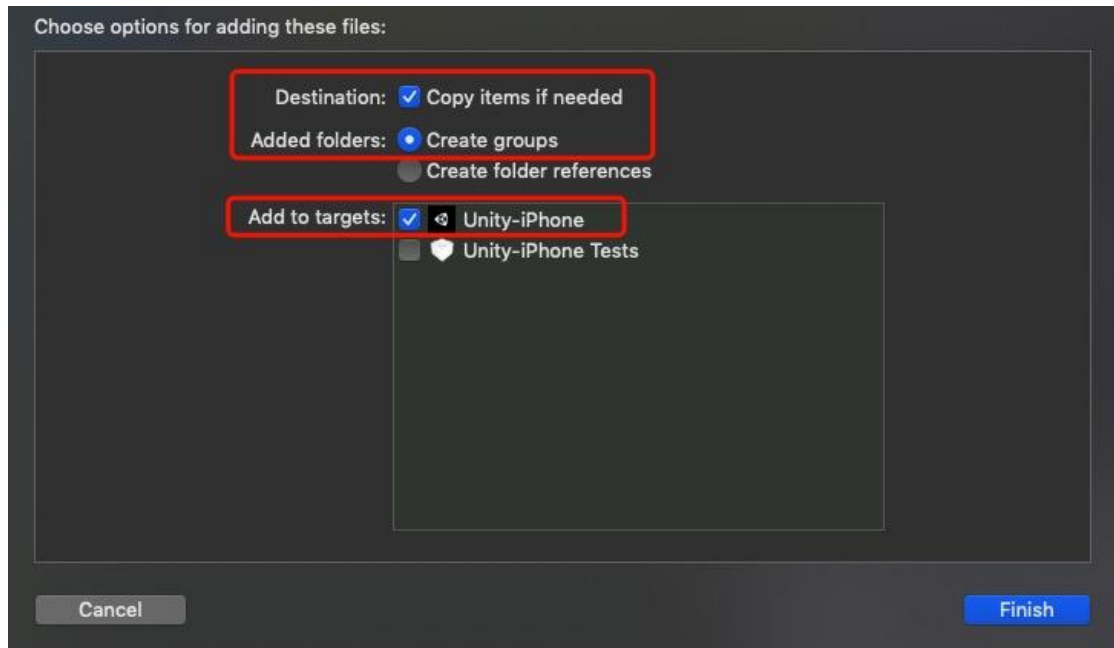
HwAdsInterface.h 和 HwAdsInterface.m 是示例代码，也是 unity 调用 iOS 这边的“桥梁”。iOS 开发的游戏，可以参考这两个类；unity 开发的游戏，**必须**将这两个类放入到 Classes 文件夹中。



1.1 iOS SDK 导入 framework

1. 导入 framework

在 .xcodeproj 同级目录新建一个 HwAds 文件夹，将 DependenceSDK.zip 和 HwAdsFramework.framework.zip 解压，放入到 HwAds 文件夹中（注意在 mac 下解压，framework 在 windows 下解压，文件容易失效）；请确保勾选了 Copy items if needed 和 Create groups，以及 Add to targets 中勾选了：Unity-iPhone



1.2 Xcode 中 info.plist 相关设置

1.2.1 添加 HTTP 权限

在工程 info.plist 中，在 NSAppTransportSecurity 配置为 true

```
<key>NSAppTransportSecurity</key>
<dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
</dict>
```

1.2.2 添加 Applovin 的 key 的配置

在 information Property List 下面，添加 key 值为：**ApplovinSdkKey**；类型为 String，Value 为
9hZx6KZPde6GlbP6OX3zn0M3HzROhgMsNxOTGLtv_-AHZzNzURwFRGJkEl6xQTnoPZFf
v7Tv5iAegTb9-WPQni

1.2.3 添加 admob 的 key 值配置

在 information Property List 下面，添加 key 值为：**GADApplicationIdentifier**；类型为 String，Value 为

ca-app-pub-XXXXXXXXXX~XXXXXX[这个值【找对接人员要】，不同的项目这个值

不同]

1.2.4 添加 facebook 相关的值

其中 appid **【找对接人员要】**

1.2.5 汇总

可以右键 info.plist Open As——Source Code

将下面这段代码复制进去；替换红颜色的三个位置的值就好。

```
<key>AppLovinSdkKey</key>
<string>9hZx6KZPde6GIbp6OX3zn0M3HzROhgMsNxoTGLtv_-AHZzNzURwFRGJk
El6xQTnoPZFfv7Tv5iAegTb9-WPQni</string>
  <key>GADIsAdManagerApp</key>
  <true/>
  <key>GADApplicationIdentifier</key>
  <string>ca-app-pub-5470400114155059~3256815974</string>
  <key>NSAppTransportSecurity</key>
  <dict>
    <key>NSAllowsArbitraryLoads</key>
    <true/>
  </dict>
  <key>CFBundleURLTypes</key>
  <array>
    <dict>
      <key>CFBundleURLSchemes</key>
      <array>
        <string>fb570422036837896</string>
      </array>
    </dict>
  </array>
  <key>FacebookAppID</key>
  <string>570422036837896</string>
  <key>FacebookDisplayName</key>
  <string>shopping mall tycoon</string>
  <key>LSApplicationQueriesSchemes</key>
  <array>
    <string>fbapi</string>
    <string>fb-messenger-share-api</string>
    <string>fbauth2</string>
    <string>fbshareextension</string>
  </array>
```

1.3 Xcode 编译选项设置

1.3.1 运行环境配置

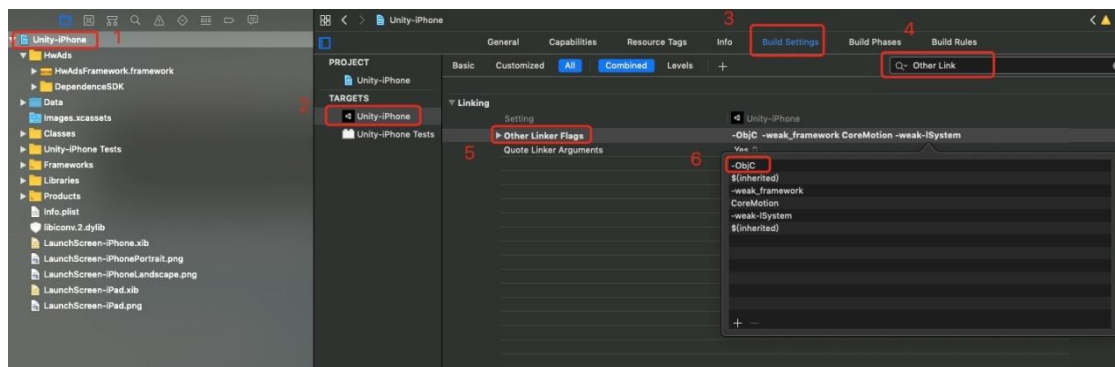
SDK 编译环境 9.0+;

支持架构: i386, x86-64, armv7, armv7s, arm64

1.3.2 添加 ObjC 链接器标记

在 Xcode 中选择项目的 Targets->Build Settings, 配置 Other Link Flags, 增加 -ObjC

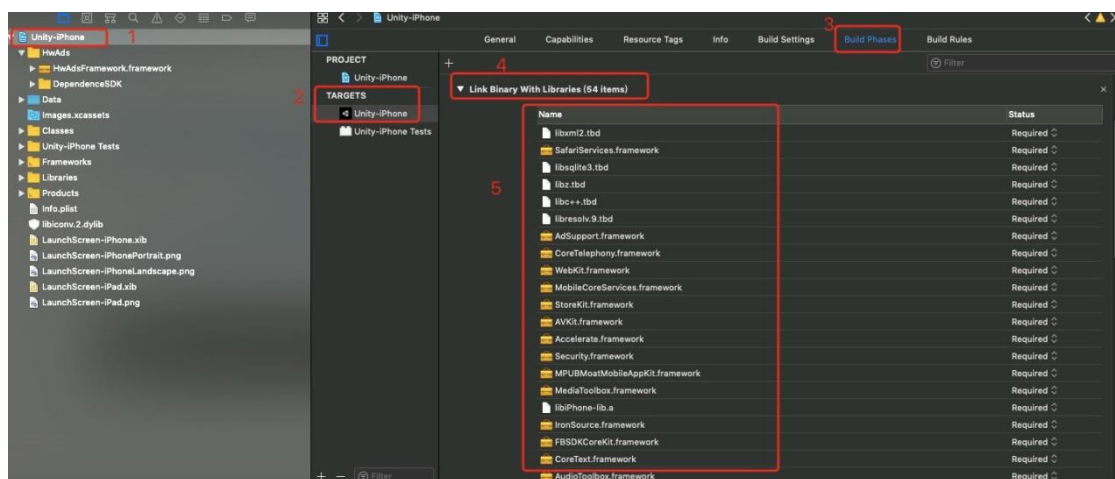
注意: O 大写, C 大写



1.3.3 添加依赖库

工程在 TARGETS 选中项目, 选择 Build Phases 中找到 Link Binary With Libraries, 点击 “+”, 依次添加下列依赖库

若, 前面导入 framework 正确勾选了, 下面很多库会自动带进来; 系统库需要手动添加



- Accelerate.framework
- AdSupport.framework
- AppLovinSDK.framework
- AppsFlyerLib.framework
- AudioToolbox.framework
- AVFoundation.framework

-
- AVKit.framework
 - BUAdSDK.framework
 - CFNetwork.framework
 - CoreGraphics.framework
 - CoreLocation.framework
 - CoreMedia.framework
 - CoreMotion.framework
 - CoreTelephony.framework
 - CoreText.framework
 - CoreVideo.framework
 - FBAudienceNetwork.framework
 - FBSDKCoreKit.framework
 - Foundation.framework
 - GoogleAppMeasurement.framework
 - GoogleMobileAds.framework
 - GoogleUtilities.framework
 - HwAdsFramework.framework
 - IronSource.framework
 - libAvid-mopub-3.6.1.a
 - libc++.tbd
 - libGDTMobSDK.a
 - libil2cpp.a
 - libilconv.2.dylib
 - libiPhone-lib.a
 - libresolv.9.tbd
 - libsqlite3.tbd
 - libxml2.tbd
 - libz.tbd
 - MediaPlayer.framework
 - MediaToolbox.framework
 - Metal.framework
 - MobileCoreServices.framework
 - MPUBMoatMobileAppKit.framework
 - nanopb.framework
 - OpenAL.framework
 - OpenGL.framework
 - QuartzCore.framework
 - SafariServices.framework
 - Security.framework
 - StoreKit.framework
 - SystemConfiguration.framework
 - UIKit.framework
 - UnityAds.framework
 - VungleSDK.framework

-
- WebKit.framework
 - WindSDK.framework

2 SDK 接口类介绍

2.1 整体 API 介绍

```
@interface HwAds: NSObject<...>
```

```
@protocol HwAdsDelegate <NSObject>
```

```
@optional
```

```
//加载成功 添加 delegate
```

```
- (void)hwAdsRewardedVideoLoadSuccess;
```

```
//加载失败
```

```
- (void)hwAdsRewardedVideoLoadFail;
```

```
//加载超时
```

```
- (void)hwAdsRewardedVideoLoadExpire;
```

```
//播放失败，不给奖励
```

```
- (void)hwAdsRewardedVideoPlayFail;
```

```
//广告展示
```

```
- (void)hwAdsRewardedVideoDidAppear;
```

```
//广告关闭
```

```
- (void)hwAdsRewardedVideoClose;
```

```
//广告被点击
```

```
- (void)hwAdsRewardedVideoClick;
```

```
//广告播放完成，给奖励，最好在这里做标记，在 close 中给
```

```
- (void)hwAdsRewardedVideoGiveReward;
```

```
@end
```

```
/**
```

```
*注册代理，这个代理是可选的
```

```
*/
```

```
@property (nonatomic, weak) id<HwAdsDelegate> hwAdsDelegate;
```

```
//单例对象
```

```
+ (id)instance;
```

```
//初始化接口，游戏中唯一需要调用的接口，传入参数为 serverURL, 【找对接人员要】
```

```
- (void)initSDK:(NSString *)serverURL;
```

//加载插屏接口，不需要游戏中调用；sdk 内部会自动加载广告，看完之后会加载下一条，即使遇到异常，也会过一段时间重新加载广告

```
- (void)loadInter;
```

//显示插屏接口，调用前，最好调用 isInterLoad 判断插屏是否加载，加载了就调用 show 方法

```
- (void)showInter;
```

```
//判断插屏是否加载，返回值为 bool
- (BOOL)isInterLoad;
//加载激励接口，不需要游戏中调用；sdk 内部会自动加载广告，看完之后会加载下一条，即使遇到异常，也会过一段时间重新加载广告
- (void)loadReward;
//显示激励方法，需要传入一个 string 类型的参数，用于区分激励的不同的位置：如：双倍传“double”，复活用“review”等等
- (void)showReward:(NSString *)tag;
//判断激励是否加载，调用显示方法前，通过这个 api 判断
- (BOOL)isReward;
//fb 打点，参数是 3 个字符串，可以根据需要将游戏中的关键的点，调用这个方法
- (void)hwFbEvent:(NSString *)category
    action:(NSString *)action
    label:(NSString *)label;
```

3 示例代码

见 HwAdsInterface.h 和 HwAdsInterface.m 两个文件

4 常见错误

- 1.错误 Multiple commands produce README.md
解决方法：将 File——Project Settings 中 build system 修改为 Legacy Build System
- 2.错误 Cannot use ‘@try’ with Objective-C exceptions disabled 的错误
解决方法：选中项目-Build Settings 选项卡-Apple Clang-Language-Objective-C Enable Objective-C Exceptions 将 No 改为 Yes