



AdView

iOS SDK

开 发 者 手 册

目录

| | |
|--------------------------|----|
| I. 注册并获取SDK | 3 |
| II. 添加SDK | 6 |
| III. 创建横幅广告 | 8 |
| IV.创建插屏广告 | 9 |
| V.创建开屏广告 | 11 |
| VI.创建原生广告 | 13 |
| VII.创建视频广告 | 15 |
| VIII.创建直投和交换广告及投放 | 17 |
| IX.配置Xcode工程 | 21 |
| X.AdView后台应用设置 | 26 |
| XI.离线配置文件 | 29 |
| XII.删除不需要的广告平台 | 30 |
| XIII.添加自定义广告平台(插屏) | 31 |
| XIV.常见问题处理 | 35 |

I. 注册并获取SDK

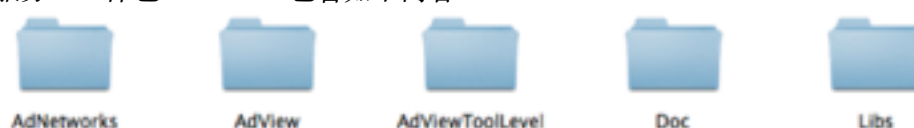
1. 访问AdView网站<http://www.adview.cn>注册AdView账号。
2. 登录后在“我的产品”页面选择“发布应用”。
3. 应用平台选择“iOS”,根据提示完成相关信息，您将获取唯一的SDK key。
4. 在“应用管理”界面，配置自己的应用：“未设置”中输入该平台的广告id，打开“开关”，设置各平台投放量总和为100%，点击“保存”。如需更多平台，点击右上角的“添加广告平台”。一般建议开启平台数为1-3个。



5. 首页-> iOS SDK下载，或者应用管理-> iOS SDK下载，能获取AdViewSDK_iOS包。



6. 一体包 iOS SDK是专为iOS开发者免费提供的移动广告优化工具，旨在帮助开发者轻松实现广告收益最大化。基于广告聚合的应用互推，帮助应用迅速扩大用户数。同时我们联合多家广告平台和应用商城，开通绿色通道服务，为广大开发者服务。一体包 iOS SDK 包含如下内容：



| 文件或目录名 | 说明 |
|-----------------|------------------------------------|
| AdView | AdView 一体包SDK开发包。 |
| AdNetworks | 一体包 SDK 支持的广告插屏开发包。 |
| Libs | 一体包依赖的开源库。 |
| Doc | 一体包SDK资料包括：更新日志、使用手册、当前版本和注意事项等文件。 |
| AdViewToolLevel | 一体包中Banner辅助集成工具。 |

说明：

1. 我们提供的SDK可以让您自由选择您喜欢的广告公司，除了竞价补余以外，其他广告平台均为第三方，所需的广告App Key，需要您到这些平台官网上注册申请，并将此配置到AdView后台对应的平台下。
2. 如果你是一个新手，对广告平台不怎么了解，也不知道选择哪家广告平台，或者不知道哪家广告平台收入稳定，**Adview**建议你可以先使用竞价看看效果。
3. 竞价和补余广告需要在后台补充市场信息，通过审核后方能获取正式广告，开始费用，在此之前都是测试广告，不产生计费
4. 开关：只有开关打开的那家平台的广告才会在应用中显示。
5. 投放量：只有开关打开的投放量才有意义，投放量各广告平台的请求优先级，一般是比例高的，优先请求。切换广告平台应同时调整投放量，所有广告状态为on的应用投放量总额应为100，否则无法正确保存
6. 横幅广告、全屏/插屏广告、开屏广告等，每页下面均有一个“保存”键，每修改完一种广告形式，都要点击该页的“保存”键，否则所做的修改无效
7. 地域优化：开启地域优化的功能是指定在国内的手机显示国内配置的广告，在国外的手机显示国外配置的广告；最大限度的满足不同的需求。当地域优化关闭的时候，就不区分国内国外了，设置为开关打开的都可以显示。
8. 如果使用视频广告，如图创建视频广告位：（原生广告同理）

注意：①完善视频广告位后，一定要点击“保存”

②创建成功的应用会生成“视频广告位ID”（使用聚合功能中的视频竞价广告不需要填写广告位ID）。视频广告位ID位置：应用管理→编辑→视频广告位模块

视频广告位 如果您要使用“视频广告”，请创建视频广告位

创建视频广告位

是否创建：☒是 ☐否

视频尺寸：  系统会根据您选定尺寸的接近宽高比进行适配，而不是严格按照尺寸

广告最长时长：☐15s ☐30s ☐45s ☐60s ☒不限

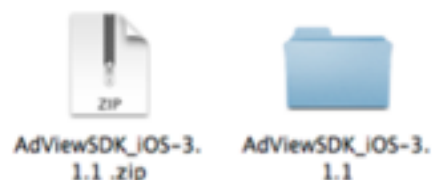
视频格式：☒MP4 ☒FLV ☒SWF

视频类型：☒非贴片视频 ☐贴片视频

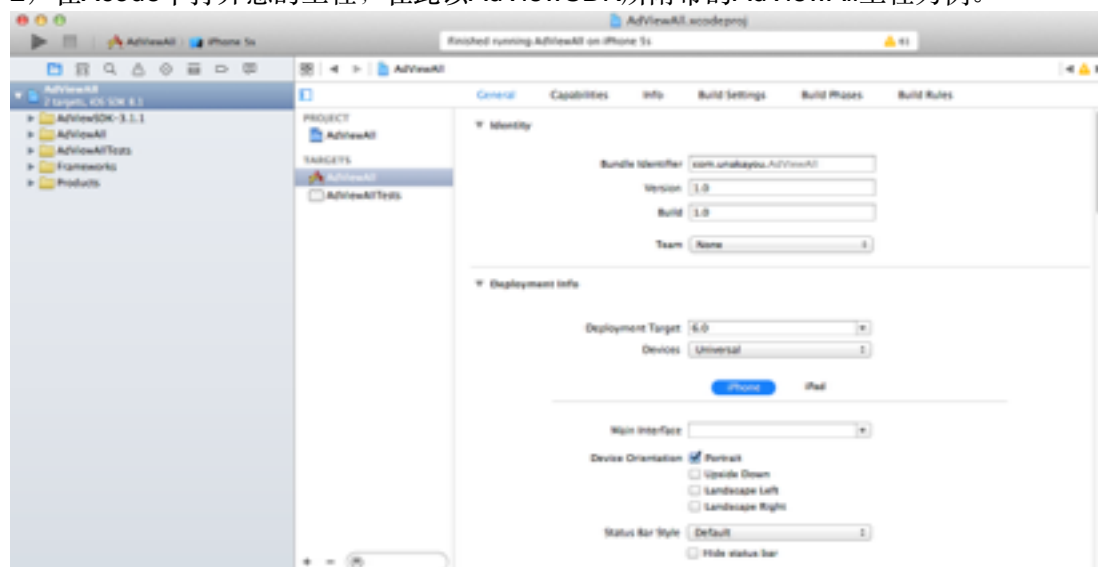
II. 添加SDK

在项目中集成一体包SDK比较容易，通常您按照以下步骤即可把AdInstl集成到您的项目中。

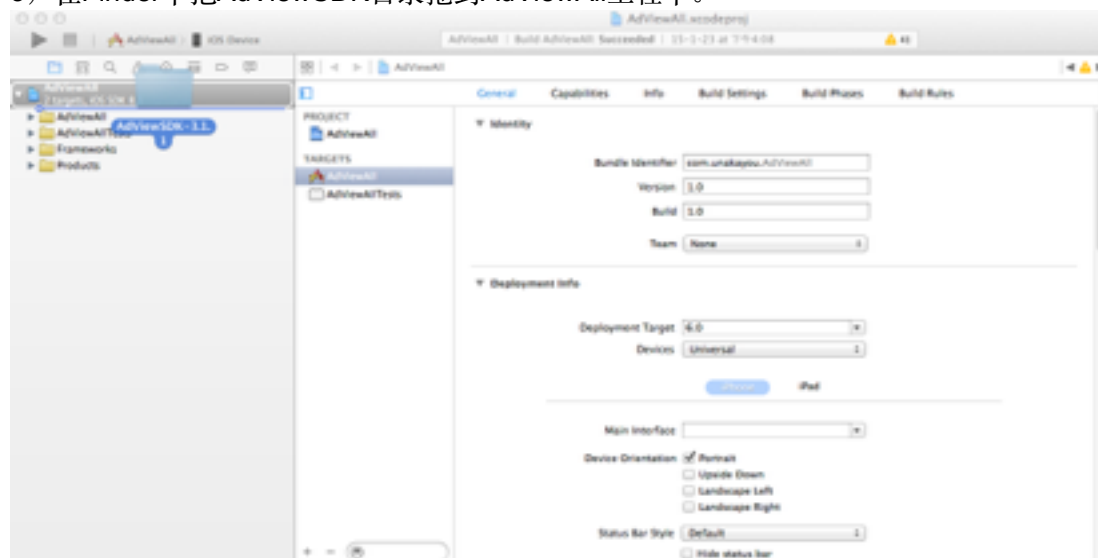
1) 从网站获取AdViewSDK开发包，在Finder中解压。您将在Finder中看到解压后的AdViewSDK，如图所示：



2) 在Xcode中打开您的工程，在此以AdViewSDK所附带的AdViewAll工程为例。

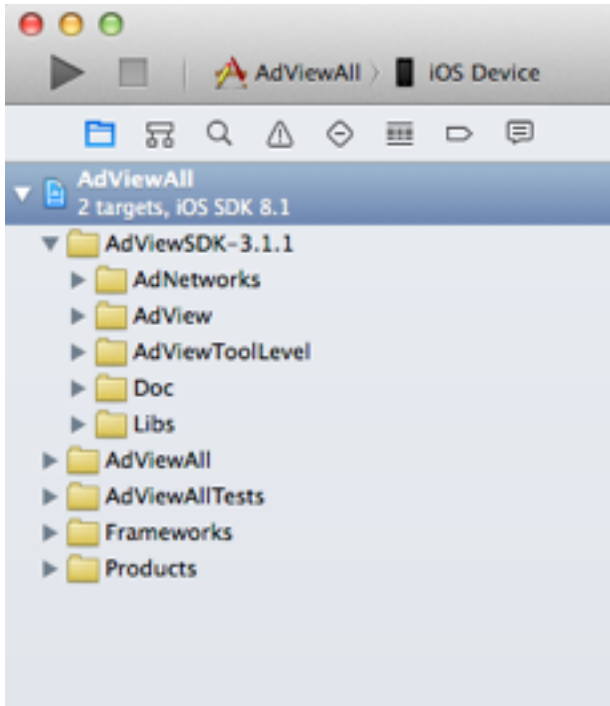


3) 在Finder中把AdViewSDK目录拖到AdViewAll工程中。



4) 在Xcode弹出的对话框中选中“Copy items into destination group's folder (if

needed)”。完成以后AdViewSDK即已经添加到项目中，如图所示：



5) 一些平台对于Xcode版本或者SDK版本有限制，不满足需求请不链接该平台库。

6) AdViewSDK开发包附带了所有支持的广告平台，但是某些广告平台之间可能会有些符号冲突此时您将不得不舍弃一些广告平台。这写广告平台的冲突请见ReadMe.txt。

7) 其他：

- AdViewSDK项目框架调整为ARC模式（3.1.0版本开始），非ARC项目使用时请给相应文件添加-fobjc-arc标签。

- AdViewSDK不直接使用地理位置信息，广告SDK可能使用。

- 部分广告平台可能重复导入某些文件导致重复编译或者冲突问题，详情请看ReadMe.txt文件。

8) 注册配置：

- 须在横幅，插屏等广告初始化之前创建AdViewConfigStore对象并注册横幅，插屏等配置。建议写在AppDelegate的- (BOOL)application:didFinishLaunchingWithOptions:方法内

- 请求广告的key必须在注册的配置之内。

```
AdViewConfigStore *cfg = [AdViewConfigStore sharedStore];
[cfg
requestConfig:@[@"SDK20111022530129m85is43b70r4iyc",@"SDK2015090
3090237sjtskjva8ppy91r"] sdkType:AdViewSDKType_Banner];
[cfg
requestConfig:@[@"SDK20141430020730kbmya6prn5qg1t0",@"SDK2015090
3090237sjtskjva8ppy91r"] sdkType:AdViewSDKType_Instl];
[cfg
requestConfig:@[@"SDK20151311010501u7hth0gexjm9gym",@"SDK2015090
3090237sjtskjva8ppy91r"] sdkType:AdViewSDKType_SpreadScreen];
```

III. 创建横幅广告

1) 在需要嵌入Banner的界面对象中实现AdView的Delegate，如下代码所示：

```
#import <UIKit/UIKit.h>
#import "AdViewView.h"
@interface BannerViewController : UIViewController<AdViewDelegate>
{
    AdViewView *adView;
}
@property (nonatomic,retain)AdViewView *adView;
@end
```

2) 实现AdViewDelegate，如下代码所示：

```
- (UIViewController*)viewControllerForPresentingModalView
{
    return [UIApplication sharedApplication].keyWindow.rootViewController ;
}
/**
 * 用户是否要求输出日志，以前是和adViewTestMode同步的，现在可单独启用
 */
- (BOOL)adViewLogMode
{
    return YES;
}
```

3) 在控制器的viewDidLoad函数中创建AdInstlManager，使用第1步所获取Application key，如下代码所示：

```
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    self.view.backgroundColor =[UIColor blueColor];
    self.adView = [AdViewView
requestAdViewViewWithAppKey:@"SDK201416200410555y9p0pjcph5f5ue"
WithDelegate:self];
    [self.view addSubview:self.adView];
}
```


IV. 创建插屏广告

1) 在需要嵌入插屏的对象中实现插屏的Delegate。如果在不同的界面用到插屏，可以在AppDelegate对象中只实现一次，提供外接方法供调用位置使用。

```
#import <UIKit/UIKit.h>
#import "AdInstlManager.h"
#import "AdInstlManagerDelegate.h"
@interface InstlViewController : UIViewController <AdInstlManagerDelegate>
@property (nonatomic, strong) AdInstlManager * adInstlManager;
@end
```

2) 实现插屏Delegate，如下代码所示：

```
-(void)adInstlManager:(AdInstlManager*)manager didGetEvent:
(InstlEventType)eType error:(NSError*)error
{
    switch (eType) {
        case InstlEventType_DidLoad:
            break;
        case InstlEventType_FailLoadAd:
            break;
        default:
            break;
    }
}
-(BOOL)adInstlTestMode {
    return NO;
}
-(BOOL)adInstlLogMode {
    return YES;
}
```

3) 在控制器viewDidLoad函数中创建AdInstlManager，如下代码所示：

```
-(void)viewDidLoad
{
    [super viewDidLoad];
    self.adInstlManager=[AdInstlManager managerWithAdInstlKey:@" SDK
201416200410555y9p0pjcp5f5ue" WithDelegate:self];
}
```

4) 在合适的位置加入广告加载代码，如下代码所示：

```
-(void)adInstlLoad:(id)sender
{
    // 因插屏广告有一定的生存周期，load后请不要间隔太久才去调用
    (showAdInstlView:)方法，以免广告失效
    BOOL bRet = [self.adInstl loadAdInstlView:self];
}
```

5) 在合适的位置加入广告显示代码，如下代码所示：

```
- (void)adInstlShow:(id)sender
{
    BOOL bRet = NO;
    if ([self.adInstl isReady])
        bRet = [self.adInstl showAdInstlView:self];
}
```

V. 创建开屏广告

- 1) 在AppDelegate中嵌入开屏对象，以及开屏对象生成，如下代码所示：

```
#import <UIKit/UIKit.h>
#import "AdSpreadScreenManager.h"
#import "AdSpreadScreenManagerDelegate.h"
@interface AppDelegate : UIResponder
<UIApplicationDelegate, AdSpreadScreenManagerDelegate>
@property (strong, nonatomic) UIWindow *window;
@property (strong, nonatomic) AdSpreadScreenManager *manager;
@end
```

- 2) AppDelegate中的Application方法中创建AdSpreadScreenManager如下代码所示：

```
- (BOOL) applicationDidFinishLaunchingWithOptions:(NSDictionary*)launchOptions
{
    self.manager=[AdSpreadScreenManager managerWithAdSpreadScreen
        Key:@"SDK20141023100445zhgsncda3wciah6" WithDelegate:self];
    [self.manager requestAdSpreadScreenView:self.window.rootViewController];
    return YES;
}
```

- 3) 实现开屏Delegate，如下代码所示：

```

/**
 * 信息回调
 */
- (void)adSpreadScreenManager:(AdSpreadScreenManager*)manager
didGetEvent:(SpreadScreenEventType)eType error:(NSError*)error{
}
/**
 * 取得配置的回调通知
 */
- (void)adSpreadScreenDidReceiveConfig:(AdSpreadScreenManager*)manager {}
/**
 * 配置全部无效或为空的通知
 */
- (void)adSpreadScreenReceivedNotificationAdsAreOff:
(AdSpreadScreenManager*)manager {}
/**
 * 是否打开测试模式，缺省为NO
 */
- (BOOL)adSpreadScreenTestMode {
    return YES;
}
/**
 * 是否打开日志模式，缺省为NO
 */
- (BOOL)adSpreadScreenLogMode {
    return NO;
}
/**
 * 是否获取地理位置，缺省为NO
 */
- (BOOL)adSpreadScreenOpenGps {
    return NO;
}
/**
 * 是否使用html5广告，缺省为NO
 */
- (BOOL)adSpreadScreenUsingHtml5
{
    return NO;
}
- (UIWindow *)adSpreadScreenWindow {
    return self.window;
}
- (NSString *)adSpreadScreenLogoImgName {
    return @"Adview_Logo.jpg";
}
- (UIColor *)adSpreadScreenBackgroundColor {
    return [UIColor whiteColor];
}
}

```

VI. 创建原生广告

1) 在需要嵌入原生的对象中实现原生的Delegate。如果在不同的界面用到原生，可以在AppDelegate对象中只实现一次，提供外接方法供调用位置使用。

```
#import "AdNativeManager.h"
#import "AdNativeManagerDelegate.h"
@interface ShowNativeViewController
()<AdNativeManagerDelegate>
@property (nonatomic, strong) AdNativeManager
*nativeManager;
```

2) 实现原生Delegate，如下代码所示：

```
- (void)requestNativeAdSucceeded:(AdNativeManager*)manager
adInfo:(NSArray*)adviewNativeAdArray{
    self.adArr = adviewNativeAdArray;
    self.adCountIndex = 0;
    self.label.text = @"请求广告成功";
    [self createNativeAdView];
}

- (BOOL)adNativeTestMode{
    return NO;
}

- (BOOL)adNativeLogMode {
    return YES;
}
```

3) 在控制器viewDidLoad函数中创建AdNativeManager，如下代码所示：

```
- (void)viewDidLoad
{
    [super viewDidLoad];
    self.nativeManager=[AdNativeManager
managerWithAdNativeKey:NATIVEKEY WithDelegate:self];
}
```

4) 在合适的位置加入广告加载代码，如下代码所示：

```
- (IBAction)requestAd:(id)sender {
    if (self.adIndex) {
        [nativeManager loadSingleAdNative:3
withIndex:self.adIndex];
    }else{
        [nativeManager loadNativeAd:3];
    }
}
```

5) 在合适的位置加入广告显示以及汇报代码，如下代码所示：

```
- (void)adNativeShow:(id)sender
{
    AdViewNativeAdInfo *info = [self.adArr objectAtIndex:self.adCountIndex];

    UIImage *iconImage = [UIImage imageData:[NSData
dataWithContentsOfURL:[NSURL URLWithString:[info.nativeAdDict
objectForKey:AdViewNativeAdIconUrl]]]];
    UIImageView *iconImgView = [[UIImageView alloc]
initWithImage:iconImage];

    [showView addSubview:iconImgView];
    [info showNativeAdWith:showView];
}
```

6) 在合适的位置加入广告点击汇报代码，如下代码所示：

```
- (void)click{
    [[self.adArr objectAtIndex:self.adCountIndex]
clickNativeAd];
}
```

VII. 创建视频广告

1) 在需要嵌入视频的对象中实现视频的Delegate。如果在不同的界面用到视频，可以在AppDelegate对象中只实现一次，提供外接方法供调用位置使用。

```
#import "AdVideoManager.h"
#import "AdVideoManagerDelegate.h"
@interface VideoViewController
()<AdVideoManagerDelegate>
@property (nonatomic, strong) AdVideoManager
*videoManger;
```

2) 实现视频Delegate，如下代码所示：

```
/**
 * 信息回调
 */
- (void)adVideoManager:(AdVideoManager*)manager didGetEvent:
(VideoEventType)eType error:(NSError*)error{
    if (eType == VideoEventType_DidLoadAd) {
        [manager showVideoAdWithController:[UIApplication
sharedApplication].keyWindow.rootViewController];
    }
}

/**
 *
 */
- (void)adVideoManager:(AdVideoManager*)manager
videoAvailable:(BOOL)avilable{}

/**
 * 是否打开测试模式，缺省为NO
 */
- (BOOL)adVideoTestMode{
    return NO;
}

/**
 * 是否打开日志模式，缺省为NO
 */
- (BOOL)adVideoLogMode{
    return YES;
}
```

3) 在控制器viewDidLoad函数中创建 AdVideoManager，如下代码所示：

```

- (void)viewDidLoad
{
    [super viewDidLoad];
    self.videoManger = [AdVideoManager
managerWithAdVideoKey:NATIVEKEY WithDelegate:self];
}

```

4) 在合适的位置加入广告加载代码，如下代码所示：

```

- (void)showVideo
{
    [self.videoManger loadVideoAd:self];
}

```

5) 在合适的位置加入广告显示以及汇报代码，如下代码所示：

```

/**
 * 信息回调
 */
- (void)adVideoManager:(AdVideoManager*)manager didGetEvent:
(VideoEventType)eType error:(NSError*)error{
    if (eType == VideoEventType_DidLoadAd) {
        [manager showVideoAdWithController:[UIApplication
sharedApplication].keyWindow.rootViewController];
    }
}

```


VIII.创建直投和交换广告及投放

1. 登录AdView官网，进入开发者后台，点击广告管理中的物料库开始创建广告物料。如图所示：



2. 点击快速创建物料按钮，开始填写一些创建物料中相应的信息。**创建H5的广告的时候必须在HTML代码中添加**

直投广告

交换广告

创建广告物料

物料库

基本信息

广告物料名

称:

备注:

广告名称和备注仅用于标识, 不会出现在广告内容中

0/200

广告物料

广告物料:

手机横版

广告形式:

o 图片 o 图文 o html5 o GIF动画

宽: 320 高: 50

点击后:

打开网页

目标URL:

您需要在“目标URL”中提供点击后要去
的网页地址。请按以下格式填写:
<http://www.adview.cn>

快速创建

广告类型:

请选择

直投广告

交换广告

ADVIEW

取消

- 进入到创建直投广告界面, 填写相应的信息后点击保存, 直投广告创建完毕。



4. 此时您可以在广告管理界面看到刚才创建的直投广告。接下来您只需要等待我们的工
员
对您的广告进行审核，审核通过的时候，您就可以投放您的广告了。



5. 进入广告投放界面，在直投广告点击设置，开始投放广告（**投放总量必须达到100%**）。



6) 创建交换广告物料及投放步骤和直投一样。

IX. 配置Xcode工程

1. 在项目设置中设置Other Linker Flags的值，并添加“-ObjC”标志。

2. 在您的工程中添加 AdViewSDK 以及各广告依赖的 framework：

- libsqlite3.tbd
- libz.tbd
- libc++.tbd —Optional
- Metal.framework —Optional
- MapKit.framework
- MobileCoreServices.framework
- libstdc++.tbd —Optional
- PassKit.framework —Optional
- Social.framework —Optional
- AudioToolbox.framework
- AddressBook.framework
- AVFoundation.framework
- AdSupport.framework—Optional
- CoreGraphics.framework
- CoreLocation.framework
- CoreMedia.framework
- CoreMotion.framework
- CoreTelephony.framework
- CoreVideo.framework
- ImageIO.framework
- MediaPlayer.framework
- MessageUI.framework
- QuartzCore.framework
- UIKit.framework
- UIKitUI.framework
- Foundation.framework
- Security.framework
- StoreKit.framework—Optional
- SystemConfiguration.framework
- UIKit.framework
- Twitter.framework
- AssetsLibrary.framework
- Accounts.framework
- WebKit.framework
- libxml2.tbd
- libz.tbd
- libz.1.2.5.tbd

附：以下为各家广告平台需要的framework（AdView的framework必须添加）。

| 广告平台 | Framework |
|--------|--|
| AdView | AdSupport.framework(optional) CoreLocation.framework AddressBook.framework libc++.tbd(optional) libsqlite3.tbd libxml2.tbd libz.tbd |

| | |
|---------------|--|
| AdChina (易传媒) | AudioToolbox.framework AVFoundation.framework CoreGraphics.framework CoreLocation.framework CoreMotion.framework CoreTelephony.framework EventKit.framework EventKitUI.framework Foundation.framework libz.tbd MapKit.framework MediaPlayer.framework MessageUI.framework QuartzCore.framework SystemConfiguration.framework StoreKit.framework UIKit.framework libconv.2.4.0.tbd libsqlite3.0.tbd libc++.tbd(optional) |
| Adwo (安沃) | AdSupport.framework(optional) AudioToolbox.framework AVFoundation.framework Audio.framework CoreLocation(optional) CoreMedia.framework CoreMotion.framework CoreTelephony.framework EventKit.framework MapKit.framework MessageUI.framework PassKit.framework(optional) QuartzCore.framework StoreKit.framework(optional) SystemConfiguration.framework Social.framework(optional) libz.1.2.5.tbd或libz.1.1.3.tbd |
| Baidu (百度) | QuartzCore.framework Security.framework AdSupper.farmework(optional) StoreKit.framework(optional) libz.tbd AudioToolbox.framework CoreMotion.framework MediaPlayer.framework CoreLocation.framework libc++.daylib(optional) CoreTelephony.framework UIKit.framework SystemConfiguration.framework Foundation.framework CoreGraphics.framework WebKit.framework |
| Chance (畅思) | libz.tbd Security.framework SystemConfigguration.framework StoreKit.framework(optional) AdSupport.framework CoreTelephony.framework |
| DianRu (点入) | libz.1.tbd QuartzCore.framework UIKit.framework CoreLocation.framework MapKit.framework SystemConfiguration.framework Foundation.framework CoreGraphics.framework AdSupport.framework(optional) |
| Domob (多盟) | libz.tbd libsqlite3.tbd CoreMedia.framework QuartzCore.framework CoreText.framework StoreKit.framework(optional) PassKit.framework(optional) Social.framework EventKit.framework AVFoundation.framework AudioToolbox.framework UIKit.framework Systemconfiguration.framework Foundation.framework |

| | |
|-----------------------|--|
| G T D M o b (广 点 通) | AdSupport.framework QuartzCore.framework SystemConfiguration.framework CoreTelephony.framework Security.framework CoreLocation.framework libz.tbd StoreKit.framework(optional) |
| GuoHe (智游汇) | SystemConfiguration.framework StoreKit.framework(optional) AdSupport.framework |
| Guomen (果盟) | QuartzCore.framework SystemConfiguration.framework UIKit.framework CoreGraphics.framework StoreKit.framework(optional) AdSupport.framework(optional) CoreTelephony.framework Foundation.framework ImageIO.framework |
| IAD | IAD.framework |
| InMobi | CoreLocation.framework EventKitUI.framework libz.tbd AudioToolbox.framework CoreGraphics.framework Foundation.framework MessageUI.framework PassKit.framework(optional) Security.framework SystemConfiguration.framework UIKit.framework EventKit.framework Social.framework AdSupport.framework AVFoundation.framework CoreTelephony.framework MediaPlayer.framework MobileCoreServices.framework QuartzCore.framework StoreKit.framework(optional) |
| Lmmob (力美) | AudioToolbox.framework CoreLocation.framework EventKit.framework MapKit.framework MessageUI.framework QuartzCore.framework AdSupport.framework(optional) SystemConfiguration.framework libz.1.2.5.tbd CFNetwork.framework CoreTelephony.framework EventKitUI.framework MediaPlayer.framework MobileCoreServices.framework Security.framework |
| Miidi (米迪) | CoreGraphics.framework QuartzCore.framework MessageUI.framework SystemConfiguration.framework AdSupport.framework(optional) |
| MobiSage (艾德思齐) | Foundation.framework StoreKit.framework(optional) MobileCoreServices.framework CoreAudio.framework AudioToolbox.framework ImageIO.framework QuartzCore.framework CFNetwork.framework CoreGraphics.framework SystemConfiguration.framework CoreLocation.framework CoreTelephony.framework MessageUI.framework AdSupport.framework(optional) CoreMotion.framework QuartzCore.framework CoreImage.framework AVFoundation.framework CoreMedia.framework Security.framework UIKit.framework QuartzCore.framework Libz.tbd |

| | |
|--------------|--|
| SmartMad | CFNetwork.framework MessageUI.framework SystemConfiguration.framework EventKiyUI.framgwork EventKit.framework MediaPlayer.framework QuartzCore.framework libz.tbd AVFoundation.framework C o r e T e l e p h o n y . f r a m e w o r k AdSupport.framework(optional) AudioToolbox.framework CoreLocaation.framework CoreMotion.framework UIKit.framework Foundation.framework CoreGraphics.framework |
| Tanx（阿里妈妈） | UIKit.framework CoreGraphics.framework QuartzCore.framework Foundation.framework CoreLocation.framework libz.tbd SystemConfiguration.framework StoreKit.framework(optional) MapKit.framework MessageUI.framework CoreTelephony.framework |
| Waps（万普） | AdSupport.framework(optional) QuartzCore.framework Security.framework CoreTelephony.framework SystemConfiguriotin.framework libz.tbd |
| WQMobile（帷千） | libsqlite3.tbd StoreKit.frameowrk UIKit.framework QuartzCore.framework S y s t e m C o n f i g u r a t i o n . f r a m e w o r k MobileCoreServices.framework MessageUI.framework MediaPlayer.framework MapKit.framework libz.tbd Foundation.framework EventKit.framework EventKitUI.framework CoreMotion.framework CoreLocation.framework CoreGraphics.framework CoreFoundation.framework FNetwork.framework AdSupport.framework(optional) CoreTelephony.framework |
| XingYun（行云） | CoreLocation.framework MapKit.framework Security.framework CFNetwork.framework MobileCoreServices.framework CoreTelephony.framework AdSupport.framework(optional) SystemConfiguration.framework ImageIO.framework QuartzCore.framework UIKit.framework Foundation.framework CoreGraphics |
| YiJiFen（易积分） | ImageIO.framework QuartzCore.framework SystemConfiguration.framework StoreKit.framework(optional) CoreTelephony.framework AdSupport.framework(optional) CoreGraphics.framework Eadver.framework Poundation.framework Ulkit.framework |
| YouMi（有米） | storekit.framework(optional) security.framework cfnetwork.framework ImageIO.framework systemconfiguration.framework CoreMotion.framework libz.tbd libsqlite3.tbd |

| | | |
|----------|---|---|
| AdColony | AdSupport.framework(optional) AVFoundation.framework EventKit.framework MessageUI.framework StoreKit.framework SystemConfiguration.framework libz.1.2.5.tbd | AudioToolbox.framework CoreTelephony.framework JavaScriptCore.framework Social.framework WebKit.framework |
|----------|---|---|

3. 在添加SDK的时候，有些文件被重复引用，需要将重复引用的文件删除或者不让其编译。重复的文件包括：OpenUDID.m 、 ZipArchive.m 、 zip.m 、 unzip.m 、 mztools.c ioapi.c 和Reachability.m。
4. 在AdView网站上配置您的各个广告插屏优先级和百分比。
5. 编译并链接您的工程，即可显示广告插屏。

X.AdView后台应用设置

1) 发布应用设置

ADVIEW

我的产品应用管理广告管理数据分析财务管理

添加应用信息

选择广告平台

SDK下载

添加应用信息

应用信息

应用名称: 测试模式 通过

应用平台: iOS Android

应用类型: 实用软件 穿戴设备 通过

iTunes ID:

广告文字设置

背景颜色: 000000

文字颜色: #ffffff

刷新时间: 不限

地域优化: ON

推广及受众描述

下一步 取消

2) 选择广告平台

yyz1 SDK-KEY:SDK20141430021046ue831q28ozje350

横幅广告 全屏/插屏广告 开屏广告

地域优化 ON 广告审核 OFF

国内用户配置 + 添加广告平台

| 广告平台 | 评分 | 设置 | 开关 | 投放量 | 优先级 | 删除平台 |
|--------|------|----|-----|------|-----|------|
| 竞价广告 | 7.69 | 设置 | ON | 100% | | |
| 百度移动 | 5.33 | 设置 | OFF | 100% | + | 删除 |
| 腾讯(微信) | 2.00 | 设置 | OFF | 100% | + | 删除 |
| 广点通 | 6.91 | 设置 | OFF | 100% | + | 删除 |
| 糯米 | 7.16 | 设置 | OFF | 100% | + | 删除 |
| InMobi | 5.69 | 设置 | OFF | 100% | + | 删除 |

3) 添加广告平台



开发者根据需要选择想要的平台；

4) 配置广告平台



- 1、APP ID设置：开发者应当先去相关的广告平台注册，获取对应的APP信息。根据各广告平台接口不同，一般只需要输入APP ID，有的还需要输入其他信息。
- 2、开关：只有开关打开的那家平台的广告才会在应用中显示。开发者可以控制广告平台状态切换不同的广告平台。
- 3、投放量：只有开关打开的投放量才有意义，投放量为对广告平台的请求比例。切换广告平台应同时调整投放量，所有广告状态为on的应用投放量总额应为100，否则无法正确保存。

5) 地理位置优化



开启地域优化的功能是指定在国内的手机显示国内配置的广告，在国外的手机显示国外配置的广告；最大限度的满足不同的需求。当地域优化关闭的时候，就不区分国内国外了，设置为开关打开的都可以显示。

XI. 离线配置文件

(1) 下载离线配置文件

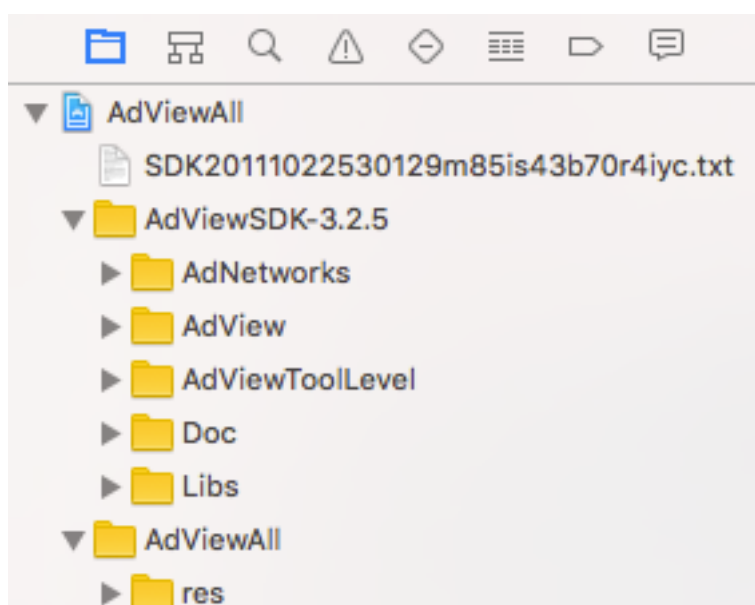
SDK-KEY

本应用的AdView SDK-KEY: SDK20111022530129m85is43b70r4iyc
嵌入SDK时使用, 此应用的专属KEY

为保证在断网或AdView服务器宕机等极端情况下, 您的广告展示不受影响。AdView建议您在您的“离线配置文件”, 并按照说明将其添加到应用源代码工程中。

[离线配置文件下载 >>](#)

(2) 放进工程目录



XII. 删除不需要的广告平台

删除不需要的广告插屏非常容易，只需要在项目工程中删除AdInstlSDK/AdNetworks组下相应的广告插屏库文件，重新编译工程即可。

在从工程中删除广告插屏时我们建议只是删除工程引用，而不要把广告插屏从硬盘上删除。保留这些广告插屏的开发库以备不时之需。

XIII. 添加自定义广告平台(插屏)

开发者有时想添加一个不是聚合里面的广告平台；Adview提供了这种需求的实现：

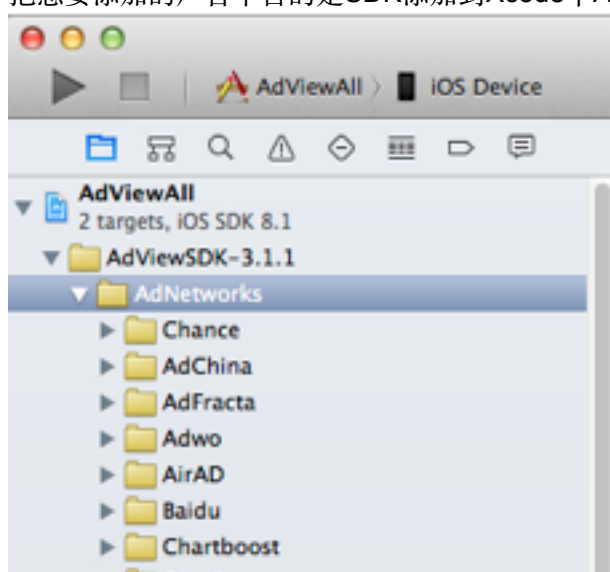
添加广告平台中有个“自定义广告平台”：



开发者应当先去相关的广告平台注册，获取对应的APP信息。根据各广告平台接口不同，一般只需要输入APP ID，有的还需要输入其他信息，如果没有可以随便填一个字符。

参考代码(行云):

1. 把想要添加的广告平台的是SDK添加到Xcode中AdNetworks下。



2. 自己创建继承于AdInstlAdNetworkAdapter的类，并实现它的Delegate
`#import "AdInstlAdNetworkAdapter.h"`
`#import <Walker/GuInitServer.h>`

```
#import <Walker/PobAppFrame.h>
@interface AdInstlAdapterXingYun : AdInstlAdNetworkAdapter<PobAppFrameDelegate>

@property (nonatomic, weak) UIViewController *rootController;
@property (nonatomic, strong) GulnitServer * gunInitServer;
@property (nonatomic, strong) PobAppFrame * pobAppFrame;
@end
```

3. 在实现文件里面实现下面两个方法：一个是networkType（获取渠道号，自定义广告一般返回AdInstlAdNetworkTypeUserDefined）。另一个是Load

```
+ (AdInstlAdNetworkType)networkType
{
    return AdInstlAdNetworkTypeXingYun;
}
//@"PobAppFrame"是嵌入广告类的类名
//sharedInstlRegistry插屏
//sharedBannerRegistry横幅
//sharedSpreadScreenRegistry开屏
+ (void)load
{
    if (NSClassFromString(@"PobAppFrame") != nil) {
        [[AdViewAdNetworkRegistry sharedInstlRegistry] registerClass:self];
    }
}
```

4. 在加载广告(loadAdInstl)方法中去实例化广告和发送广告请求

```
- (BOOL)loadAdInstl:(UIViewController *)controller
{
    Class XingYunInstlClass = NSClassFromString(@"PobAppFrame");
    Class XingYunInstlServer = NSClassFromString(@"GulnitServer");
    if(nil == XingYunInstlClass) return NO;
    if (nil == XingYunInstlServer) return NO;
    if (controller == nil) return NO;
    self.rootController = controller;
    //从后台接受AppID
    NSString *applD = self.networkConfig.pubId;
    //实例化
    self.gunInitServer = [[XingYunInstlServer alloc] init];
    [self.gunInitServer setInfo:applD Channel:@" User_info:@"];
    //请求广告
    self.pobAppFrame = [[XingYunInstlClass alloc] orientation:@"Portrait"
andDelegate:self];
    //向AdView服务器发送请求汇报，方便开发者查看
    [self.adInstlManager adapter:self requestString:@"req"];
    return YES;
}
```

5. 在展示广告的方法中实现广告展示

```
- (BOOL)showAdInstl:(UIViewController *)controller
{
    #if __IPHONE_OS_VERSION_MAX_ALLOWED >= __IPHONE_5_0
    if (nil == controller || ![controller isViewLoaded] || [controller isBeingDismissed]) {
        [self.adInstlManager adapter:self requestString:@"fail"];
        [self.adInstlManager adapter:self didGetEvent:InstlEventType_FailLoadAd error:nil];
        return NO;
    }
    #else
    if (nil == controller || ![controller isViewLoaded]) {
        [self.adInstlManager adapter:self requestString:@"fail"];
        [self.adInstlManager adapter:self didGetEvent:InstlEventType_FailLoadAd error:nil];
        return NO;
    }
    #endif
}
```



```

    if (self.pobAppFrame)
    {
        //广告展示
        [self.pobAppFrame showPobFrame:self.rootController];
        return YES;
    }
    return NO;
}

```

6. 实现嵌入广告SDK中的代理方法在这些相应的方法中添加广告状态和汇报数据

```

//插屏弹出成功
- (void)showPobFrameSuccess
{
    [self.ADINSTLMANAGER Adapter:self didGetEvent:InstlEventType_WillPresentAd error:nil];
    [self.adInstlManager adapter:self requestString:@"show"];
}
//插屏弹出失败
- (void)showPobFrameFail
{
    [self.adInstlManager adapter:self didGetEvent:InstlEventType_FailLoadAd error:nil];
    [self.adInstlManager adapter:self requestString:@"fail"];
}
//插屏关闭
- (void)closePobAppFrame
{
    if (self.pobAppFrame) {
        self.pobAppFrame = nil;
    }

    if (self.gunInitServer) {
        self.gunInitServer = nil;
    }

    if (self.rootController) {
        self.rootController = nil;
    }

    [adInstlManager adapter:self didGetEvent:InstlEventType_DidDismissAd error:nil];
}
//获取数据成功
- (void)initPobFrameSuccess
{
    [self.adInstlManager adapter:self didGetEvent:InstlEventType_DidLoadAd error:nil];
    [self.adInstlManager adapter:self requestString:@"suc"];
}
//获取数据失败
- (void)initPobFrameFail
{
    [self.adInstlManager adapter:self didGetEvent:InstlEventType_FailLoadAd error:nil];
    [self.adInstlManager adapter:self requestString:@"fail"];
}

```

7. 在stopBeingDelegate方法中实现数据清空。

```

- (void)stopBeingDelegate
{
    AWLogInfo(@"xingyun stop being delegate");
    if (self.pobAppFrame) {
        self.pobAppFrame = nil;
    }
    if (self.gunInitServer) {
        self.gunInitServer = nil;
    }
}

```

```
    }  
    if (self.rootController) {  
        self.rootController = nil;  
    }  
}
```

XIV. 常见问题处理

问：为什么AdInstl无法显示广告？

答：最大可能性的原因在于您没有在我的项目中添加额外的链接选项，使广告插屏的开发库能正常加载到您的应用程序中。您可以参考 [“在程序界面中嵌入AdViewSDK一体包”](#) 的第 5 步。

问：工程在编译能够成功，但是在链接的时候发生错误，这是为什么？

答：由于Objective-C语言的特性所限制，所有参与链接的静态库不允许出现重复的函数或者变量等符号。如果冲突发生在广告插屏之间，这个比较容易解决，您只需要注意冲突发生在那些广告插屏之间即可。