

# 万普平台iOS版SDK开发者手册

## (Ver1.3.3)

## 平台介绍

万普世纪移动营销服务平台(以下称为“万普平台”)的iOS版SDK提供了一套现成的开发包及Demo源代码,便于开发者在iOS应用中方便的集成万普平台的各项功能,包括万普统计、虚拟货币、推荐列表、互动广告、插屏广告等功能。

本文档描述了标准版SDK的用途与用法,并提供了示例代码。

## 引入SDK包

步骤1:下载iOS版SDK包,将Frameworks目录中的 WapsOffer.framework 复制到自定义路径。

步骤2:引入WapsOffer.framework,在工程中选择 -> TARGETS->Summary->Linked Frameworks and Libraries,点“+”,点击“Add Other”指向到下载的WapsOffer.framework  
在所有调用万普SDK接口的代码文件中引入头文件:

```
#import "WapsOffer/AppConnect.h"
```

其他需要添加的依赖包如下(请在系统包中自行寻找添加):

CoreTelephony.framework

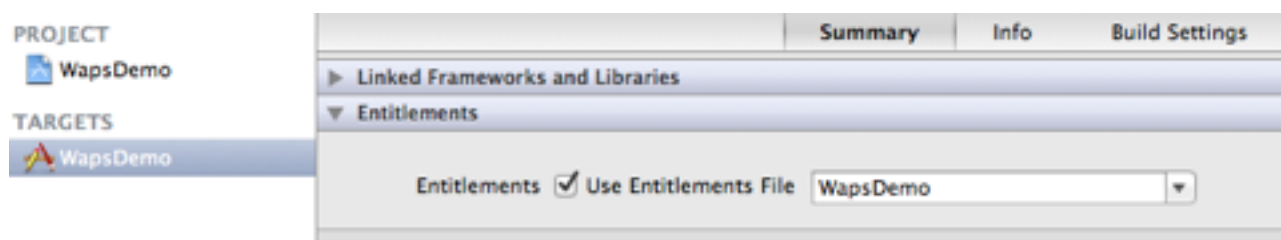
SystemConfiguration.framework

Libz.dylib

步骤3: (仅适用越狱应用)为支持iOS6.0以上越狱广告,需要进行如下配置:

1.复制下载包中的WapsDemo.entitlements 到工程目录,并引入到工程中。

在工程中选择TARGETS->Summary->Entitlements,将Entitlements选项选中,并在Use Entitlements File下拉框中选中WapsDemo (如下图)。亦可自己命名Entitlements文件,按照WapsOffer.entitlements中的属性配置。



注意: 步骤3会导致应用在未越狱设备上无法运行,且无法通过AppStore审核。因此,发布到AppStore 的版本务必不可添加步骤3,发布到其他越狱渠道的版本则需要添加步骤3的配置。

# 加入接口代码

## 1.数据统计接口

该接口是所有其他接口能正常使用的基础，在每次应用启动时，必须调用该接口，才能保证获得准确的统计数据。

在程序的启动方法中加入(必须)：（WAPS\_ID在万普后台添加应用后获取）

`[AppConnect getConnect:@"WAPS_ID" pid:@"appstore"];` //指定WAPS\_ID和发布渠道编号PID  
或者(两种初始化方式只能选择一种)：

`[AppConnect getConnect:@"WAPS_ID"];` //不指定PID参数的方式初始化（不推荐）

连接成功的回调方法(可选)：

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onConnectSuccess:) name:WAPS_CONNECT_SUCCESS object:nil];
```

//连接成功处理：

```
-(void)onConnectSuccess:(NSNotification*)notifyObj{
    NSLog(@"连接成功");
}
```

连接失败的回调方法(可选)：

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onConnectFailed:) name:WAPS_CONNECT_FAILED object:nil];
```

//连接失败处理：

```
-(void)onConnectFailed:(NSNotification*)notifyObj{
    NSLog(@"连接失败");
}
```

## 2.虚拟货币接口

在应用中合理设置虚拟货币及消费机制，可促进用户参与应用内购买或参与广告活动，增强应用粘性，大幅提升收益。如果您的应用开启了虚拟货币功能，需要使用该接口和服务器同步用户的虚拟货币余额。

### (1)获取用户虚拟货币

步骤 1: 发起从服务器端获取用户点数/虚拟货币余额的请求：

`[AppConnect getPoints];`

步骤 2:指定处理获取用户积分的回调方法：

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(getUpdatedPoints:) name:WAPS_GET_POINTS_SUCCESS object:nil];
```

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(getUpdatedPointsFailed:) name:WAPS_GET_POINTS_FAILED object:nil];
```

//获取积分成功处理方法：

```
-(void)getUpdatedPoints:(NSNotification*)notifyObj{
    WapsUserPoints *userPoints = notifyObj.object;
    NSString *pointsName=[userPoints getPointsName];
    int pointsValue=[userPoints getPointsValue];
}
```

//获取积分失败处理方法:

```
-(void)getUpdatedPointsFailed:(NSNotification*)notifyObj{
    NSLog(@"获取积分失败");
}
```

## (2)花费用户虚拟货币

花费（扣除）用户的虚拟货币接口(调用此方法的响应结果将通过getUpdatedPoints回调方法返回):

```
[AppConnect spendPoints:(int)amount];
```

## (3)奖励用户虚拟货币

花费（扣除）用户的虚拟货币接口(调用此方法的响应结果将通过getUpdatedPoints回调方法返回):

```
[AppConnect awardPoints:(int)amount];
```

# 3.推荐列表接口

推荐列表（也称Offer，广告墙）是万普平台提供的一种集中展示型广告。开发者可在应用中合适的位置加入“推荐应用”、“免费赚积分”等类似字样的功能，获取更高的广告收益或参与流量交换。添加如下代码，即可显示万普平台推荐应用列表：

```
[AppConnect showOffers:viewController]; //支持横竖屏自动切换
```

或

```
[AppConnect showOffers]; //仅支持竖屏显示
```

//从积分墙返回的事件回调方法（可选）：

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onOfferClosed:) name:WAPS_OFFER_CLOSED object:nil];
-(void)onOfferClosed:(NSNotification*)notifyObj{
    NSLog(@"Offer列表已关闭");
}
```

# 4.互动广告接口

互动广告是一个显示在应用内固定位置高度为50像素广告条，将自动显示万普平台提供的Banner广告。使用下面一行代码即可显示万普互动广告条：

```
[AppConnect displayAd:viewController];
```

以上默认情况下，广告条将显示在屏幕顶端，默认为320\*50像素。如需在其他位置显示广告条，可使用以下方法在指定位置创建广告条：

```
[AppConnect displayAd:viewController showX:(int)x showY:(int)y];
```

# 5.插屏广告接口

插屏广告是在1.2版本后新加入的广告形式，将弹出显示万普平台提供的插屏广告。下面是使用方法：

步骤1 初始化(预先加载)广告数据：

```
[AppConnect initPopAd];
```

**注意：**请在 [AppConnect getConnect:@"WAPS\_ID" pid:@"appstore"];方法之后调用。

步骤2 显示插屏广告：

1.普通应用中使用：

```
[AppConnect showPopAd:viewController];
```

## 2.Cocos2d游戏引擎中使用:

引入头文件:

```
#import "WapsOffer/WapsPopAdController.h"
```

```
[[[CCDirector sharedDirector] openGLView] addSubview:[[WapsPopAdController alloc]
init].view];
```

### 步骤3 处理插屏广告显示状态的回调方法(可选):

```
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onPopAdShow:) name:WAPS_POPAD_SHOW object:nil];
[[NSNotificationCenter defaultCenter] addObserver:self
selector:@selector(onPopAdClosed:) name:WAPS_POPAD_CLOSED object:nil];
```

```
- (void)onPopAdShow:(NSNotification*)notifyObj
{
    NSLog(@"%@", @"插屏广告开启");
}

- (void)onPopAdClosed:(NSNotification*)notifyObj
{
    NSLog(@"%@", @"插屏广告关闭");
}
```

## 6.在线配置接口

开发者可以通过万普平台管理后台的“在线配置”功能设置各种参数值，便于在线修改应用的各项配置。其中参数key为在线配置的参数ID，通过次方法可获取到对应的值。

注意：该方法将在统计器初始化成功时返回所有参数值，缓存到本地调用。因此在线参数的初始化需要在数据统计接口的连接成功回调函数中进行。

//计数器连接成功的回调方法

```
-(void)onConnectSuccess:(NSNotification*)notifyObj{
    NSLog(@"连接成功");
    //启动并连接成功后获取在线配置
    NSString* value=[[AppConnect getConfigItems] objectForKey:@"key"]; //key设置为在线参
数ID
}
```

## 7.用户反馈接口

用户反馈接口的作用便于开发者收集到用户对应用的使用反馈，用户反馈的内容开发者可在万普平台管理后台的“用户反馈”中进行管理，也可通过管理后台给用户及时反馈。

调用方法：

```
[AppConnect showFeedBack:viewController];
```