

# **亿动智道应用内置广告开发包使用说明**

---

## **SmartMad In-App Ad SDK for iOS**

**V3.0.5**  
**Madhouse Inc.**

## 目录

目录 .....	1
1. 适用范围 .....	3
2. 名词解释 .....	3
3. 广告类型 .....	4
3.1. 固定广告位 .....	4
3.1.1. 旗帜广告 .....	4
3.1.2. 全屏广告 .....	5
3.1.3. 炫动广告 .....	5
3.1.4. 视频广告 .....	6
3.2. 浮动广告位 .....	6
3.2.1. 插页广告 .....	6
3.2.2. 富媒体插页广告 .....	7
4. 广告植入说明 .....	7
4.1. 准备 .....	7
4.2. 建立工程项目并添加文件 .....	8
4.3. 配置工程项目 .....	8
4.3.1. 添加工程项目所需的framework 和库文件 .....	8
4.3.2. 配置工程项目编译链接器 .....	9
4.4. 设置全局性广告配置信息 .....	9
4.4.1. 通过plist 配置文件设置信息 ( 选择 ) .....	10
4.4.2. 通过编写代码设置信息 ( 选择 ) .....	10
4.4.3. 固定位广告请求间隔时间设置规则 .....	11
4.5. 添加固定位旗帜广告 .....	11
4.5.1. Design 方式 : 通过Interface Builder 添加广告 ( 选择 ) .....	11
4.5.2. Coding 方式 : 通过Objective-C 代码添加广告 ( 选择 ) .....	12
4.5.3. 固定位旗帜广告委托事件说明 ( 选择 ) .....	13
4.6. 添加浮动位插页广告 .....	14
4.6.1. Coding 方式 : 通过Objective-C 代码添加广告 .....	14
4.6.2. 浮动位插页广告事件说明 ( 选择 ) .....	14
4.7. 广告必要参数规范 ( 重要 ) .....	15
4.7.1. 广告位标识 : AdSpaceId .....	15

---

4.7.2.	广告尺寸类型 : AdSize.....	15
4.7.3.	广告动画类型 : AdAnimationType .....	15
5.	广告植入高级技巧.....	16
5.1.	广告预缓存机制应用.....	16
5.1.1.	Coding 方式 : 通过Objective-C 代码调用广告预缓存机制.....	16
5.1.2.	广告预缓存必要参数说明.....	17
5.1.3.	广告预缓存事件说明.....	17
5.2.	提供信息辅助广告精确定向.....	17
5.2.1.	提供终端用户信息.....	17
5.2.2.	提供关键字信息.....	18
5.3.	广告事件(SMAdEventCode)说明 .....	18

## 1. 适用范围

本文档主要介绍“亿动智道”移动广告平台(SmartMad)为开发者和第三方合作伙伴提供的基于 iOS 移动手机操作系统 ( 4.3/5.x/6.x/7.0.x ) 的移动应用程序 ( iPod Touch/iPhone 和 iPad ) 中广告植入的方法和规范,并简单的描述了投放广告的类型和展示形式,有助于开发人员快速的植入广告到移动应用程序中。本文档主要面向具有一定 iOS 开发经验的人员编写,关于 iOS 编程技术的介绍不属于本文档的介绍范围。

## 2. 名词解释

名词	全称	释义
iOS	Apple iOS	Apple公司的移动操作系统。
Xcode	Xcode for Developer	Apple 为开发者提供的iOS开发工具,集成了多种开发和测试工具于一体。
IB	Interface Builder	Xcode开发工具中所附带的界面生成器工具,可以方便的生成UI并和程序相关联。
MRAID	Mobile Rich Media Ad Interface Definitions	MRAID 是 IAB 移动营销中心的卓越项目。MRAID 是针对移动应用程序进行富媒体广告投放而定义的一个通用型 API。
ORMMA	Open Rich Media Mobile Advertising	ORMMA 是一个在广告行业内倡导的开放式跨平台的富媒体广告规范。
SmartMad	SmartMad Advertising Platform	“亿动智道”移动广告网络平台。
Ad SDK	Advertising System Development kit	SmartMad提供的用于移动平台应用程序的广告系统植入开发包。
AppId	Application Identity	SmartMad中注册的用于标识移动平台应用程序的标识码。
AdSpaceId	Advertising Position Identity	SmartMad中根据AppId分配的用于标识应用程序中广告位置的标识码。
App Package Name	Application Package Name	SmartMad中和AppId、AdSpaceId进行绑定,用于校验独立移动应用程序的唯一标识。
Refresh Interval	Advertising Request Interval	广告请求间隔,开发者可以在SDK接口中设置或在SmartMad广告平台网站中设置。
Ad Size	Advertising Size	广告规格尺寸, SmartMad中定义的广告位的尺寸规则。

名词	全称	释义
Banner Ad Animation	Banner Advertising Animation Type	Banner广告过场动画 ,SmartMad中定义的广告切换之间执行的动画类型。
Interstitial Ad Animation	Interstitial Advertising Animation Type	Interstitial广告过场动画。SmartMad中定义的广告切换之间执行的动画类型。
Debug	Advertising Debug Model	Debug模式，也称作广告调试模式。开发者在Debug模式下可以获得所有类型的测试广告，并且在IDE的控制台中能获取有益的状态或错误提示。注意：测试模式下无法计费。
App	Application	Mobile Application，即移动应用程序，在下文中泛指所有基于 iOS 建立的“移动应用程序”。

### 3. 广告类型

目前 SmartMad 为 iOS 移动应用程序提供了 2 种布局接入方式 ( In-Style 和 Float Window ) , 共计 6 种广告展示形式，支持 MRAID 1.0 和 ORMMA Level1-2 开放式富媒体接口，为富媒体广告创意和富媒体广告交换提供了强有力的支撑。

#### 3.1.固定广告位

固定广告位基于应用程序布局中，位置相对固定，也被称为样式内广告位（ In-Style ）或内联式广告位（ Inline ），具体包含以下广告形式：

##### 3.1.1. 旗帜广告

旗帜广告，即 Banner 广告。分为图形 Banner 广告、文字链 Banner 广告和富媒体 Banner 广告，可以将该广告视图放于任何可以容纳的位置（容纳的容器或位置必须在屏幕内并且尺寸等于或大于广告视图，推荐放在界面的顶部或底部）。针对每个 App，系统都提供 2 个广告位，且在同一屏幕仅可同时使用 2 个不同的广告位。



### 3.1.2. 全屏广告

全屏广告，即 Full Screen 广告。是一种基于 Banner 广告形式，使用 MRAID/ORMMA 中 Expand 方式扩展到整个屏幕进行广告展示的广告类型。全屏广告当前会在首次调用时展示，使用在 App 启动时调用可类似于开屏广告（Splash）进行展示。



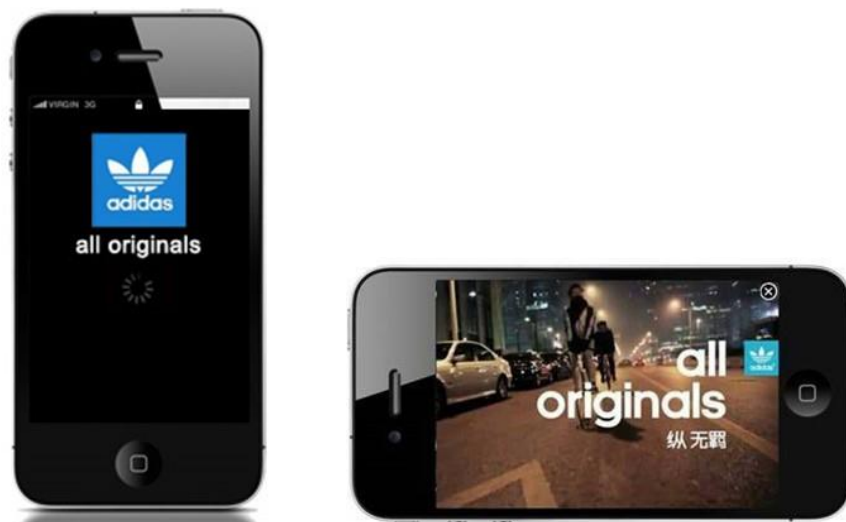
### 3.1.3. 炫动广告

炫动广告，也被称为 Crazy Ad。是一种基于 Banner 广告，使用 MRAID/ORMMA 中 Expand 方式扩展到整个屏幕并动态展示广告的富媒体广告形式（Rich Media Ad）。



### 3.1.4. 视频广告

视频广告，是一种基于 Banner 广告，使用 MRAID/ORMMA 中 Expand 方式扩展到半屏或全屏，使用流媒体视频在移动终端展示的富媒体广告形式。



## 3.2. 浮动广告位

浮动位广告位以窗口方式浮动于应用程序布局之上，因为无需对应用布局进行调整，也被称为浮动窗广告位（float window），具体包含以下广告形式：

### 3.2.1. 插页广告

插页广告，即 Interstitial Ad。是一种开发者可以在界面过场时请求并根据需要进行展示的间质性广告形式。插页广告有区别于固定位旗帜广告的 API，调用形式灵活可控。



### 3.2.2. 富媒体插页广告

富媒体插页广告，是一种插页广告的富媒体版本。其展示内容更据视觉效果，且含有丰富的交互行为，有利于广告的转化和效果。

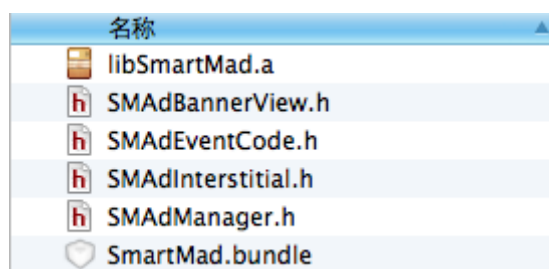


## 4. 广告植入说明

应用程序进行广告植入将添加 SmartMad Ad SDK 所包含的代码和文件，以下开发示例说明将假设在以 Mac OS X 10.8.4 和 Xcode 4.6.3(包含 iOS SDK 6.1)搭建的开发环境下。

### 4.1. 准备

在亿动智道移动广告平台网站下载 SmartMad In-App Ad SDK for iOS 发布版本压缩包，解压缩后将会在 SmartMad 目录中看到以下文件：

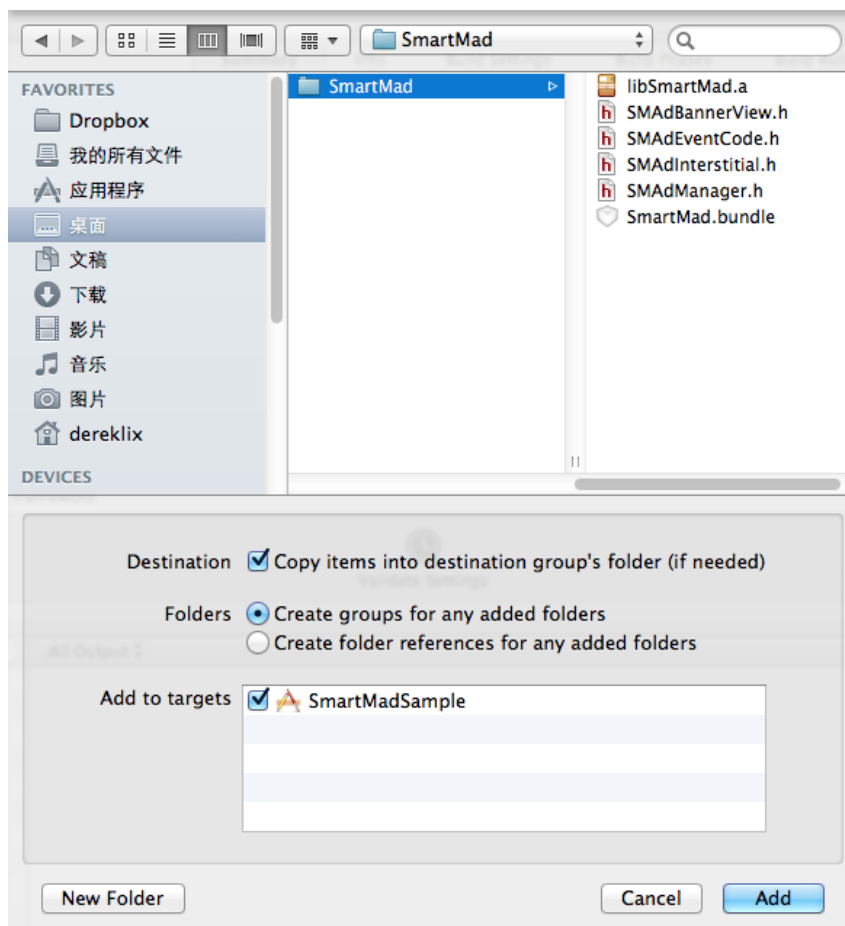


目录中除了 SmartMad In-App Ad SDK 的主库文件“libSmartMad.a”外，还包括主库正常工作时所必须的“SmartMad.bundle”资源包。



## 4.2. 建立工程项目并添加文件

使用 Xcode 中的模板功能建立一个新的工程项目，并在放置代码的目录上点击右键。右键菜单中选中“Add Files to ...”，在弹出窗口中找到之前解压出的 SmartMad SDK 目录，点击“Add”按钮完成添加，具体如下图：



**注意：**如果需要将 SDK 中的文件拷贝到新建的工程项目中，需要选中 **Destination** 项。

## 4.3. 配置工程项目

SmartMad SDK 文件添加完成后，需要对工程项目进行相应的配置，具体配置方法如下：

### 4.3.1. 添加工程项目所需的 framework 和库文件

选中工程项目的“TARGETS”，在“Build Phases”标签的“Link Binary With Libraries”中通过“+”号按钮添加所需的 Framework 和库文件。所需的 frameworks 和 libraries 如下图：

▼ Linked Frameworks and Libraries		
StoreKit.framework	Optional	⬆ ⬇ ⬆
CFNetwork.framework	Required	⬆ ⬇ ⬆
MessageUI.framework	Required	⬆ ⬇ ⬆
SystemConfiguration.framework	Required	⬆ ⬇ ⬆
EventKit.framework	Required	⬆ ⬇ ⬆
EventKitUI.framework	Required	⬆ ⬇ ⬆
MediaPlayer.framework	Required	⬆ ⬇ ⬆
QuartzCore.framework	Required	⬆ ⬇ ⬆
libz.1.2.5.dylib	Required	⬆ ⬇ ⬆
AVFoundation.framework	Required	⬆ ⬇ ⬆
CoreTelephony.framework	Required	⬆ ⬇ ⬆
CoreLocation.framework	Required	⬆ ⬇ ⬆
AdSupport.framework	Optional	⬆ ⬇ ⬆
AudioToolbox.framework	Required	⬆ ⬇ ⬆
CoreMotion.framework	Required	⬆ ⬇ ⬆
UIKit.framework	Required	⬆ ⬇ ⬆
Foundation.framework	Required	⬆ ⬇ ⬆
CoreGraphics.framework	Required	⬆ ⬇ ⬆
libSmartMad.a	Required	⬆ ⬇ ⬆
+ -		

注意:如果应用程序需要部署到 iOS6.0 之前的设备上，必须将 AdSupport.framework 和 StoreKit.framework 设置为 “Optional”。

4.3.2. 配置工程项目编译链接器

选中工程项目的 “TARGETS” ,在 “Build Settings” 标签的 “Linking” 中修改 “Other Linker Flags” 项，添加 “-all\_load -ObjC” 内容，完成后如下图：

Order File		
► Other Linker Flags	-all_load -ObjC	-all_load -ObjC
► Path to Link Map File	<Multiple valu...	

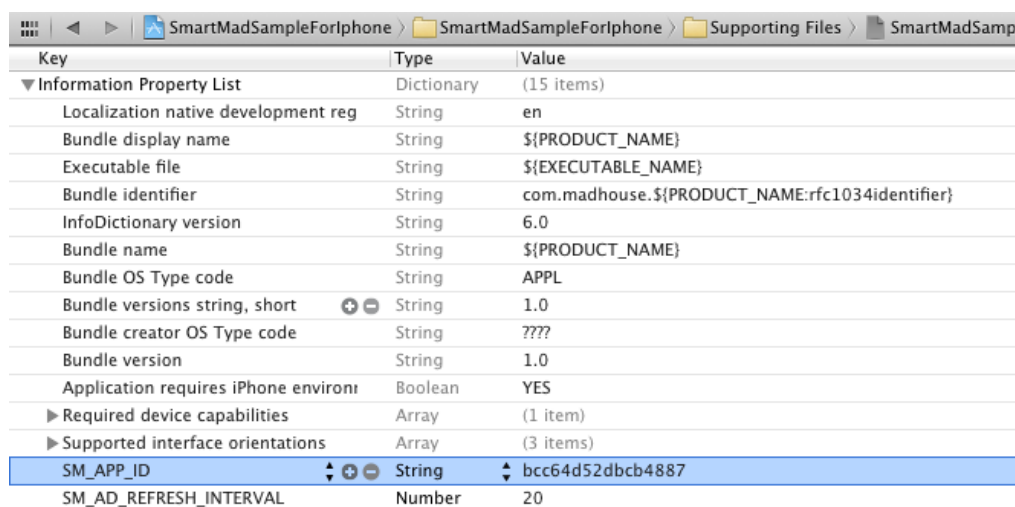
提示：“-ObjC” 使 Objective-C 编写的静态类别库在编译链接的时候不会报错；“-all\_load” 将强制链接器加载所有包含非 ObjC 的目标文档。

4.4.设置全局性广告配置信息

添加广告容器之前，需要设置一些全局性广告配置信息（AppId 和 Refresh Interval），以便能正常进行广告投放。具体配置方法如下：

#### 4.4.1. 通过 plist 配置文件设置信息（选择）

打开工程项目的 plist 配置文件，通过右键菜单中“Add Row”选项新添加一行，键值“Key”输入字符串“SM\_APP\_ID”；类型“Type”选择“String”；数值“Value”输入在亿动智道移动广告平台中申请媒体时分配的“应用程序/网站 ID”。重复以上操作添加一行，键值“Key”输入字符串“SM\_AD\_REFRESH\_INTERVAL”；类型“Type”选择“Number”；数值“Value”输入设定的固定位广告请求（广告刷新）间隔时间，具体见“固定位广告请求间隔时间设置规则”所述内容。当设置完成后如下图：



Key	Type	Value
▼ Information Property List	Dictionary	(15 items)
Localization native development region	String	en
Bundle display name	String	\${PRODUCT_NAME}
Executable file	String	\${EXECUTABLE_NAME}
Bundle identifier	String	com.madhouse.\${PRODUCT_NAME:rfc1034identifier}
InfoDictionary version	String	6.0
Bundle name	String	\${PRODUCT_NAME}
Bundle OS Type code	String	APPL
Bundle versions string, short	String	1.0
Bundle creator OS Type code	String	???
Bundle version	String	1.0
Application requires iPhone environment	Boolean	YES
▶ Required device capabilities	Array	(1 item)
▶ Supported interface orientations	Array	(3 items)
SM_APP_ID	String	bcc64d52dbcb4887
SM_AD_REFRESH_INTERVAL	Number	20

注意：如果不添加“SM\_AD\_REFRESH\_INTERVAL”键值，固定位广告刷新时间将会设置为默认值。

#### 4.4.2. 通过编写代码设置信息（选择）

除了可以在工程项目的 plist 配置文件中添加广告全局配置信息外，还可以在编写代码的时候进行全局配置信息的设置。设置信息必须在实例化广告视图之前加入以下代码（粗体字部分）：

```
// 设置亿动智道移动广告平台中获取的应用程序/网站ID
[SMAdManager setApplicationId:@"Application ID"];
// 设置固定位广告请求(广告刷新)的间隔时间
[SMAdManager setAdRefreshInterval:60];
// 设置开发(Debug)模式
[SMAdManager setDebugMode:YES];
```

注意：开发模式仅可以通过编写代码方式进行设置，默认值为“false”表示正式计费投放模式；值为“true”表示开发模式，将会展示 Debug 广告并在 IDE 的控制台中显示提示信息或调试信息，但是开发模式不会进行计费。

### 4.4.3. 固定广告请求间隔时间设置规则

固定广告请求间隔时间即广告刷新时间，该值直接影响广告的展示（刷新）频率。亿动智道移动广告平台对该值的定义如下：

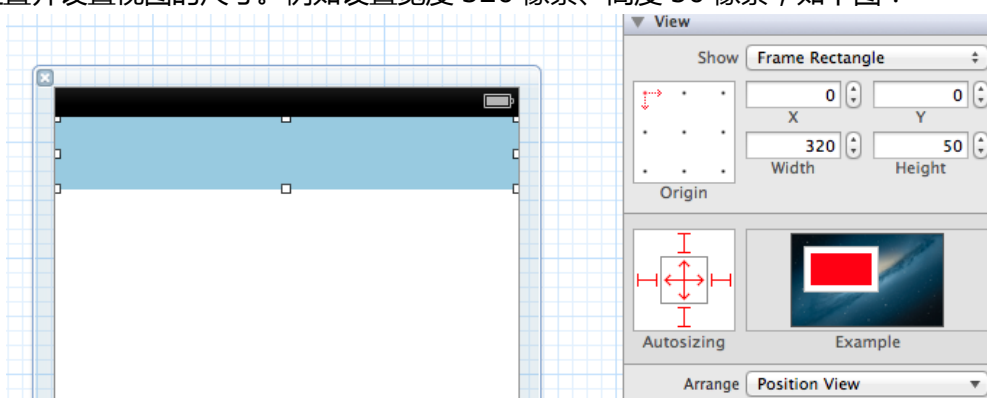
类型	数值（单位:秒）	说明
特殊值	0	默认值，以系统的默认时间为准
	1	仅请求 1 次广告，适合于聚合平台使用
区间值	20~600	最小 20，最大 600，如果取值超过此区间则自动使用默认值 0

提示：广告刷新时间根据经验并不是越小越好，展示时间过短会影响广告浏览者对广告内容产生兴趣的概率，一般取值在 45-60 秒之间时广告被点击的概率会上升到最大。

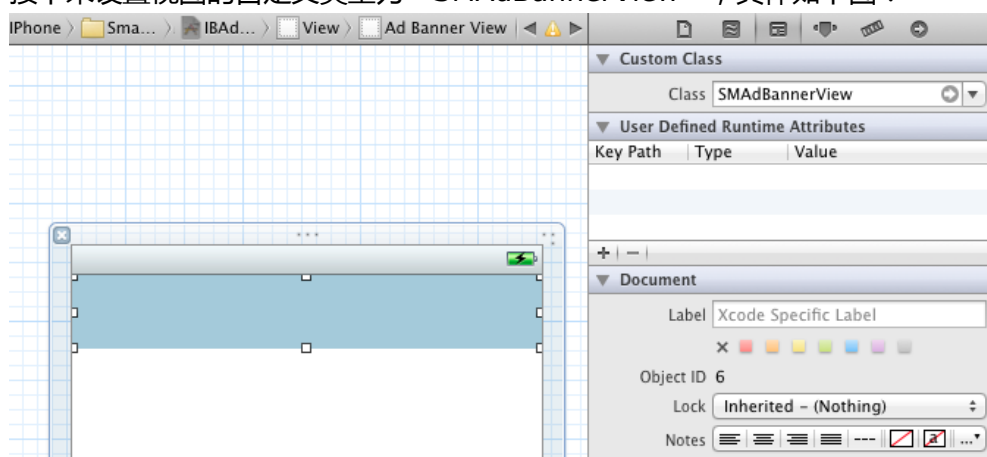
## 4.5.添加固定旗帜广告

### 4.5.1. Design 方式：通过 Interface Builder 添加广告（选择）

Step 1: 在工程项目中找到需要添加广告的“.xib”文件，拖拽出一个“UIView”放置于需要展示广告的位置并设置视图的尺寸。例如设置宽度 320 像素、高度 50 像素，如下图：



Step 2: 接下来设置视图的自定义类型为“SMAdBannerView”，具体如下图：



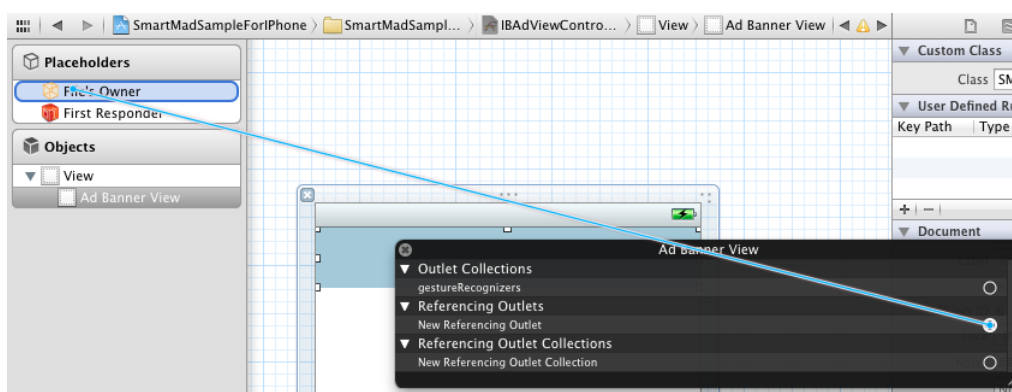
Step 3: 然后在 xib 源代码文件中添加 SAdBannerView 控件的 Outlets 属性 添加以下代码 (粗体字部分) :

```
// 添加到.h 文件
#import "SAdBannerView.h"
@interface IBAViewController:UIViewController
@property(nonatomic,retain)IBOutlet SAdBannerView* banner;

// 添加到.m 文件
@synthesize banner
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view from its nib.
    self.banner.adSpaceId = @"ad position id";
    self.banner.rootViewController = [[UIApplication sharedApplication].keyWindow
    rootViewController];
}
```

注意：“AdSpaceId” 属性值请使用亿动智道移动广告平台建立媒体时所分配的“广告位”。

Step 4: 最后设置 SAdBannerView 类型的 UIView 和 File's Owner 的 Outlets 进行连接，如下图：



#### 4.5.2. Coding 方式：通过 Objective-C 代码添加广告（选择）

如果不习惯于使用 IB 来设计界面，而是通过 Objective-C 编写代码来实现程序 UI，可以在需要添加广告的容器中添加以下代码（粗体字部分）：

```
// 添加到.h 文件
#import <UIKit/UIKit.h>
#import "SAdBannerView.h"

@interface ProgrammingBuildAdViewController:UIViewController {
    // Declare SmartMad ad view member variables.
    SAdBannerView* _banner;
}
}
```

```
// 添加到.m 文件
- (void)viewDidLoad {
    [super viewDidLoad];
    // Do any additional setup after loading the view.
    _banner = [[SMAdBannerView alloc] initWithAdSpaceId:@"ad position id"];
    _banner.rootViewController = [[UIApplication sharedApplication].keyWindow
rootViewController];
    [self.view addSubview:_banner];
}

- (void)dealloc {
    if (_banner) {
        [_banner removeFromSuperview];
        [_banner release];
        _banner = nil;
    }
    [super dealloc];
}
```

注意：“AdSpaceId”为在亿动智道移动广告平台上申请媒体时分配的广告位标识。

4.5.3. 固定位旗帜广告委托事件说明（选择）

添加固定位旗帜广告时，可以通过一些委托事件接收广告当前状态。固定位旗帜广告委托事件类 SMAdBannerViewDelegate 定义的具体事件函数如下表：

委托事件	说明
adBannerViewDidReceiveAd:(SMAdBannerView*)adView	旗帜广告接收成功
adBannerView:(SMAdBannerView*)adView didFailToReceiveAdWithError:(SMAdEventCode*)errorCode	旗帜广告接收失败
adBannerViewWillPresentScreen:(SMAdBannerView*)adView	旗帜广告将要添加到屏幕上
adBannerViewWillDismissScreen:(SMAdBannerView*)adView	旗帜广告将要从屏幕上移除
adBannerViewDidDismissScreen:(SMAdBannerView*)adView	旗帜广告已经从屏幕上移除
adBannerViewWillLeaveApplication:(SMAdBannerView*)adView	旗帜广告被点击后应用切换到后台，例如：调用系统浏览器
adDidClick	旗帜广告已经被点击
adWillExpandAd:(SMAdBannerView *)adView	旗帜广告将要被扩展
adDidCloseExpand:(SMAdBannerView*)adView	旗帜广告已经关闭扩展
appWillSuspendForAd:(SMAdBannerView*)adView	应用将被旗帜广告遮挡，例如：调用内置浏览器或全屏广告
appWillResumeFromAd:(SMAdBannerView*)adView	应用将从被广告遮挡状态恢复

建议：应用被广告遮挡时应暂停逻辑执行，直到终端用户关闭广告解除了遮挡状态，再恢复逻辑执行。

## 4.6.添加浮动位插页广告

### 4.6.1. Coding 方式：通过 Objective-C 代码添加广告

浮动位插页广告仅能通过 Objective-C 编写代码的形式进行添加，可以在需要添加插页广告的容器类中插入以下代码：（粗体字部分）：

```
// 添加到.h 文件
#import "SMAdInterstitial.h"
@interface InterstitialViewController : UIViewController {
    SMAdInterstitial* _interstitial;
}
@end

// 添加到.m 文件
- (void)viewDidLoad {
    _interstitial = [[SMAdInterstitial alloc] initWithAdSpaceId: @"ad position id"];
    _interstitial.delegate = self;
    [_interstitial requestAd];
}

// 添加到.m 文件中需要调用显示插页广告的位置
[_interstitial presentFromRootViewController:[[UIApplication
sharedApplication].keyWindow rootViewController]];
```

注意：插页广告的展示需要配合插页广告请求事件，未请求成功时调用插页广告展示将无法响应。

### 4.6.2. 浮动位插页广告事件说明（选择）

添加浮动位插页广告时，可以通过一些委托事件接收广告当前状态。浮动位插页广告委托事件类 SMAdInterstitialDelegate 定义的具体事件函数如下表：

委托事件	说明
adInterstitialDidReceiveAd:(SMAdInterstitial*)ad	插页广告接收成功
adInterstitial:(SMAdInterstitial*)adview didFailToReceiveAdWithError:(SMAdEventCode*)errorCode	插页广告接收失败
adInterstitialDidClick	插页广告将已经被点击
adInterstitialWillPresentScreen:(SMAdInterstitial*)ad	插页广告将要添加到屏幕上
adInterstitialWillDismissScreen:(SMAdInterstitial*)ad	插页广告将要从屏幕上移除
adInterstitialDidDismissScreen:(SMAdInterstitial*)ad	插页广告已经从屏幕上移除
adInterstitialWillLeaveApplication:(SMAdInterstitial*)ad	插页广告被点击后应用切换到后台，例如：调用系统浏览器

建议：插页广告是浮动位广告形式，当添加到屏幕上后，会遮挡当前运行的应用程序，开发者应暂停当前应用的逻辑执行，直到终端用户关闭了广告窗口，插页广告被从屏幕移除之后，再恢复应用的逻辑执行。



## 4.7. 广告必要参数规范 (重要)

### 4.7.1. 广告位标识 : AdSpaceId

- 广告位标识是开发者在亿动智道移动广告平台中建立媒体时分配的一个 8 位字符串型数字。广告位标识与“应用程序/网站 ID”和“包名称”(Bundle Id)互为绑定关系,故一款应用程序仅能使用申请时对应的广告位,否则广告请求将无效。
- 广告位使用时还要遵循以下原则:相同的广告位在同一屏幕(页面)中仅允许使用一次;不相同的广告位在同一屏幕中使用不能超过两个。

### 4.7.2. 广告尺寸类型 : AdSize

- 固定位旗帜广告的广告尺寸类型分类如下表:

设备类型	广告尺寸	值	枚举类型 ( SMAdBannerSizeType )
iPod Touch/iPhone	Auto	0	PHONE_AD_BANNER_MEASURE_AUTO
iPad	300x250	7	TABLET_AD_BANNER_MEASURE_300X250
	468x60	8	TABLET_AD_BANNER_MEASURE_468X60
	728x90	9	TABLET_AD_BANNER_MEASURE_728X90

- 浮动位插页广告的广告尺寸类型分类如下表:

枚举类型 ( SMAdInterstitialSizeType )	说明
AD_INTERSTITIAL_MEASURE_AUTO	插页广告尺寸类型由广告投放系统根据终端设备屏幕尺寸自动决定,此选项为默认设定
AD_INTERSTITIAL_MEASURE_UMAP	插页广告尺寸类型使用“MMA Universal Mobile Ad Package 2.0 - UMAP”标准,仅使用 300x250 的插页尺寸。

### 4.7.3. 广告动画类型 : AdAnimationType

广告切换时,为了能更好的吸引终端用户的眼球,SmartMad SDK 提供了一系列的动画类型供开发者选用,具体动画类型说明如下:



➤ 固定位旗帜广告过场动画类型

枚举类型 ( SAdBannerAnimationType )	说明
BANNER_ANIMATION_TYPE_NONE	旗帜广告切换无过场动画效果
BANNER_ANIMATION_TYPE_RANDOM	旗帜广告切换随机选择过场动画效果
BANNER_ANIMATION_TYPE_FADEINOUT	旗帜广告切换使用淡入淡出动画效果
BANNER_ANIMATION_TYPE_FLIPFROMLEFT	旗帜广告切换使用从左向右翻转动画效果
BANNER_ANIMATION_TYPE_FLIPFROMRIGHT	旗帜广告切换使用从右向左翻转动画效果
BANNER_ANIMATION_TYPE_CURLUP	旗帜广告切换使用向上卷起动画效果
BANNER_ANIMATION_TYPE_CURLDOWN	旗帜广告切换使用向下卷起动画效果
BANNER_ANIMATION_TYPE_SLIDEFROMLEFT	旗帜广告切换使用从左向右滑动动画效果
BANNER_ANIMATION_TYPE_SLIDEFROMRIGHT	旗帜广告切换使用从右向左滑动动画效果

➤ 浮动位插页广告显示关闭动画类型:

枚举类型 ( SAdInterstitialAnimationType )	说明
INTERSTITIAL_ANIMATION_TYPE_NONE	插页广告显示关闭无动画效果
INTERSTITIAL_ANIMATION_TYPE_POPUP	插页广告显示关闭使用弹出动画效果
INTERSTITIAL_ANIMATION_TYPE_FADEINOUT	插页广告显示关闭使用淡入淡出动画效果

## 5. 广告植入高级技巧

### 5.1. 广告预缓存机制应用

广告预缓存机制是 SmartMad SDK 为开发者提供的一种在应用程序启动时根据自身需要，调用获取广告缓存物料加速正式广告投放的广告通用型技术。

#### 5.1.1. Coding 方式：通过 Objective-C 代码调用广告预缓存机制

使用广告预缓存功能必须在实例化广告容器之前完成，具体代码如下（粗体字部分）：

```
// 添加到.m 文件
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [SAdManager enableAdPrecache: @"ad position id" adUnitType:SAdUnitType];
    [SAdManager setPrecacheDelegate:SAdPrecacheDelegate];
}
```

**注意：**广告预缓存功能当前仅支持应用程序启动后在请求正式广告前执行一次，且请求必须指定需要预缓存的广告位和广告类型。广告预缓存功能仅对广告物料尺寸较大的广告生效(例如：富媒体广告)，且处于节省终端用户流量的考虑，本功能仅在 WIFI 请求时才能正常激活。

5.1.2. 广告预缓存必要参数说明

- 广告位(SMAdSpaceId)  
亿动智道移动广告平台中申请应用程序媒体时分配的广告位。预缓存使用的广告位必须和正式广告请求的广告位相同，这样预缓存加速才会有效果。
- 广告位类型(SMAdUnitType)  
亿动智道移动广告平台中的广告位当前是包含固定位旗帜广告和浮动位插页广告两种类型，且开发者可以根据自己的情况进行选择，所以调用广告预缓存时开发者也必须根据在平台中的设置传入需要相应的类型参数，具体参数设置见下表：

枚举类型	说明
BANNER_AD_UNIT_TYPE	请求广告位中的旗帜广告预缓存单元
INTERSTITIAL_AD_UNIT_TYPE	请求广告位中的插页广告预缓存单元

5.1.3. 广告预缓存事件说明

SmartMad SDK 提供了广告预缓存功能执行时状态委托事件，方便开发者实时的掌握广告预缓存的动态信息，具体委托事件说明见下表：

委托事件(SMAdPrecacheDelegate)	说明
adPrecacheStarted	广告预缓存已经开始
adPrecacheCompleted	广告预缓存完成
adPrecacheError:(SMAdEventCode*)errorCode	广告预缓存错误，代码见广告事件

建议：开发者可以根据广告预缓存完成的状态开始正式广告的请求。

5.2.提供信息辅助广告精确定向

5.2.1. 提供终端用户信息

开发者如果希望提高广告的转化率，可以将一些仅用于广告投放服务的终端用户信息提交到亿动智道移动广告平台，我们将根据所提供的信息为你进行具有优化效果的广告精准定向投放。这些信息使用 JSON 格式进行传输，具体调用代码如下（粗体字部分）：

```
// 添加到.m 文件
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [SMAdManager setUserInformation:@"jsonUserInfo"];
}
```

具体可以提交的用户信息内容如下表：

内容项	说明
{"age": "string"}	终端用户年龄信息
{"gender": boolean}	终端用户性别信息，false代表女性，true代表男性
{"birthday": "string"}	终端用户生日信息
{"hobby": "string"}	终端用户爱好信息
{"city": "string"}	终端用户所在城市信息
{"zipcode": "string"}	终端用户所在地邮政编码信息
{"work": "string"}	终端用户工作类型信息

注意：用户信息内容的提交调用需要在正式广告请求调用之前完成。

5.2.2. 提供关键字信息

开发者可以参与并提高广告转化率的另一种方法是将应用的类型或者用于应用程序的关键字提交到亿动智道移动广告平台，我们将根据所提供的关键字为你进行具有优化效果的广告精准定向投放。具体调用代码如下（粗体字部分）：

```
// 添加到.m 文件
- (BOOL)application:(UIApplication *)application
didFinishLaunchingWithOptions:(NSDictionary *)launchOptions {
    [SMAdManager setKeywords: @"keywords"];
}
```

注意：关键字的提交调用需要在正式广告请求调用之前完成，如果有多个关键字使用“,”进行分隔。

5.3.广告事件(SMAdEventCode)说明

SmartMad SDK 在广告正式请求或广告预缓存请求时 ,通过定义的事件接口会返回事件状态代码，这些状态代码的具体定义说明如下表：

枚举类型	枚举值	说明
SUCCEED	0	执行成功
INVALID_ID	1	唯一标识校验没有通过（例如：广告位）
AD_TOO_MUCH_ON_SCREEN	2	使用了过多的广告位（例如：超过2个）
INVALID_AREA_OR_BE_COVERD	4	广告视图加载到了窗口之外
NETWORK_ERROR	6	网络错误（例如：移动设备无法检测到网络）
INVALID_REQUEST	7	无效的广告请求
NO_FILL	8	有效的广告请求，但无广告填充
INTERNAL_ERROR	9	内部错误