



AdView iOS SDK 开发者手册

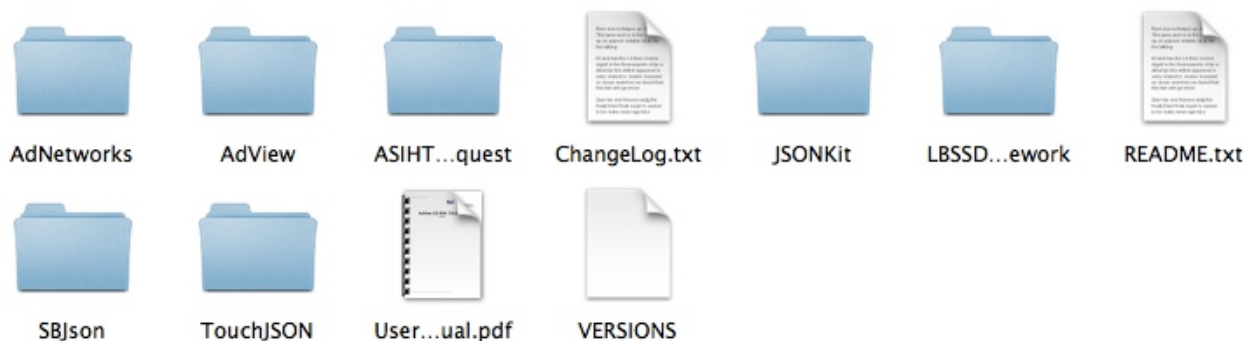
v1.5.x

I. AdView iOS SDK 介绍	3
II. 在项目中集成 AdView	4
III. 在程序界面中嵌入 AdView	7
IV. 删除不需要广告平台	10
V. 常见问题处理	11
VI. Cocos2d和Cocos2d-x集成说明	12
VII. Unity3D集成指南	13
VIII.Flex集成指南	15
IX. Flex AdView ANE SDK说明	22

I. AdView iOS SDK 介绍

AdView 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。

AdView iOS SDK 包含如下内容：

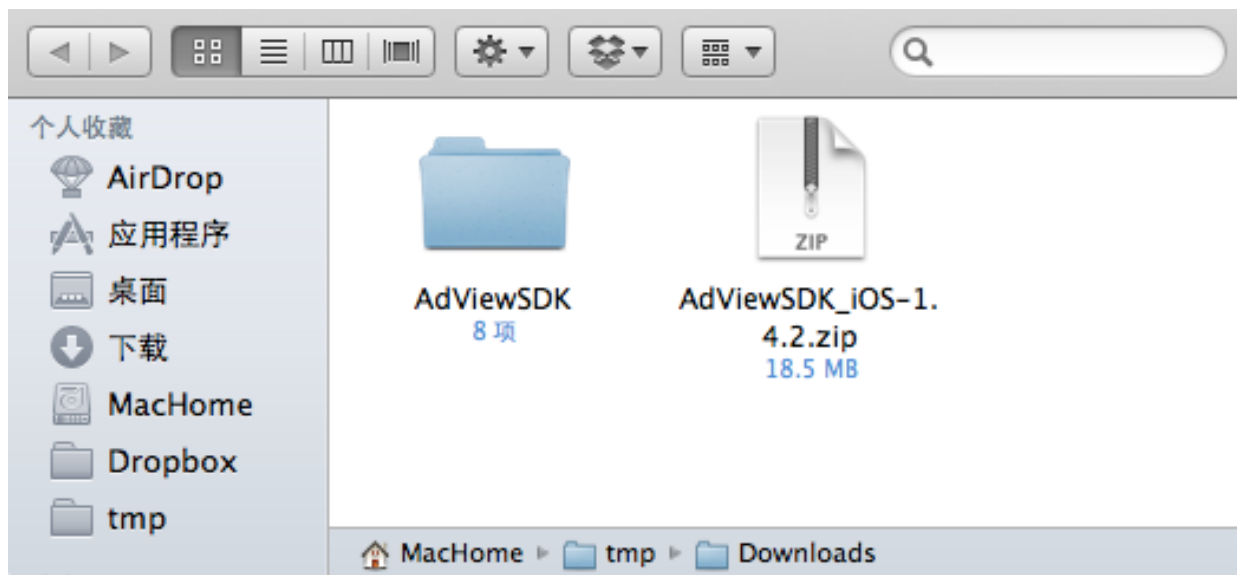


文件或目录名	说明
AdNetworks	AdView SDK 支持的广告平台开发包.
AdView	AdView SDK 开发包.
ASIHTTPRequest	部分广告平台依赖的ASI开源库.
SBJson	部分广告平台依赖的 json-framework 源代码.
TouchJSON	部分广告平台依赖的 TouchJSON 源代码.
JSONKit	部分广告平台依赖的 JSONKit 源代码.
LBSSDK.framework	聚赢平台依赖的框架.
ChangeLog.txt	更新日志.
README.txt	广告平台版本，符号冲突，和一些使用限制上的说明.
UserManual.pdf	使用手册.
VERSIONS	当前版本号.

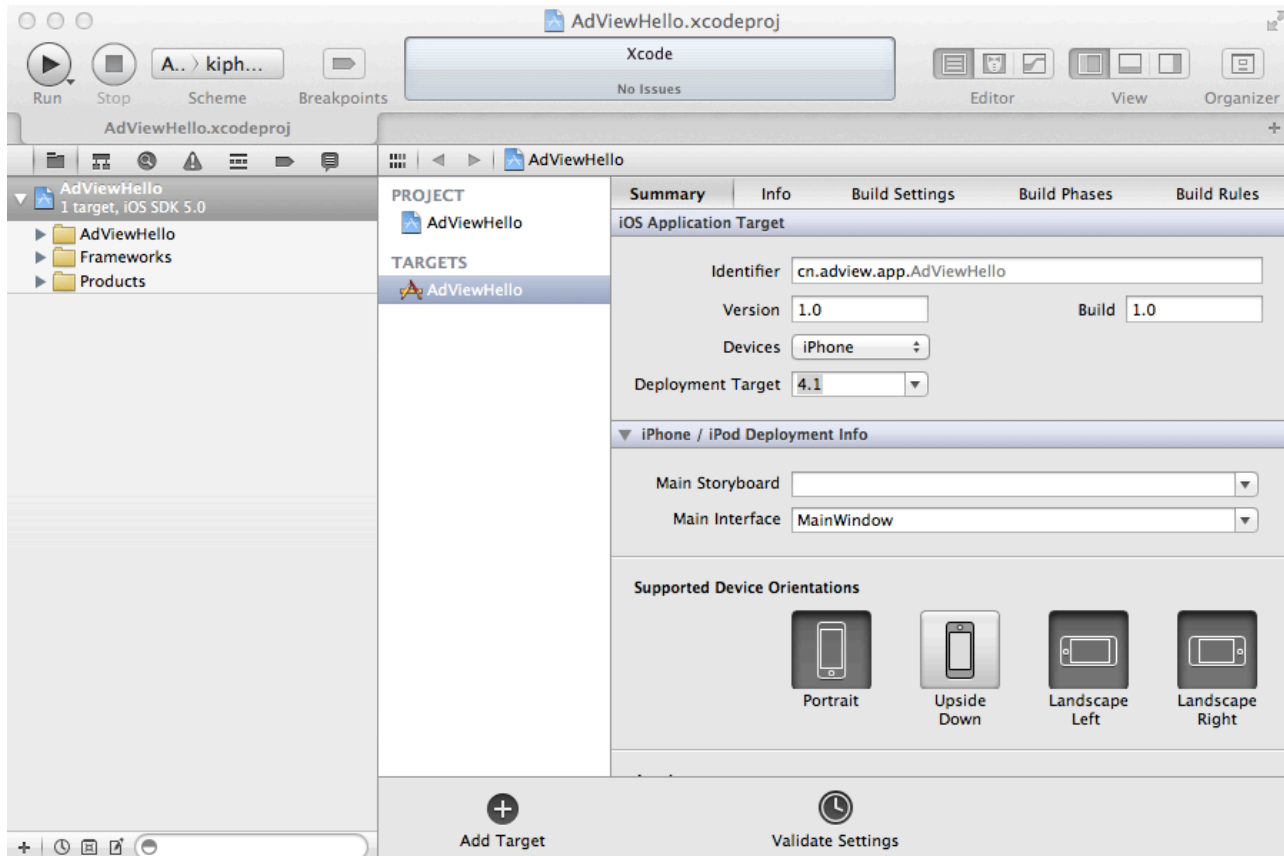
II. 在项目中集成 AdView

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

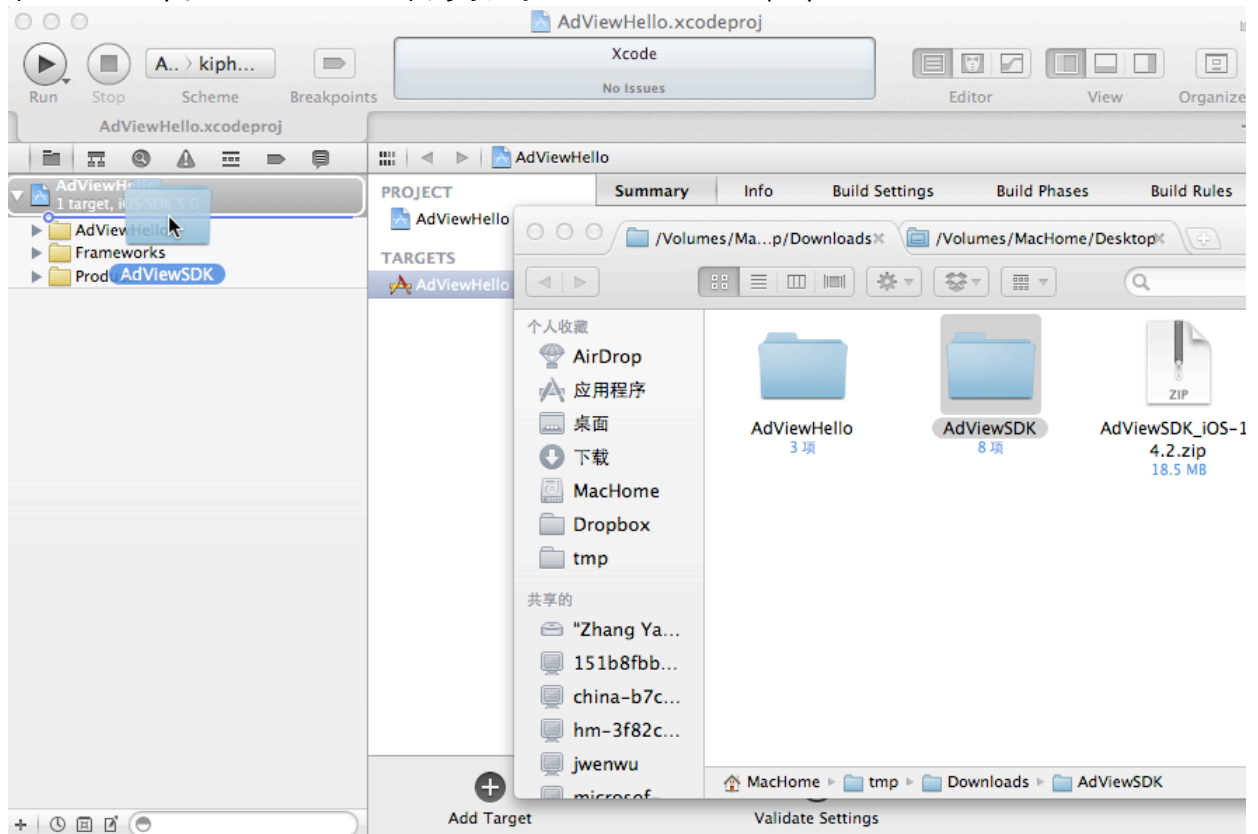
1. 从网站获取 AdView SDK 开发包, 在 Finder 中解压. 您将在 Finder 中看到解压后的 AdView SDK. 如图所示.



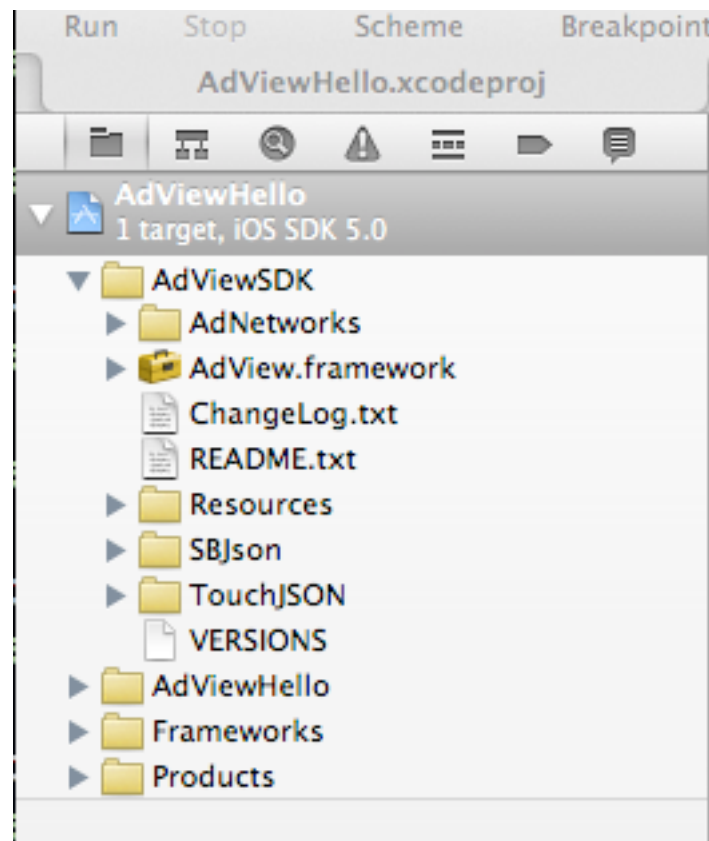
2. 在 Xcode 中打开您的工程. 在此以 AdView SDK 所附带的 AdViewHello 工程为例.



3. 在 Finder 中把 AdViewSDK 目录拖到 AdViewHello 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 AdView SDK 即已经添加到项目中. 如图所示.



5. 若需要使用离线配置功能，从网站“我的产品”——“ios”——对应应用——“广告优化”页面，点击“离线配置下载”，下载名字如“SDK20111022530129m85is43b70r4iyc.txt”的文件，放入到项目目录下，加入到项目，确认该文件被选中会被打包到app。并确认这个名字的txt文件和要填入的 `adViewApplicationKey` 委托函数返回的字符串是一致的。在无法从网上取得配置的情况下，adview会优先查看是否有上次取得的配置的备份，若没有，才会使用离线配置数据。
6. AdView SDK开发包附带了所有支持的广告平台. 但是不幸的是某些广告平台之间会有一些符号冲突. 所以您不得不忍痛割爱一些广告平台. 这些广告平台的冲突请见 README.txt.
7. 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：
 - AirAD仅支持Xcode 4.2, IOS 4.0以上。
 - Admob要求Xcode 4.2以上。
8. 其他：
 - 不同广告平台库做链接时可能冲突，详见README.txt.
 - AdView不直接使用地理位置信息，广告平台SDK可能使用.

III. 在程序界面中嵌入 AdView

1. 首先您需要现在 AdView 网站上注册帐号并创建您的应用程序, 获取 application key.
2. 在需要嵌入 AdView 的界面对象中实现 AdView 的 Delegate, 如下代码所示:

```
#import "AdViewView.h"
#import "AdViewDelegateProtocol.h"

@interface SimpleViewController : UIViewController <AdViewDelegate>
{
.....
    AdViewView *adView;
.....
}

.....
@property (nonatomic, retain) AdViewView* adView;
.....
@end
```

```
@implement SimpleViewController
.....

- (NSString*) adViewApplicationKey
{
    return @"SDK20111022530129m85is43b70r4iyc"; // 在 AdView 网站申请的 key.
}

- (UIViewController*) viewControllerForPresentingModalView {
    return [((AdViewHelloAppDelegate *)[[UIApplication sharedApplication] delegate])
        navigationController]; // 全屏广告附着的控制器.
}

- (BOOL) adViewTestMode
{
    return NO; //在测试时返回 YES;
}

//以下两个方法若不需要baidu广告可不实现
- (NSString *)BaiDuApIDString {
    return @"2f952126"; //返回baidu广告网站的adview key
}

- (NSString *)BaiDuApSpecString {
    return @"debug"; //返回实际计费键值, 在测试时返回 @"debug"
}

.....
@end
```

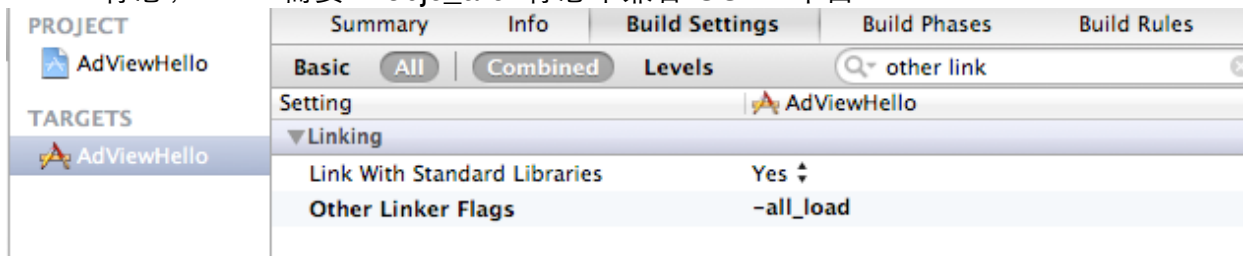
3. 实现 AdView Delegate, 如下代码所示.
4. 在控制器的 viewDidLoad 函数中创建 AdViewView 并开始请求广告, 如下代码所示.

```

- (void) viewDidLoad
{
    .....
    self.adView = [AdViewView requestAdViewViewWithDelegate:self];
    [self.view addSubview:self.adView];
    .....
}

```

5. 在项目设置中设置 Other Linker Flags 的值, 并添加 “-all_load” 标志。另VPON需要“-lxml2”标志；AirAD需要“-fobjc_arc”标志来兼容IOS 4.x平台。



6. 若想要对广告条的大小做定制, 可实现委托函数 PreferBannerSize, adview将根据返回值选择最接近的大小做适配, 如果不实现, 将自动选择, iphone选择320宽的小尺寸, ipad将选择宽在700以上的大尺寸。若广告平台不支持, 将只支持广告平台的缺省尺寸。

```

- (void) PreferBannerSize
{
    return AdviewBannerSize_320x50;
}

```

7. 若希望能打开各广告平台对于Gps的支持, 可实现委托函数 adGpsMode, 返回YES将打开。

```

- (BOOL) adGpsMode
{
    return YES;
}

```

8. 在您的工程中添加 AdView 以及各广告平台依赖的 framework.
以下 framework 为 AdView 以及各广告平台所依赖

- libiconv.dylib
- libsqlite3.dylib
- libstdc++.dylib
- libz.dylib
- AudioToolbox.framework
- AVFoundation.framework

- CFNetwork.framework
- CoreGraphics.framework
- CoreLocation.framework
- CoreMedia.framework
- CoreMotion.framework
- CoreTelephony.framework
- CoreVideo.framework
- EventKit.framework
- Foundation.framework
- iAd.framework
- ImageIO.framework
- MapKit.framework
- MediaPlayer.framework
- MessageUI.framework
- MobileCoreServices.framework
- QuartzCore.framework
- Security.framework
- SystemConfiguration.framework
- UIKit.framework

9. 在 AdView 网站上配置您的各个广告平台.

10. 编译并链接您的工程, AdView 即可显示广告.

IV. 删除不需要广告平台

删除不需要的广告平台非常容易, 只需要在项目工程中删除 `AdViewSDK/AdNetworks` 组下相应的广告平台库文件, 重新编译工程即可.

在从工程中删除广告平台时我们建议只是删除工程引用, 而不要把广告平台从硬盘上删除. 保留这些广告平台的开发库以备不时之需.

V. 常见问题处理

1. 为什么 AdView 无法显示广告？
最大可能性的原因在于您没有为您的项目中添加额外的链接选项，使广告平台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AdView](#)”的第 5 步。
2. 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？
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平台之间，这个比较容易解决，您只需要注意冲突发生在那些广告平台之间即可。已知的广告平台冲突发生在以下几个：
 - AdChina, WinAd 和 Wooboo 都包含了 Reachability，所以其中任何两个库共存的情况下，会发生符号冲突。这个办法暂时无法解决，您只能使用其中一个广告平台。
 - Casee 与 SBJson 的冲突。由于 Casee 已经包含 SBJson 部分的代码，所以您可以从工程中删除 AdViewSDK/SBJson 组的所有源代码。
3. 工程添加 -all_load 标志后在链接外部第三方静态库时发生冲突。
发生这种错误时，原因的确是出于 -all_load 标志的问题，要避免这个问题，请确认您的 Xcode 版本在 3.2.6 以上。并且删除 -all_load 标志，而使用 -force_load 逐个添加您需要链接的第三方静态库以及 AdView/AdNetworks 组下的广告平台开发库。
4. 为什么找不到 AdViewView.h 头文件？
AdView 开发库从 1.4.3a 开始不再使用 framework 的封装方式。请使用 #import “AdViewView.h”方式包含头文件
5. 为什么运行 AirAD 广告在 4.x 的手机上崩溃？
AirAD 库为包含 ARC 的外部库，需要加入 -fobjc_arc 的 linker flag 参数。

VI. Cocos2d和Cocos2d-x集成说明

1. 请参考例子AdViewCocos2dHello和AdViewCocos2dxHello工程。
2. 广告平台对Cocos2d-x的兼容性不如Cocos2d，请开发者根据实际需要和运行情况选择广告平台。
3. Cocos2d-x集成AdView后，点击某些广告平台的广告再返回时，可能弹出虚拟键盘，且无法取消（可以切换到某些广告平台后点击返回时被取消，比如AdMob）。一个修复的方法是，修改EAGLView类的- (BOOL)canBecomeFirstResponder方法，让它返回NO，请参考AdViewCocos2dxHello工程，开发实际项目时，可以根据实际需要进行修改。

VII. Unity3D集成指南

1. 生成Unity3D IOS工程。
2. 将AdViewSDK和AdViewToolLevel文件夹拷贝到工程目录下。在xcode中将它们都加入，加入AdViewSDK需要的FrameWork和链接标志（尤其是-all_load）。去掉AdViewSDK中不想要的广告平台，去掉AdViewSDK中可能引起冲突的JSONKit.m和SBJsonParser.m文件。
3. 试编译一次，可能出现链接问题，如果出现很多undefined symbol问题，可能是因为Unity3D的库中，有些代码调用了一些外部类或函数，实际上是这些函数不会使用到，Unity3D的库也就没有包含。碰到这种情况，如果不多的话，可以自己写个空的同名同参同返回类型的函数模拟，以便链接通过，可参考AdViewToolLevel/unity3d/SimMethod.cpp文件。如果很多，就只能不使用-all_load标志，目前AdViewSDK可能出现系列问题，待解决。
4. 第3步编译成功，则表示集成AdViewSDK是可行的。后面的步骤将实际加入AdView。

5. 找到AppController.mm文件，加入以下代码：

```
#import "AdViewController.h"
```

```
void setupAdView(UINavigationController *viewController, UIView *view)
{
    AdViewController *controller = [AdViewController sharedController];
    [controller setAdViewKey:@"SDK20111022530129m85is43b70r4iyc"];
    [controller setModeTest:YES Log:YES];
    [controller setOrientationUp:NO Down:NO Left:NO Right:YES];
    [controller setAdBannerSize:AdviewBannerSize_Auto];
    [controller setAdRootController:viewController];

    [controller loadView];
    [view addSubview:controller.adView];
}
```

```
void releaseAdView()
{
    [AdViewController deleteController];
}
```

6. 在OpenEAGL_UnityCallback函数中的


```
[_window makeKeyAndVisible];
[view release];
```

加入：

```
setupAdView(controller, view);
```

变成：

```
[_window makeKeyAndVisible];
setupAdView(controller, view);
[view release];
```

7. 在AppController的dealloc方法里首行加入：


```
releaseAdView();
```

8. 编译运行，可以看到广告成功的出来了：



9. 修改广告位置可以调用 `AdViewController`的`setAdPosition`方法进行，`setAdHidden`可以按需要隐藏广告。

10.至此，AdViewSDK集成完成。依场景移动或隐藏显示广告请开发者自己完成。

VIII. Flex集成指南

1. AdView为Flex开发者提供了ANE库，在Flex开发IOS应用时可使用。以后步骤是使用方法，写的比较细，若有Flex使用ANE经验，可不观看，直接拷贝例子程序代码使用AdView ANE库。
2. 打开Flash Builder 4.6.0， 打开菜单“文件”——“新建”——“Flex手机项目”，弹出指南窗口。



3. 如上例填写想要的项目名，按“下一步”，到下一个页面。

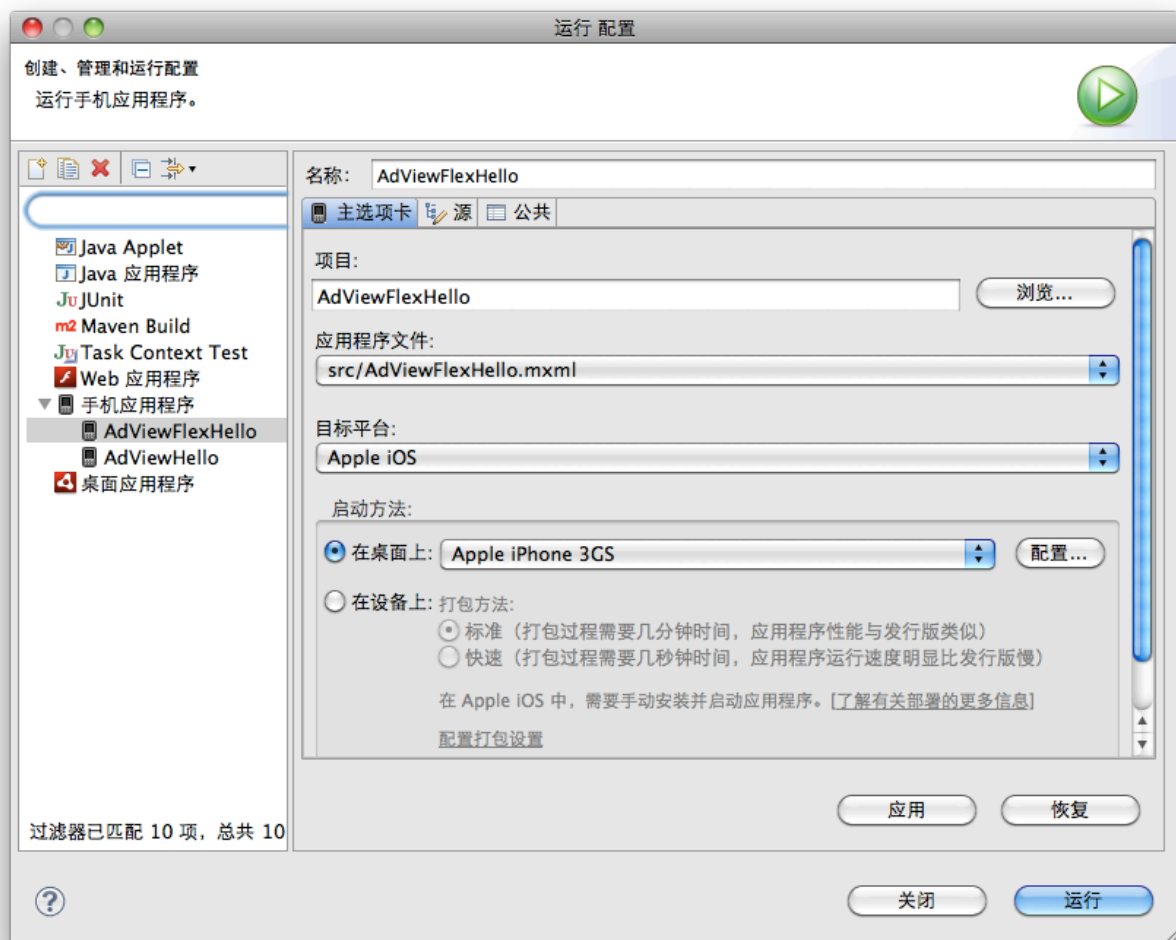


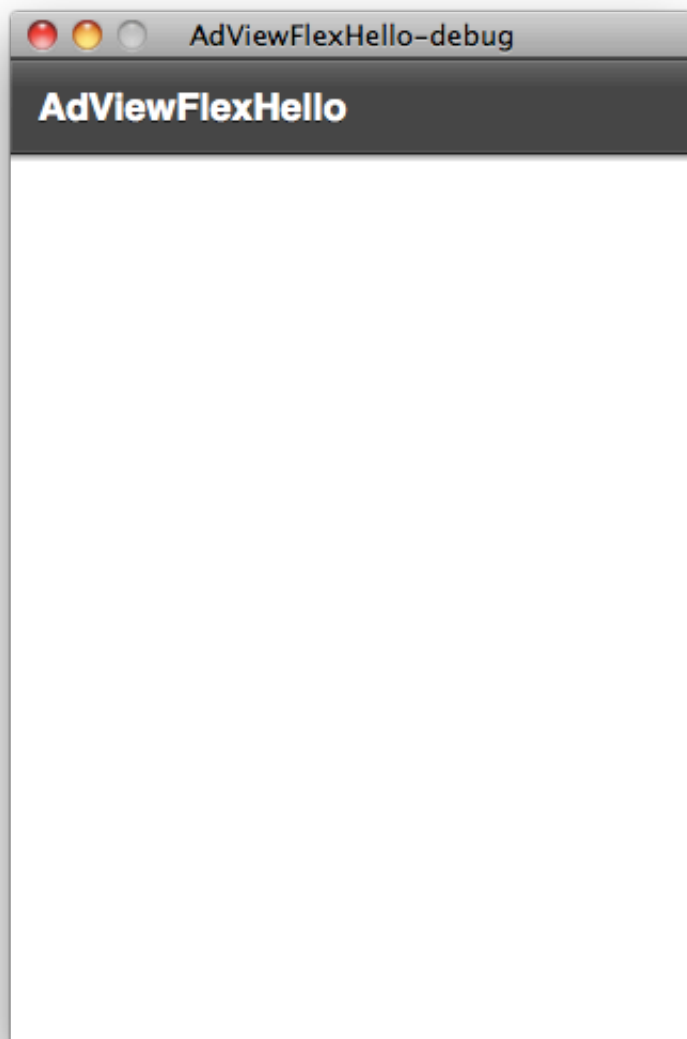
4. 如上图，选中Apple iOS，改变初始视图标识。按完成，等待Flash Builder建立项目。

5. 选中刚建立的项目，如下图，单击右键，弹出菜单，选中“运行方式”——“运行 配置...”，弹出配置对话框。

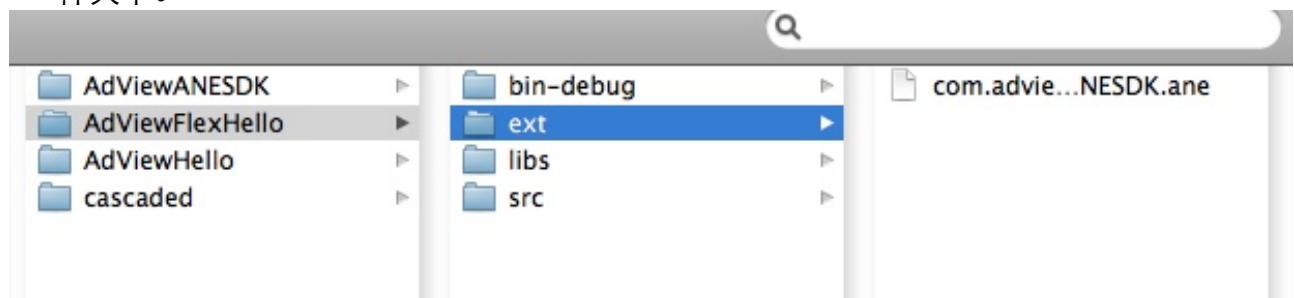


6. 将对话框配置如下图，按“运行”按钮，Flash模拟器将运行。建立Flex项目的步骤已经完成，下一步将是往Flex项目加入AdView的ANE库。





7. 到项目目录下，新建“ext”文件夹，将com.adview.AdViewANESDK.ane文件复制到该文件夹下。



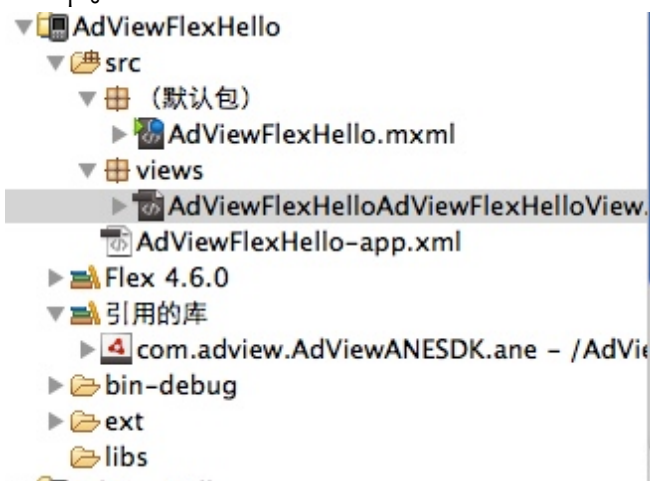
8. 回到Flash Builder界面，选中项目，单击右键，选择菜单中的“刷新”命令，可刷出“ext”目录。
9. 单击右键，选择菜单中的“属性”命令，弹出对话框。选择“Flex 构建路径”，选中“本机扩展”。
10. 单击“添加ANE...”，弹出“添加ANE”对话框，单击“浏览...”，弹出“选中ANE文件”对话框，选择步骤7复制的ANE文件，单击“打开”，退回到“添加ANE”对话框，可以看到文本框已经显示了选中ANE文件的相对路径。



11. 单击“确定”，回到属性对话框，可以看到本机扩展的列表中已经包含选择的ANE文件。选择Apple iOS SDK右边的浏览，选中正确的SDK目录，如下图。将LBSSDK.framework复制到该目录下，最后打ipa包时需要用到。



12. 单击“确定”，等待Builder更新设置，可以看到项目将ANE文件加入到了“引用的库”标签下。



13. 双击src/views/目录下的mxml文件，可打开编辑。（发现文件名字不太对，选中，单击右键选中“重命名...”命令，可改名，Builder会完成改名的所有工作。）修改代码为如下：
 <?xml version="1.0" encoding="utf-8"?>

```

<s:View xmlns:fx="http://ns.adobe.com/mxml/2009"
        xmlns:s="library://ns.adobe.com/flex/spark" title="AdViewFlexHello"
        viewActivate="setupAdView(event)"
        viewDeactivate="releaseAdView(event)">
    <fx:Declarations>
        <!-- 将非可视元素（例如服务、值对象）放在此处 -->
    </fx:Declarations>
    <fx:Script>
        <![CDATA[
            import com.adview.AdViewListener;
            import com.adview.AdViewSDK;

            import spark.events.ViewNavigatorEvent;

            protected function setupAdView(event:ViewNavigatorEvent):void
            {
                try {
                    var adViewSDK:AdViewSDK = AdViewSDK.sharedAdViewSDK();
                    var listener:AdViewListener = new AdViewListener();

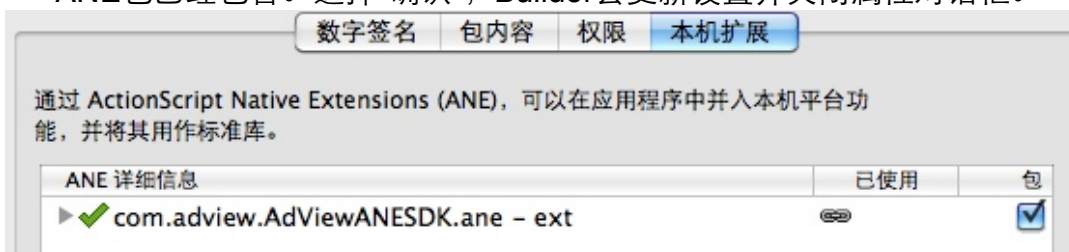
                    adViewSDK.setAdViewListener(listener);
                    adViewSDK.setupAdView
("SDK20111022530129m85is43b70r4iyc",
                        AdViewSDK.AdviewBannerSize_Auto, true, true);

                    adViewSDK.moveAdView(AdViewSDK.AdViewAlignCenter, 0);
                } catch (e:Error) {}
            }

            protected function releaseAdView(event:ViewNavigatorEvent):void
            {
                try {
                    var adViewSDK:AdViewSDK = AdViewSDK.sharedAdViewSDK();
                    adViewSDK.releaseAdView();
                } catch (e:Error) {}
            }
        ]]>
    </fx:Script>
</s:View>

```

14. 为项目选中右键命令“属性”，打开“Flex构建打包”——“Apple iOS”——“本机扩展”，确保ANE包已经包含。选择“确认”，Builder会更新设置并关闭属性对话框。



15. 选择右键菜单“运行方式”——“1手机应用程序”，等待Builder编译并运行模拟器，运行成功，但不会有任何变化，AdView并不支持MacOs模拟器，因此无法实际显示广告。下面步骤将打包真机ipa文件，可以安装ipa文件来在真机测试。
16. 为项目选中右键命令“属性”，打开“Flex构建打包”——“Apple iOS”——“数字签名”，选中有效的证书和配置文件，如何生成证书和配置文件请自己查询文档。单击“确定”返回。
17. 选择右键菜单“运行方式”——“运行配置...”，在弹出的对话框中选中为“在设备上”，如下图。单击“运行”按钮。



18. 弹出“证书密码”对话框，输入密码，单击“确定”，Builder弹出“正在打包”对话框，等待完成。
19. 打包完成，没有错误，可在项目“bin-debug”目录下找到ipa文件，即为真机ipa包。将ipa包安装到真机，运行。
20. 如下图所示，成功获取广告并展示。



21. FlexANE目录下的AdViewHello是一个更完善的例子项目，可参考。

IX. Flex AdView ANE SDK说明

- SDK下载地址:

http://www.adview.cn/download/sdk/AdViewSDK_FlexANE_IOS-1.5.3.zip

- com.adview.AdViewSDK类:

说明:

AdView接口类。

常量:

```
public static const AdviewBannerSize_Auto:int = 0;
public static const AdviewBannerSize_320x50:int = 1;
public static const AdviewBannerSize_300x250:int = 2;
public static const AdviewBannerSize_480x60:int = 3;
public static const AdviewBannerSize_728x90:int = 4;

public static const AdViewAlignLeft:int = 0;
public static const AdViewAlignTop:int = 0;
public static const AdViewAlignCenter:int = -1;
public static const AdViewAlignRight:int = -2;
public static const AdViewAlignBottom:int = -2;
```

函数:

1. 获取AdViewSDK实例:

```
public static function sharedAdViewSDK():AdViewSDK
```

2. 设置监听事件对象:

```
public function setAdViewListener(l:AdViewListener):void
```

3. 创建广告:

```
public function setupAdView(adviewKey:String, adSize:int,
testMode:Boolean, logMode:Boolean):void
```

4. 释放广告:

```
public function releaseAdView():void
```

5. 设置广告位置:

```
public function moveAdView(x:int, y:int):void
```

6. 设置广告显示或者隐藏:

```
public function setAdViewHidden(bHidden:Boolean):void
```

7. 取下一个广告平台的广告, 主要用于测试:

```
public function rolloverAd():void
```

8. 重新获取当前广告平台的广告, 主要用于测试:

```
public function requestNewAd():void
```

- com.adview.AdViewListener类:

说明:

通过继承本类, 覆盖didGotNotification函数以实现回调事件的处理。

常量:

无

函数:

1. 获取广告通知的回调:

```
public function didGotNotification(code:String, content:String):void
```

回调code参数可以有以下值:

```
show, click, failGot, presentModal, dismissModal, gotConfig, toGetAd, adsAreOff
```