

MobFox iOS SDK Integration Guide 4.1.6

The following document provide detailed instructions on how to integrate the MobFox iOS SDK 4.1.6 into your iOS 4.3 to iOS 6.1.3 Projects based on a simple Demo app. The MobFox SDK is capable of displaying both traditional Banner Ads and (Rich Media) Interstitials.

Step 1: Set Up your Application at the MobFox Dashboard

- Log into your MobFox account or sign up at <http://www.mobfox.com/signup>
- At the MobFox Dashboard navigate to "Sites & Apps"
- Click on "Create New Site/App" and enter the details
- Once created you will be provided with a unique Publisher ID

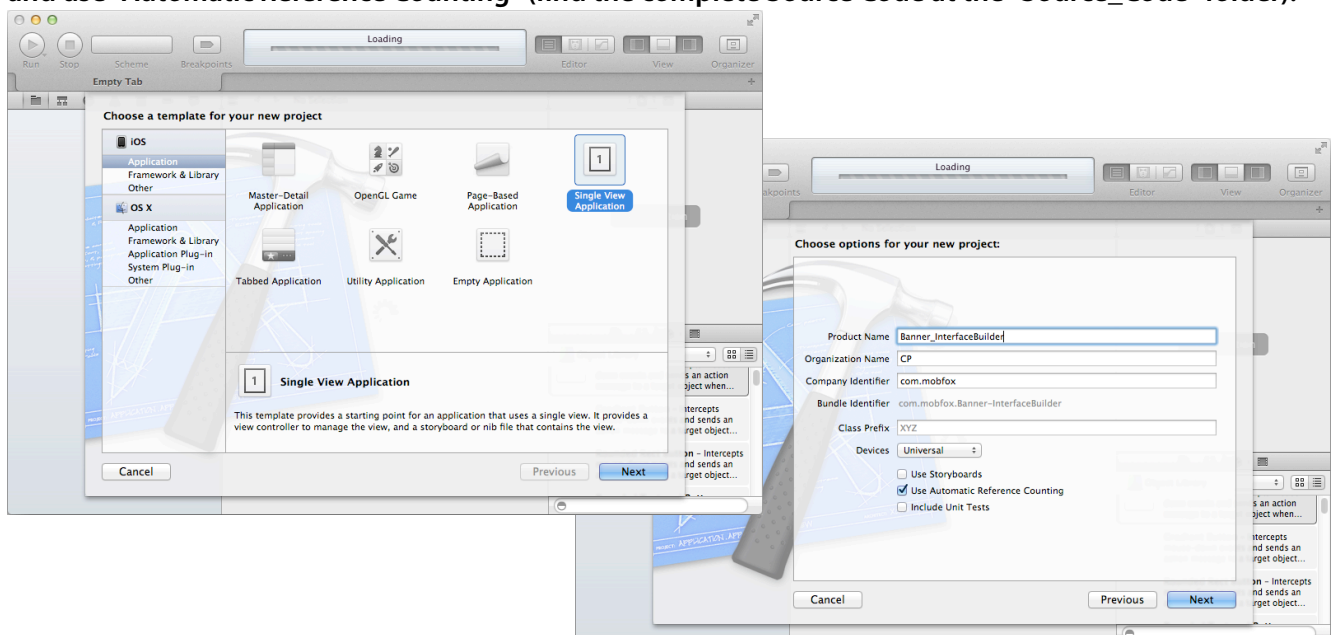
Step 2: Download the SDK

The downloaded ZIP contains the following files:

- PDF Documentation "MobFox iOS SDK Setup Guide 4.1.6.pdf" (this document)
- The SDK Framework itself: MobFox.framework
- Demo Application "MobFoxDemo" folder:
 - Request Banner Ads and Video/Interstitial Ads with Buttons
 - Based on Xcode 4.6.1
- "Source_Code" folder with 3 additional here described Coding and Customization options

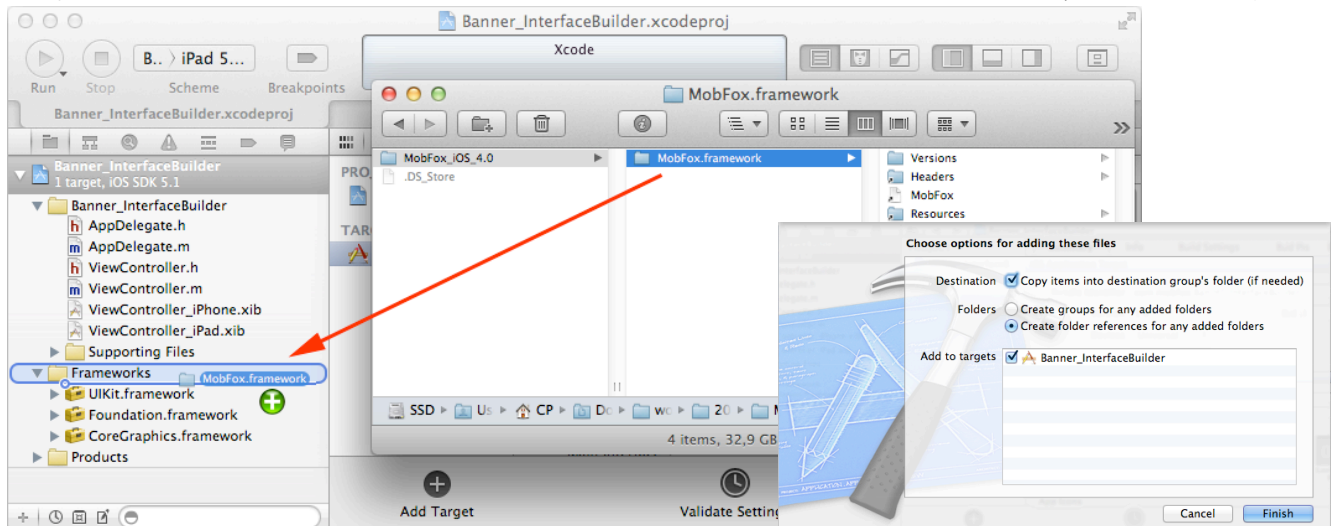
Step 3: Create and define a new iOS Project in Xcode

Create a New Project "Banner_InterfaceBuilder", select "Single View Application", use no "Storyboard" and use "Automatic Reference Counting" (find the complete Source Code at the "Source_Code" folder).

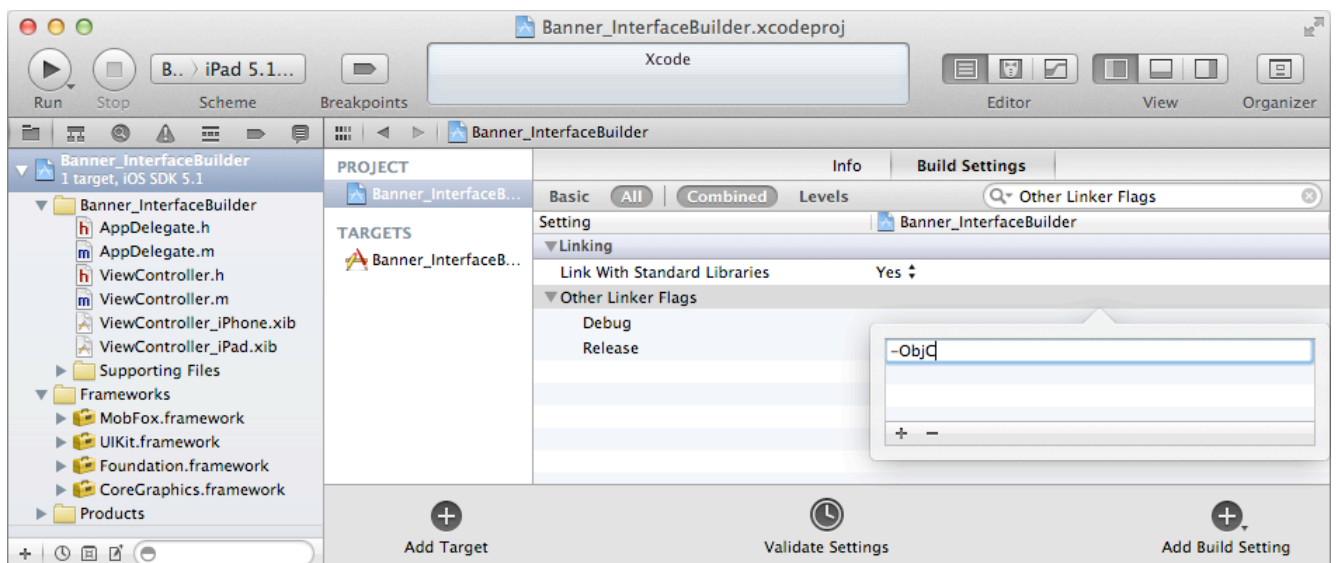


Step 4: Add the SDK to your project "Frameworks" folder

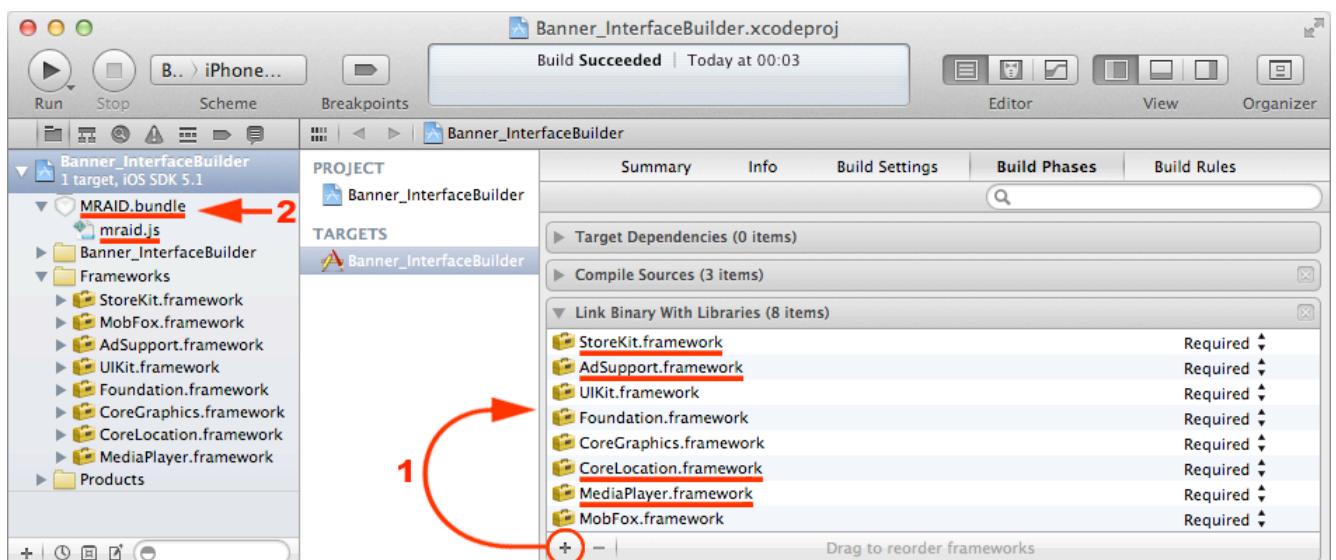
Drag & Drop the MobFox Framework from the Finder into the Frameworks Folder in your Xcode project



Under "Project" > "Build Settings" search for "Other Linker Flags" and add "-ObjC" to load objective-C code from libraries. Note: For non ARC Projects add an additional line "-fobjc-arc".



1) Add mandatory Frameworks and 2) add MRAID.bundle into root.



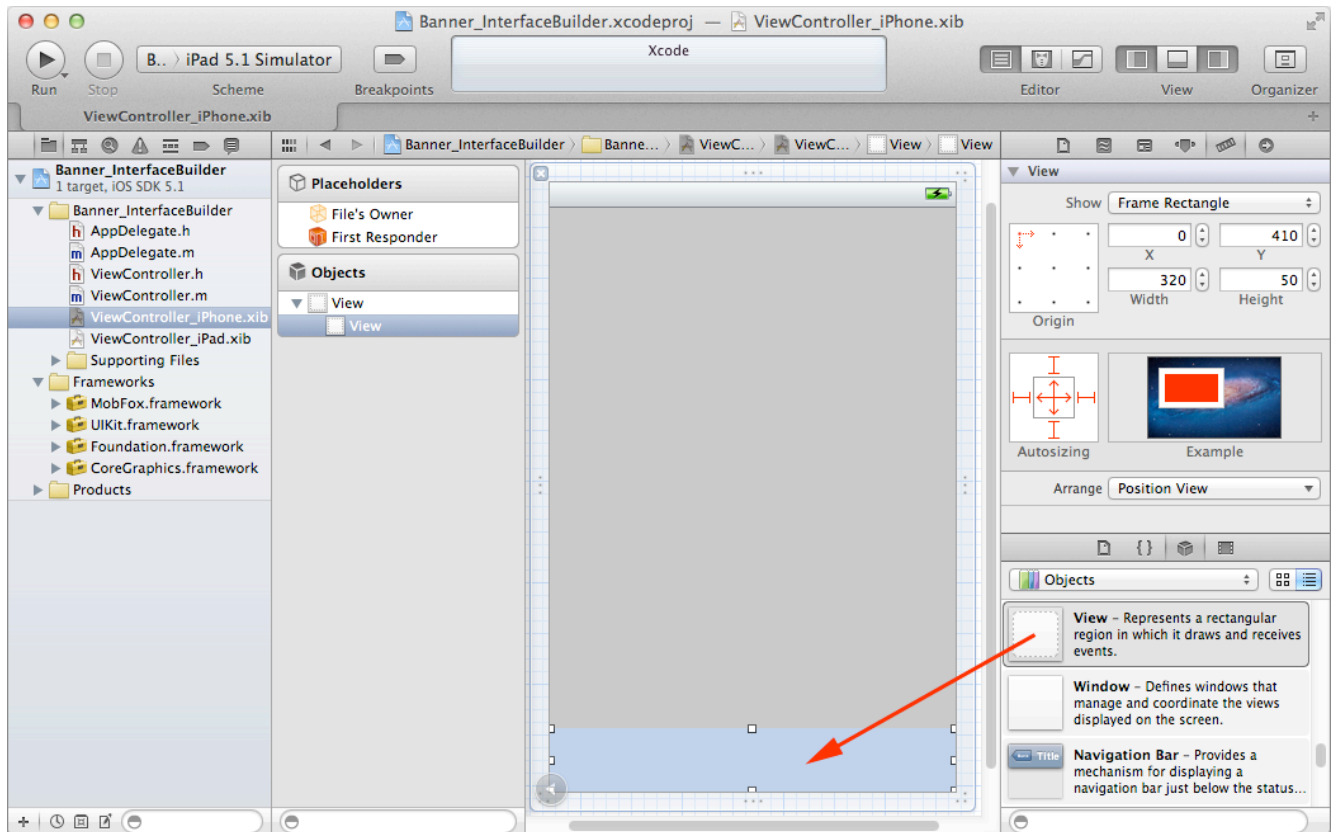
To display Banner Ads please continue with "Step 5".

To display MobFox:vAds, our high-paying Premium ad format, please jump to "Step 6".

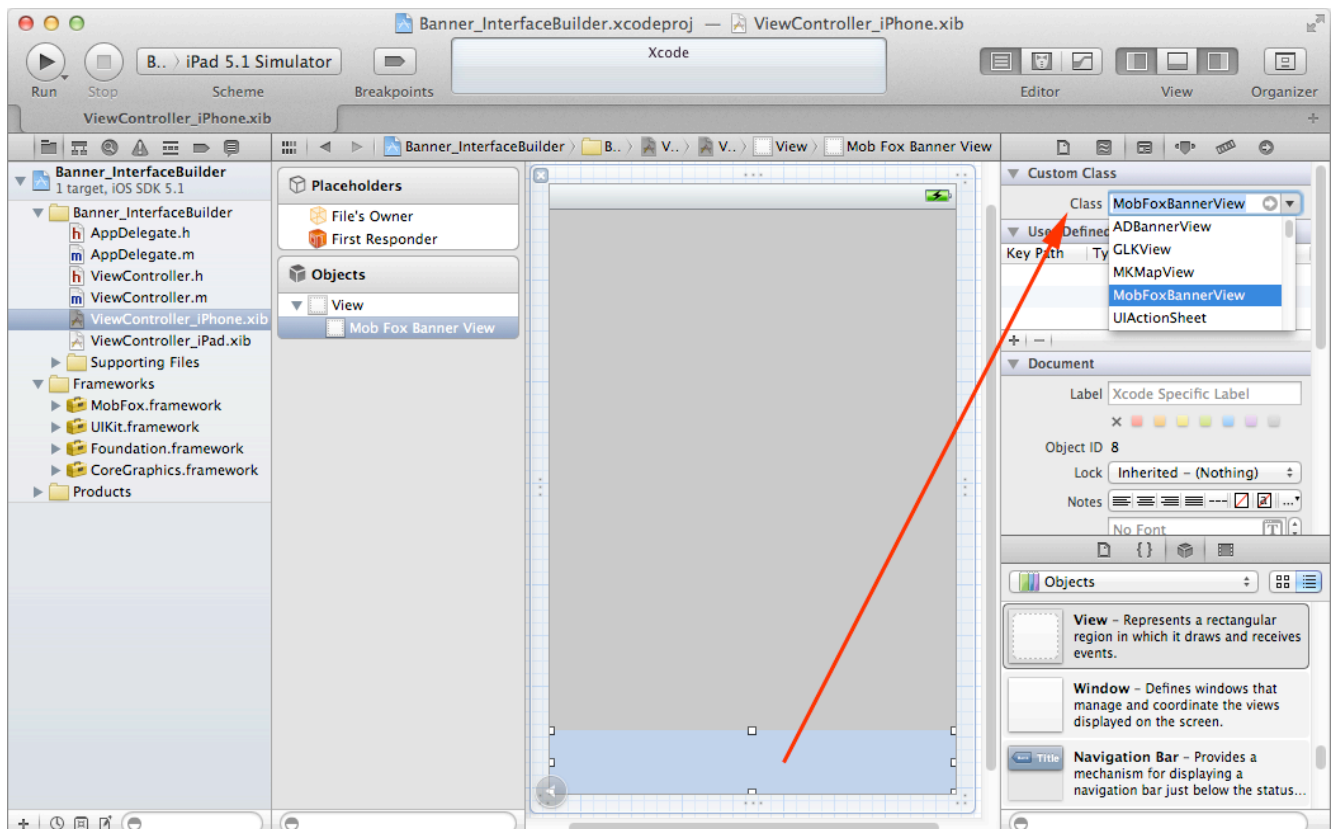
Step 5: Request and display Banner Ads using Interface Builder

iPhone: Go to the "ViewController_iPhone.xib" and add a View sized 320*50

iPad: Go to the "ViewController_iPad.xib" and add a View sized 728x90



Set the Class of this View to "MobFoxBannerView". Do the same for the "ViewController_iPad.xib".

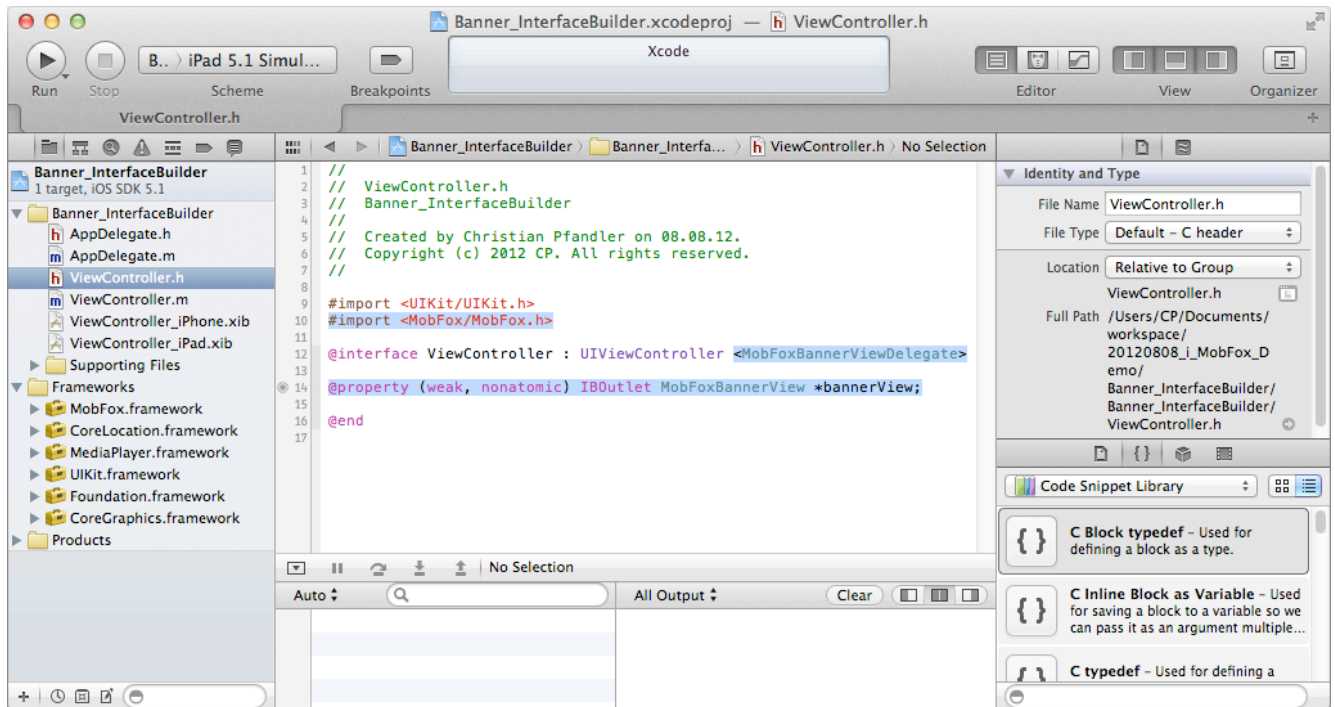


Jump to ViewController.h and add the lines (copy/paste from "Source_Code" folder).

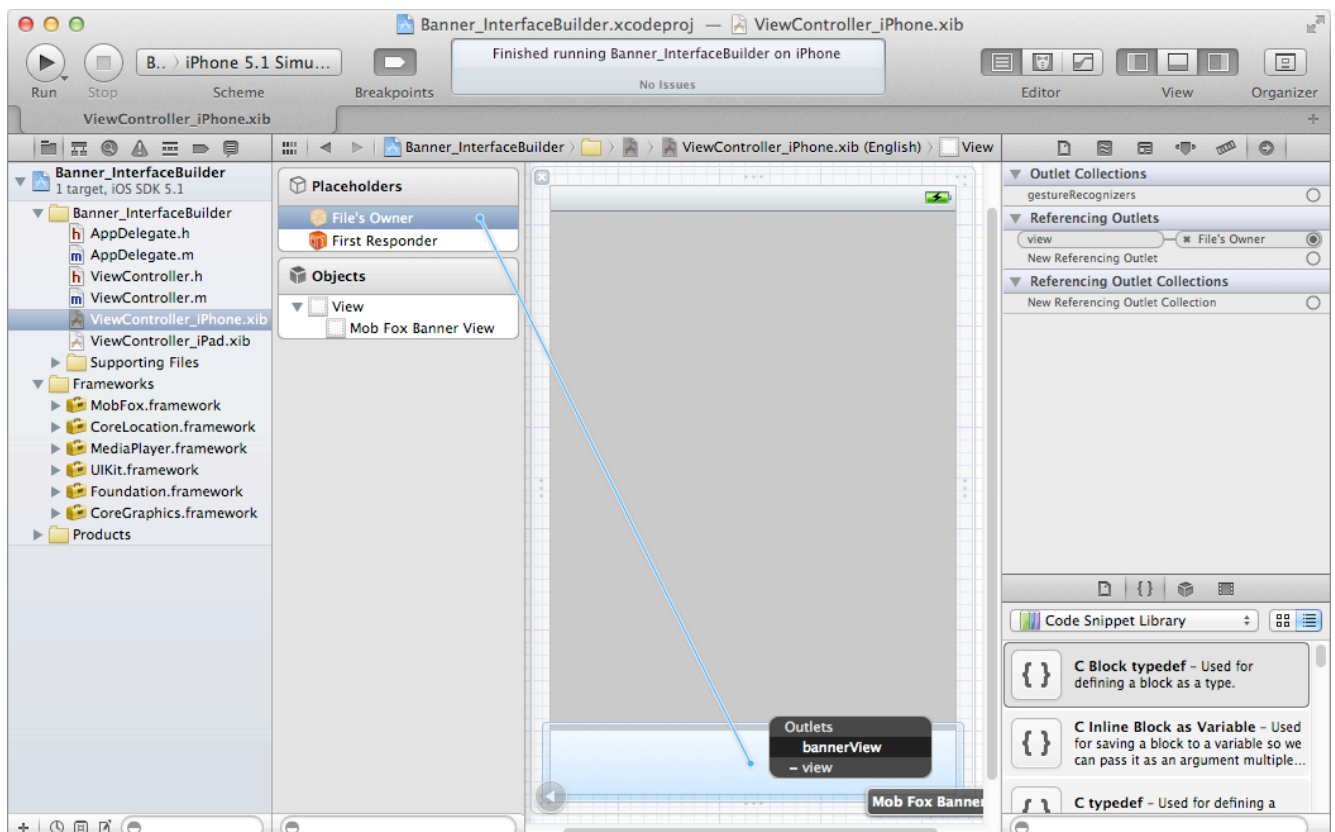
Note the property for the bannerView to reference it from code. Don't forget to @synthesize in .m file.

```
#import <UIKit/UIKit.h>
#import <MobFox/MobFox.h>

@interface ViewController : UIViewController <MobFoxBannerViewDelegate>
@property (weak, nonatomic) IBOutlet MobFoxBannerView *bannerView;
@end
```



In the "ViewController_iPhone.xib" assign by ctrl-drag the "File's Owner" to the BannerView and select "bannerView". Do the same for the "ViewController_iPad.xib".



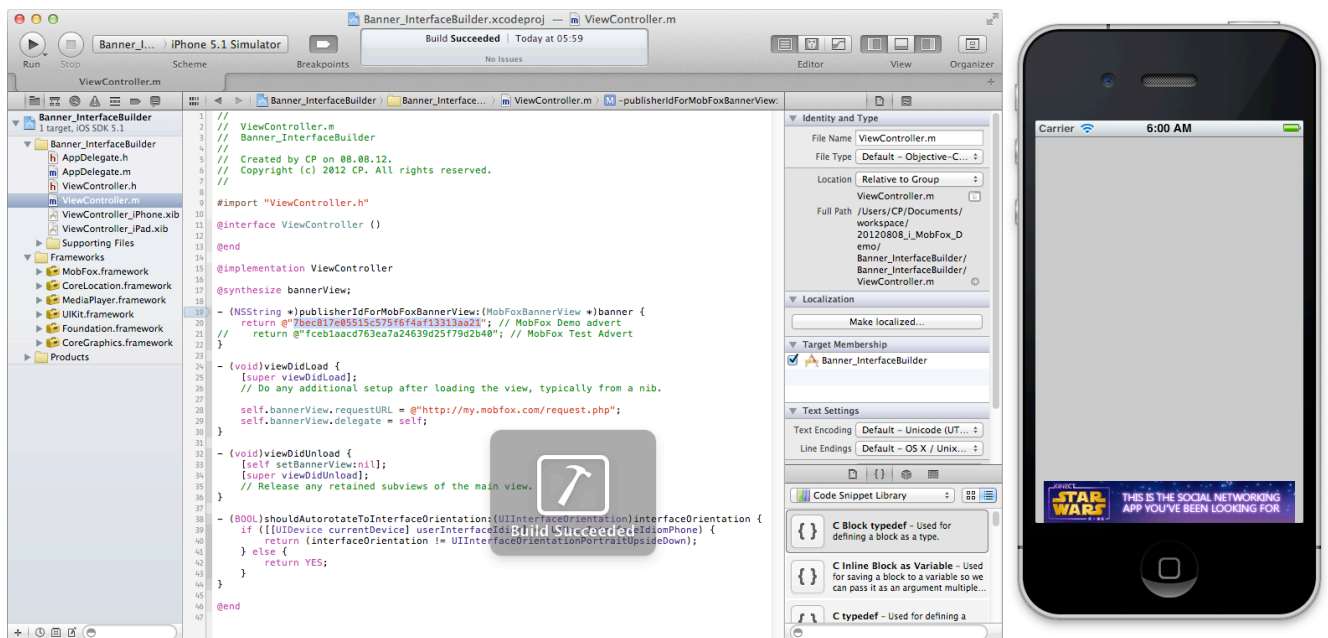
In ViewController.m add the following lines (copy/paste from "Source_Code" folder).

```
- (NSString *)publisherIdForMobFoxBannerView:(MobFoxBannerView *)banner {  
    return @"ENTER_PUBLISHER_ID_HERE"; // Enter Publisher ID here  
}
```

The final step is to set the RequestURL where the BannerView should get the Ads and set the bannerView delegate to self. This will initiate an ad request.

```
- (void)viewDidLoad {  
    [super viewDidLoad];  
    // Do any additional setup after loading the view, typically from a nib.  
  
    self.bannerView.requestURL = @"http://my.mobfox.com/request.php";  
    self.bannerView.delegate = self;  
}
```

Build with your own Publisher ID and compare it with the provided one in the "Source_Code" folder.



Please find in the "Source_Code" folder two additional Demos including Banner Ads:

MobFoxDemo_Banner

- The quickest and simplest way to add a Banner Ad in your app by just using code
- Shows several customizations over the Banner_InterfaceBuilder version:
 - The banner view is created and placed below the bottom of the view
 - This allows to animate it into place once an ad is successfully retrieved
 - The refresh animation property is set to [UIViewAnimationTransitionCurlDown](#).
 - The background Color is set to dark gray to match the UI

MobFoxDemo

- The best way to test your PublisherID to request Banner or Video/Interstitial Ads
- A more advanced approach which allows the Banner to be added and removed dynamically
- Use this approach when integrating both Banner and Interstitial Ads into the same view
- Use where it is important to hide banner adverts while showing interstitial Ads

Note: Instead of requesting an Ad using a button use the code under the - (IBAction)requestBannerAdvert

If you don't want to add vAds, our Premium advertising format, please jump to "Step 7".

Step 6: Request and display MobFox:vAds

MobFox:vAd is our new full-screen ad format. vAds are sold exclusively to well known brands and CPM payouts to developers start at \$10 CPM. To implement vAds please follow the steps below:

Create a New Project called "MobFoxDemo_VideoInterstitial", follow Step 3, Step 4 and these steps touched on here to reach the provided state you find in the provided "Source_Code" folder.

Ad these lines to ViewController.h (copy/paste from "Source_Code" folder). Note the property for the videoInterstitialViewController to reference it from code. Don't forget to `@synthesize` in .m file.

```
#import <UIKit/UIKit.h>
#import <MobFox/MobFox.h>

@interface ViewController : UIViewController <MobFoxVideoInterstitialViewControllerDelegate>
@property (nonatomic, strong) MobFoxVideoInterstitialViewController
    *videoInterstitialViewController;

- (IBAction)requestInterstitialAdvert:(id)sender;
@end
```

Ad these lines to ViewController.m to create the Interstitial/Video Advert view Controller and add the view(copy/paste from "Source_Code" folder).

```
- (void)viewDidLoad {
    [super viewDidLoad];

    // Create, add Interstitial/Video Ad View Controller and add view to view hierarchy
    self.videoInterstitialViewController = [[MobFoxVideoInterstitialViewController alloc] init];
    // Assign delegate
    self.videoInterstitialViewController.delegate = self;
    // Defaults to NO. Set to YES to get locationAware Adverts
    self.videoInterstitialViewController.locationAwareAdverts = YES;
    // Add view. Note when it is created is transparent, with alpha = 0.0 and hidden
    // Only when an ad is being presented it become visible
    [self.view addSubview:self.videoInterstitialViewController.view];
}
```

Continue to add an IBAction to the ViewController.h file (copy/paste from "Source_Code" folder)

```
- (IBAction)requestInterstitialAdvert:(id)sender {
    if(self.videoInterstitialViewController) {
        self.videoInterstitialViewController.requestURL = @"http://my.mobfox.com/vrequest.php";
        [self.videoInterstitialViewController requestAd];
    }
}
```

Jump to ViewController.xib and add a Button attached to the IBAction. Instead of requesting an Ad using a button use the code under the – (IBAction)requestInterstitialAdvert

Continue to modify ViewController.m by adding the following Video/Interstitial delegate methods (copy/paste from "Source_Code" folder).

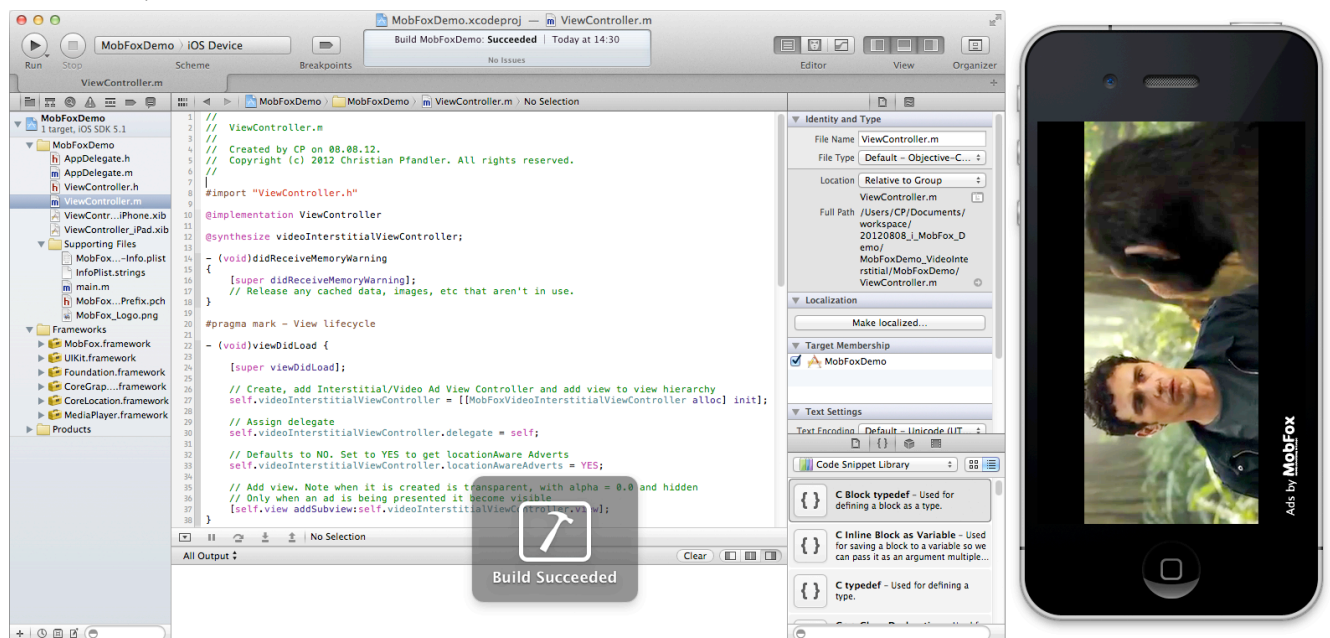
```
#pragma mark MobFox Interstitial Delegate Methods

- (NSString *)publisherIdForMobFoxVideoInterstitialView:
    (MobFoxVideoInterstitialViewController *)videoInterstitial {
    return @"ENTER_PUBLISHER_ID_HERE"; // Enter Publisher ID here
}

- (void)mobfoxVideoInterstitialViewDidLoadMobFoxAd:(MobFoxVideoInterstitialViewController *)
    videoInterstitial advertTypeLoaded:(MobFoxAdType)advertType {
    NSLog(@"MobFox Interstitial: did load ad");
    // Means an advert has been retrieved and configured.
    // Display the ad using the presentAd method and ensure you pass back the advertType
    [videoInterstitial presentAd:advertType];
}

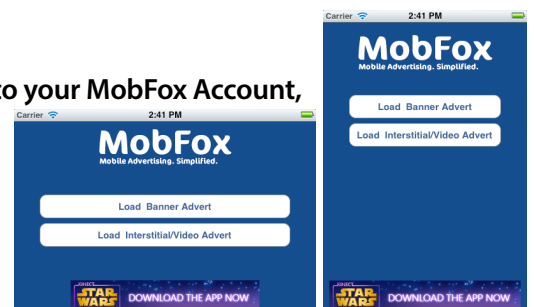
- (void)mobfoxVideoInterstitialView:(MobFoxVideoInterstitialViewController *)
    banner didFailToReceiveAdWithError:(NSError *)error {
    NSLog(@"MobFox Interstitial: did fail to load ad: %@", [error localizedDescription]);
}
```

Build with your own Publisher ID and compare it with the provided one in the "Source_Code" folder.



Step 7: Finished!!!

You will see test ads. Once ready to receive live ads, please log into your MobFox Account, navigate to „Sites & Apps“, and click on „Request Activation“. We will review and activate your app for live ads within 24 hours. Note: Please always clear memory before showing Ads and restore your apps orientation after an Ad is shown. If you would like to test your SDK integration with one of our Test Publisher IDs, please click [here](#)



iPhone 5, iOS 6 and Xcode 4.5 compatibility and new Properties

MobFox SDK now works with the new iPhone 5 in "letterbox" and takes full advantage of the new 4" display (if you include a Default-568@2x.png launch image).

MobFox SDK 4.1.6 has been compiled with Xcode 4.5 and includes armv7 and armv7s (iPhone 5) versions for this devices and above: iPhone 3GS, iPad 2 and iPod Touch (Sept 2010). Minimum Deployment Target is now iOS 4.3. Please note that Xcode 4.5 no longer includes a iOS 4.3 Simulator so testing should be done on an actual iOS 4.3 device.

On iOS 6.X the SDK also uses the new identifierForAdvertising instead of OpenUDID or MAC hashing.

BannerView - allowDelegateAssignmentToRequestAd:

Previous assigning the delegate caused a requestAd. Now gain finer control by setting the property allowDelegateAssignmentToRequestAd to NO, assign the delegate and use the requestAd method to load an advert when ready. Use the previous by setting it to YES.

BannerView - mobfoxBannerViewDidLoadRefreshedAd:

This is called each time when a Banner Ad is refreshed

BannerView - requestAd:

This is used to manually request an Ad

- Please hide Ad banner if no ad is retrieved (displaying an empty area is against Apple guidelines)
- Ads may be smaller than the actual view. Set banner background color to match your UI or "clearColor"
- To implement both Banner & vAds refer to the provided MobFoxDemo Source Code

Explanation of Methods for Banner Ads

Mandatory! You must implement and set the Publisher ID

```
- (NSString *)publisherIdForMobFoxBannerView:(MobFoxBannerView *)banner;
```

Optional. Please use this methods for a smoother integration and result

```
// Called if an Ad has been successfully retrieved and displayed the first time.
// Not called when an adView receive a "refreshed" Ad
- (void)mobfoxBannerViewDidLoadMobFoxAd:(MobFoxBannerView *)banner;

// Called if an existing Ad view receives a "refreshed" Ad
- (void)mobfoxBannerViewDidLoadRefreshedAd:(MobFoxBannerView *)banner;

// Called if no banner is available or there is an error
- (void)mobfoxBannerView:(MobFoxBannerView *)banner
    didFailToReceiveAdWithError:(NSError *)error;

// Called when user taps on a banner
- (BOOL)mobfoxBannerViewActionShouldBegin:(MobFoxBannerView *)banner
    willLeaveApplication:(BOOL)willLeave;

// Called when the modal web view will be displayed
- (void)mobfoxBannerViewActionWillPresent:(MobFoxBannerView *)banner;

// Called when the modal web view is about to be cancelled
// Restart any foreground activities paused as part of mobfoxBannerViewActionWillPresent:
- (void)mobfoxBannerViewActionWillFinish:(MobFoxBannerView *)banner;

// Called when the modal web view is cancelled and the user is returning to the app
- (void)mobfoxBannerViewActionDidFinish:(MobFoxBannerView *)banner;

// Called when a user tap results in Application Switching
- (void)mobfoxBannerViewActionWillLeaveApplication:(MobFoxBannerView *)banner;
```

Explanation of Methods for Video/Interstitial Ads

Mandatory! You must implement and set the Publisher ID

```
- (NSString *) publisherIdForMobFoxVideoInterstitialView:
    (MobFoxVideoInterstitialViewController *) videoInterstitial;
```

Optional. Please use this methods for a smoother integration and result

```
// Called if a Video/Interstitial Ad has been successfully retrieved and
// is ready to display via - (void)presentAd:(MobFoxAdType)advertType:
- (void)mobfoxVideoInterstitialViewDidLoadMobFoxAd:(MobFoxVideoInterstitialViewController *)
    videoInterstitial advertTypeLoaded:(MobFoxAdType)advertType;

// Called if no Video/Interstitial is available or there is an error
- (void)mobfoxVideoInterstitialView:(MobFoxVideoInterstitialViewController *)
    banner didFailToReceiveAdWithError:(NSError *)error;

// Sent immediately before Video/Interstitial is shown to the user. At this point
// pause any animations, timers or other activities that assume user interaction and
// save app state, much like on UIApplicationDidEnterBackgroundNotification.
// Remember that the user may press Home or touch links to other apps like App Store or
// iTunes within the interstitial, leaving your app.
- (void)mobfoxVideoInterstitialViewActionWillPresentScreen:
    (MobFoxVideoInterstitialViewController *)videoInterstitial;

// Sent immediately before interstitial leaves the screen. At this point
// restart any foreground activities paused as part of interstitialWillPresentScreen:
- (void)mobfoxVideoInterstitialViewWillDismissScreen:(MobFoxVideoInterstitialViewController *)
    videoInterstitial;

// Sent when the user has dismissed interstitial and it has left the screen.
- (void)mobfoxVideoInterstitialViewDidDismissScreen:(MobFoxVideoInterstitialViewController *)
    videoInterstitial;

// Called when a user tap results in Application Switching
- (void)mobfoxVideoInterstitialViewActionWillLeaveApplication:
    (MobFoxVideoInterstitialViewController *)videoInterstitial;
```