



IOS Unity Plugin v3.0  
**Publishers**



# iOS Unity Plugin v3.0

Publishers



## Introduction

This document outlines the iOS Unity Plugin v3.0 that allows displaying mobile ads, mobile Editorials, and interstitials (**simple** and **HTML5** versions) inside iOS applications developed by **Unity**.

The current version of the iOS SDK is compatible with iOS 4.3 and above.

Today users expect the same high quality experience regardless of device and if they don't get it, they blame your brand. This is why we use **Akamai number 1 CDN world wide** to display your ads.

## 1. Download iOS SDK

Download the Ad4game iOS SDK publishers for Unity. Decompress the zip file and extract the files to your development computer. The iOS SDK for Unity is provided as a single framework file (**A4GPubUnity.framework**), making it easy to include in your iOS project.

## 2. Include the A4GPubUnity.framework

Build your Unity project for iOS, then open the generated xcode project.

To include the A4GPubUnity.framework to your xcode project, you need to select your target project, and then go to **Build Phases → Link Binary With Libraries**, then click on the **+** sign → **Add Other**, and select the A4GPubUnity.framework saved on your computer.

## 3. Include the AdSupport.framework

You need to include the AdSupport.framework to your project target to enable the Advertiser Identifier functionality. Go to **Build Phases → Link Binary With Libraries**, then click on the **+** sign to add the AdSupport.framework.

Then, change the value of AdSupport.framework from Required to **Optional**.

## 4. Include the SystemConfiguration.framework

You need to include the SystemConfiguration.framework to your project target to enable the detection of the connection type. Go to **Build Phases → Link Binary With Libraries**, then click on the **+** sign to add the SystemConfiguration.framework.

## 5. Display banners

To display ads on your iOS project, you need to follow the steps bellow:

- You need to add this line into your **appController.h** file

`@property (strong , nonatomic) UIWindow *window;`

### In your appController.mm file

- You need to import the A4GPubUnity.framework, after the last #import.  
`#import <A4GPubUnity/A4GPubUnityViewController.h>`
- Go to **startUnity** method and do these steps :
- Instantiate the window declared inside appController.h file.
- Instantiate an object of A4GPubUnityViewController.
- Add a imageView and Button with the same size of the ad to display (the button need to be above the imageView and need to be transparent).
- call the different method to display ads. The methods are:
  - \* **setZoneID** method allows you to set the zoneID.
  - \* **getRefresh** method to refresh the display of banners.
  - \* **loadAds** method to get banners from ad4game servers.

Your code need to be similar to this:

```
- (void) startUnity:(UIApplication*)application
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    A4GPubUnityViewController *a4g = [[A4GPubUnityViewController alloc] init];

    UIImageView *zoneImage = [[UIImageView alloc] initWithFrame:CGRectMake(20, 20, 300, 250)];
    UIButton *button = [UIButton buttonWithType:UIButtonTypeCustom];
    button.frame = CGRectMake(20, 20, 300, 250);
    [button setBackgroundColor:[UIColor clearColor]];
    [button addTarget:a4g action:@selector(loadClick)
        forControlEvents:UIControlEventTouchUpInside];
}
```

```
[a4g setZoneID:@"16209"];  
[a4g getRefresh];  
[a4g loadAds:zoneImage];  
[self.window addSubview:zoneImage];  
[self.window addSubview:button];  
[self.window makeKeyAndVisible];
```

## 6. Display Editorials

To display mobile Editorials on your iOS project developed by Unity, you need to follow the steps bellow:

- You need to add this line into your **appController.h** file  
`@property (strong , nonatomic) UIWindow *window;`

### In your appController.mm file

- You need to import the A4GPubUnity.framework, after the last #import.  
`#import <A4GPubUnity/A4GPubUnityViewController.h>`  
`#import <A4GPubUnity/A4GEdito.h>`
- Go to **startUnity** method and do these steps :
- Instantiate the window declared inside appController.h file.
- Instantiate an object of A4GEdito.
- You need to add a **UIButton** if you want to launch the editorials when the user clicks on that button.
- call the different method to display mobile Editorials. The methods are:
  - \* **setZoneID** method allows you to set the zoneID.
  - \* **showA4GEditos** method, to display directly the Editorials when the app started.
  - \* **showA4GEditosWithStartTime** method, to display the Editorials after a certain time.
  - \* **showA4GEditosWithStartAndEndTime** method, to display the Editorials after a certain time, and close them automatically after a certain time.

Your code need to be similar to this:

```
- (void) startUnity:(UIApplication*)application
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    A4GEdito *a4gEdito = [[A4GEdito alloc] init];

    [a4gEdito setZoneID:@"32593"]; // set ZoneID

    UIButton *button = [UIButton buttonWithType:UIButtonTypeRoundedRect];
    button.frame = CGRectMake(20, 20, 50, 50); // Create a button where x=20, y=20, width=50, height=50;

    [button setTitle:@"Editorials" forState:UIControlStateNormal];

    [button addTarget:a4gEdito action:@selector(showA4GEditos)
        forControlEvents:UIControlEventTouchUpInside];

    [self.window addSubview:button];
    [self.window makeKeyAndVisible];

    [a4gEdito showA4GEditos]; // Show Editorials when the app started.

    [a4gEdito showA4GEditosWithStartTime : 10]; // Show Editorials after 10 seconds.

    [a4gEdito showA4GEditosWithStartAndEndTime : 10 withArg2 : 20]; // Show Editorials after 10 seconds and
    close it automatically after 20 seconds.
```

## 8. Display Interstitials

To display mobile interstitials on your iOS project developed by Unity, you need to follow the steps bellow:

- You need to add this line into your **appController.h** file

```
@property (strong , nonatomic) UIWindow *window;
```

### In your appController.mm file

- You need to import the A4GPubUnity.framework, after the last #import.

```
#import <A4GPubUnity/A4GPubUnityViewController.h>
```

```
#import <A4GPubUnity/A4GInterstitials.h>
```

- Go to **startUnity** method and do these steps :
- Instantiate the window declared inside appController.h file.
- Instantiate an object of A4GInterstitials.

- call the different method to display mobile interstitials. The methods are:
  - \* **setZoneID** method allows you to set the zoneID.
  - \* **showA4GInterstitials** method, to display directly the interstitials when the app started.
  - \* **showA4GInterstitialsWithStartTime** method, to display the interstitials after a certain time.
  - \* **showA4GInterstitialsWithStartAndEndTime** method, to display the interstitials after a certain time, and close them automatically after a certain time.

Your code need to be similar to this:

```
- (void) startUnity:(UIApplication*)application
{
    self.window = [[UIWindow alloc] initWithFrame:[[UIScreen mainScreen] bounds]];
    A4GInterstitials *a4gInter = [[A4GInterstitials alloc] init];

    [a4gInter setZoneID:@"32593"]; // set ZoneID

    [a4gInter showA4GInterstitials]; // Show Interstitials when the app started.

    [a4gInter showA4GInterstitialsWithStartTime : 10]; // Show interstitials after 10 seconds.

    [a4gInter showA4GInterstitialsWithStartAndEndTime : 10 withArg2 : 20]; // Show Interstitials after 10 seconds
    and close them automatically after 20 seconds.

    [self.window makeKeyAndVisible];
}
```