

# iOS mOcean SDK Developer Guide

---

For iOS Version 2.9.0

# Table of Contents

---

System Requirements .....	5
Feature list .....	6
Installing the Ad SDK .....	7
Add the Latest version of SDK library project to your app.....	7
Add Frameworks and libraries to your project .....	8
Set Target Dependencies .....	9
Install Documentation .....	10
Simple AdView integration .....	11
Simple AdInterstitialView integration.....	12
Customize AdView .....	13
Customize view appearance. ....	13
Customize content. ....	13
Customize AdInterstitialView .....	14
Customize view appearance. ....	14
Customize content. ....	14
Troubleshooting.....	15
Duplicate symbol compile error.....	15
AdView Class Reference.....	16
Overview .....	16
Tasks .....	16
Initializing an AdView Object.....	16
Configuring the AdView .....	16
Loading the AdView Content .....	17
Filtering the AdView Content.....	17
Install Notification.....	17
Debug the AdView.....	17
Setting the Delegate .....	18
Properties .....	18
adServerUrl .....	18

additionalParameters .....	18
advertiserId .....	18
animateMode .....	19
area.....	19
carrier.....	19
city .....	20
closeButton.....	20
contentAlignment.....	20
contentSize.....	20
country .....	21
defaultImage .....	21
delegate.....	21
groupCode.....	22
internalOpenMode.....	22
isLoading.....	23
keywords.....	23
latitude .....	23
logMode .....	23
longitude.....	24
maxSize .....	24
metro.....	24
minSize .....	25
premium.....	25
region.....	25
site .....	26
testMode .....	26
textColor .....	26
track .....	27
type.....	27
updateTimeInterval.....	27
zip .....	28
zone.....	28
<b>Instance Methods .....</b>	<b>28</b>

initWithFrame:site:zone: .....	28
update.....	29
AdInterstitialView Class Reference .....	30
Overview .....	30
Tasks .....	30
Customizing AdInterstitialView Closing .....	30
Setting the Delegate .....	30
Properties .....	30
autocloseInterstitialTime.....	30
closeButton.....	31
delegate.....	31
showCloseButtonTime .....	31
AdViewDelegate Protocol Reference.....	33
Overview .....	33
Tasks .....	33
Instance Methods .....	33
adDidEndFullScreen:.....	33
adShouldOpen:withUrl:.....	34
adWillStartFullScreen:.....	34
didClosedAd:usageTimeInterval:.....	35
didFailToReceiveAd:withError:.....	35
didReceiveAd: .....	36
didReceiveThirdPartyRequest:content: .....	36
ormmaProcess:event:parameters: .....	37
willReceiveAd:.....	37

## System Requirements

---

- Intel based Mac
- iOS 4.0 - 5.0 SDK
- XCode 4

## Feature list

---

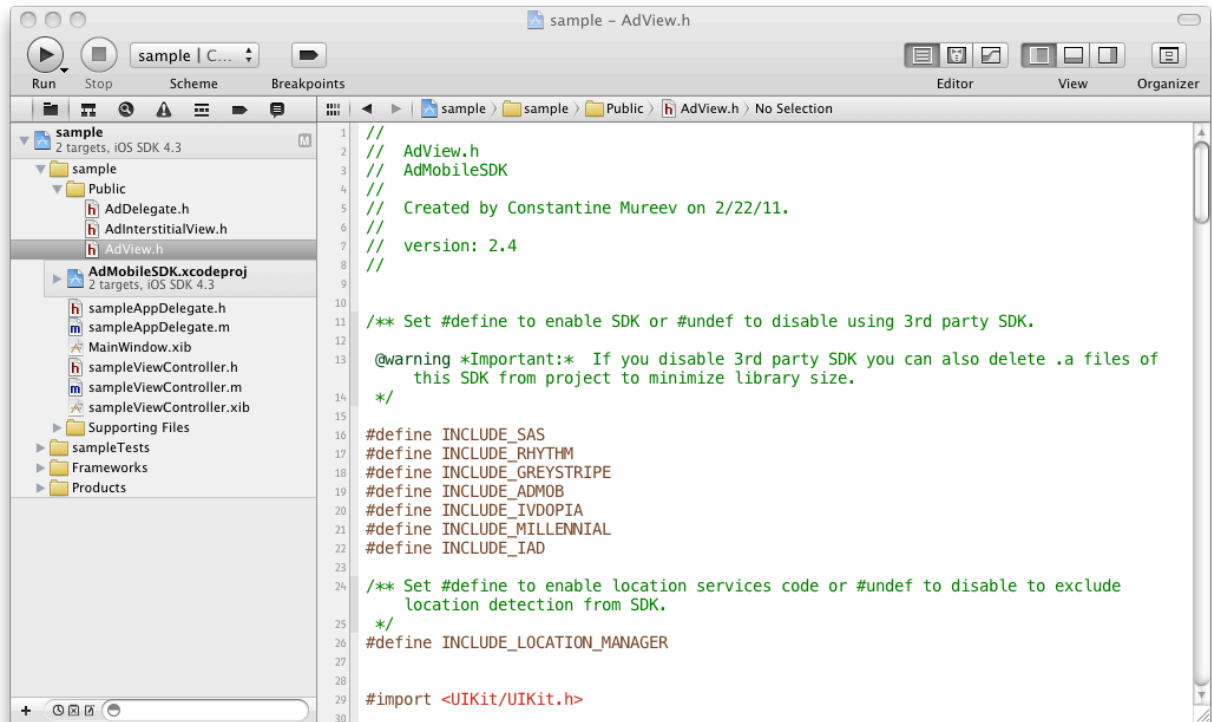
- **HTML / JS ads**  
SDK support displaying web ads using UIWebView component.
- **Video ads**  
SDK support playing video ads in clickable MPMoviePlayerController component.
- **Location auto detect**  
SDK could automatically detecting user location
- **User-Agent auto detect**  
SDK automatically detects device User-Agent
- **Ad visibility tracking**  
SDK automatically detects ads visibility for controlling updates and video playback
- **Logging**  
SDK support logging.
- **Content caching**  
All content cached
- **Internal browser**  
SDK contains built-in browser for displaying ads in application
- **ORMMA/MRAID level 1,2,3**  
SDK support all ORMMA levels. [See more.](#)

# Installing the Ad SDK

## Add the Latest version of SDK library project to your app

Add the AdView.h, AdInterstitialView.h, AdDelegate.h header files from Public folder in SDK sources and library project.

Select Classes, then right-click and select Add Files to “Project Name”...

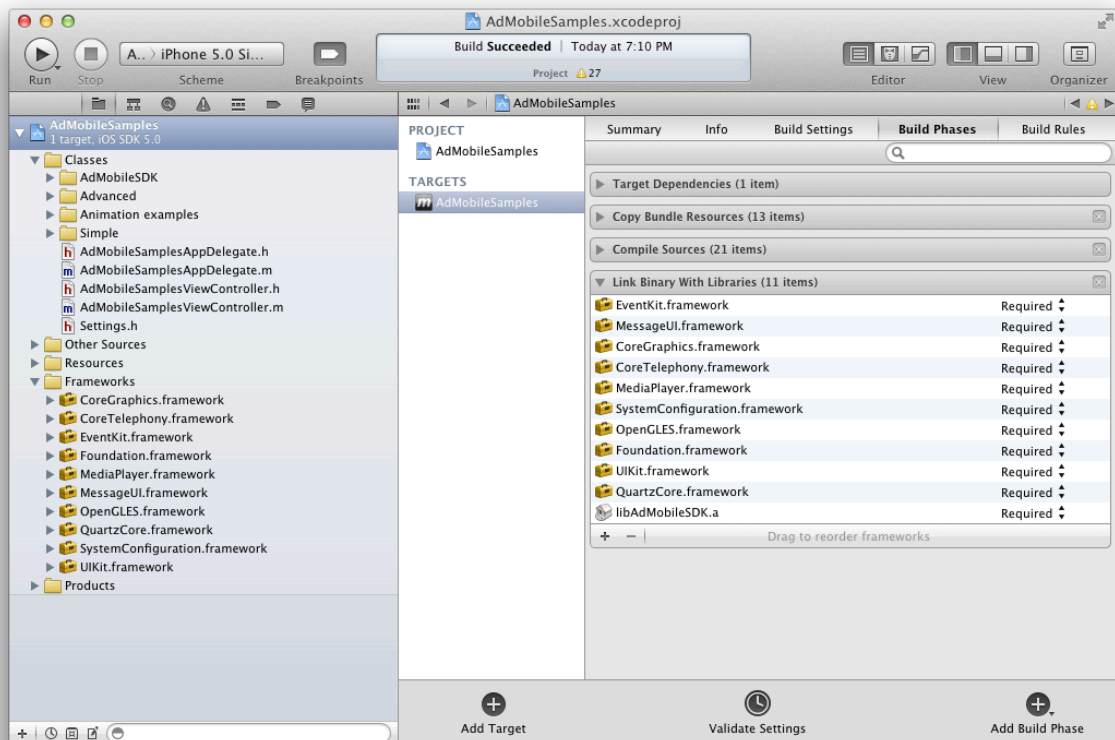


## Add Frameworks and libraries to your project

Add following frameworks and libraries to you project:

- **libAdMobileSDK.a**
- CoreGraphics
- CoreTelephony
- CoreMotion
- CoreTelephony
- EventKit
- MediaPlayer
- MessageUI
- QuartzCore
- SystemConfiguration

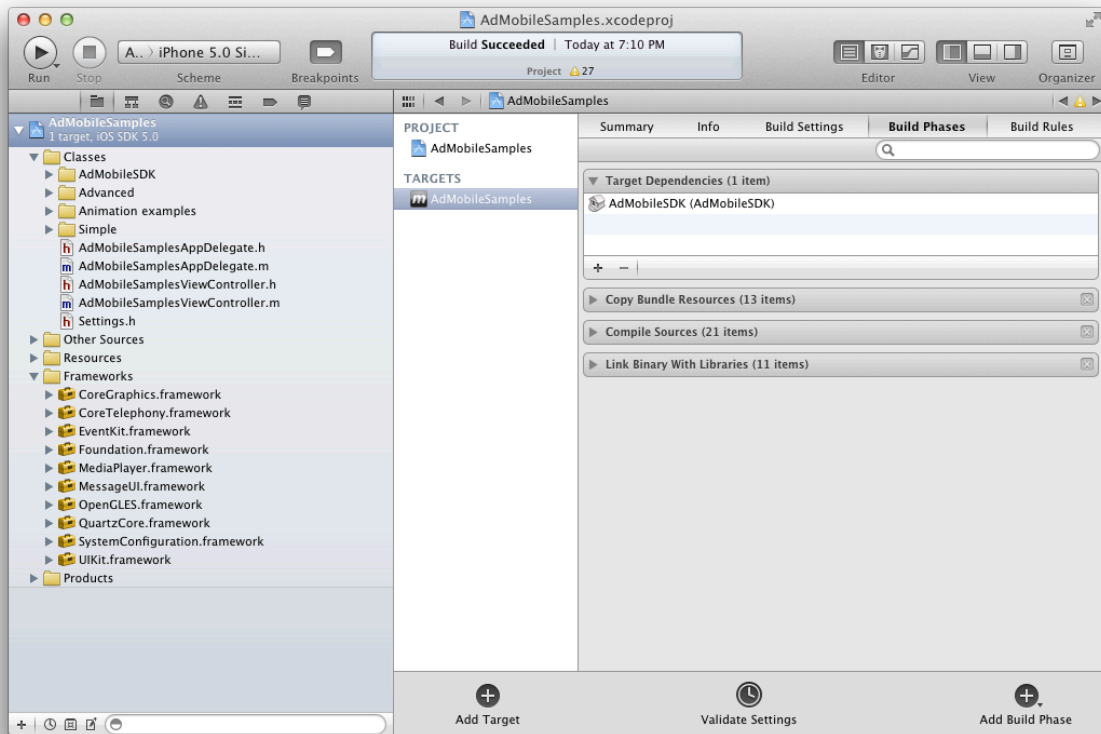
Select Project, Target, Build Phases, then add frameworks in “Link Binary With Libraries” section





## Set Target Dependencies

Select Project, Target, Build Phases, then add AdMobileSDK target in “Target Dependencies” section.



## **Install Documentation**

SDK includes XCode documentation in “Documentation\Reference” folder.

Run installdocset.sh script in Terminal.app to copy documentation to XCode or you can use web based version of documentation in html folder

## Simple AdView integration

---

To add AdView into your application just import AdView.h

```
#import "AdView.h"
```

And put initialize code in (void)viewDidLoad method

```
- (void)viewDidLoad {  
    AdView *adView = [[AdView alloc] initWithFrame:CGRectMake(0, 0, 320, 50)  
site:8061 zone:20249];  
    [self.view addSubview:adView];  
    [adView release];  
  
    [super viewDidLoad];  
}
```

See Also:

- For more code samples see AdMobileSamples application.

## Simple AdInterstitialView integration

---

To add AdInterstitialView into your application just import AdInterstitialView.h

```
#import "AdInterstitialView.h"
```

And put initialize code in (void)viewDidLoad method

```
- (void)viewDidLoad {  
    AdInterstitialView *adView = [[AdInterstitialView alloc]  
initWithFrame:CGRectMake(0, 20, 320, 460) site:8061 zone:16112];  
    adView.minSize = CGSizeMake(320, 460);  
  
    [self.navigationController.view addSubview:adView];  
    [adView release];  
  
    [super viewDidLoad];  
}
```

For more customization you can set custom close button, delay for showing button or auto close interstitial time.

See Also:

- For more code samples see AdMobileSamples application.

## Customize AdView

---

### Customize view appearance.

1. By default in AdView enabled animation and internalOpenMode.

To disable default animation set animateMode property to NO. **Important:** you need to disable animation if you specify your own.

2. Also you can set defaultImage property to show image while ad loading.

3. Set contentAlignment property to enable auto alignment content in center vertically and horizontally.

4. Also you can apply default UIView customization such as animation, background color, support different orientations and so on.

### Customize content.

1. By default in SDK disabled location auto detect. If you want to filter content by user location set latitude and longitude property of AdView.

Also you could enable auto detect location by replacing “#undef INCLUDE\_LOCATION\_MANAGER” with «#define INCLUDE\_LOCATION\_MANAGER” in AdView.h and adding CoreLocation framework.

**Important:** However do not use Core Location just for advertising, make sure it is used for more beneficial reasons as well. It is both a good idea and part of Apple's guidelines.

2. To add additionally filtration you can set keywords, city, carrier, metro, etc.

## Customize AdInterstitialView

---

### Customize view appearance.

1. By default for interstitial created rounded white close button. You can customize button by setting your own.
2. To enable auto close interstitial timer set time interval `autoCloseInterstitial` property, in seconds.
3. You can delay close button displaying by setting `showCloseButtonTime` property.

### Customize content.

See customize section of `AdView`

# Troubleshooting

---

## Duplicate symbol compile error

If you got compile error like this:

```
/Users/william/Library/Developer/Xcode/DerivedData/theScoreMobile-  
testApp/Build/Products/Debug-iphoneos/libAdMobileSDK.a(SBJSON.o)"
```

It means that your project and SDK contains files with same name. In most cases this error due to duplication some framework. In this case – SBJSON.

To solve this problem you need to remove .m files for all files than cause this error. In this case – keep SBJSON.h and remove SBJSON.m and so on...

# AdView Class Reference

---

Inherits from    [UIView](#)

---

Declared in    [AdView.h](#)

---

## Overview

---

You use the [AdView](#) class to embed advertisement content in your application. To do so, you simply create an [AdView](#) object and add it to a [UIView](#). An instance of [AdView](#) (or simply, an ad) is a means for displaying advertisement information from ad publisher [site](#). To choose ad publisher [site](#) set parameters in the ad initialization method [initWithFrame:site:zone:](#) or use the properties [site](#) and [zone](#). Use the [adServerUrl](#) property to choose ad publisher server.

Ad handles the rendering of any content in its area: HTML, Video, Gif animation or JavaScript. Ad also handles any interactions with that content. So you can link from the banner to ad publisher [site](#). Use the [internalOpenMode](#) property to set open mode for ad publisher [site](#). To control all interactions implement [AdViewDelegate](#) protocol.

Ad always tries to load the content after creation. The time interval between load requests is managed using the property [updateTimeInterval](#). Also you can [update](#) ad content immediately using the [update](#) method. Use the [isLoading](#) property to find out if ad is in the process of loading.

To configure ad visual appearance use the properties [defaultImage](#), [textColor](#) or [contentAlignment](#). To manage the ad animation use the property [animateMode](#).

Advanced ad customization is supported. So you can filter the ad content using the [premium](#) property. Use the properties [minSize](#) and [maxSize](#) to configure ad content size in server response. Also you can set the search parameters or any other parameters using the properties [keywords](#) and [additionalParameters](#) accordingly.

To debug ad behavior use the properties [testMode](#) and [logMode](#).

Set the [delegate](#) property to an object conforming to the [AdViewDelegate](#) protocol if you want to listen the processing of ad content.

## Tasks

---

### Initializing an AdView Object

- [– initWithFrame:site:zone:](#)

### Configuring the AdView

- [site](#) property



- [zone](#) *property*
- [adServerUrl](#) *property*
- [defaultImage](#) *property*
- [animateMode](#) *property*
- [closeButton](#) *property*
- [contentAlignment](#) *property*
- [internalOpenMode](#) *property*
- [track](#) *property*
- [textColor](#) *property*

## Loading the AdView Content

- [isLoading](#) *property*
- [updateTimeInterval](#) *property*
- [– update](#)
- [contentSize](#) *property*

## Filtering the AdView Content

- [premium](#) *property*
- [type](#) *property*
- [keywords](#) *property*
- [minSize](#) *property*
- [maxSize](#) *property*
- [additionalParameters](#) *property*
- [latitude](#) *property*
- [longitude](#) *property*
- [country](#) *property*
- [region](#) *property*
- [city](#) *property*
- [area](#) *property*
- [metro](#) *property*
- [zip](#) *property*
- [carrier](#) *property*

## Install Notification

- [advertiserId](#) *property*
- [groupCode](#) *property*

## Debug the AdView

- [testMode](#) *property*
- [logMode](#) *property*

## Setting the Delegate

- [delegate](#) *property*

## Properties

---

### adServerUrl

Publisher server url.

```
@property (retain) NSString *adServerUrl
```

#### **Discussion**

Publisher server url.

The default value is <http://ads.AdMobile.mobi/ad> .

#### **Declared In**

AdView.h

### additionalParameters

Custom request parameters.

```
@property (retain) NSDictionary *additionalParameters
```

#### **Discussion**

Custom request parameters.

Use this property to add custom request parameters.

The default value is nil.

**Note:** All keys and objects is need to be kind of NSString Class. For example:

```
[NSDictionary dictionaryWithObject:@"value" forKey:@"key"]
```

#### **Declared In**

AdView.h

### advertiserId

Id of the advertiser for install notification.

```
@property (assign) NSInteger advertiserId
```

#### **Discussion**

Id of the advertiser for install notification.

**Note:** Install notification enabled only if [advertiserId](#) and [groupCode](#) are specified.

### ***Declared In***

AdView.h

## **animateMode**

A Boolean value that determines whether ad animate mode is enabled.

@property BOOL animateMode

### ***Discussion***

A Boolean value that determines whether ad animate mode is enabled.

Setting the value of this property to YES enables ad animate mode and setting it to NO disables this mode.

The default value is YES.

### ***Declared In***

AdView.h

## **area**

Area code of a user. For US only.

@property (retain) NSString \*area

### ***Discussion***

Area code of a user. For US only.

The default value is nil.

### ***Declared In***

AdView.h

## **carrier**

User [carrier](#).

@property (retain) NSString \*carrier

### ***Discussion***

User [carrier](#).

The default value is nil.

### ***Declared In***

AdView.h

## city

City of the device user (with state). For US only.

```
@property (retain) NSString *city
```

### **Discussion**

City of the device user (with state). For US only.

The default value is nil.

### **Declared In**

AdView.h

## closeButton

Close button.

Set this value to customize close button appearance and behaviour.

By default closed button set by SDK with hidden property YES.

**Note:** If you set set UIButton then you need implement close logic too.

**Note:** If you want simply enable default close button set hidden property to NO.

### **Declared In**

AdView.h

## contentAlignment

A Boolean value that determines whether content alignment center vertically and horizontally.

```
@property BOOL contentAlignment
```

### **Discussion**

A Boolean value that determines whether content alignment center vertically and horizontally.

Setting the value of this property to YES enables auto wrapping server response content in HTML table with alignment

The default value is NO.

**Important:** You need to disable animation if you specify your own.

### **Declared In**

AdView.h

## contentSize

Size of the ad content to be shown.

@property (readonly) CGSize contentSize

### **Discussion**

Size of the ad content to be shown.

Use this property to get the actual size of the ad content. Property value updated after ad content downloaded.

**Note:** If size unavailable (Millennial, Greystripe, iVdopia and other 3rd party SDKs) property returns CGRectZero.

### **Declared In**

AdView.h

## country

Country of visitor. It overrides [country](#) detected by IP. It is ISO 3166 to be used for specifying [country](#) code.

@property (retain) NSString \*country

### **Discussion**

Country of visitor. It overrides [country](#) detected by IP. It is ISO 3166 to be used for specifying [country](#) code.

The default value is nil.

### **Declared In**

AdView.h

## defaultImage

Image for unloaded ad state.

@property (retain) UIImage \*defaultImage

### **Discussion**

Image for unloaded ad state.

Settings the value of this property determines ad default image for unloaded state. In this state the content of ad is invisible and ad default image is displayed. Without connection to the internet default image also will be displayed.

The default value is nil.

### **Declared In**

AdView.h

## delegate

The receiver's [delegate](#).

```
@property (assign) id<AdViewDelegate> delegate
```

### Discussion

The receiver's [delegate](#).

The [AdView](#) is sent messages when content is processing. The [delegate](#) must adopt the [AdViewDelegate](#) protocol. The [delegate](#) is not retained.

**Important:** Before releasing an instance of [AdView](#) for which you have set a [delegate](#), you must first set its [delegate](#) property to nil. This can be done, for example, in your dealloc method.

### See Also

- [AdViewDelegate](#) Protocol Reference for the optional methods this [@property delegate](#) may implement.

### Declared In

AdView.h

## groupCode

Group code for install notification.

```
@property (retain) NSString *groupCode
```

### Discussion

Group code for install notification.

**Note:** Install notification enabled only if [advertiserId](#) and [groupCode](#) are specified.

### Declared In

AdView.h

## internalOpenMode

A Boolean value that determines whether ad internal browser is enabled.

```
@property BOOL internalOpenMode
```

### Discussion

A Boolean value that determines whether ad internal browser is enabled.

Setting the value of this property to NO disables internal browser, so after linking from the banner the ad publisher [site](#) will be opened in Safari.

To handle opening/closing internal browser use [AdView delegate](#) or [viewWillAppear/viewWillDisappear](#) methods of [UIViewController](#).

The default value is YES.

### Declared In

AdView.h

## isLoading

A Boolean value that determines whether ad is in the process of loading.

```
@property (readonly) BOOL isLoading
```

### **Discussion**

A Boolean value that determines whether ad is in the process of loading.

### **Declared In**

AdView.h

## keywords

Keywords for search ads.

```
@property (retain) NSString *keywords
```

### **Discussion**

Keywords for search ads.

Use this property to search ads. The values are delimited by commas.

The default value is nil.

### **Declared In**

AdView.h

## latitude

User location [latitude](#) value.

```
@property (retain) NSString *latitude
```

### **Discussion**

User location [latitude](#) value.

Use this property to set [latitude](#). The value @" " will stop coordinates auto-detection and coordinates will not be sent to server. Any other values also will stop coordinates auto-detection but coordinates will be sent to server.

The default value is auto-detected by locationManager and sent to server.

### **Declared In**

AdView.h

## logMode

AdLogMode value that determines log level.

```
@property AdLogMode logMode
```

Copyright © 2011 mOcean Mobile. All Rights Reserved.  
136 Baxter St, New York, NY 10013

## Discussion

AdLogMode value that determines log level.

```
typedef enum {      AdLogModeNone = 0,      AdLogModeErrorsOnly = 1,  
AdLogModeAll = 2,  } AdLogMode;
```

Setting the value of this property to AdLogModeNone disables ads logging.

AdLogModeErrorsOnly – enables logging errors only. AdLogModeAll – enables logging errors and infos.

The default value is AdLogModeErrorsOnly.

## Declared In

AdView.h

## longitude

User location [longitude](#) value.

```
@property (retain) NSString *longitude
```

## Discussion

User location [longitude](#) value.

Use this property to set [longitude](#). The value @"" will stop coordinates auto-detection and coordinates will not be sent to server. Any other values also will stop coordinates auto-detection but coordinates will be sent to server.

The default value is auto-detected by locationManager and sent to server.

## Declared In

AdView.h

## maxSize

Maximal size of the ad content to be shown.

```
@property CGSize maxSize
```

## Discussion

Maximal size of the ad content to be shown.

Use this property to set the maximal size of the ad content and server response will be close to this size.

## Declared In

AdView.h

## metro

Metro code of a user. For US only.



```
@property (retain) NSString *metro
```

### **Discussion**

Metro code of a user. For US only.

The default value is nil.

### **Declared In**

AdView.h

## minSize

Minimal size of the ad content to be shown.

```
@property CGSize minSize
```

### **Discussion**

Minimal size of the ad content to be shown.

Use this property to set the minimal size of the ad content and server response will be close to this size.

### **Declared In**

AdView.h

## premium

Ad [premium](#) filter.

```
@property AdPremium premium
```

### **Discussion**

Ad [premium](#) filter.

```
typedef enum {      AdPremiumNonPremium = 0,      AdPremiumPremium,  
AdPremiumBoth, } AdPremium;
```

Use this property to filter the content of ad by [premium](#) status.

The default value is AdPremiumNonPremium.

### **Declared In**

AdView.h

## region

Region of visitor. ISO 3166-2 is used for United States and Canada and FIBS 10-4 is used for other countries.

```
@property (retain) NSString *region
```

### **Discussion**

Region of visitor. ISO 3166-2 is used for United States and Canada and FIBS 10-4 is used for other countries.

The default value is nil.

### **Declared In**

AdView.h

## **site**

Id of the publisher [site](#).

```
@property (assign) NSInteger site
```

### **Discussion**

Id of the publisher [site](#).

Settings the value of this property determines the id of the publisher [site](#), so switching between publisher sites is possible. The default value is copied from parameter [site](#) of ad initialization method.

### **Declared In**

AdView.h

## **testMode**

A Boolean value that determines whether ads test mode is enabled. Setting the value of this property to YES enables ads test mode and setting it to NO disables ads test mode.

```
@property BOOL testMode
```

### **Discussion**

A Boolean value that determines whether ads test mode is enabled. Setting the value of this property to YES enables ads test mode and setting it to NO disables ads test mode.

The default value is NO.

### **Declared In**

AdView.h

## **textColor**

Color of ad text links.

```
@property (retain) UIColor *textColor
```

### **Discussion**

Color of ad text links.

The default value is nil.

**Note:** Alpha value ignored.

### ***Declared In***

AdView.h

## track

A Boolean value that determines whether ad [track](#) is enabled.

@property BOOL track

### ***Discussion***

A Boolean value that determines whether ad [track](#) is enabled.

If set to YES, the ad server will send a client side impression tracking pixel with each ad, regardless of if the campaign has this property set or not. Impressions will not be counting if this pixel does not render on the device.

The default value is NO.

### ***Declared In***

AdView.h

## type

Ad [type](#) filter.

@property AdType type

### ***Discussion***

Ad [type](#) filter.

```
typedef enum {      AdTypeTextOnly = 1,      AdTypeImagesOnly = 2,
AdTypeImagesAndText = 3,      AdTypeRichmedia = 4,
AdTypeRichmediaAndText = 5,      AdTypeRichmediaAndImages = 6,      AdTypeAll
= 7, } AdType;
```

Use this property to filter the content of ad by [type](#).

### ***Declared In***

AdView.h

## updateTimeInterval

Update time interval, in seconds.

@property NSTimeInterval updateTimeInterval

### ***Discussion***

Update time interval, in seconds.

The value of this property determines time interval between ads updating. This interval is counted after finish loading content, so the ad will start updating only after loading is finished and time interval is passed.

Setting value in range from 0 to 5 will apply 5 seconds to prevent too fast ad updates.

Setting to 0 will stop updates. All positive values enable updates.

The default value is 120.

### **Declared In**

AdView.h

## zip

Zip/Postal code of user (note: parameter is all caps). For US only.

```
@property (retain) NSString *zip
```

### **Discussion**

Zip/Postal code of user (note: parameter is all caps). For US only.

The default value is nil.

### **Declared In**

AdView.h

## zone

Id of the publisher [zone](#).

```
@property (assign) NSInteger zone
```

### **Discussion**

Id of the publisher [zone](#).

Settings the value of this property determines the id of the publisher [zone](#), so switching between publisher zones is possible. The default value is copied from parameter [zone](#) of ad initialization method.

### **Declared In**

AdView.h

## Instance Methods

---

### initWithFrame:site:zone:

Initializes and returns an [AdView](#) object having the given frame, [site](#) and [zone](#).

–

```
(id)initWithFrame:(CGRect)frame site:(NSInteger)site zone:(NSInteger)zone
```

## **Parameters**

### *frame*

A rectangle specifying the initial location and size of the ad view in its superview's coordinates.

### *site*

A value that specifies the id of ad publisher [site](#).

### *zone*

A value that specifies the id of ad publisher [zone](#).

## **Return Value**

Returns an initialized [AdView](#) object or nil if the object could not be successfully initialized.

## **Discussion**

Initializes and returns an [AdView](#) object having the given frame, [site](#) and [zone](#).

## **Declared In**

AdView.h

## **update**

Starts to [update](#) the ad content immediately.

– (void)update

## **Discussion**

Starts to [update](#) the ad content immediately.

Call this method if you want [update](#) the ad content immediately (for example, after setting [site](#) and [zone](#) or changing [adServerUrl](#)). If ad is in the process of loading it will be interrupted.

## **Declared In**

AdView.h

# AdInterstitialView Class Reference

---

**Inherits from**    AdView : UIView

---

**Declared in**    AdInterstitialView.h

---

## Overview

---

The AdInterstitialView class is subclassing of AdView with advanced customization parameters. An instance of AdInterstitialView is a means for full-screen displaying ads with specific closing.

## Tasks

---

### Customizing AdInterstitialView Closing

- `showCloseButtonTime` *property*
- `autocloseInterstitialTime` *property*
- `closeButton` *property*

### Setting the Delegate

- `delegate` *property*

## Properties

---

### autocloseInterstitialTime

Auto close interstitial time interval, in seconds.

@property NSTimeInterval autocloseInterstitialTime

#### **Discussion**

Auto close interstitial time interval, in seconds.

Setting to 0 will disable auto closing interstitial.

The default value is 0.

#### **Declared In**

AdInterstitialView.h

## closeButton

Close button.

```
@property (retain) UIButton *closeButton
```

### **Discussion**

Close button.

Set this value to customize close button appearance and behaviour.

By default closed button set by SDK.

**Note:** If you set set UIButton then you need implement close logic too.

### **Declared In**

AdInterstitialView.h

## delegate

The receiver's delegate.

```
@property (assign) id<AdInterstitialViewDelegate> delegate
```

### **Discussion**

The receiver's delegate.

The AdInterstitialView is sent messages when content is processing. The delegate must adopt the AdInterstitialViewDelegate protocol. The delegate is not retained.

**Important:** Before releasing an instance of AdInterstitialView for which you have set a delegate, you must first set its delegate property to nil. This can be done, for example, in your dealloc method.

### **See Also**

- [AdInterstitialViewDelegate Protocol Reference](#) for the optional methods `this@property delegate` may implement.

### **Declared In**

AdInterstitialView.h

## showCloseButtonTime

Show close button delay time interval, in seconds.

```
@property NSTimeInterval showCloseButtonTime
```

### **Discussion**

Show close button delay time interval, in seconds.

Setting to 0 will show close button immediately.

The default value is 0.

***Declared In***

AdInterstitialView.h



# AdViewDelegate Protocol Reference

---

---

**Conforms to**    NSObject

---

**Declared in**    AdDelegate.h

---

## Overview

---

The [AdViewDelegate](#) protocol defines methods that a delegate of a [AdView](#) object can optionally implement to receive notifications from ad.

## Tasks

---

- [– willReceiveAd:](#)
- [– didReceiveAd:](#)
- [– didReceiveThirdPartyRequest:content:](#)
- [– didFailToReceiveAd:withError:](#)
- [– adWillStartFullScreen:](#)
- [– adDidEndFullScreen:](#)
- [– adShouldOpen:withUrl:](#)
- [– didClosedAd:usageTimeInterval:](#)
- [– ormmaProcess:event:parameters:](#)

## Instance Methods

---

### adDidEndFullScreen:

Sent after an ad view finished displaying internal browser.

– (void)adDidEndFullScreen:(id) *sender*

#### **Parameters**

*sender*

The ad view has finished displaying internal browser.

#### **Discussion**

Sent after an ad view finished displaying internal browser.

### **Declared In**

AdDelegate.h

## **adShouldOpen:withUrl:**

Sent before an ad view will start to open URL.

– (BOOL)adShouldOpen:(id)*sender* withUrl:(NSURL \*)*url*

### **Parameters**

*sender*

The ad view that is about to open URL.

*url*

The URL that should be opened in internal or external browser.

### **Return Value**

Returns YES to allow SDK open browser otherwise returns NO.

### **Discussion**

Sent before an ad view will start to open URL.

Implement this method with return NO value if you want to control opening ads by your self.

This method is optional. If you do not implement this method, the SDK accept YES as return value.

**Important:** This method may not calling on 3rd party SDKs (Millennial, Greystripe, iVdopia, ...)

### **Declared In**

AdDelegate.h

## **adWillStartFullScreen:**

Sent before an ad view will start to display internal browser.

– (void)adWillStartFullScreen:(id)*sender*

### **Parameters**

*sender*

The ad view that is about to display internal browser.

### **Discussion**

Sent before an ad view will start to display internal browser.

**Important:** This method called after [adShouldOpen:withUrl:](#) returns YES or not implemented.

**Important:** This method is not called on opening ads in Safari (internalOpenMode set to NO). To handle this behaviour implement UIApplicationDelegate protocol

### **Declared In**

AdDelegate.h

## **didClosedAd:usageTimeInterval:**

Sent after an ad view closed and tracks the usage time of ad interstitial view.

-

```
(void)didClosedAd:(id)sender usageTimeInterval:(NSTimeInterval)usageTimeInterval
```

### **Parameters**

*sender*

The ad view was closed.

*usageTimeInterval*

The usage time interval of ad view.

### **Discussion**

Sent after an ad view closed and tracks the usage time of ad interstitial view.

### **Declared In**

AdDelegate.h

## **didFailToReceiveAd:withError:**

Sent if an ad view failed to load ad content.

```
- (void)didFailToReceiveAd:(id)sender withError:(NSError *)error
```

### **Parameters**

*sender*

The ad view that failed to load ad content.

*error*

The error that occurred during loading.

### **Discussion**

Sent if an ad view failed to load ad content.

The same method can be invoked if the server does not currently advertising.

### **Declared In**

AdDelegate.h

## **didReceiveAd:**

Sent after an ad view finished loading ad content.

– (void)didReceiveAd:(id) *sender*

### **Parameters**

*sender*

The ad view has finished loading.

### **Discussion**

Sent after an ad view finished loading ad content.

### **Declared In**

AdDelegate.h

## **didReceiveThirdPartyRequest:content:**

Sent if SDK received client side third party campaign.

– (void)didReceiveThirdPartyRequest:(id) *sender* content:(NSDictionary \*) *content*

### **Parameters**

*sender*

The ad view that receive 3rd party ad content.

*content*

The dictionary of strings (key/values) for third party. You can start processing this content from @"type" key for determining ads type.

### **Discussion**

Sent if SDK received client side third party campaign.

Generally, this method called if the SDK can't display ads due to reliance on a third party. For example: AdMob, Rhythm or SmartAdServer

**Important:** The SDK will never display ads which trigger this method! They are used to call the third party SDKs in your app.

### **Declared In**

AdDelegate.h

## ormmaProcess:event:parameters:

Sent after an ad process ORMMA command.

```
- (void)ormmaProcess:(id)sender event:(NSString *)event parameters:(NSDictionary *)parameters
```

### **Parameters**

*sender*

The ad view that is about to process ORMMA event.

*event*

The string with name of the event.

*parameters*

The Dictionary with parameters from event.

### **Discussion**

Sent after an ad process ORMMA command.

**Important:** Implement this method only if you want add additional logic for event. By default SDK already implements all methods and your code could conflict with SDK

### **Declared In**

AdDelegate.h

## willReceiveAd:

Sent before an ad view will begin loading ad content.

```
- (void)willReceiveAd:(id)sender
```

### **Parameters**

*sender*

The ad view that is about to load ad content.

### **Discussion**

Sent before an ad view will begin loading ad content.

### **Declared In**

AdDelegate.h