iOS mOcean SDK

Developer Guide

For iOS SDK Version 2.11

Table of Contents

Table of Contents 2

What’s New in 2.11 3

Implementation Changes 3

System Requirements 3

Feature List 4

Installing the Ad SDK 5

MASTAdView Use Case 7

Simple Ad Integration 8

Interstitial Ad Integration 10

MASTAdView Customization 11

MASTAdView Interstitial Ad Customization 12

Ad Content Customization 12

Content Updates 12

Troubleshooting 13

# What’s New in 2.11

* Calling update is now required to start ad loading
* Location detection support reworked
* Replaced usage of UIDevice uniqueIdentifier
* Added support for custom wrapper/injection HTML/JavaScript
* New sample application

# Implementation Changes

* Implementations MUST now call update on instances of MASTAdView in order for it to update ad content. For most implementations this can be done in the viewWillAppear: method of the owning view controller.
* Using the device unique identifier from UIDevice has been deprecated by Apple. To send a unique ID create one within the application and set the udid property on MASTAdView.
* On iOS targets less than 4.2, comment out the first three lines in MASTLocationManager deviceLocationAvailable and MASTLocationManager deviceHeadingAvailable. The authorizationStatus method is only available in iOS version 4.2 and higher.

# System Requirements

* Intel based Mac
* iOS 4.2 or higher
* Xcode 4.2 or higher

# Feature List

* **Rich media**

SDK supports ORMMA level 1.1 and MRAID level 1.0. [See more](http://code.google.com/p/ormma/).

* **HTML/ JS ads**

SDK supports displaying web ads using UIWebView component.

* **Location auto detect**

SDK can automatically detect user location.

* **User-Agent auto detect**

SDK automatically detects device User-Agent.

* **Internal browser**

SDK contains built-in browser for displaying ads in application.

* **Ad visibility tracking**

SDK automatically detects ads visibility for controlling updates.

* **Logging**

SDK supports logging.

# Installing the Ad SDK

## Add the SDK headers and library project

Create a group to place the SDK content. Add the MASTAdView.h and MASTAdDelegate.h header files from the SDK Public folder into the new group. Add the SDK project to the new group. Note: If the SDK project is already open in another Xcode project (nested or directly) the SDK project will be locked and unavailable to open within the project.

## Add frameworks and libraries

Add the following frameworks and libraries to you project:

* Foundation
* UIKit
* CoreGraphics
* CoreLocation
* CoreMotion
* CoreTelephony
* EventKit
* MediaPlayer
* MessageUI
* QuartzCore
* SystemConfiguration
* libAdMobileSDK.a

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

## Set target dependencies

Select Project, Target, Build Phases, then add AdMobileSDK target to the Target Dependencies section.



## Install documentation

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

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

# MASTAdView Use Case

1. A simple banner that is used for the embedding of the existing form and occupies a small area in it.
2. Interstitial is used for the full screen banner display. It’s displayed only once and after closing it becomes inactive.
   1. Full screen banner is shown as a pop up on top of the main screen of the window and blocks all user’s actions with the application.
   2. Full screen banner is shown as a pop up on top of the main screen of the window and doesn’t block all user actions with the application. As an example, the user can switch tabs.
   3. If several displays are switched on, the banner will be displayed on the main screen only.

# Simple Ad Integration

To add MASTAdView into your application just import MASTAdView.h

#import "MASTAdView.h"

And put initialization code in viewDidLoad method.

- (void)viewDidLoad

{

MASTAdView \*\_adView = [[MASTAdView alloc] initWithFrame:CGRectMake(0, 0, self.view.frame.size.width, 50.0) site:19829 zone:88269];

[self.view addSubview:\_adView];

[\_adView release];

[super viewDidLoad];

}

See Also:

* For more code samples examine the Samples application.



Zones may contain ads of different sizes to support various device types. The SDK by default will send the frame size (scaled for Retina as needed) in the ad request but is not guaranteed to only receive ads of that size. The minSize and maxSize parameters can be used to tune the ads received from the server based on the requirements of the device. This allows zone content to contain ads for the two iPhone/iPod as well as the two iPad resolutions.

MASTAdView \*\_adView = [[MASTAdView alloc] initWithFrame:CGRectMake(0, 0, self.view.frame.size.width, 100.0) site:19829 zone:88269];

Example usage of the minSize property to prevent non-Retina ads on Retina equipped devices:

- (void)viewDidLoad

{

MASTAdView \*\_adView = [[MASTAdView alloc] initWithFrame:CGRectMake(0, 0, self.view.frame.size.width, 100.0) site:19829 zone:88269];

CGFloat scale = [UIScreen mainScreen].scale;

adView.minSize = CGSizeMake(self.view.frame.size.width\*scale, 50\*scale);

[self.view addSubview:\_adView];

[\_adView release];

[super viewDidLoad];

}

If the minSize or maxSize properties are used they must be updated post-rotation. Below is an example of handling rotation updates.

- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation

{

CGFloat scale = [UIScreen mainScreen].scale;

adView.minSize = CGSizeMake(self.view.frame.size.width\*scale, 50\*scale);

[\_adView update];

}

# Interstitial Ad Integration

To add interstitial ad into your application just import MASTAdView.h.

#import " MASTAdView.h"

And put initialization code in the viewDidLoad method.

- (void)viewDidLoad

{

[super viewDidLoad];

MASTAdView \*\_adView = [[MASTAdView alloc] initWithFrame:CGRectMake(0.0, 0.0, self.view.frame.size.width, self.view.frame.size.height) site:19829 zone:88269];

\_adView.minSize = CGSizeMake(self.view.frame.size.width, self.view.frame.size.height);

\_adView.showCloseButtonTime = 5;

\_adView.autocloseInterstitialTime = 15;

[self.navigationController.view addSubview:adView];

[adView release];

}

Either the showCloseButtonTime or autocloseInterstitialTime MUST be set to something greater than -1.

- (void)didRotateFromInterfaceOrientation:(UIInterfaceOrientation)fromInterfaceOrientation

{

adView.minSize = CGSizeMake(self.view.frame.size.width, self.view.frame.size.height);

[\_adView update];

}

See Also:

* For more code samples examine the Samples application.

# MASTAdView Customization

## Customize view appearance

1. To enable animation on ad changes set the isAdChangeAnimated property to YES.
2. Ad links are opened in Safari by default. To enable the internal browser set the internalOpenMode property to YES.
3. To center ad content in the view vertically and horizontally set the contentAlignment property to YES. Note: Rich media ads do not support this property.
4. Default UIView customization such as animation, background color, orientation/sizing masks, etc. can be used as well.
5. The defaultImage, autoCollapse and showPreviousAdOnError properties determine the ad view behavior on update failures. Note: MASTAdView is always visible by default.



# MASTAdView Interstitial Ad Customization

## Customize view appearance

1. By default the ad view includes a close button. The button can be replaced by setting the closeButton property to a custom button. If this property is used to set a custom button then the implementation MUST also implement the close logic. To display the default button set the hidden property of the default closeButton to NO.
2. To enable auto close behavior of the ad set the autoCloseInterstitial property to a positive value.
3. To enable showing the close button after a delay set the showCloseButtonTime to a positive value.

# Ad Content Customization

## Content filtering

The content can be filtered by setting the various content filtering properties for keywords, city, carrier, dma, etc. See the API documentation for the full list.

## Location support

The latitude and longitude properties can be set manually or can be set automatically by the SDK based on the location services available to the device.

If the SDK services are used for location the user will be prompted for location permission. See the Location Detection Configuration section of the API documentation for usage of the SDK location services.

# Content Updates

MASTAdView updates content only by the following methods:

1. Calling the update method. Use this after initializing and during display of the owning view controller. Also required after calling stopEverythingAndNotifyDelegateOnCleanup.
2. Every interval of the update timer AFTER update has been called. If stopEverythingAndNotifyDelegateOnCleanup the timer is halted.

Troubleshooting

## Duplicate symbol compile error

If you get a link 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 the SDK contain files with same name. In most cases this error appears due to the duplication of 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…

## Ad content loading issues

1. Verify the specified content zone has ad content.
2. Implement the ad view’s delegate and debug any ad download failure errors.
3. Enable verbose console logging by setting the logMode property to AdLogModeAll.
4. Enable simple test banners by setting the testMode property to YES.

API Documentation

The API documentation can be found in the Documentation folder of the SDK package or the docs folder in the source tree. Both HTML and Xcode documentation packages are available.