Mobile SDK for iPhone/iPad

Index

Introduction and basics

The following documentation contains a detailed description for integration and implementation of the G+J EMS iOS Software Development Kits (SDK). The SDK supports the ORMMA Standard. Following advertising formats and features can be realized with the SDK:

- Standard advertisement
- Interstitials
- Rich media
- Keyword targeting
- Geo targeting
- Custom targeting
- IP targeting
- Unique user / Frequency capping
- In app browser

SDK modules

The SDK is structured in several modules:

GuJBaseSDK: contains base functionality for loading and displaying ads. Native device features can also be used for advertisement e.g. location based services, messaging, video and audio, maps etc.

ORMMA Module: extends the GuJBaseSDK and realizes the ORMMA conform handling of ads. Get further information about ORMMA here http://www.ormma.org

GuJmOcean Module: extends the GuJ ORMMA SDK and realizes the delivery of ads of the performance network optimobile. Get further information about mOcean on http://code.google.com/p/mocean-sdk-ios

GuJ SMARTSTREAM Module: extends the GuJ mOcean SDK and realizes backfill for interstitial placements with SMARTSTREAM inventory.

GuJAdView Context: is a kind of wrapper and contains all mentioned SDKs (Base SDK, ORMMA SDK, mOcean SDK und SMARTSTREAM SDK). Therefore all modules can be easy integrated by opening this module in the project tree.

The special software architecture and -design of the SDK allows the developer the enabling of ads with just a few lines of code.

Supported iOS versions

The SDK supports iOS 5.0 and higher. It was tested on the following devices and iOS versions.

Hardware

- iPhone 3GS
- iPhone 4
- iPhone 4S
- iPhone 5
- iPhone 5S
- iPad 1
- iPad 2
- iPad 3
- iPad Air

iOS version

- iOS 5.x
- iOS 6.x
- iOS 7.x

The starter package

The SDK starter package contains the following components:

- libGuJAdViewContext.aforiOS
- libGuJAdViewContextSimulator.aforthesimulator
- ORMMAResourceBundle.bundle
- VideoAdLib.bundle for the SMARTSTREAM Video SDK
- Required Header-files:
- GuJAdViewContext.h
- Optional Header-files
- GuJAdViewControllerDelegate.h

The libraries **libGUJAdViewContext.a** and **libGUJAdViewContextSimulator** are „fat“ libraries. Content of different architectures (armv7, armv7s, i386) are collected to one big library.

libGUJAdViewContext (see GuJAdViewContext) contains following libs:

- libGuJBaseSDK
- libGuJORMMASDK
- libGuJmOceanSDK
- libGuJXAXISSDK
- libAdMobileSDK (external mOcean SDK)
- libAdSDKLib (external smartstream SDK)

Requirements

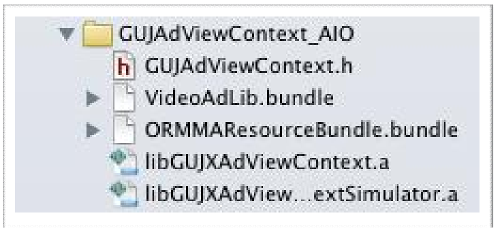
The SDK was compiled in Xcode 4.6.2 with ARC (Automatic Reference Counting). In case of working with another Xcode version you have to make sure the Apple LLVM 4.2 compiler is available. In case of compiling without ARC, the -fobj-arc flag has to be set in the compiler settings.

First steps – project setup And IDE configuration

General settings

Xcode Setup

To integrate the SDK, please add the starter package files to your project folder. To get a better overview, it is recommended to group the files according to their affiliation as follows:



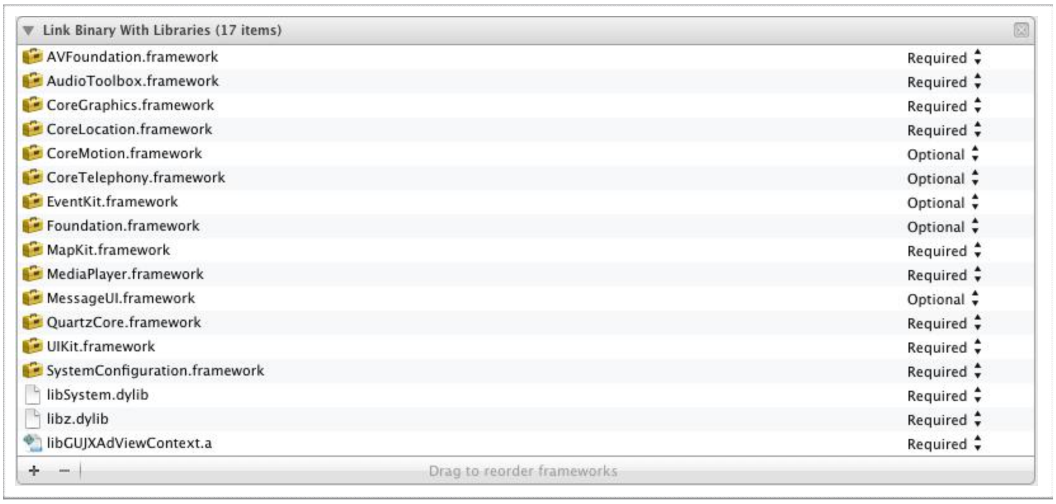
```
▼ GUJAdViewContext_AIO
      GUJAdViewContext.h
    ▶ VideoAdLib.bundle
    ▶ ORMMAResourceBundle.bundle
      libGUJXAdViewContext.a
      libGUJXAdView...extSimulator.a
```

Linked libraries and frameworks

This step is one of the most important integration steps! Please make sure that all required libraries and frameworks are added to your project.

Add the libraries and frameworks in the “Build Phase” tab of your project.

- Open project navigator
- Open project
- Select “Build Phase”.
- Expand “Link Binary With Libraries”



▼ Link Binary With Libraries (17 items)

AVFoundation.framework	Required
AudioToolbox.framework	Required
CoreGraphics.framework	Required
CoreLocation.framework	Required
CoreMotion.framework	Optional
CoreTelephony.framework	Optional
EventKit.framework	Optional
Foundation.framework	Optional
MapKit.framework	Required
MediaPlayer.framework	Required
MessageUI.framework	Optional
QuartzCore.framework	Required
UIKit.framework	Required
SystemConfiguration.framework	Required
libSystem.dylib	Required
libz.dylib	Required
libGUJXAdViewContext.a	Required

Please add the following iOS frameworks to your list. These frameworks are mandatory!

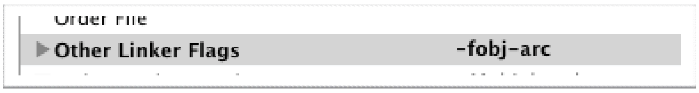
Framework	Used by
AVFoundation.framework	ORMMA SDK, mOcean SDK, SMARTSTREAM SDK
AudioToolbox.framework	SMARTSTREAM SDK
CoreGraphics.framework	mOcean SKD
CoreLocation.framework	ORMMA SDK, mOcean SDK, SMARTSTREAM SDK
CoreMotion.framework	mOcean SKD
CoreTelephony.framework	mOcean SKD, SMARTSTREAM SDK
EventKit.framework	ORMMA SDK, mOcean SDK
MapKit.framework	ORMMA SDK, mOcean SDK
MediaPlayer.framework	ORMMA SDK, mOcean SDK, SMARTSTREAM SDK
MessageUI.framework	ORMMA SDK, mOcean SDK
QuarzCore.framework	mOcean SKD
SystemConfiguration.framework	mOcean SDK, SMARTSTREAM SDK
libSystem.dylib	mOcean SDK, SMARTSTREAM SDK
libz.dylib	SMARTSTREAM SDK

Additional settings for testing, compiling and debugging

Add linker flags

If your project does not use ARC, do not forget to add the "-fobj-arc" flag to the GuJ libraries:

- Open the project navigator
- Open your project
- Select "Build Settings"
- Scroll to "Linking" Options



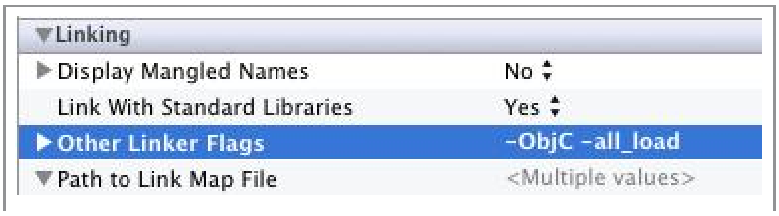
Order File

▶ Other Linker Flags	-fobj-arc

Regardless to the project targets and of the project setup, it is recommended to add the compiler flags "-ObjC -all_load" to "Other Linker Flags":

Open the project navigator

- Open your project
- Select “Build Settings”
- Scroll to “Linking” Options

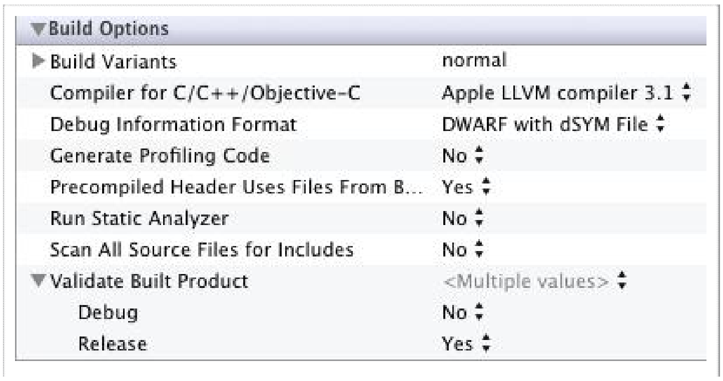


Linking	
Display Mangled Names	No
Link With Standard Libraries	Yes
Other Linker Flags	-ObjC -all_load
Path to Link Map File	<Multiple values>

Compiler LLVM

Make sure that you are compiling with Apple >= LLVM 4.2 compiler.

- Open project navigator
- Open your project
- Select “Build Settings”
- Scroll to “Build Options”



Build Options	
Build Variants	normal
Compiler for C/C++/Objective-C	Apple LLVM compiler 3.1
Debug Information Format	DWARF with dSYM File
Generate Profiling Code	No
Precompiled Header Uses Files From B...	Yes
Run Static Analyzer	No
Scan All Source Files for Includes	No
Validate Built Product	<Multiple values>
Debug	No
Release	Yes

Notes for targeting and debugging

Please make sure that your compatibility target is iOS 5.0 or higher.

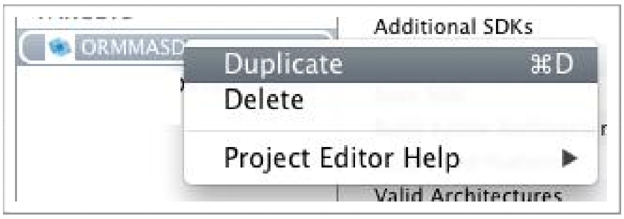
Device and simulator targets

If you are planning to use the simulator for testing, it will be required to add the simulator libraries to your project. It is possible to choose two different targets.

- Open project navigator
- Open your project

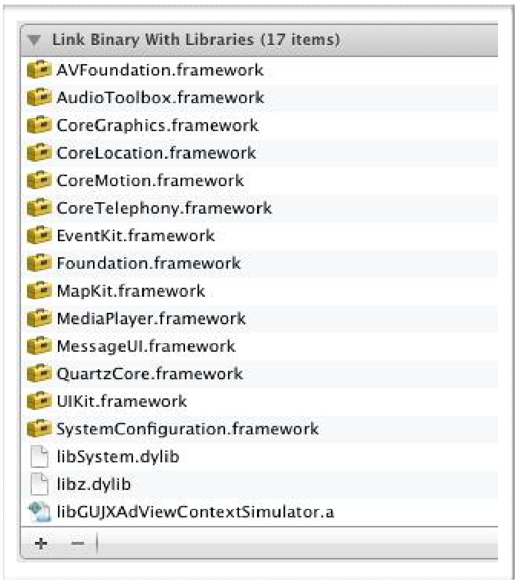
To setup a simple device and test environment proceed as follows:

- Create a device target
- Copy the created device target to the simulator targets



Delete the device libraries from the simulator target’s list and add the libraries with the suffix “simulator”.

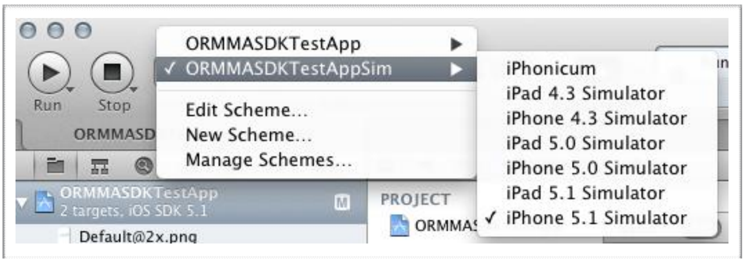
- Open project navigator
- Open project
- Select “Build Phase”.
- Expand “Link Binary With Libraries”



Link Binary With Libraries (17 items)

AVFoundation.framework
AudioToolbox.framework
CoreGraphics.framework
CoreLocation.framework
CoreMotion.framework
CoreTelephony.framework
EventKit.framework
Foundation.framework
MapKit.framework
MediaPlayer.framework
MessageUI.framework
QuartzCore.framework
UIKit.framework
SystemConfiguration.framework
libSystem.dylib
libz.dylib
libGUJXAdViewContextSimulator.a

Please start the different targets via selection of the respective scheme in the Xcode dropdown.

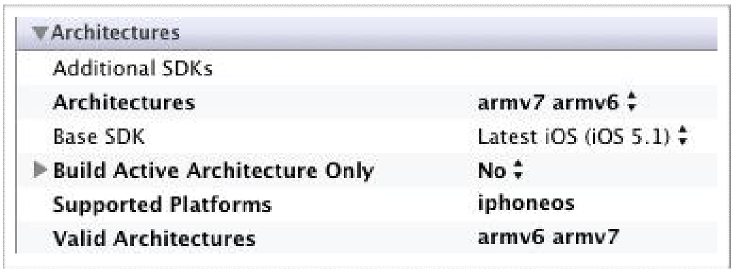


Furthermore you can edit the build settings.

Device target selection

Please select the device target as followed:

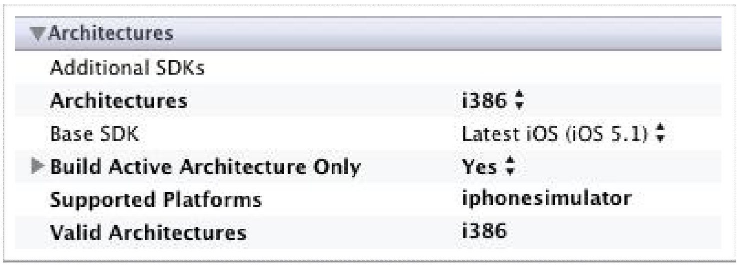
- Open project navigator
- Open project
- Select ”Build Settings”
- Select device target
- Scroll to “Architectures”



▼Architectures	
Additional SDKs	
Architectures	armv7 armv6 ⇕
Base SDK	Latest iOS (iOS 5.1) ⇕
▶ Build Active Architecture Only	No ⇕
Supported Platforms	iphoneos
Valid Architectures	armv6 armv7

Simulator target selection

Selecting the simulator target is equivalent to the “Device target selection”; please choose a simulator target instead of a device target.



▼Architectures	
Additional SDKs	
Architectures	i386 ⇕
Base SDK	Latest iOS (iOS 5.1) ⇕
▶ Build Active Architecture Only	Yes ⇕
Supported Platforms	iphonesimulator
Valid Architectures	i386

Integration and implementation of the SDK

Ad integration guideline -how to add the GuJ SDK to the GuJAdView

Import the **GuJAdViewContext.h** to your **UIViewController** or to your **UIViewheaderfile**.

GuJAdView Methods (Version 2.0)

From version 2.0 on the developer is able to control the **GuJAdView** with the following methods:

```
-(void)show;
```

Shows the successfully loaded **GuJAdView** immediately.

```
 -(void)showInterstitialView;
```

Shows the successfully loaded **GuJAdView Interstitial** immediately.

```
-(void)hide;
```

Hides the **GuJAdView** (also interstitial) immediately.

These methods only take effect if the developer has added the **GuJAdView** (not the interstitial) as a Parent-View’s SubView.

```
-(NSString*)adSpaceId;
```

Additionally the **GuJAdView** was extended by a reference to the current AdSpaceID. This is useful to identify different AdViews.

GuJAdView With GCD-Blocks (Version 2.0)

From version 2.0 on the **GuJAdView** supports Grand-Central-Dispatch (GCD) blocks. For this reason the following type in the **GuJAdViewContext.h** header file was defined:

```
typedef BOOL (^adViewCompletion)(GuJAdView* _adView, NSError* _error);
```

The developer can access the loaded GuJAdView (_adView) and possible error messages (_error) form the **GuJAdViewContext** within the block. It is mandatory to define a Boolean value that returns the visibility of the AdView.

- If the Boolean value returns YES(true/1), the **GuJAdView** will be displayed.
- If the Boolean value returns NO(false/0), the **GuJAdView** will not be displayed. The developer must ensure in a following programming step that the **GuJAdView** will be displayed.

Example:

```
[[GuJAdViewContext instanceForAdspaceId:@"12345"] adView:^BOOL(GuJAdView *_adView,
NSError *_error) {
      if( _error != nil ) {
          NSLog(@"AdViewError: %@",_error);
          return NO;
      } else {
          [self.view addSubview:_adView];
          return YES;
      }}];
```

The following AdView calls additionally contain this Grand-Central-Dispatch (GCD) block:

- (void)adView:(adViewCompletion)completion;
- (void)adViewWithOrigin:(CGPoint)origin completion:(adViewCompletion)completion;
- (void)adViewForKeywords:(NSArray*)keywords completion:(adViewCompletion)completion;
- (void)adViewForKeywords:(NSArray*)keywords origin:(CGPoint)origin completion:(adViewCompletion)completion;
- (void)interstitialAdViewWithCompletionHandler:(adViewCompletion)completion;

(!)Note:

The delegate method "**adViewController:canDisplayAdView**" will **NOT be executed** if the mentioned GCD block calls have been used.

GuJAdView with mOcean backfill

To activate the backfill for an adView, please add a valid site and zone ID during the **GuJAdViewContext** initialization.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId" site:1234 zone:1234]
Or with delegate for notification:
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId" site:1234 zone:1234 delegate:self]
```

The **mOcean backfill** is automatically activated when a site and zone ID was delivered. If no mOcean site and zone ID was delivered, the call will remain without consequences. If you would like to deactivate the backfill manually, please implement the following lines of code.

```
Aktivate:
[myAdViewContext setMOceanBackFill:YES]
Deaktivate:
[myAdViewContext setMOceanBackFill:NO]
```

Instantiate GuJAdView without keywords

The following example describes how to instantiate an adView without any keywords in your app. Please instantiate the **GuJAdViewContext** and add the AdContainer to your ViewController's **viewDidLoad** method.

```
[self.viewaddSubview:
[[GuJAdViewContex tinstanceForAdspaceId:@"AdSpaceId"] adView]
];
> OR GCD BLOCK:
[self.viewaddSubview:[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"]
adView:^BOOL(GuJAdView *_adView, NSError *_error) {
     [self.view addSubView:_adView];
      return YES;
   }];
];
```

Instantiate GuJAdView with keywords

The following example describes how to instantiate an adView including keywords. These keywords are delivered via ad-request and can be used as targeting option. The Ad server checks the entered campaign keywords and matches them with the AdView keywords.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray
arrayWithObjects:@"key1",@"key2", nil]];
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray
arrayWithObjects:@"key1",@"key2", nil] completion:^BOOL(GuJAdView *_adView, NSError *_error)
{
        return YES;
}];
```

Positioning of GuJAdViews

There are two different ways for positioning the AdView in general. Both methods are described as followed.

Positioning of a GuJAdView with CGPoint information

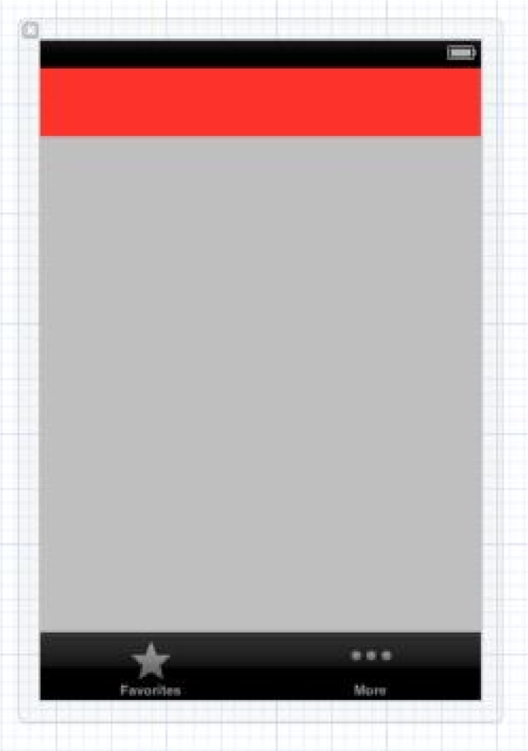
You can deliver a CGPoint via GuJAdViewContext to get access to the position of the ads. The CGPoint x-value is ignored, because the container is centered automatically by the SDK. The y-value y is delivered to the AdView.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f,
20.0f)];
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f,
20.0f) completion:^BOOL(GuJAdView *_adView, NSError *_error) {
        return YES;
}];
```

Positioning a GuJAdView through a view replacement

AdView example with Xcode interface builder view template. With the use of the Xcode interface builder the developer can use a UIView as a GuJAdView template in an interface file. This allows you to have a comfortable overview of the look and feel of your app including the GuJAdView for ad delivery.

Drag and drop a UIView to your interface and fill it with a color for highlighting if you like.



The red space defines the future position for ad delivery.

The following code describes how this UIView is converted to a GuJAdView without losing the position in the interface. The variable “containerView” references to the UIView template (red space) in the InterfaceBuilder.

```
- (void)_loadAdview {
adViewCTX_ = [GuJAdViewContextinstanceForAdspaceId:@"12345" delegate:self];
CGRect frame = containerView.frame;
[containerViewremoveFromSuperview];
containerView = [adViewCTX_ adViewWithOrigin:CGPointMake(0.0, frame.origin.y)];
[self.view addSubView:containerView];
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewWithOrigin:CGPointMake(0.0f,
frame.origin.y) completion:^BOOL(GuJAdView *_adView, NSError *_error) {
   [self.viewaddSubview:_adView];
   return YES;
}];}
```

Positioning of a GuJAdView with keywords

The following code shows the implementation of an AdView with keywords in a defined position. For further details please refer to “Positioning of a GuJAdView with CGPoint information”.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray
arrayWithObjects:@"key1",@"key2", nil] origin:CGPointMake(0.0f, 20.0f)];
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"] adViewForKeywords:[NSArray
arrayWithObjects:@"key1",@"key2", nil] origin:CGPointMake(0.0f, 20.0f)
completion:^BOOL(GuJAdView *_adView, NSError *_error) {
        [self.view addSubview:_adView];
        return YES;
}];
```

Deactivation of GEO-location services for GuJAdView

It is possible to switch the geo localization on and off: use the following lines of code to enable/disable the GEO location services.

(!) Note: From Version 2.0 the static methods enable- and disableLocationService are not static anymore.

```
Aktiviate:
[myAdViewContext enableLocationService]

Deaktivate:
[myAdViewContext disableLocationService]
```

Compiling and starting

After compiling and launching the app you will see advertisement in the defined GuJAdView position.

(!) Note: If you decide to implement the GuJAdView without using the delegate methods of the GuJAdViewControllerDelegate, you won’t be able to analyze the current ad status. This is because you can neither determine whether an ad is loaded or not, nor identify any kinds of exceptions while loading.

For further details please refer to “Notification usage and handling with GuJAdViewControllerdelegate”.

Notification usage and handling with GuJAdViewControllerDelegate

The GuJAdViewControllerDelegate communicates with the GuJAdView and delivers several controller messages to your app. Furthermore it allows you to control the ad delivery.

A variety of delegate methods become optional through the usage of GCD blocks, because the developer gets equivalent information about the delegate methods. It is nevertheless generally advisable – especially for debugging – to use the delegates.

To enable the usage of these delegate methods please add the optional header file **”GUJAdViewControllerDelegate.h”** to your project and implement the GuJAdViewControllerDelegateprotocol into the respective class of the header file (which shall receive the delegate notifications):

```
@interface ViewController : UIViewController<GuJAdViewControllerDelegate>
```

If you implement the **GuJAdViewControllerDelegate** protocol, you will have to instantiate the **GuJAdViewController** with “delegate”, for receiving delegate notifications.

```
[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId" delegate:self]
```

Furthermore you have to add following delegate methods to your UIViewController class.

```
- (void)adViewController:(GuJAdViewController*)adViewController
didConfigurationFailure:(NSError*)error;
- (void)bannerView:(GuJAdView*)bannerView didFailLoadingAdWithError:(NSError*)error;
- (void)interstitialView:(GuJAdView*)interstitialView didFailLoadingAdWithError:(NSError*)error;
```

Additionally you can implement the following delegate methods to collect additional information of the GuJAdView.

```
- (BOOL)adViewController:(GuJAdViewController*)adViewController
canDisplayAdView:(GuJAdView*)adView;
- (void)bannerViewInitialized:(GuJAdView*)bannerView;
- (void)bannerViewDidShow:(GuJAdView*)bannerView;
- (void)bannerViewDidHide:(GuJAdView*)bannerView;
- (void)bannerViewWillLoadAdData:(GuJAdView*)bannerView;

- (void)bannerViewDidLoadAdData:(GuJAdView*)bannerView;

- (void)bannerView:(GuJAdView*)bannerView receivedEvent:(GuJAdViewEvent*)event;
```

If the configuration of the adViewController is incorrect, the delegate method **adViewController:didConfigurationFailure**: will be called.

You will get detailed information about several errors by analyzing the NSError. NSError is a parameter of this method. In many cases the error caused by a missing AdSpaceId.

If an error occurred while loading an ad the delegate method **bannerView:didFialLoadingAdWithError** will be called.

Optional you have the following possibilities to identify errors by

- Reloading the ad
- Deleting and re-implementing the AdViewController
- Doing nothing

When the GuJAdView is loaded and no GCD-Blocks are used and the upcoming GuJAdView is just before visibility, the delegate method **(BOOL)adViewController:canDisplayAdView** is executed. In this method the developer can control the ad display:

- Returning YES(true/1), the AdView will be displayed immediately.
- When returning NO(false/0), the AdView will not be displayed until the developer calls the AdView [AdView show]

If the GuJAdView is loaded and added to your app as a SubView, the delegate method **bannerViewInitialized**: will be called. This allows you to collect detailed information about the GuJAdView. This information can be read by parameter bannerView of this method.

If the AdView started to load the ads, the delegate method **bannerViewWillLoadAdData**: will be called. Then the app is connecting to the adserver.

If the GuJAdView successfully finished loading the ad, the delegate method **bannerViewDidLoadAdData** will be called. You can check if the AdView was loaded completely and correctly; if **bannerViewDidLoadAdData** failed, you will be able to detect errors during the loading process. In case of an error the method won't be called.

If the ad is displayed in the ViewController, the delegate method **bannerViewDidShow:** will be called. Here you are able to change the visibility of an ad.

If the ad won't be displayed or was displayed and is now set to visible=false, the delegate method **bannerViewDidHide**: will be called. In this case you can change the invisibility of the ad. This is useful to fill the free space with new content or new UIElements.

If an ad is receiving a system message, the message will be forwarded to delegate method **bannerView:receivedEvent:.** System messages can be message from 3rd party suppliers like mOcean or XAXIS.

Interstitialintegration

Viewdelegate interstitial notifications

An interstitial is displayed during the switching from one view to another. For controlling the interstitial entry and exit points you have to implement the following delegate methods. Otherwise you won't be able to go to your target view after showing an interstitial.

Add the following *GuJAdViewControllerDelegate* method to your UIViewController:

```
- (void)interstitialViewInitialized:(GuJAdView*)interstitialView;

- (void)interstitialViewWillLoadAdData:(GuJAdView*)interstitialView;

- (void)interstitialViewDidLoadAdData:(GuJAdView*)interstitialView;

- (void)interstitialViewWillAppear;

- (void)interstitialViewDidAppear;

- (void)interstitialViewWillDisappear;

- (void)interstitialViewDidDisappear;
-  (void)interstitialViewReceivedEvent:(GuJAdViewEvent*)event;
```

interstitialViewDidFailLoadingWithError is deprecated in Version 2.0 and will be deleted in further Versions. Please use interstitialViewdidFailLoadingAdWithError: instead.

If an error occurred while loading the delegate method, **interstitialViewDidFailLoadingWithError:** will be called.
You will get detailed information about several errors by analyzing the NSError. NSError is a parameter of this method. In case of an error the interstitialView won't be displayed.

If the InterstitialView is loaded, the delegate method **interstitialViewWillAppear:** will be called immediately. Start here to disable the user interaction for the current view while displaying the interstitial.

If the AdView started to load the ad, the delegate method **interstitialViewWillLoadAdData:** will be called. Then the app is connecting to the ad-server.

If the GuJAdView successfully finished loading the ad, the delegate method **interstitialViewDidLoadAdData** will be called. If **interstitialViewDidLoadAdData** failed, you will be able to detect errors during the loading process. In case of an error the method won't be called.

If the InterstitialView successfully finished loading the interstitial, the delegate method **interstitialViewDidAppear** will be called.

If the InterstitialView has closed (after 7seconds in general), the delegate method **interstitialViewWillDisappear** is called. At this time the app should be prepared for loading all contents of the target view and be able to display this target view.

If the InterstitialView has closed, the delegate method **interstitialViewDidDisappear** is called. This call gives you the signal to display the target view.

If an ad is receiving a system message, the message will be forwarded to delegate method **interstitialView:receivedEvent:** System messages can be message from 3rd party suppliers like mOcean or XAXIS.

InterstitialView implementation

The easiest way to implement the Interstitialview is using the GuJAdViewContext as followed:

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdView]
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"]
interstitialAdViewWithCompletionHandler:^BOOL(GuJAdView *_adView, NSError *_error) {
   return YES;
}];
```

Optional you can deliver keywords within the InterstitialView. This allows G+J EMS to target interstitial campaigns in dependency of these keywords as target.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdViewForKeywords:
[NSArray arrayWithObjects:@"key1",@"key2", nil]
];
> OR GCD BLOCK:
[[GuJAdViewContext instanceForAdspaceId:@"AdSpaceId"]
interstitialAdViewForKeywords:[NSArray arrayWithObjects:@"key1",@"key2", nil]
completion:^BOOL(GuJAdView *_adView, NSError *_error) {
        return YES;
}];
```

In case of a keyword mismatch (keyword parameter ⇔ keywords in adserver) no InterstitialView will be displayed and the delegate method **interstitialViewDidFailLoadingWithError**: will be called.

```
[[GuJAdViewContextinstanceForAdspaceId:@"AdSpaceId"] interstitialAdViewForKeywords:
[NSArrayarrayWithObjects:@"key1",@"key2", nil]
];
```

Startstitial handling

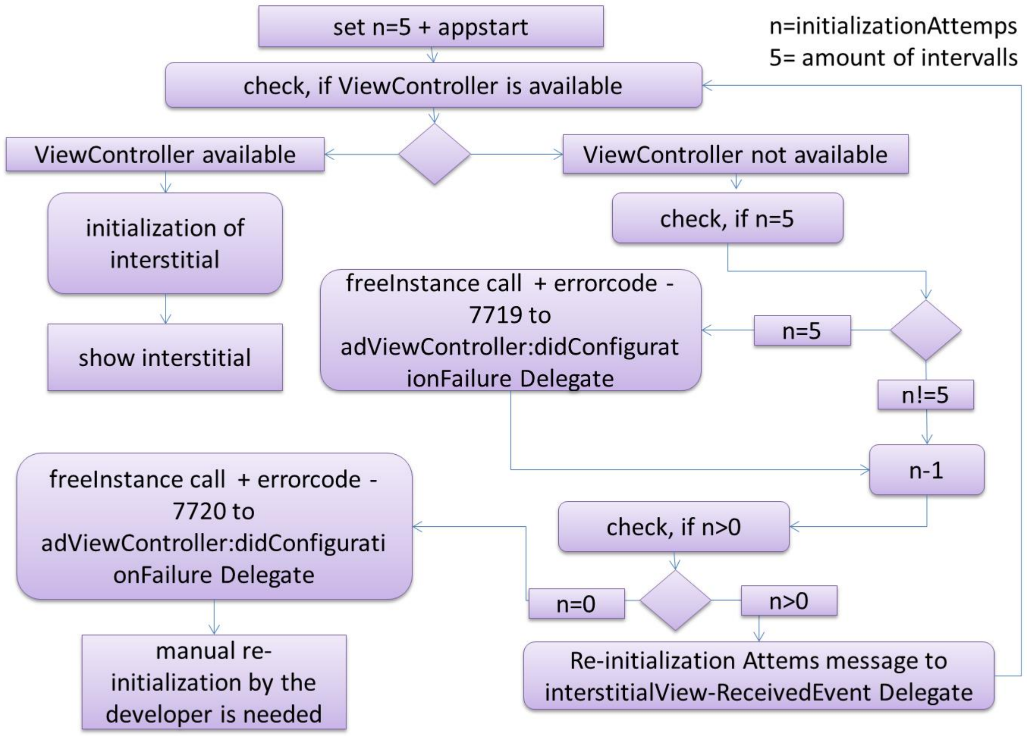
When presenting a Start-interstitial without using GCD-Block-Calls, the AdView appears first when a ViewController (RootViewController) becomes responsible. It takes different time after start of an app until the ViewController is available. To avoid loading of interstitials before this ViewController is accessible the SDK provides an initialization-waiting queue. This is needed for each interstitialView going to be handled in the app delegate class.

The initialization-waiting-queue will check whether the ViewController becomes available or not. The interval is one (1) second. The developer is able to set the amount of intervals in an initializationAttemps property as followed:

```
[gujAdViewContext_ initalizationAttempts:5];
```

For example: the number “5” causes five (5) checks for a valid ViewController. In case of an unsuccessful search for a ViewController the SDK is going to quit. This process is realized as calling the *freeInstance* method. After this, the developer should do a re-initialization programmatically.

In the initialization-waiting queue the SDK fires several error codes and events to different delegates. In the following graphic the control flow for this queue is described.


set n=5 + appstart
n=initializationAttemps
5= amount of intervalls
check, if ViewController is available
ViewController available
ViewController not available
initialization of interstitial
check, if n=5
freeInstance call + errorcode -7719 to adViewController:didConfigurationFailure Delegate
n=5
show interstitial
n!=5
n-1
freeInstance call + errorcode -7720 to adViewController:didConfigurationFailure Delegate
check, if n>0
n=0
n>0
manual re-initialization by the developer is needed
Re-initialization Attems message to interstitialView-ReceivedEvent Delegate

- **Fehlercode -7719** (Missing (Root)ViewController) is sent to the **“adViewController:didConfigurationFailure:”** delegate, when the ViewController availability is first time checked. In case of developers wish for manual initialization the developer can react on this error-code.

- During the initialization-waiting queue the SDK sends the information “Re-initialization Attempt” to the “**interstitialViewReceivedEvent**” for each intervallic check

- In case of last time checking the availability of the ViewController and the initialization failure **the errorcode-7720** is sent to the **“adViewController:didConfigurationFailure:”** delegate. After receiving this code the developer should manually re-initialize the InterstitialView

Add own requestheader and requestparameter

It is possible to define own request header and request parameter for the ad-request. This is needed e.g. for the delivery of the current temperature for temperature targeting. With the following lines of code you are able to define and implement them.

- TheRequestHeaderFieldis a parameter delivered as http-Header in the ad-request.
- TheRequestParameter is a parameter delivered as QueryString as a part of the request URL.

```
[[GuJAdViewContextinstance] addAdServerRequestHeaderField:@"name" value:@"key1"]
[[GuJAdViewContextinstance] addAdServerRequestParameter:@"name" value:@"key1"]
```

The parameter should set before sending the ad-request to the ad-server. This means the parameter should declared before calling adView:, adViewWithOrigin:, adViewForKeywords:, adViewForKeywords:origin, interstitialAdView orinterstitialAdViewForKeywords.

Error codes

Error Code	SDK - Internal Name	Description
-1000	GUJ_ERROR_CODE_GENERAL_UNDEFINED	An undefined/unknown error occured. This might be an invalid ad view response, iOS-related or a programmatically related error.
-1004	GUJ_ERROR_CODE_UNABLE_TO_COMPLETE	A function or request was unable to be completed by the SDK. This can occure when the AdView creation process or the SDK initialization process has been interrupted. This can also occure when the ad server request is interrupted by a connection time out.
-1005	GUJ_ERROR_CODE_COMMAND_FAILED_OR_UNKNOWN	This error code occures if an ORMMA-command failed.
-1006	GUJ_ERROR_CODE_ UNAVAILABLE	This error code occures if an ORMMA function and/or a command and/or a resource is not available.
-1007	GUJ_ERROR_CODE_FAILED_ TO_ASSIGN_OBJ	This error code occures if the SDK can not assign an ad view object to a third party SDK and/or to a third party function.
-1109	GUJ_ERROR_CODE_ORMMA_CALL_UNHANDLED	This error code occures if an ORMMA-call/-function is available but cannot be handled by the SDK.
1	GUJ_ERROR_CODE_ADSPACE_ID	This error code occures if the ad-space-ID is not configured correctly.
1005	GUJ_ERROR_CODE_SERVER_ERROR	This error code occures if the ad-server responses in an unknown format.
1006	GUJ_ERROR_CODE_ INCORRECT_AD_FORMAT	This error code occures if the ad-server responses with an unknown ad-format.
1008	GUJ_ERROR_CODE_INVALID_AD_FORMAT_HEADER	This error code occures if the ad-server responses an empty body, a server error or an buggy flight-header-code.
1009	GUJ_ERROR_CODE_MISSING_ADCONFIGURATION	This error code occures if the ad-configuration is buggy.

Error Code	SDK - Internal Name	Description
1010	GUJ_ERROR_CODE_INVALID_AD_SERVER_RESPONSE	This error code occures if the ad-server response has valid flights-headers, but an invalid ad body format.
22	GUJ_ERROR_CODE_MOCEAN_AD_FAILD_LOADING	This error code occures if the SDK cannot load the mOcean SDK binaries.
400	GUJ_ERROR_CODE_CORE_LOCATION	This error code occures if the core-location framework is buggy or cannot be loaded and/or be initialized.
2003	GUJ_ERROR_CALENDAR_UNAVAILABLE	If the calendar framework is erroneous or can not be loaded and/or cannot be initialized.

Known ORMMA issues

(ISSUE 9000.001)

The ORMMA Javascript (V1.1) delivers no response according to several native function callbacks like video or audio, maps or events. In this case the AdDesigner has to implement additional javascript-handler to be able to cause actions after relevant notifications of the SDK.

(ISSUE 9000.002)

When implementing the click-to-calendar (ormma.createEvent()) functionality the AdDesigner has to implement a notification to give the app-user a signal when a new entry was created in their calendar.

(ISSUE 9000.003)

The delivered ads should contain information about the size of the ad e.g. width: 300px and height: 50px. Otherwise the SDK won't be able to calculate the size of the Ormmaview. The default Ormmaview size and height will be 1px to 1px. Therefore no ad can be displayed and an error with error-code: 9002 would be thrown.

Technical contact persons

Arne Steinmetz

Head Product & Technology
+49 40 3703-7384
steinmetz.arne@ems.guj.de

Daniel Gerold

Product Manager Mobile
+49 221 533-4981
gerold.daniel@ems.guj.de

Jens Jensen

Product Manager Mobile
+49 40 3703-7475
jensen.jens@ems.guj.de