

Greystripe iPhone SDK v2.4

Overview

Welcome! This document will help get you up and running with the Greystripe iPhone SDK 2.4. The Greystripe SDK provides you a simple way to monetize your application via full-screen ads. Once integrated with the SDK, your application will communicate with Greystripe's system to retrieve and display ads. Greystripe handles all ad sales, both in the US and internationally, and provides comprehensive backend accounting and reporting.

API Diffs

For developers updating from the Greystripe SDK 2.3, please note the following API diffs (new developers should skip to Setup & Enrollment):

Modified `GSInit()`

	Declaration
From	<code>void GSInit(GSValue a_appID);</code>
To	<code>void GSInit(const char * a_appID)</code>

While the method signature has changed, you simply now must call `GSInit()` with your app ID wrapped in quotes.

Modified `GSDisplayAd()`

This function will now return `YES` if and only if an ad will be displayed. In SDK 2.3 it was possible for this function to return `YES` but then not have an ad display if the user's device had no network connection.

Modified `GSSetRelativeRotation()`

Calls to this function while the ad display is open will now be ignored in order to ensure the most coherent ad viewing experience.

Added `GSSDKVersion()`

This new function is useful in debugging to ensure that you are actually using the version of the Greystripe SDK that you think you are using. Because of the way XCode automatically updates the library search paths for your project, if you are working with multiple builds (ex. 2.x and 3.x) of the Greystripe SDK, you may unknowingly end up using a different version of our SDK than you think. Simply logging the results of this function will help you verify that you are using the correct version.

Setup & Enrollment

You should already have signed up through Greystripe's developer website and registered the apps you plan to ad-enable. For each app you register you are given a Test GS App ID and a Production GS App ID. The Test GS App ID is used so you can go through a comprehensive set of test ads to ensure your app works with the full range of ad types offered by Greystripe, including our Rich Media ads.

When you release to the app store, you must release using the Production GS App ID, which will display paying ads. When your app first becomes live on the app store, you will see non-premium ads until your app is screened by the Greystripe account management team. This approval process is done to ensure your apps receive appropriate ads and to protect our advertisers against inappropriate content. If you have questions about the status of your app's approval for our higher-paying ads, contact support@greystripe.com.

Developing Your Application

XCODE SETUP

Greystripe supports iPhone OS 2.x and 3.x via two separate builds of its SDK. Depending on your target OS, you must use the appropriate build:

- Use Greystripe SDK 2.4 (OS 2.x) for application builds targeting iPhone OS 2.x.
- Use Greystripe SDK 2.4 (OS 3.x) for application builds targeting iPhone OS 3.x.

Unless you are using features specifically targeted to iPhone OS 3.0 or greater, you will reach the broadest audience by targeting a 2.x iPhone OS version. **Note: Using the wrong Greystripe SDK for your target iPhone OS will result in application failure.**

Along with this document, you should have received a Greystripe SDK ZIP file, which you should unzip to your development tree. To install the Greystripe SDK into your XCode project:

1) From the Project menu choose "Add to Project", and add the following files to your project:

- libGreystripeSDK.a
- GreystripeSDK.h
- GreystripeDelegate.h

Make sure you have removed any old Greystripe files so the new ones are used.

2) The Greystripe SDK requires no additional resource files, but it does require a number of Apple's frameworks to be included. Right-click on your project in the Groups & Files view, and click "Add", and click "Existing Frameworks...". From the "/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/[YOUR_BASE_SDK]/System/Library/Frameworks" folder add the following if they're not already in your project:

- CoreGraphics.framework
- MediaPlayer.framework
- OpenAL.framework
- OpenGL.framework
- QuartzCore.framework
- SystemConfiguration.framework

3) From `"/Developer/Platforms/iPhoneOS.platform/Developer/SDKs/[YOUR_BASE_SDK]/usr/lib"` add the following if they're not already in your project:

- `libz.dylib`
- `libsqlite3.dylib`

INITIALIZATION

```
void GSInit(const char * a_appID);
```

To initialize the Greystripe client within your app, call the `GSInit` function at startup. Typically, this should be done within your `UIApplicationDelegate`'s `applicationDidFinishLaunching:` method. This call will complete initialization in the background and download any updates as well as an ad, if necessary.

Make sure you test `GSInit` with your Production GS App ID before submitting to verify you can see ads. If you accidentally enter an incorrect ID, you will be shown an error ad.

DISPLAYING ADS

```
BOOL GSDisplayAd();
```

To show an ad, simply call `GSDisplayAd()`. To ensure users do not have to wait for ads to load over the network, the SDK will download ads in the background. This process does not affect application performance. `GSDisplayAd` returns `YES` if an ad will be displayed, and `NO` if no ad could be displayed. There are two reasons for an ad not to be displayed: there was not enough time to download an ad or the device does not currently have a network connection.

When you invoke `GSDisplayAd`, the Greystripe client will take control of the view and display a full screen ad that has been downloaded in the background. The ad will be displayed in the orientation specified to `GSSetRelativeRotation` (see below) or by default in portrait mode. When the user is finished interacting with the ad, control will be returned to your application.

Display Frequency

Downloading an ad can take as long as 30 seconds or more depending on network strength and size of the ad, and we recommend showing at most one ad for every two minutes of use. Ads should always be displayed at clear junctures in your application, such as between levels in the case of a game.

We recommend you display one ad after your application's initial menu. If your app has no menu we recommend displaying an ad at launch. This helps to create a consistent user experience and it also makes our advertisers happy.

Display Considerations

When you invoke `GSDisplayAd`, you should take care not to perform other display operations, such as showing `UIAlertViews` or `UIActionSheets`. You should also be sure to stop any graphics display, pause your app, and turn off any sounds. Some ad click-through options such as links to iTunes will close your application, so auto-saving for your users is a must. Clicks to normal websites are handled by a browser bundled with the SDK (`UIWebView`), so in most cases after clicking on an ad the user can return directly to the app without leaving.

Connectivity Failures

In order for click-throughs and advertiser tracking to work, Greystripe only shows ads when the device has a network connection. If `GSDisplayAd` is called when no network connection is available, then it returns `NO` and does not attempt to display an ad. In this case the user will see nothing.

SPECIFYING A MANUAL AD ROTATION (OPTIONAL)

```
void GSSetRelativeRotation(float a_rotation);
```

The default orientation for ads is portrait mode. If your application uses a different orientation, you will need to specify an ad rotation. For landscape ads, you should specify either 90 or 270 as the value for `GSSetRelativeRotation`. If your app supports multiple orientations, you will need to call `GSSetRelativeRotation` each time your app changes orientation to ensure the ads display correctly.

EVENT NOTIFICATION

```
void GSSetDelegate(void * a_delegate);

@protocol GreystripeDelegate<NSObject>
@optional
- (void) greystripeDisplayWillOpen;
- (void) greystripeDisplayWillClose;
- (void) greystripeDidReceiveMemoryWarning;
@end
```

In order to ensure proper ad performance, you should register to receive notifications from the Greystripe SDK when ads are displayed. There are three notifications:

- `greystripeDisplayWillOpen` is sent whenever Greystripe is about to display an ad. Upon receiving this message, the application must pause any animation, sounds, and other processor-intensive activities.
- `greystripeDisplayWillClose` is sent whenever the Greystripe display is about to close. Upon receiving this message, the application should resume any animations, sounds, and other activities that were halted in response to `greystripeDisplayWillOpen`.
- `greystripeDidReceiveMemoryWarning` is sent as a courtesy to applications that may not use `UIViewController`s to control their primary view, and as such will not receive the notification on their own. In the event that an application receives this message, the app should free up as much of its own memory as possible, such as caches that aren't absolutely necessary. **Note: do not respond to this message by calling `GSDealloc`, as doing so will result in undefined behavior of the Greystripe SDK.**

SHUTDOWN

```
void GSDealloc();
```

During application shutdown, you **must** shut down the Greystripe client by invoking `GSDealloc`. Typically, this should be done within your `UIApplicationDelegate`'s `applicationWillTerminate` method, which is guaranteed to be called when your application exits. **Note: if you fail to call `GSDealloc` when you shut down your game, caching of the ads and reporting within Greystripe's system will not function correctly.**