



SDK Extension
for GameMaker Studio 1.4
and GameMaker Studio 2.1
(Android & iOS)

~ API Documentation v2.2.1 ~

by
Josep Gonzalez

INDEX	0
Introduction	3
GameMaker: Studio version tested	3
Current SDK version	3
Minimum OS version	3
OS version tested	3
Games using Unity Ads Extension:	3
Methods	4
unity_ads_initialize(gameId String)	4
unity_ads_initialize_with_test_mode(gameId String, testMode Double)	4
unity_ads_is_initialized()	4
unity_ads_is_ready()	4
unity_ads_is_ready_with_placement_id(placementId String)	5
unity_ads_is_supported()	5
unity_ads_set_debug_mode(enable Double)	5
unity_ads_get_debug_mode()	5
unity_ads_show()	5
unity_ads_show_with_placement_id(placementId String)	6
unity_ads_get_version()	6
unity_ads_get_placement_state()	6
unity_ads_get_placement_state_with_placement_id(placementId String)	6
unity_ads_has_internet_connection()	7
unity_ads_set_gdpr_consent(enable Double)	7
Delegates	8
UADidFinish	8
UADidStart	8
UADidReady	8
UADidError	9
Placement States	10
UAPlacementStateReady	10
UAPlacementStateNotAvailable	10
UAPlacementStateDisabled	10
UAPlacementStateWaiting	10
UAPlacementStateNoFill	10
Finish States	11
UAFinishStateError	11
UAFinishStateCompleted	11

UAFinishStateSkipped	11
Errors	11
UAEErrorNotInitialized	12
UAEErrorInitializedFailed	12
UAEErrorInvalidArgument	12
UAEErrorVideoPlayerError	12
UAEErrorInitSanityCheckFail	12
UAEErrorAdBlockerDetected	12
UAEErrorFileIoError	12
UAEErrorDeviceIdError	12
UAEErrorShowError	13
UAEErrorInternalError	13
Usage	13
IMPORTANT	15
Contact	17
Thank you for your purchase!	18

Introduction

Unity Ads SDK Extension lets you display mobile ads from **Unity Ads** for Android and iOS devices. You can use just this extension to display ads, or you can even use it with other ad extensions all together.

GameMaker: Studio version tested

- v1.4.1804
- v2.1.4.295

Current SDK version

- **iOS:** 2.2.1 (Apr. 23, 2018)
- **Android:** 2.2.1 (Apr. 23, 2018)

Minimum OS version

- **iOS:** 6.0 or later
- **Android:** 4.1 (API level 16) or later

OS version tested

- **iOS:** 7.1.2, 8.4.1, 9.3.5, 10.3.3, 11.2.6
- **Android:** 4.1 (API 16), 4.4 (API 19), 5.1.1 (API 22), 6.0.1 (API 23), 7.0 (API 24), 7.1.1 (API 25), 8.0 (API 26)

Games using Unity Ads Extension:

- **iOS**
 - Devilicious: <https://itunes.apple.com/app/id1156278022?mt=8>
 - Iron Snout+: <https://itunes.apple.com/app/id1162897879?mt=8>
 - Pixel Car Racer: <https://itunes.apple.com/app/id1068808996?mt=8>
- **Android**
 - Devilicious: <https://play.google.com/store/apps/details?id=com.igonfer.devilicious>
 - Iron Snout+: <https://play.google.com/store/apps/details?id=com.snoutup.ironsnoutsteamed>
 - Pixel Car Racer: <https://play.google.com/store/apps/details?id=com.StudioFurukawa.PixelCarRacer>

(Is your game using the Unity Ads Extension? Do you want to place your game here? Contact me and I'll place it in the list above)

Methods

unity_ads_initialize(gameId String)

Initializes the Unity Ads SDK using the specified game ID.

Argument	Description
gameId	The platform specific game identifier. This can be found listed under your project in the Unity Ads dashboard .

Returns: N/A

unity_ads_initialize_with_test_mode(gameId String, testMode Double)

Initializes the Unity Ads SDK using the specified game ID and it may enable the test mode.

Argument	Description
gameId	The platform specific game identifier. This can be found listed under your project in the Unity Ads dashboard .
testMode	Enables Test Mode when set to (1) . Test Mode defaults to being disabled when a value is not specified. While Test Mode is enabled, only test ads will be shown. Test ads do not generate any stats or revenue.

Returns: N/A

unity_ads_is_initialized()

Arguments: N/A

Returns: Double. Returns **(1)** if the Unity Ads SDK is initialized, **(0)** if not.

unity_ads_is_ready()

Arguments: N/A

Returns: Double. Returns **(1)** if the specified ad placement is ready to show an ad campaign. If a placement is not specified, the default ad placement will be used instead.

unity_ads_is_ready_with_placement_id(placementId String)

Argument	Description
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in the Unity Ads dashboard .

Returns: Double. Returns **(1)** if the specified ad placement is ready to show an ad campaign. If a placement is not specified, the default ad placement will be used instead.

unity_ads_is_supported()

Arguments: N/A

Returns: Double. Returns **(1)** if Unity Ads is supported by the current device. Call this method before attempting to initialize Unity Ads.

unity_ads_set_debug_mode(enable Double)

Argument	Description
enable	Logs are verbose when set to (1) , and minimal when (0) .

Returns: N/A

unity_ads_get_debug_mode()

Arguments: N/A

Returns: Double. Returns **(1)** if *Debug Mode* is enabled.

unity_ads_show()

Shows an ad campaign using the default placement specified in the [Unity Ads dashboard](#).

After the ad is shown, the method **unityAdsDidFinish** will be called when the ad placement is closed. This callback method can be used to reward players for opting-in to watch video ads.

Arguments: N/A

Returns: N/A

unity_ads_show_with_placement_id(placementId String)

Shows an ad campaign using the specified ad placement.

After the ad is shown, the method **unityAdsDidFinish** will be called when the ad placement is closed. This callback method can be used to reward players for opting-in to watch video ads.

Argument	Description
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in the Unity Ads dashboard .

Returns: N/A

unity_ads_get_version()

Arguments: N/A

Returns: String. Returns the *Unity Ads SDK* version as a string.

unity_ads_get_placement_state()

Returns the state of the default ad placement specified in the [Unity Ads dashboard](#).

Arguments: N/A

Returns: Double. Returns a **UAPlacementState** constant.

unity_ads_get_placement_state_with_placement_id(placementId String)

Returns the state of the specified ad placement ID.

Argument	Description
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in the Unity Ads dashboard .

Returns: Double. Returns a **UAPlacementState** constant.

unity_ads_has_internet_connection()

Arguments: N/A

Returns: Double. Returns **(1)** if the device has internet connection or **(0)** if not.

unity_ads_set_gdpr_consent(enable Double)

Versions 2.0 and above will automatically present users with an opportunity to opt-out of targeted advertising, with no implementation needed from the publisher. On a per-app basis, the first time a Unity ad appears, the user will see a banner with the option to opt-out of behaviorally targeted advertising. Thereafter, the user can click an information button to receive the opt-out again.

The exception to this is when a publisher or mediator manually requests a user opt-in and sends a flag to Unity (see section on **Manual implementation**, below). In these cases, the Unity opt-out banner will not appear with the ad shown. Please note that users can still request opt-out or data deletion, and access their data at any time by clicking the information icon in a Unity Ad.

Argument	Description
enable	Request consent to users when (1) , and don't when (0) .

Returns: N/A

Delegates

UADidFinish

This method is called after the ad placement is closed. For each call to show, there will be a call to **UADidFinish**, including all failure scenarios.

Argument	Description
type	The constant type of the Delegate method that the Social Asynchronous Event has received.
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in the Unity Ads dashboard .
finishState	The UAFinishState constants describes the result of the ad shown. The value of the finish state is passed to the UADidFinish callback method when the ad is closed.

UADidStart

This method is called at the start of video playback for the ad campaign being shown.

Note: If a video fails to start after the call to show, this method may not be called. In this case, Unity Ads will skip ahead to call **UADidFinish** with a result of **UAFinishStateError**.

Argument	Description
type	The constant type of the Delegate method that the Social Asynchronous Event has received.
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in the Unity Ads dashboard .

UADidReady

This method is called when the specified ad placement becomes ready to show an ad campaign.

Argument	Description
type	The constant type of the Delegate method that the Social Asynchronous Event has received.
placementId	The ad placement identifier. This can be found listed under the iOS or Android platform of your project in

	the Unity Ads dashboard .
--	---

UADidError

In addition to error logs, this method is called when an error occurs with Unity Ads. This method can be used to assist in debugging efforts, as well as for collect statistics on various failure scenarios.

Argument	Description
type	The constant type of the Delegate method that the Social Asynchronous Event has received.
error	The Unity Ads error that occurred. See UAError constants.
message	A description of the error that occurred.

Placement States

The **UAPlacementState** constants describe the readiness of the specified ad placement to show an ad. If an ad placement is not specified, the default ad placement is referenced instead.

UAPlacementStateReady

A campaign is available and ready to be shown.

UAPlacementStateNotAvailable

Either the SDK is not yet initialized, or the specified placement ID is invalid. For a placement ID to be valid, it must be listed among the placements for the iOS platform of your project in the Unity Ads dashboard. Note: Each platform has its own list of placements. When using the same custom placement ID for both Android and iOS, the placement must be added to both platforms.

UAPlacementStateDisabled

The specified placement is currently disabled for the iOS platform of your project. Placement settings can be updated from the Unity Ads dashboard.

UAPlacementStateWaiting

The specified placement is in the process of becoming ready.

UAPlacementStateNoFill

There are no campaigns currently available.

Finish States

The **UAFinishState** contains describes the result of the ad shown. The value of the finish state is passed to the **UADidFinish** callback method when the ad is closed.

UAFinishStateError

An error occurred that prevented the ad from being shown.

UAFinishStateCompleted

The ad was shown from beginning to end without error and without having been skipped.

UAFinishStateSkipped

The ad was skipped before reaching the end.

Errors

The enumerator **UAEError** is used to describe errors that may occur with Unity Ads. Errors are reported by implementations of the **Delegate methods** received in the **Asynchronous Social Events**.

UAEErrorNotInitialized

An attempt to show an ad was made before Unity Ads was initialized.

UAEErrorInitializedFailed

Some condition prevented Unity Ads from being initialized.

UAEErrorInvalidArgument

An invalid value was passed to the Unity Ads API.

UAEErrorVideoPlayerError

An error was encountered during video playback.

UAEErrorInitSanityCheckFail

The device environment sanity check failed. Unity Ads will not function.

UAEErrorAdBlockerDetected

An ad blocker was detected running on device. Unity Ads will not function.

UAEErrorFileIoError

An error occurred while attempting to write a file to device storage.

UAEErrorDeviceIdError

The device ID was not found or was invalid. Unity Ads will not function.

UAEErrorShowError

An attempt to show an ad failed before the video started.

UAEErrorInternalError

Some other internal error occurred.

Usage



Unity Ads SDK Extension comes with a great example project to show how you should use all the functions included in the extension.

- Use the function **unity_ads_initialize** or **unity_ads_initialize_with_test_mode** to initialize the *Unity Ads SDK*.
- (Optional) Use the functions **unity_ads_is_ready** or **unity_ads_is_ready_state_with_placement_id** to know if the sdk is ready to show an ad campaign. The function **unity_ads_is_ready** uses the default *placementId* that you've set in the *Unity Ads Dashboard website*.
- Use the functions **unity_ads_show** or **unity_ads_show_with_placement_id** to display a video ad or a rewarded video ad. The function **unity_ads_show** uses the default *placementId* that you've set in the *Unity Ads Dashboard website*.
- (Optional) Use the functions **unity_ads_get_placement_state** or **unity_ads_get_placement_state_with_placement_id** to know the state of a specific ad placement ID. The function **unity_ads_get_placement_state** uses the default *placementId* that you've set in the *Unity Ads Dashboard website*.
- (Optional) Use the **unity_ads_has_internet_connection** function to know if there is *Internet Connection* or not and the game will be able to get Ads or initialize the SDK.

You'll find the code in the following objects:

- Rooms **r_main**: initialization code for all ads in both platforms (Android & iOS).
- Objects **o_game**: object core of the main room. The most important event is **Social**, where you'll find all the delegate events are handled.
- Objects **o_button**: almost every function is represented by a button to let you run the script inside the **Left released** event.



- The delegate methods will trigger a **Social asynchronous event**. Check the **Example Project** to know how you can handle the **Social asynchronous event** properly.

IMPORTANT

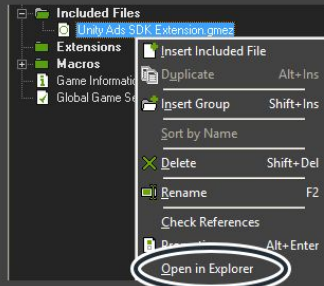
- The **Example Project** uses two example gamelids. Please, replace them by your own gamelids (one for the iOS game and the other for the Android game). Create an account if you don't have it: [Unity Ads homepage](#).
- Setup a **New Game** to display ads: [Unity Ads New Game](#).
- If you've never used before Unity Ads, check the official **Getting Started** guide: [Unity Ads Getting Started](#).
- **Extension file** is in the **Included Files** folder, IMPORT it to the extensions first:

Import Extension

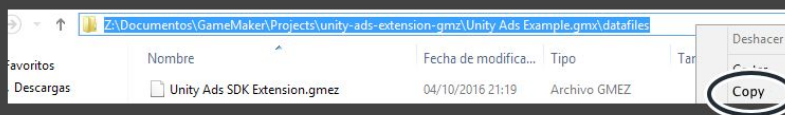
Unity Ads SDK Extension.gmez

by @j_gonfer

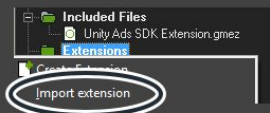
① Right click, select “Open in Explorer”



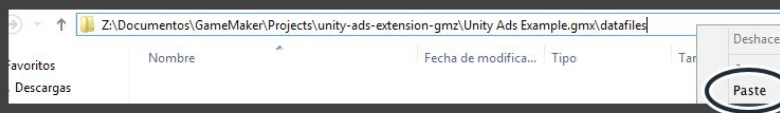
② Copy the directory path



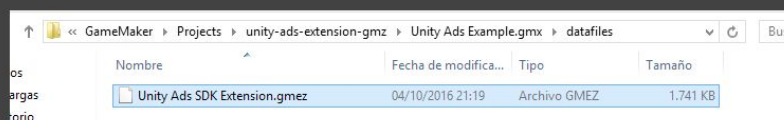
③ Right click, select “Import extension”



④ Paste the directory path into the path field



⑤ Select the “Unity Ads SDK Extension.gmez”



1. Right click on **Unity Ads SDK Extension.gmez** in **Included files** and select **Open in Explorer** option.
2. Copy the directory path from the Path field in the Explorer.
3. Right click on **Extensions** and select **Import extension** option.

4. Paste the directory path that you have in the Clipboard into the Path field in the Explorer.
5. Select the **Unity Ads SDK Extension.gmez** file.

Contact

Feel free to write me if you have any question or problem, or just to say 'hello!'.

- **Trello:** Check the current state of this project in the public [Trello board](#). I'm using this great tool to manage new tasks, versions and changelogs in real time.
- **Twitter:** Follow me on Twitter ([@j_gonfer](#)) to get the latest updates of the *Unity Ads SDK Extension* or even to know things related to GameMaker: Studio and other stuff.



Thank you for your purchase!

Last but not least, if you enjoy this extension it'd be great getting a comment from you in the YoYo Marketplace and help other users like you to find this Extension.