Google Services Manual

Contents

Introduction

This PDF manual is designed for you to use as a reference to the different Google Services functions, and as such does *not* contain tutorials on how to set up the API in your games. If you wish information on setting up, general use, etc… then please see the following YoYo Games Knowledge Base articles:

- Android: Google Play Services – Logging In
- Android: Google Play Services – Leaderboards

- Android: Google Play Services – Achievements

This manual *only* covers the Google Play Services (Logging in, leaderboards, and achievements), as Google Mobile Ads have their own manual included with the extension and are also covered in the following YoYo Games Helpdesk documentation:

- iOS and Android: Google Consent SDK
- iOS and Android: Google Mobile Ads Setup

The Asynchronous Social Event

When using the Google Play Services extension in your projects, you will be calling different functions that will trigger "callbacks" from the Google API. What this means is that certain functions will be run but won't actually give a result until sometime in the future, which could be the next step, or it could be a few seconds later. This result, when it comes, is called the "callback" and is the Google API responding to something you've done. This callback is dealt with in the **Asynchronous Social Event**.

This event will always have a DS map in the GML variable `async_load`, and this map can be parsed to get the required information. Each function will generate different callbacks, but they will all have certain key/value pairs in common, which we'll list here:

- **"id"** – This is the event ID key and it will hold a CONSTANT with the ID of the event that has been triggered. For example, if it's an event to tell you a score has been posted, then the constant will be `GooglePlayServices_EVENT_ID_POST_SCORE`. See the different functions for details about the constant returned for each.

- **"type"** – This is the event type key and it will hold a string with the type of event that has been triggered. For example, if it's an event to tell you a score has been posted, then the string will be "`achievement_post_score_result`". See the individual functions for specific details of the string returned for each.

- **"status**" – This is the status of the call that triggered the async event. In general this will be 1 for a success and 0 for a fail.

The rest of the key/value pairs in the map will depend on the function that triggered the Async Event, and you should check the individual functions listed in the rest of this manual for exact details.

Extension Functions

The rest of this manual is taken up with the different functions and constants that are included as part of the Google Play Services Extension or in the manual. Note that the function **`GooglePlayServices_Init()`** – which is included in the extension functions – *should not be used in your game code*. This function simply initialises the Google Play Services API, and is called automatically by the extension as part of the initialisation when your game is run.

GooglePlayServices_Status

Description

This function will check to see if the Google Play Services API is installed on the user device and has been initialised correctly. This API should be installed and running on all Android phones by default but may have been removed or disabled by the user which is why this check can be performed. The function will return one of the constants listed below depending on the actual API status.

Syntax

```
GooglePlayServices_Status();
```

Returns

Constant

Constant	Error Code	Description
GooglePlayServices_SUCCESS	0	The connection to the Google Play Services API was successful.
GooglePlayServices_SERVICE_MISSING	1	The Google Play Services API is missing on the device
GooglePlayServices_SERVICE_UPDATING	18	Google Play Services API is currently being updated on this device.
GooglePlayServices_SERVICE_VERSION_UPDATE_REQUIRED	2	The installed version of Google Play Services API is out of date.
GooglePlayServices_SERVICE_DISABLED	3	The installed version of the Google Play Services API has been disabled on this device.
GooglePlayServices_SERVICE_INVALID	9	The version of the Google Play Services API installed on the device is not authentic.

Example

```
if GooglePlayServices_Status() == GooglePlayServices_SUCCESS
    {
    global.API_status = true;
    }
else
    {
    alarm[0] = room_speed;
    global.API_status = false;
    }
```

achievement_login_status

Description

This function will check to see if the user is already logged into Google services, in which case it will return true, and if not then it will return false. If it returns true, there is no need to call the `achievement_login()` function.

Syntax

```
achievement_login_status();
```

Returns

Boolean

Example

```
if !achievement_login_status()
   {
   achievement_login();
   }
```

achievement_login

Description

This function logs the user into the Google Play leaderboard and achievement service. If the service is not available, the user is logged into a "pretend" server for the service and all achievements and scores are stored on the device so that when the actual service is available, these details can be uploaded.

The function will trigger a **Social Asynchronous Event** callback which contains the **async_load** DS map populated with the relevant key/value pairs. The id key of this DS map is used to identify the correct callback (there can be more than one trigger function for any given asynchronous event) and will be paired with the constant **achievement_our_info** as well as a number of other key/value pairs. The exact contents of the map are shown below:

- "**id**" - For this function it should be **achievement_our_info**
- "**name**" - The name of the user that is currently logged in (a string).
- "**playerid**" - The unique player id for the user that is currently logged in.

Syntax

```
achievement_login();
```

Returns

N/A

Example

```
if !achievement_login_status()
   {
   achievement_login();
   }
```

achievement_logout

Description

This function logs the user out of the Google Play leaderboard and achievement service. This will stop all further achievements and scores from being recorded.

Syntax

```
achievement_logout();
```

Returns

N/A

Example

```
achievement_logout();
```

achievement_available

Description

This function will return **true** if the user is currently connected to the internet and the Google Play leaderboard and achievement services are available, otherwise it will return **false**.

Syntax

```
achievement_available();
```

Returns

Boolean

Example

```
if achievement_available()
    {
    achievement_post("ach_Level1", 100);
    }
```

achievement_post

Description

You can use this function to send your achievements to the Google Play leaderboard and achievement service. You give the unique ID of the achievement (this is the ID that was assigned to it when you set up the achievement) and the percentage that the player has completed towards getting it (0 - None, 100 - Completed).

The function will trigger a Social Asynchronous Event where the returned async_load DS map will have the following key/value pairs:

- "**type**" - This is the type of event that has been fired, which will be the string "achievement_post_result" for this function.
- "**id**" - This is the ID of the event, and, for this function, it will return a GML constant **GooglePlayServices_EVENT_ID_POST_ACHIEVEMENT**.
- "**achievement_id**" - This is the unique ID of achievement that has been updated.
- "**status**" - This will be 1 if the achievement was updated successfully, or 0 if the request failed

Syntax

```
achievement_post(achievement, percent);
```

Argument	Description	Data Type
achievement	The ID of the achievement.	String
percent	The percentage of the achievement completed.	Real

Returns

N/A

Example

```
if achievement_available()
    {
    achievement_post("CgkI9tjW74AEEAIQAw", 100);
    }
```

achievement_increment

Description

Games can have achievements with no completion value and so you can use this function to increment those achievement by a given amount. You give the ID of the achievement as a string (this is the unique achievement ID that got assigned when you set it up on the Google Play dashboard), and then you give the value to increment by.

The function will trigger a **Social Asynchronous Event** where the returned **async_load** DS map will have the following key/value pairs:

- "**type**" - This is the type of event that has been fired, which will be the string "achievement_increment_result" for this function.
- "**id**" - This is the ID of the event, and, for this function, it will return a GML constant **GooglePlayServices_EVENT_ID_INCREMENT_ACHIEVEMENT**.
- "**achievement_id**" - This is the unique ID of achievement that has been incremented.
- "**status**" - This will be 1 if the achievement was incremented successfully, or 0 if the request failed
- "**complete**" - This will be 1 if the achievement has been completed by this increment, or 0 if the request failed

Syntax

```
achievement_increment(achievement, value);
```

Argument	Description	Data Type
achievement	The ID of the achievement.	String
value	The amount to add on.	Real

Returns

N/A

Example

```
if !instance_exists(obj_Enemy)
    {
    achievement_increment("CgkI9tjW74AEEAIQAw", global.EnemiesKilled);
    }
```

GooglePlayServices_AchievementReveal

Description

This function will reveal a hidden achievement so that the player can see what it is on their Google Play pages for the game. You supply the achievement ID as a string, and the function will trigger a **Social Asynchronous Event** where the returned **async_load** DS map will have the following key/value pairs:

- "**type**" - This is the type of event that has been fired, which will be the string "achievement_reveal_result" for this function.
- "**id**" - This is the ID of the event, and, for this function, it will return a GML constant **GooglePlayServices_EVENT_ID_REVEAL_ACHIEVEMENT**.
- "**achievement_id**" - This is the unique ID of achievement that has been revealed.
- "**status**" - This will be 1 if the achievement was revealed successfully, or 0 if the request failed

Syntax

```
GooglePlayServices_AchievementReveal(achievement);
```

Argument	Description	Data Type
achievement	The ID of the achievement.	String

Returns

N/A

Example

```
if achievement_available()
    {
    achievement_reveal("CgkI9tjW74AEEAIQAw");
    }
```

achievement_reset

Description

This function will reset all achievements back to their initial values for the game. This function is provided as a debug function and it is not recommended that you permit the end-user to do this in your games.

Syntax

```
achievement_reset();
```

Returns

N/A

Example

```
if debug_mode
   {
   if keyboard_check_pressed(vk_space) achievement_reset();
   }
```

achievement_post_score

Description

You can use this function to send your score to the chosen leaderboard and achievement service. You supply the ID of the score table as a string (this is the unique leaderboard ID that got assigned when you set up the leaderboard), and then you give the actual score value. The function will trigger a **Social Asynchronous Event** where the returned async_load DS map will have the following key/value pairs:

- "**type**" - This is the type of event that has been fired, which will be the string "**achievement_post_score_result**" for this function.
- "**id**" - This is the ID of the event, and, for this function, it will return a GML constant **GooglePlayServices_EVENT_ID_POST_SCORE**.
- "**leaderboard_id**" - This is the unique ID of leaderboard that has been updated.
- "**status**" - This will be 1 if the scores were updated successfully, or 0 if the request failed.
- "**daily_new_best**" - This will be 1 if the score submitted was the user's best in the last day, or 0 otherwise.
- "**daily_best**" - This will hold the user's top score from the last day of playing.
- "**weekly_new_best**" - This will be 1 if the score submitted was the user's best in the last week, or 0 otherwise.
- "**weekly_best**" - This holds the user's top score from the last week.
- "**all_time_new_best**" - This will be 1 if the score submitted was the user's best of all time, or 0 otherwise.
- "**all_time_best**" - This holds the user's top score of all time.

Cont…/

Cont…/

Syntax

```
achievement_post_score(leaderboard_id, score);
```

Argument	Description	Data Type
leaderboard_id	The ID of the scoreboard to post the score to.	String
score	The score to send	Real

Returns

N/A

Example

```
if achievement_available()
    {
    achievement_post_score(global.LeaderboardID, global.Scr);
    }
```

achievement_show_achievements

Description

This function will open the Google Play achievements page overlay. Please note that this is an *asynchronous* function, ie: your game will continue to run in the background while the achievements page is being shown. As such, you should be careful where you use this and make sure to pause the game or only permit it to be shown in areas of your game where it will not interfere with the game-play.

Syntax

```
achievement_show_achievements();
```

Returns

N/A

Example

```
if achievement_available()
	{
	global.Pause = true;
	instance_create_layer(0, 0, "Controllers", obj_Pause);
	achievement_show_achievements();
	}
```

achievement_show_leaderboards

Description

This function will open the Google Play leaderboards page overlay. Please note that this is an *asynchronous* function, ie: your game will continue to run in the background while the leaderboards page is being shown. As such, you should be careful where you use this and make sure to pause the game or only permit it to be shown in areas of your game where it will not interfere with the game-play.

Syntax

```
achievement_show_leaderboards();
```

Returns

N/A

Example

```
if achievement_available()
    {
    global.Pause = true;
    instance_create_layer(0, 0, "Controllers", obj_Pause);
    achievement_show_leaderboards();
    }
```

achievement_show_challenge_notifications

Description

With this function you can show, or suppress, the various different "toast" pop-up notifications relating to challenges. When the arguments are set to `true` these messages will appear, informing the player of any local or remote challenges received as well as those challenges that have been completed, while setting them to `false` will suppress these notifications.

Syntax

```
achievement_show_challenge_notifications(receive, local, remote);
```

Argument	Description	Data Type
receive	Show challenges received pop-up (true) or not (false).	Boolean
local	Show completed local challenges pop-up (true) or not (false).	Boolean
remote	Show completed remote challenges pop-up (true) or not (false).	Boolean

Returns

N/A

Example

```
achievement_show_challenge_notifications(true, false, false);
```

achievement_load_leaderboard

Description

This function will return a status string which you can use to display the status of your interstitial ads. The possible return values are:

- "**Not Ready**" - No interstitial has been loaded
- "**Ready**" - The interstitial has been loaded and is ready to show

Syntax

```
achievement_load_leaderboard(ident, minindex, maxindex, filter);
```

Argument	Description	Data Type
ident	The unique ID of the leaderboard as shown on the developer dashboard.	String
minindex	The starting index value to get the leaderboard data from.	Integer
maxident	The maximum index value to get the leaderboard data from.	Integer
filter	Whether to filter the results to those that are on your friends list or to all players (see the Description below)	

Returns

String

Example

```
if GoogleMobileAds_InterstitialStatus() == "Ready"
    {
    GoogleMobileAds_ShowInterstitial();
    }
```