Google Play Licensing Extension

Contents

Introduction

Google Play offers a licensing service that lets you enforce licensing policies for applications that you publish on Google Play. With Google Play Licensing, your application can query Google Play at run time to obtain the licensing status for the current user, then allow or disallow further use as appropriate.

When you target the Android platform, you get the possibility to upload your game to the Google Play app store. Using Google Play then gives you the option to add Google Licencing to your project, which is a protective measure to help reduce the risk of people running your game from sources other than Google Play.

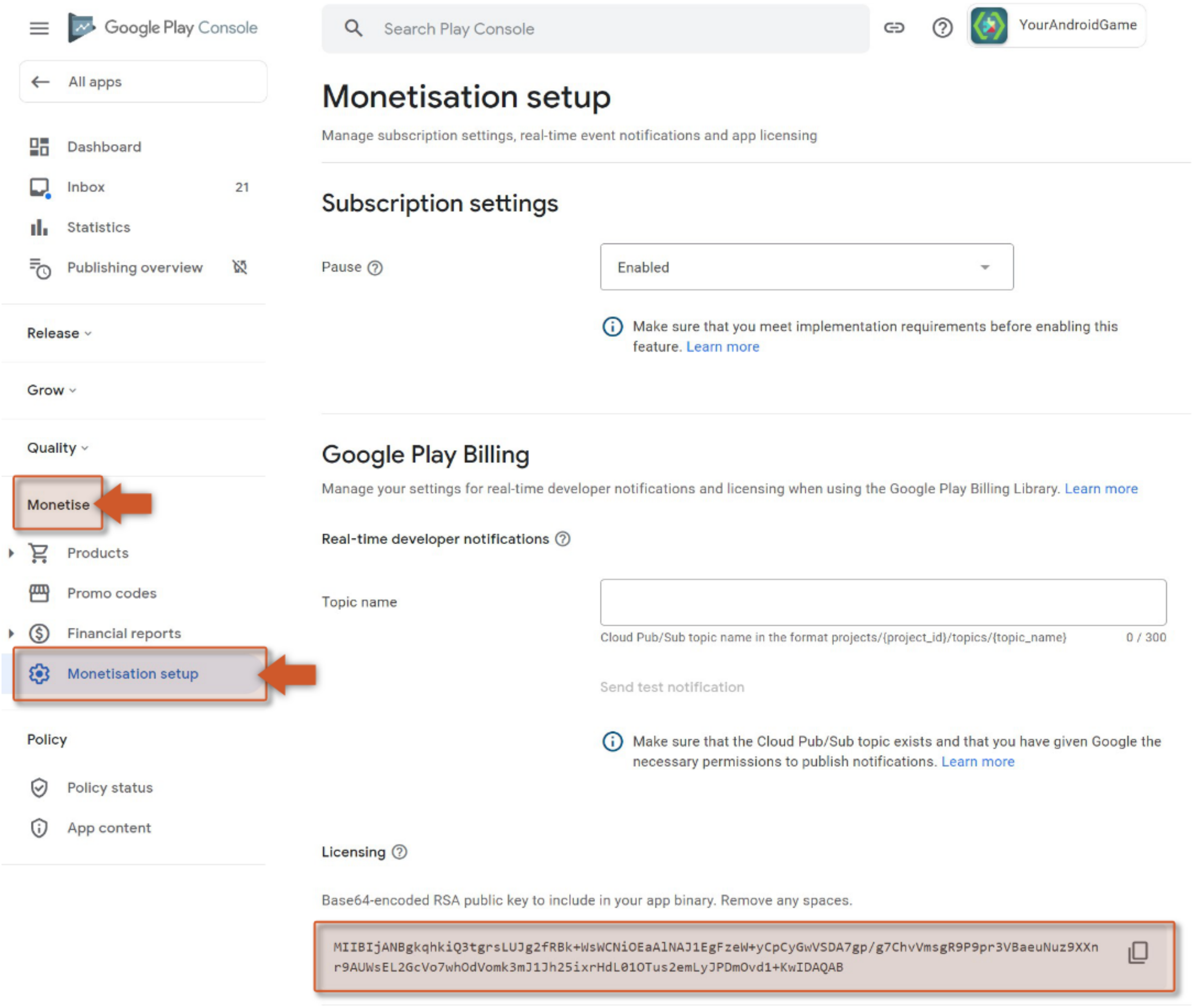
Extension’s Features

- Protect your game from unauthorized distribution.

Setup

This extension provides no in-game functions and only requires some initial setup. To correctly set up your game please follow these steps:

1. Once you have the extension installed, you then need to set up licensing (more info) for that go to your Google Play dashboard and set up an **App Listing** for the game (from your Google Play console click the Create Application button and follow the on-screen prompts). You then need to get the Public Key from the **Monetise** → **Monetization Setup** section of the Developer Console and then copy the Licensing Public Key from the box at the bottom.



Monetisation setup

Manage subscription settings, real-time event notifications and app licensing

Subscription settings

Pause: Enabled

Make sure that you meet implementation requirements before enabling this feature. Learn more

Google Play Billing

Manage your settings for real-time developer notifications and licensing when using the Google Play Billing Library. Learn more

Real-time developer notifications

Topic name:

Cloud Pub/Sub topic name in the format projects/{project_id}/topics/{topic_name} 0 / 300

Send test notification

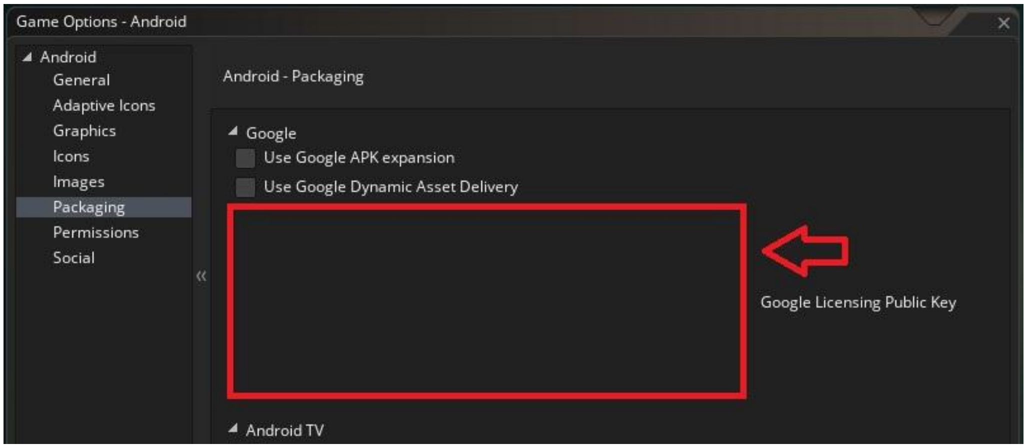
Make sure that the Cloud Pub/Sub topic exists and that you have given Google the necessary permissions to publish notifications. Learn more

Licensing

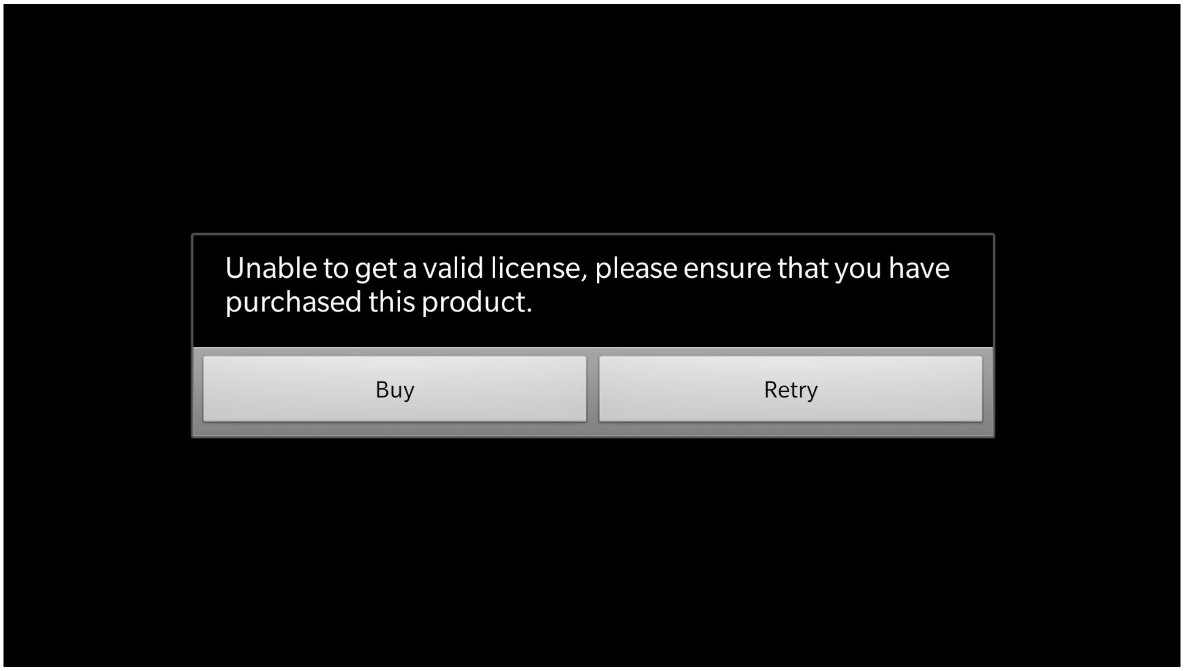
Base64-encoded RSA public key to include in your app binary. Remove any spaces.

```
MIIBIjANBgkqhkiQ3tgrsLUJg2fRBk+WsWCNiOEaAlNAJ1EgFzeW+yCpCyGwVSDA7gp/g7ChvVmsgR9P9pr3VBaeuNuz9XXn
r9AUWsEL2GcVo7whOdVomk3mJ1Jh25ixrHdL01OTus2emLyJPDmOvd1+KwIDAQAB
```

2. You need to copy that to the clipboard then go back to GameMaker and add it into the **Google Licencing Public Key** section of the **Android Game Options**.

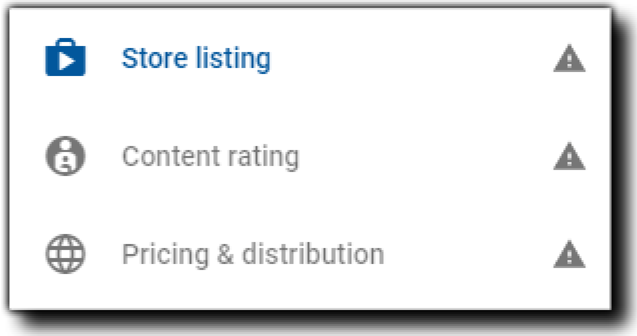


3. Now, with Google Licensing enabled, testing your game will show a message saying that the game could not be licensed and would you like to retry the connection or go to the store to buy the game.


Unable to get a valid license, please ensure that you have purchased this product.

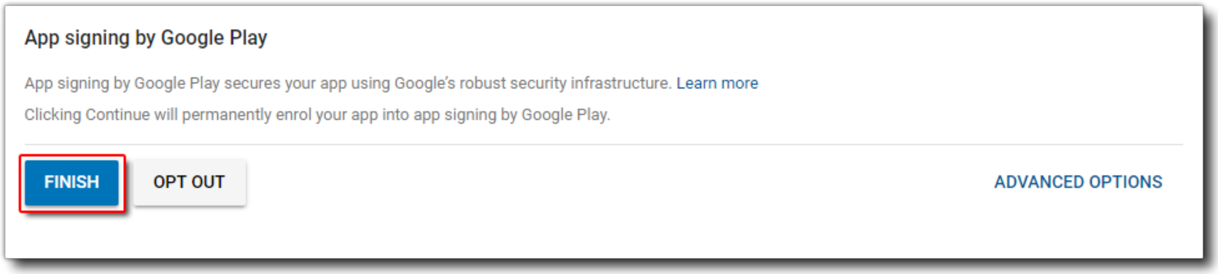
It looks like the licensing works, but you can't test the game anymore... to fix this so you can continue developing (and in the process see that the licensing works) you need to create an **Alpha** build of the project and upload it to the store.

4. Before you can test that the licensing works and keep playing your game on your devices, you will need to upload an APK of the project as an **Alpha** or an **Internal Test** build. To do this, you should first fill in the minimum required details for your app listing on the console, shown in the following image.



Some of these sections require you to upload an APK first, so go to the section **App Releases** and click the **Manage Alpha** (or the Manage Internal Test) text. On the next page you want to create a **New Alpha/Test Release**.

[IMPORTANT] At this point you will be asked if you want to use Google Play App Signing for your game. You should select **CONTINUE/FINISH** for this, as it is required for the extension to work.



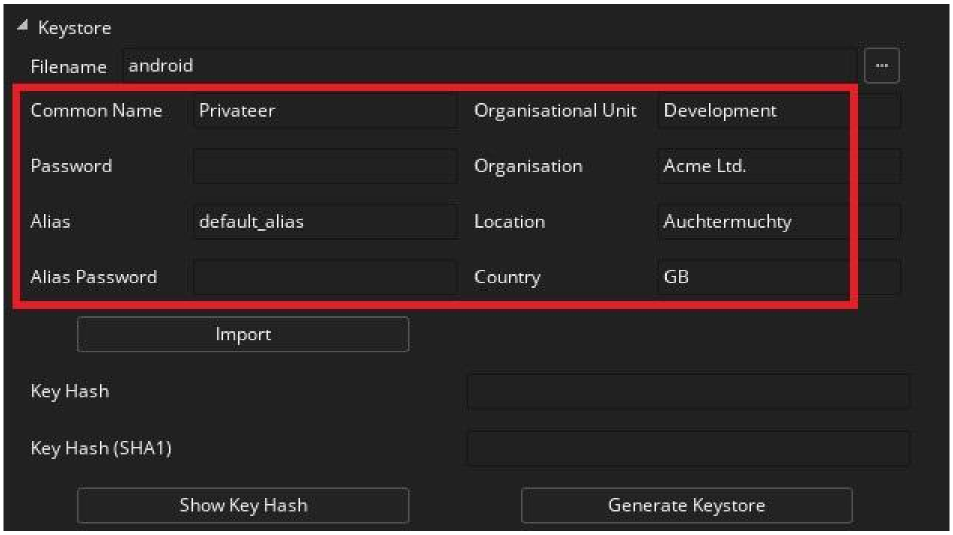
App signing by Google Play

App signing by Google Play secures your app using Google's robust security infrastructure. Learn more

Clicking Continue will permanently enrol your app into app signing by Google Play.

5. Once you opt into the App Signing program, you can then upload an APK as an Alpha/Test build (if you don't know how to make an APK then please see here). Once it's uploaded you should then continue to fill out the rest of the relevant app listing details (like content rating or pricing), and then when all is correct (you will see green ticks beside the relevant items) you can return to the Alpha/Test section, review the details and publish.

   [IMPORTANT] When you opt in to the Google Licensing, you will need to sign each app with a **UNIQUE KEYSTORE**. This is a security measure implemented by Google as part of the licensing system. Don't forget that you need to keep this **.keystore** file along with its information stored in your hard drive in order to be able to update your application.



Keystore

Filename	android		
Common Name	Privateer	Organisational Unit	Development
Password		Organisation	Acme Ltd.
Alias	default_alias	Location	Auchtermuchty
Alias Password		Country	GB
Key Hash			
Key Hash (SHA1)			

Note that once you publish the Alpha/Test build, it can take a few hours for it to be accepted and another few hours for it to become "live" and available for testing.

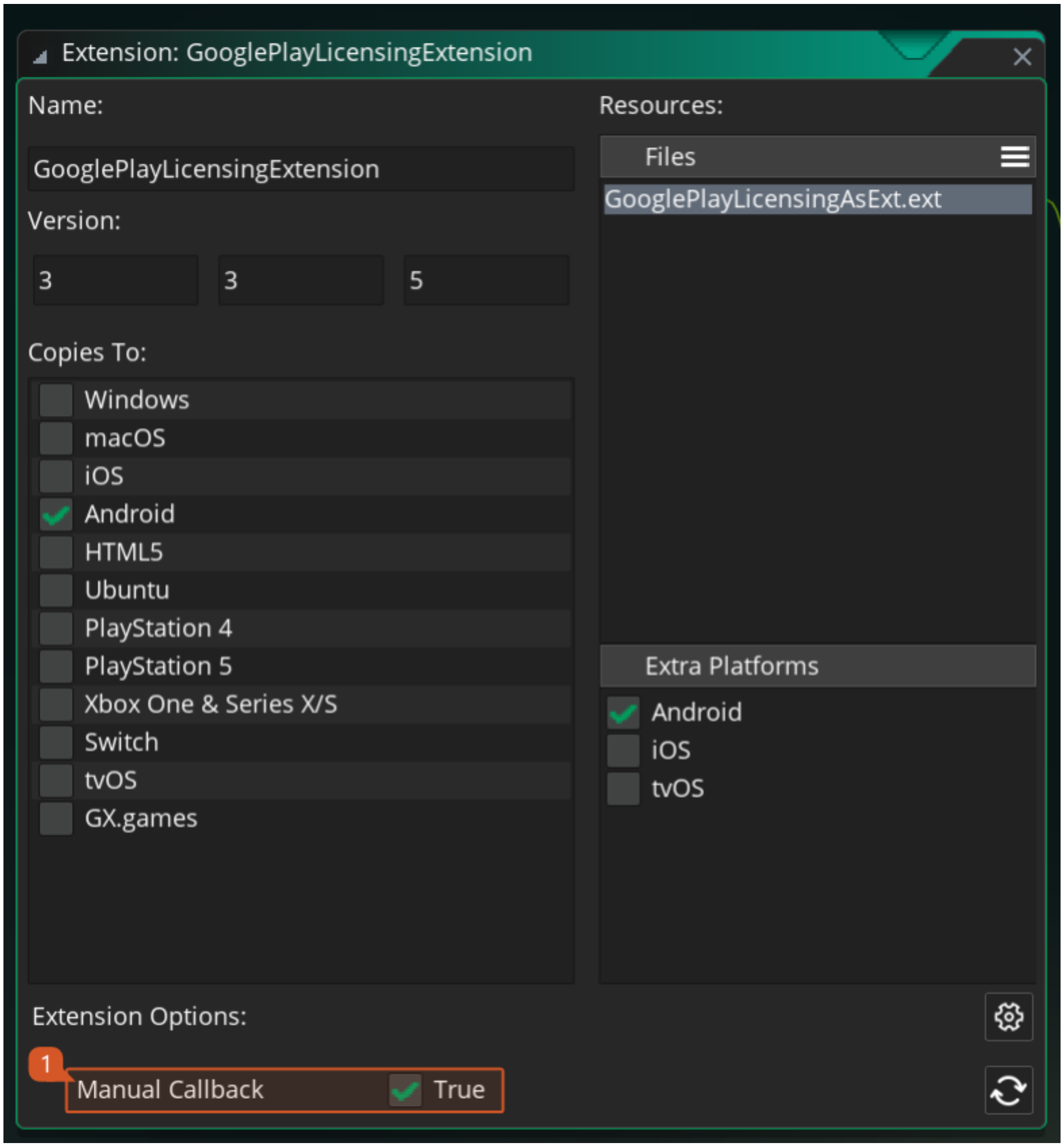
6. Once the app is live as an Alpha on the Play store, you should delete any test APK's created by GameMaker that are on your device, and then download the version from the Google Play Store. This should be run once to ensure that licencing is working as required. You should be able to run the game now from GameMaker Studio 2 on your Android device and the previous message about retrying or buying will not be shown, indicating that the project has been licensed correctly.

   [NOTE] If you still get the licensing warning message, make sure that the email used to register the device is part of the defined **Testers** (as a developer your email should be automatically added, more info)

Extension Options

This new version of the extension allows developers to manually handle the license validation response. To enable this feature access the extension options panel by navigating to the extension asset in the asset browser and double clicking it.

Now you should be able to enable or disable the Manual Callback flag of the extension.



Extension: GooglePlayLicensingExtension

Name:

GooglePlayLicensingExtension

Version:

3 3 5

Copies To:

- □ Windows
- □ macOS
- □ iOS
- ☑ Android
- □ HTML5
- □ Ubuntu
- □ PlayStation 4
- □ PlayStation 5
- □ Xbox One & Series X/S
- □ Switch
- □ tvOS
- □ GX.games

Resources:

Files

GooglePlayLicensingAsExt.ext

Extra Platforms

- ☑ Android
- □ iOS
- □ tvOS

Extension Options:

1 Manual Callback ☑ True

Functions

This new version of the extension will require developers to manually make the validation of the license using a provided function and also allows them to handle the response in a way they see fit.

`GooglePlayLicensing_CheckLicensing()`

Description: This function triggers the validation process of the license that will result in a callback event being executed when the process is finished. If the resulting Async Social Event returns a **not allowed** flag with a *GooglePlayLicensingExtension_RETRY* (reason code) the developer should try to check the license again as this indicates there was a connection error.

Note: The asynchronous event will not be triggered if the **Manual Callback** option is disabled under Extension Options.

Returns: N/A

Triggers: ASYNC SOCIAL EVENT

type: "GooglePlayLicensingExtension"

allowed: {*boolean*} whether or not the user is licensed to use the app.

reason: {*real*} the reason code **if the user is not allowed** (see LicensingErrorReason)

Constants

LicensingErrorReason

→ GooglePlayLicensingExtension_RETRY