

Howto solve “Android apk is built Debuggable” when uploading to Google Play Console

Introduction

When we upload an app to Google Play, we have 2 options to sign it for ourselves or let Google sign it for us. The second option is the most comfortable, since it is Google who stores our private key to sign the applications, but still requires that we sign our apk with a private key that Google calls upload key.

We must provide Google with the public key corresponding to this private key, this allows google to make sure that we are the author of the apk. If we do not follow these steps, Google extracts the public key from the first apk that we upload to the playStore. In our case, generated by GDeveloper when compiling, so we do not have it!!.

Why does Google say that our apk is debuggable? Because, it really is debuggable!!. There is an entry in the AndroidManifest.xml where the option debuggable is set to TRUE. The easy solution is unpack the apk generated by Gdeveloper, modify the androidManifest.xml, repack the apk and sign it. In this document I will explain this solution.

First

If we don't have the upload key, we must contact Google and tell them that we have lost the keystore where the private key to sign the applications is stored.

To do this follow this link

<https://support.google.com/googleplay/android-developer/contact/otherbugs>

Second

You will get an email from Google in about 2-3 business days explaining how to generate a new upload key. They refer to Android Studio and command line, I'll go for command line. You must have java installed on your system, as keytool is a java utility.

Here's how to generate and register a new upload key:

1. Use the following command line to generate a new key:

```
keytool -genkeypair -alias upload -keyalg RSA -keysize 2048 -validity 9125 -keystore keystore.jks
```

- This key must be a 2048 bit RSA key and have 25-year validity.

2. Export the certificate for that key to PEM format:

```
keytool -export -rfc -alias upload -file upload_certificate.pem -keystore keystore.jks
```

3. Reply to Google email and attach the upload_certificate.pem file.





Third

You will get an email from Google announcing when your new key will be valid. For me it were 2 days. Until then you won't be able to make updates to your app (we couldn't anyway). From this day you won't be able to upload APKs signed with the old key, only apps signed with the new upload key will be valid.

Fourth

Compile your game from GDevelop as usual and download your APK.

Export project to a standalone game

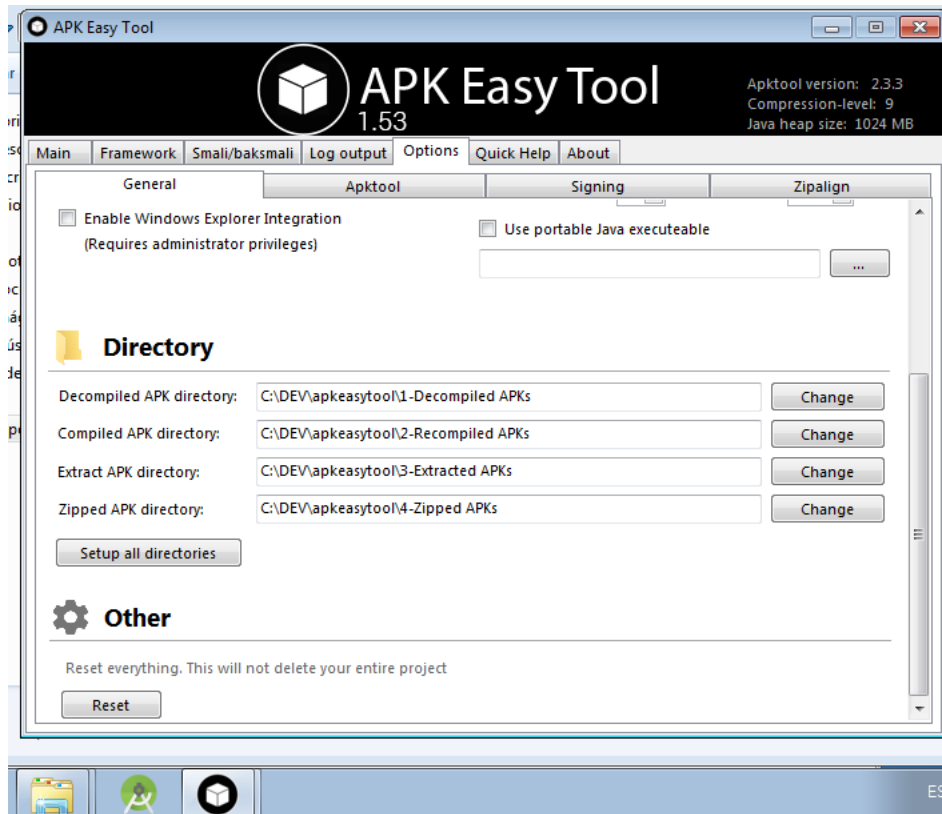
-  **Android (& iOS coming soon)**
Package your game for Android directly from GDevelop. iOS support is coming soon!
-  **Facebook Instant Games**
Package your game as a Facebook Instant Games that can be played on Facebook Messenger.
-  **Web (upload online)**
Upload your game online directly from GDevelop and share the link to players. Play to your game using your browser on computers and mobile phones.
-  **Windows/macOS/Linux (beta)**
Package your game as an app for Windows, macOS or Linux directly from GDevelop.

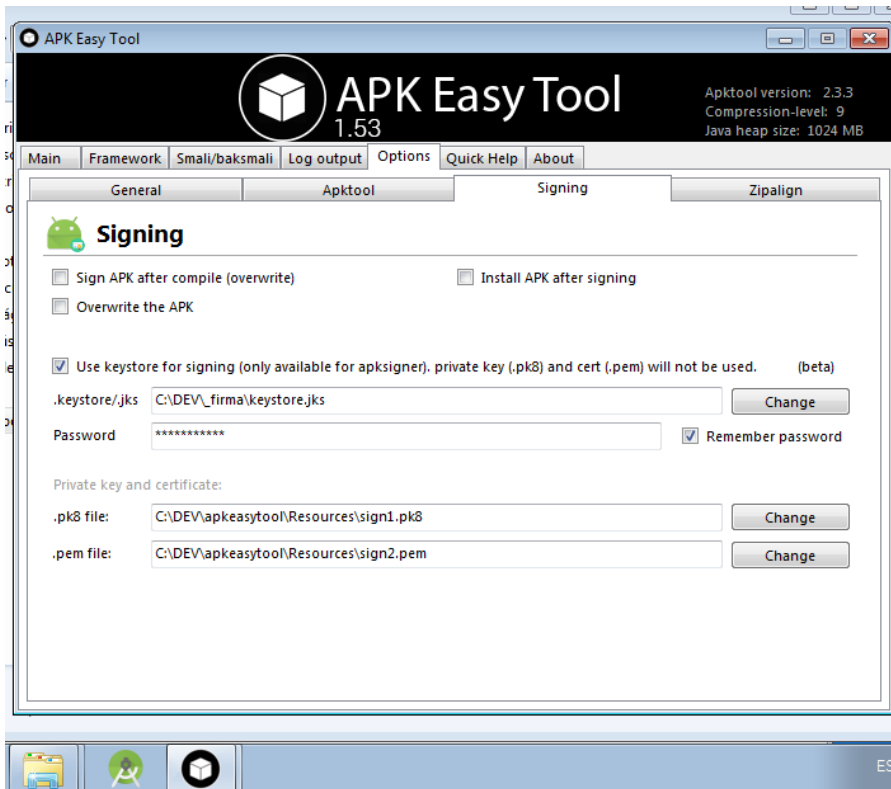
Advanced

Fifth

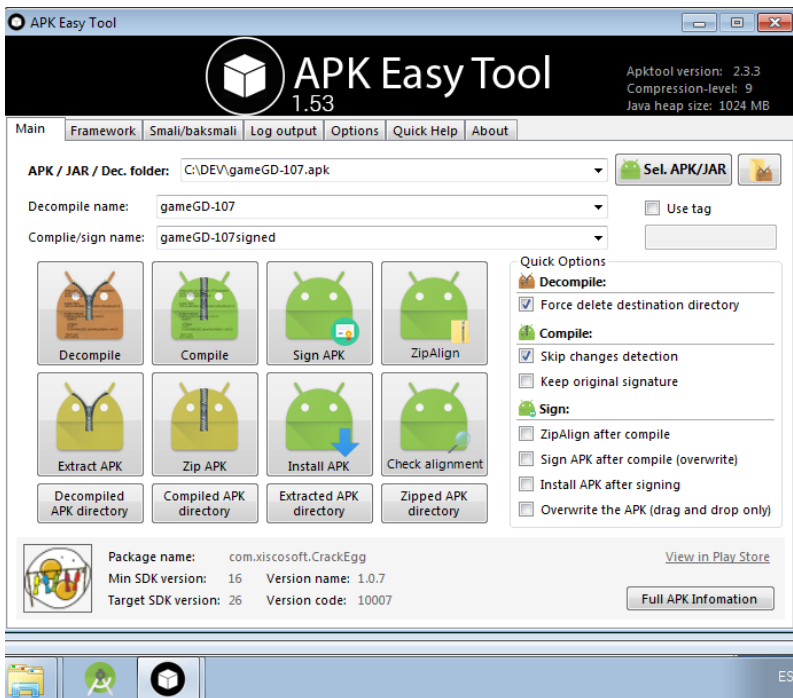
Download and unzip APK Easy Tool (requires .Net Framework 4.0 to run)

Run APK Easy Tool and setup the required folders and keystore location.

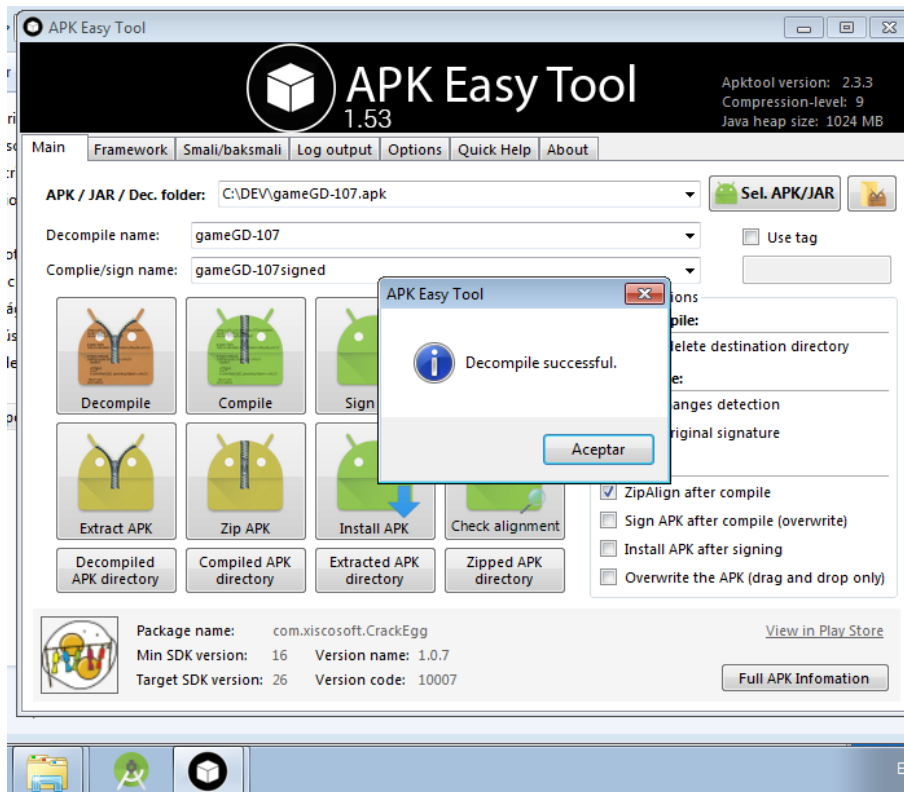




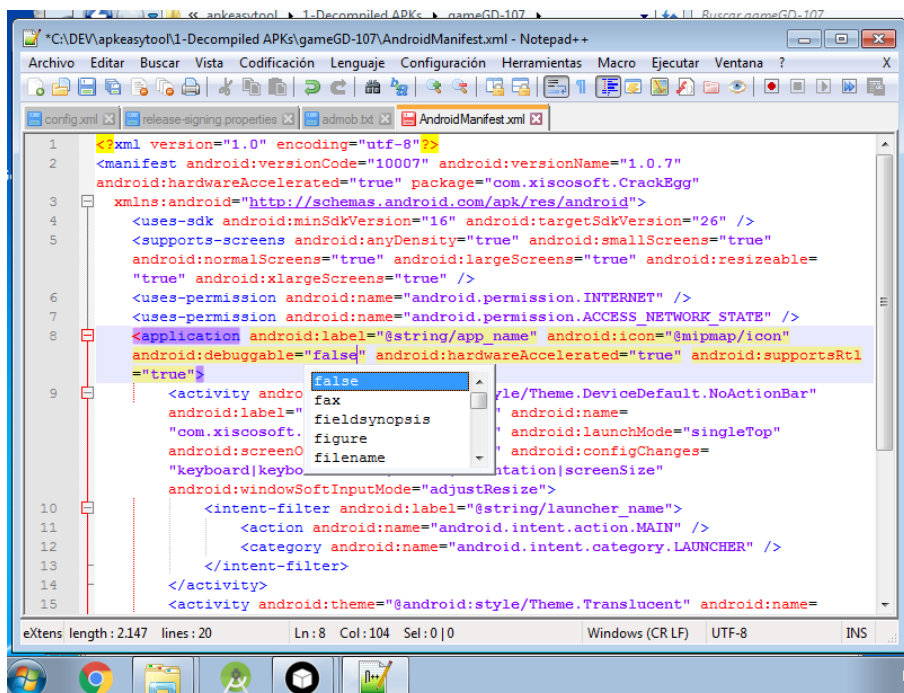
Select you APK



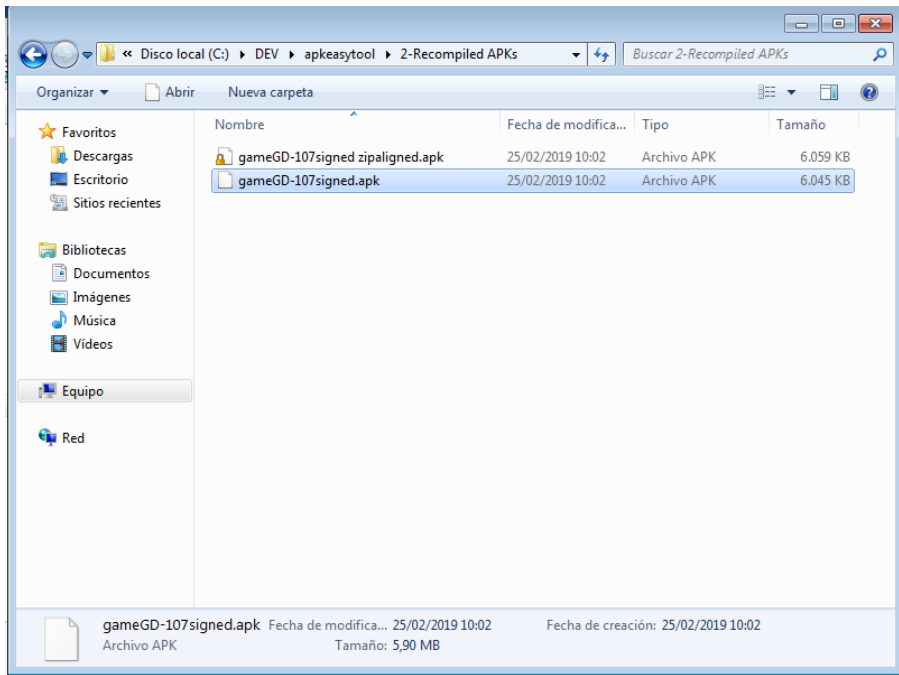
Decompile APK



Modify AndroidManifest.xml. Change android:debuggable from true to false



Compile APK, ZipAlign APK (I don't know why but Google complains about file not aligned) and finally Sign APK with your key!!. The file with the keylock is the signed apk.



That's all! Now upload to Google Play!!

Document Created by Xisco

xisdroid@gmail.com – 25/02/2019