**How to Sign and Publish a Phonegap App in the Google Play Store**

**Update**- I’ve now figured out an easier way to do this in one command instead of 5 ([Android Automation](http://www.adamwadeharris.com/android-automation))

So you’ve got your awesome app programmed, tested, added a cool icon, and you want to get it out to the world? It’s pretty easy to get android apps on your phone when testing, but you have to jump through a few hoops to actually get it in the Google Play store.

It was pretty hard to find a good guide out there on the Internet about what to do to finally get your app to submit to the Google Play Store for Android. The [official documentation](http://developer.android.com/tools/publishing/publishing_overview.html) is pretty wordy and somewhat vague. These instructions are specifically for apps built with Phonegap, because that’s what I used, but it might be helpful for other people as well.

**Make sure your app is good to go**

Make sure you’ve set your versionName in www/config.xml and versionCode  platforms/android/AndroidManifest.xml. The reason you have to set the versionName in config.xml inside the www folder is because phonegap will overwrite the versionName in the androidManifest file with whatever is in config.xml. Google Play won’t accept the app unless the versionCode is different than the previous versions in the store (preferably larger). versionCode is an integer value, so just increment it by 1 each time you submit regardless of whether it’s a major or minor update. versionName isn’t used for anything except for displaying to users and it’s a string so you can name it whatever you want. For example, you could set it to 1.0.3 while versionCode might be 3. (<http://developer.android.com/tools/publishing/versioning.html#appversioning>)

<manifest android:hardwareAccelerated="true" android:versionCode="3" android:versionName="1.0.3" android:windowSoftInputMode="adjustPan" package="com.compay.app" xmlns:android="http://schemas.android.com/apk/res/android">

Also, make sure you set debuggable to false in AndroidoManifest.xml in the application tag like this: android:debuggable=”false”

**Create a keystore file**

Create a keystore file and set a password. I won’t go into a lot of detail about how to actually do this. Just make sure you don’t lose this file. If you lose it, and you have to create a new one, then it will become a new app when you try to add it to the Google Play Store. (<http://developer.android.com/tools/publishing/app-signing.html#cert>)

Always use a different keystore file for each app because it’s your private key for uploading apps to the store. If you ever decide to transfer your app to another developer, you’ll have to give them the keystore file, and if you also use that keystore for other apps, then you have a security issue. (<http://developer.android.com/tools/publishing/app-signing.html#secure-key>)

Put the keystore file somewhere on your computer. It doesn’t really matter where.

**Tell ant where your keystore file is for this app**

Then you just need to tell ant where the keystore file is by going to your android project folder (For phonegap it’s in platforms/android) and create an ant.properties file and put the following in it:

key.store=/Users/username/Documents/path/to/my-release-key.keystore

key.alias=app\_name

Where key.store equals the path to the keystore file starting at the C Drive if you’re on windows and it’s a relative path starting from the location of the ant.properties file if you’re on mac (example: key.store=../../../../lifeUnlimited-release-key.keystore), and key.Alias is whatever you setup when you created the keystore.

**Build your app**

Open up the command prompt, and navigate to your project and run phonegap build.

phonegap build android

in platforms/android/bin you should have:

AppName.ap\_

AppName.ap\_.d

AppName-debug.apk

AppName-debug-unaligned.apk

AppName-debug-unaligned.apk.d

**Sign in release mode**

Then navigate to the android directory and run ant release:

cd platforms/android

ant release

It might prompt you for your keystore password and the password for the alias ‘app\_name’. Enter your keystore password for both of them.

In platforms/android/bin you should now also have release versions of the app:

AppName-release.apk AppName-release-unaligned.apk AppName-release-unsigned.apk AppName-release-unsigned.apk.d

Now move into the bin directory and run these jarsigner commands:

cd bin

jarsigner -verbose -sigalg SHA1withRSA -digestalg SHA1 -keystore viplusKey.keystore Viplus-release-unsigned.apk Viplus

Enter your keystore password

jarsigner -verify -verbose -certs Viplus-release-unsigned.apk

If you get a warning like this ignore it: Warning: This jar contains entries whose certificate chain is not validated.

Then run this zipalign command to create the final apk file:

zipalign -v 4 Viplus-release-unsigned.apk Viplus.apk

it will say: Verification successful

And your final apk (AppName.apk) will be created in the bin directory.

(<http://developer.android.com/tools/publishing/app-signing.html#releasemode>)

Then you can upload to Google Play.

I hope this helps. Let me know if you have any questions.