

## Obtaining a Google Maps API Key

v1.0

By GoNorthWest

15 December 2011

If you are creating an Android application that uses maps in it (which is a very cool feature to have!), you're going to need a Google Maps API key from, you guessed it, Google! The process can be a bit confusing, and is documented here:

<http://code.google.com/android/add-ons/google-apis/mapkey.html>

Let's take a few minutes and walk through the process step-by-step. We'll do this on a Windows XP box, but the steps are essentially the same on a Mac or Linux box. You're going to do the following:

- Get the MD5 Fingerprint of your Signing Certificate
- Get the MD5 Fingerprint of your SDK Debug Certificate
- Register your fingerprints (not the ones on your hand!) with the Google Maps Service
- Add the Maps API Keys to your application

Before you go any further, you might want to read a bit about Signing Android Apps, which I have written up in this document:

[http://dl.dropbox.com/u/4993711/Signing\\_Android\\_Apps\\_v1.0.pdf](http://dl.dropbox.com/u/4993711/Signing_Android_Apps_v1.0.pdf)

That document talks about your debug and release keystore, how to set them up, and how to use them to sign your apps for development and distribution. We'll go through part of that process again here, but there are more details in the other document.

Let's begin by making sure the appropriate keystores are in place. When you installed your Android SDK, it automatically created a debug.keystore, which you use for, you guessed it, debug signing! The default locations for the debug.keystore appear to be these locations for the following platforms:

**Windows:** C:\Documents and Settings\\.android

**Mac:** \Users\\android

If you don't find anything at these locations on your computer, just do a search for 'keystore' and you should find what you're looking for.

The debug.keystore ships with the following default parameters:

- Keystore name: "debug.keystore"
- Keystore password: "android"
- Key alias: "androiddebugkey"
- Key password: "android"

CN: "CN=Android Debug,O=Android,C=US"

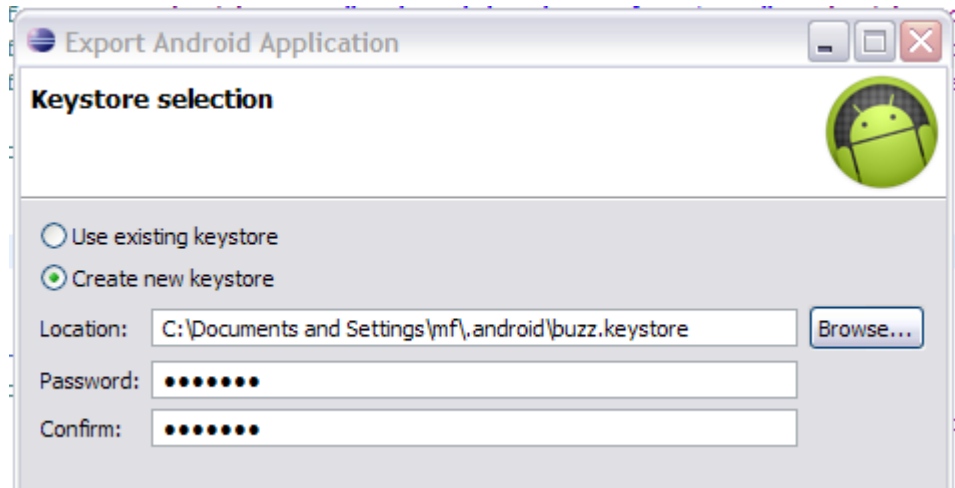
If, for some reason, your debug.keystore does not exist, or become corrupted, you can create a new one by doing a build of your application, and it will automatically be created.

In addition to the debug.keystore, you'll also need a release keystore created in order to get your Google Maps API key. There are basically two ways to create it – via the command line, and by building your application for release, and creating the keystore at that time.

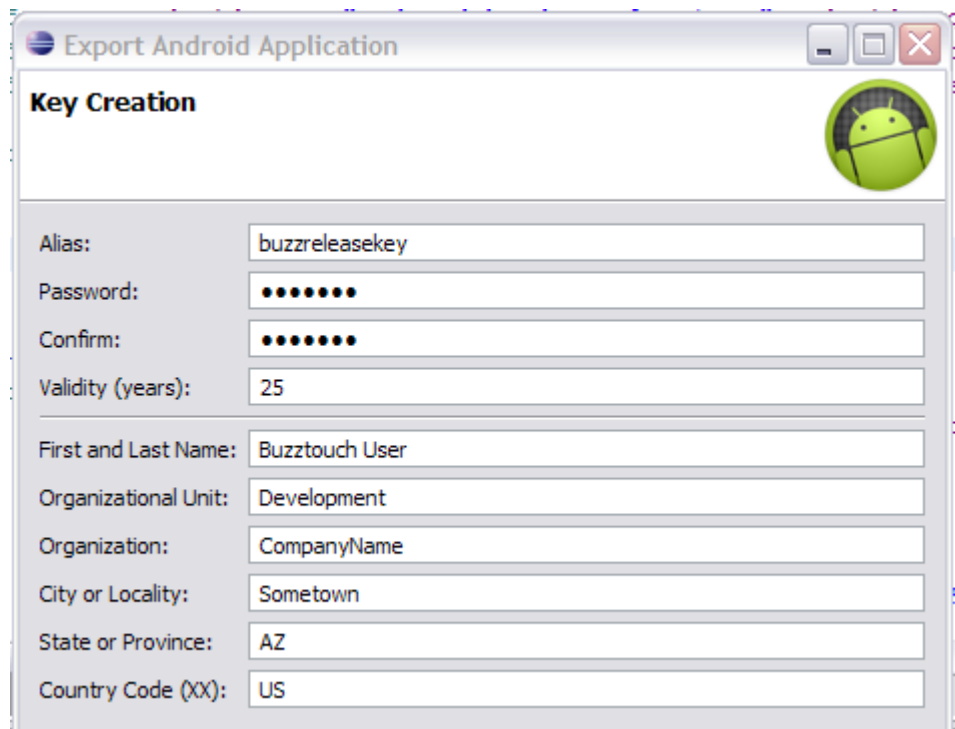
1. **Command Line Method.** Open a command prompt (Windows) or terminal session (Linux/Mac) and issue the following command:

```
a. keytool -genkey -v -keystore you-decide.keystore -alias  
alias_name -keyalg RSA -keysize 2048 -validity 10000
```

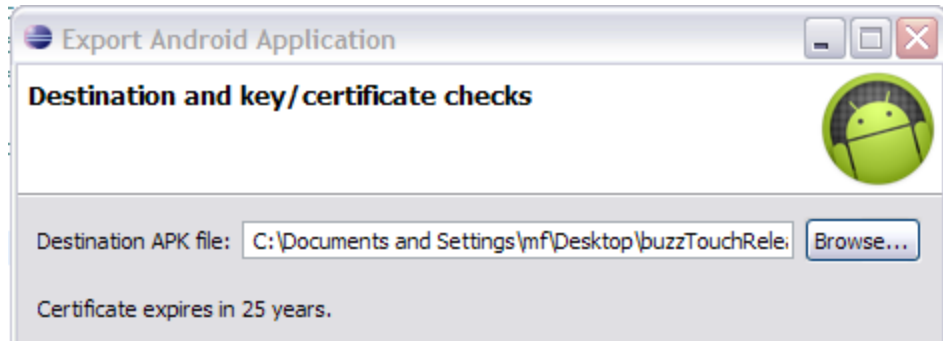
2. **Release Build Method.** With this method, you will build your app in release mode, and create the keystore along the way.
  - a. Make sure that **android:debuggable="false"** is set in your AndroidManifest.xml file. This file is located in the root of your project folder that you got from Buzztouch.
  - b. Highlight your project in Package Explorer, right-click, and select **Export**.
  - c. Select **Android...Export Android Application** and then **Next**.
  - d. The Project name should already be filled in, since you are working directly from that project. Hit **Next**.
  - e. Select **Create new keystore** and fill in a keystore location and passwords (keep them strong!).



- f. Provide the requested information. Use a strong password, and pick a nice validity duration (I used 25 years). This can be the release key you use forever, and it identifies you as the signer of the application. So, keep it legit, and keep it secure! Hit **Next**.



- g. Now you'll want to select a destination for your .apk file. As always, make sure it's somewhere you remember. You can also give your package a name at this step.



- h. Hit **Finish**, and the release package for your application will be built! You'll also now have a release keystore good for 25 years, and you can use it to request your Google Maps API key!

So, this has been a bit of a long process to get ready to request our MD5 fingerprints, but we're ready for that now. This is a command line procedure, and is accomplished as follows:

1. **Get the MD5 Fingerprint of your Signing Certificate.** From the command line or terminal session, enter the following command.

```
a. keytool -list -alias your_release_key_alias -keystore
your_release.keystore -storepass your_password -keypass
your_password
```

The result of this command should look something like this:

Certificate fingerprint (MD5): 91:EF:A3:23:D9:4C:C2:1D:7F:AB:9C:49:DC:36:D9:3B

2. **Get the MD5 Fingerprint of your SDK Debug Certificate.** From the command line or terminal session, enter the following command.

```
a. keytool -list -alias androiddebugkey -keystore
debug.keystore -storepass android -keypass android
```

The result of this command should look something like this:

Certificate fingerprint (MD5): G1:8E:31:07:09:22:7F:A1:A1:47:87:10:90:C6:53:6C

At this point, you have all the information you need to request your Google Maps API debug and release keys! Follow these steps, and you'll be up and running soon!

1. Go to <http://code.google.com/android/maps-api-signup.html>
2. If you don't have a Google account, use the link on the page to set one up.

3. Read the Android Maps API Terms of Service carefully. If you agree to the terms, indicate so using the checkbox on the screen.
4. Paste the MD5 certificate fingerprint of the certificate that you are registering into the appropriate form field.
5. Click "Generate API Key"

The website will generate a code that looks something like this:

0oYoJTnGQYNBciyMASfbutxEAZBqAsEjshrbplw

Get an API key for both your debug and release MD5 fingerprints and write them down for safe keeping. I created a file that contains my MD5 fingerprints (along with the command I used to get them) and the corresponding API key. That way I have all the info I need in one place.

You're now ready for the last step, which is configuring your Android application with the correct keys. Double check the Readme.txt file that comes with your Android project package from Buzztouch, but at the time of this writing, the location for placing your keys is in the res\values\string.xml file. If you open it up, you'll see the following two lines near the top:

```
<string name="googleMapsAPIKeyDebug">yourGoogleMapsKeyForDebug</string>
```

```
<string name="googleMapsAPIKeyRelease">yourGoogleMapsKeyForRelease</string>
```

Paste your keys in the appropriate line, and you're good to go!

Comments? Post them in the forum or email me at [MrkFleming@gmail.com](mailto:MrkFleming@gmail.com).

Revision Log			
v1.0	12/15/11	Initial release of document.	GoNorthWest