

Signing Android Apps

v1.0

By GoNorthWest

15 December 2011

If you are creating an Android application, Google requires that those applications are signed with a certificate. This signing process is significantly easier than what Apple makes iOS developers go through, as Android developers can generate their own certificates for signing applications, using commonly available tools like *keystore*, which comes with your Java JDK. Google requires that **all** applications be signed, be it a development or release application. Each requires a different certificate, so let's take a look at each.

Signing For Development/Debug

When you installed your Android SDK, you got with it a `debug.keystore` that can be used to sign your application for debug purposes. The default locations for the `debug.keystore` appears to be these locations for the following platforms:

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

Mac: `\Users\\.android`

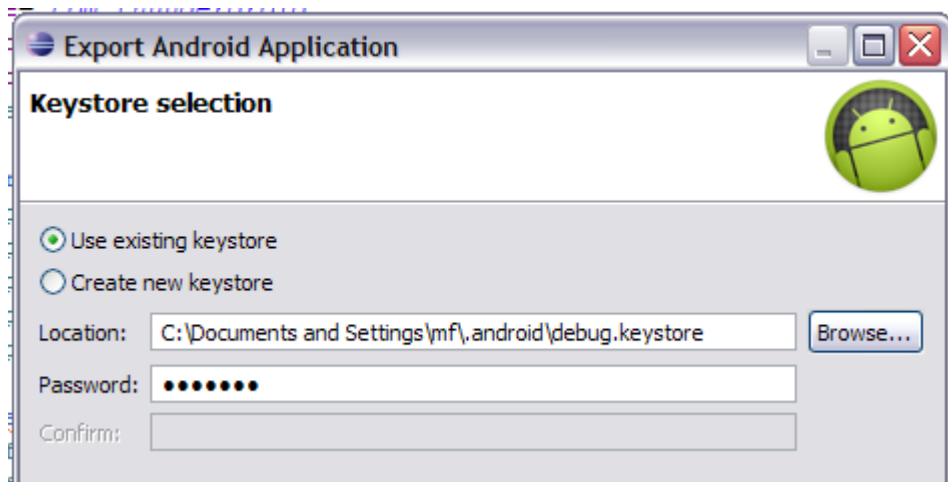
If you don't find anything at these locations on your computer, just do a search for 'keystore' and you should find what you're looking for.

The `debug.keystore` ships with the following default parameters:

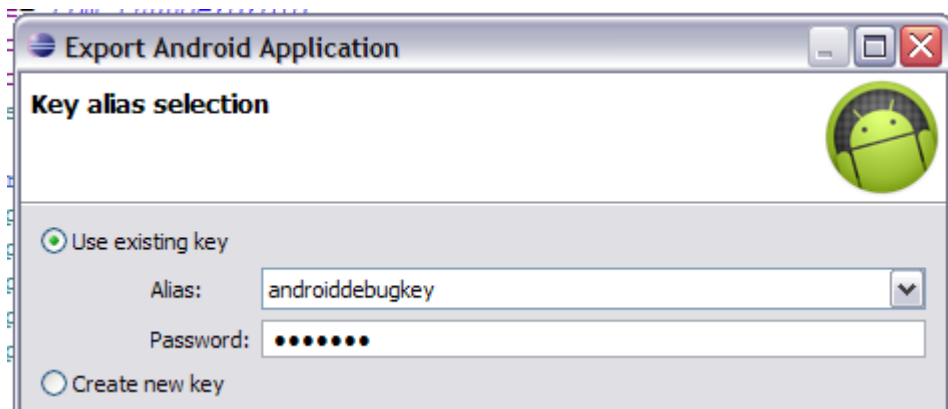
- Keystore name: "debug.keystore"
- Keystore password: "android"
- Key alias: "androiddebugkey"
- Key password: "android"
- CN: "CN=Android Debug,O=Android,C=US"

This information becomes important when you go to sign your application for debug. Once you have your application ready for testing, and you want to create a debug package, you can follow these steps:

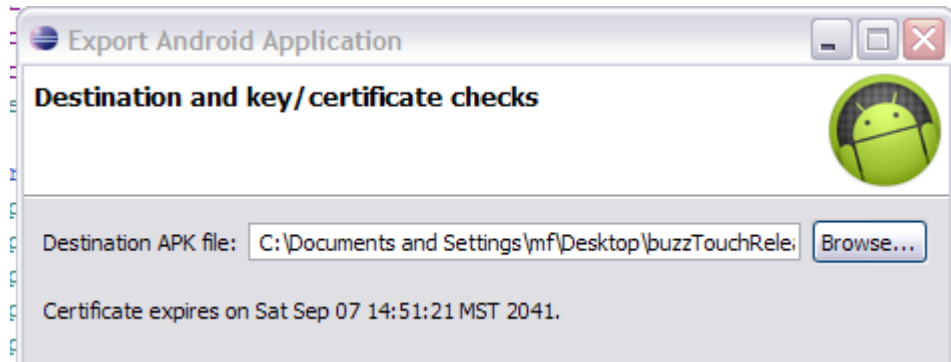
- Make sure that **android:debuggable="true"** is set in your AndroidManifest.xml file. This file is located in the root of your project folder that you got from Buzztouch.
- Highlight your project in Package Explorer, right-click, and select **Export**.
- Select **Android...Export Android Application** and then **Next**.
- The Project name should already be filled in, since you are working directly from that project. Hit **Next**.
- Select **Use Existing Keystore** and hit Browse to navigate to your debug.keystore, and select it. Then enter your password (android), and hit **Next**.



- Select the **androiddebugkey** alias and use **android** as the password. Select **Next**.



- 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. Notice that in this example, the expiration date for the debug key is in 2041. So no worries!



- Hit **Finish**, and the debug package for your application will be built! Congratulations on building and signing your Android application! You can now distribute it in whichever way you want (email, the web, etc), so long as it doesn't go into a store. Debug apps are not allowed in stores (meaning, if `debuggable="true"` is set, it won't get accepted).

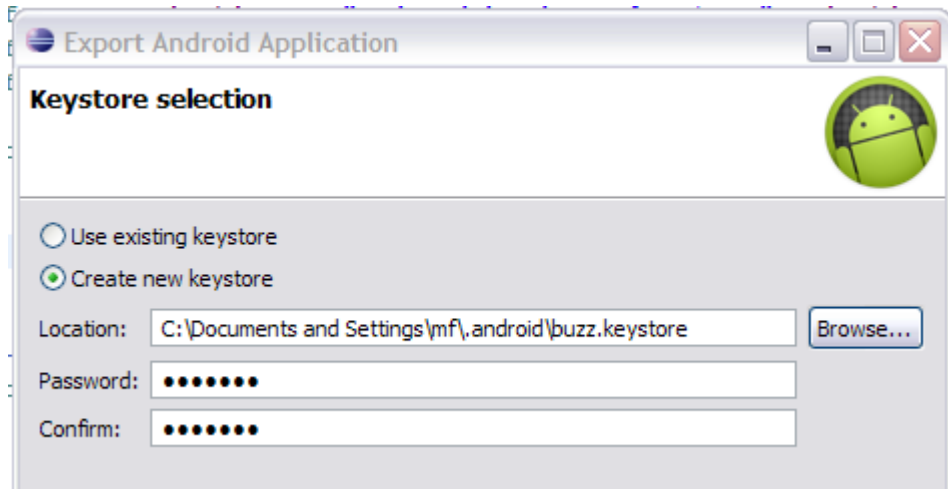
Keep in mind that you don't have to use the default `debug.keystore` that comes with your Android SDK. I'm not entirely sure why you would want to create your own...perhaps you want people to specifically know it came from you, but the option exists to create it during this same process. Just select **Create New Keystore** instead of using the existing keystore, and you'll be prompted through the process.

Signing For Release/Distribution

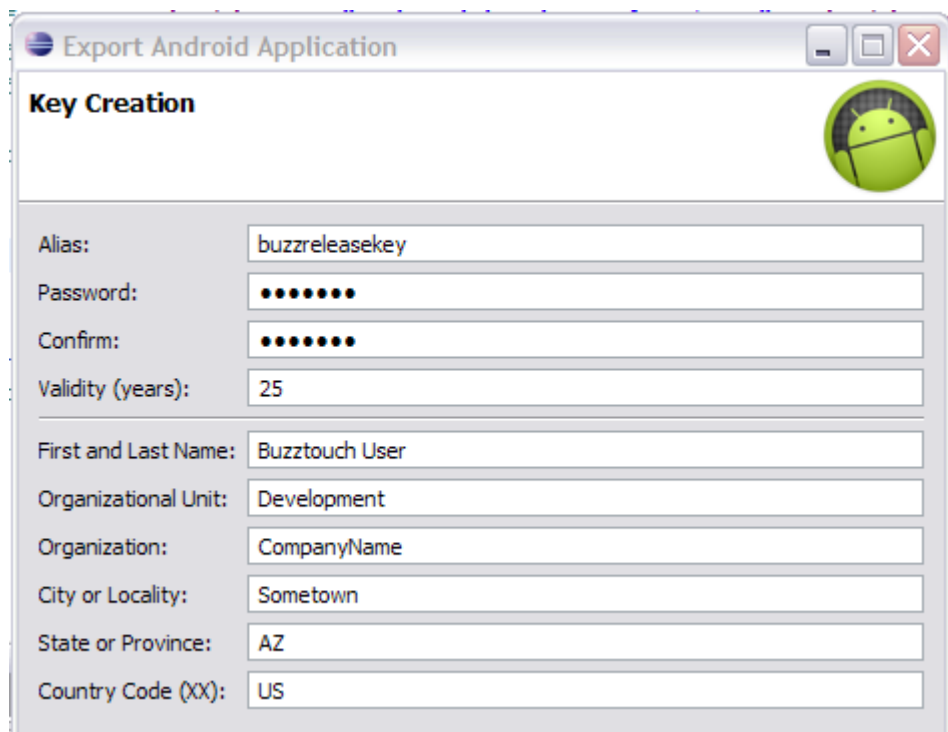
OK, you've done all your testing, everything is in order, and you're ready to put your store in one of the various marketplaces. Cool! The process for signing your application for release/distribution is the same as for debug/development, but in this case you use a `release.keystore` instead. You can create a `release.keystore` during the export process.

- 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.
- Highlight your project in Package Explorer, right-click, and select **Export**.
- Select **Android...Export Android Application** and then **Next**.
- The Project name should already be filled in, since you are working directly from that project. Hit **Next**.

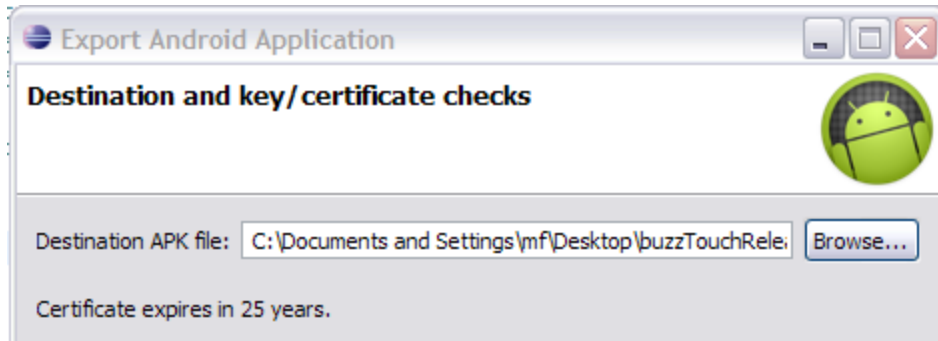
- Select **Create new keystore** and fill in a keystore location and passwords (keep them strong!).



- 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**.



- 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.



- Hit **Finish**, and the release package for your application will be built! Congratulations on building and signing your Android application! You are now ready to officially place it in the Marketplace!

That pretty much sums up the Android signing process! The two keystores you created, debug and release, will be used again in a different tutorial on getting a Google Maps API key. You should make a backup of our release key, as it identifies you as the author of your app, and should not be compromised.

For more information, see <http://developer.android.com/guide/publishing/app-signing.html>

Comments? Post them in the forum or email me at MrkFleming@gmail.com.

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