

# Signing and Packaging Android Applications

*Updated March, 2014*

This document will show you how to sign and package Android applications. This process is required to publish your applications in Google Play (the Android Market).

## Copyright Notices:

*Java, Java Development Kit (JDK)* and related terms are all copyright by Oracle Corporation. Please see <http://www.oracle.com> for more details. *Android, Android Software Development Kit, and Android Development Tools* and related terms are all copyright by Google, Inc.

This document, *TeenCoder™: Java Programming, TeenCoder™: Android Programming* and related terms are copyright by Homeschool Programming, Inc. This document may not be transmitted or reproduced without written permission except under terms of your purchased course license.

## Disclaimer:

Homeschool Programming, Inc, and their officers and shareholders, assume no liability for damage to personal computers or loss of data residing on personal computers arising due to the use or misuse of this course material. Always follow instructions provided by the manufacturer of 3rd party programs that may be included or referenced by our courses.

Note: All red highlights below are added for emphasis and are not present on the actual pages.

## Let's Get Started!

All Android applications must be digitally signed in order to be placed in the Google Play market. This certification provides some assurance to your users that you are a trusted programmer.

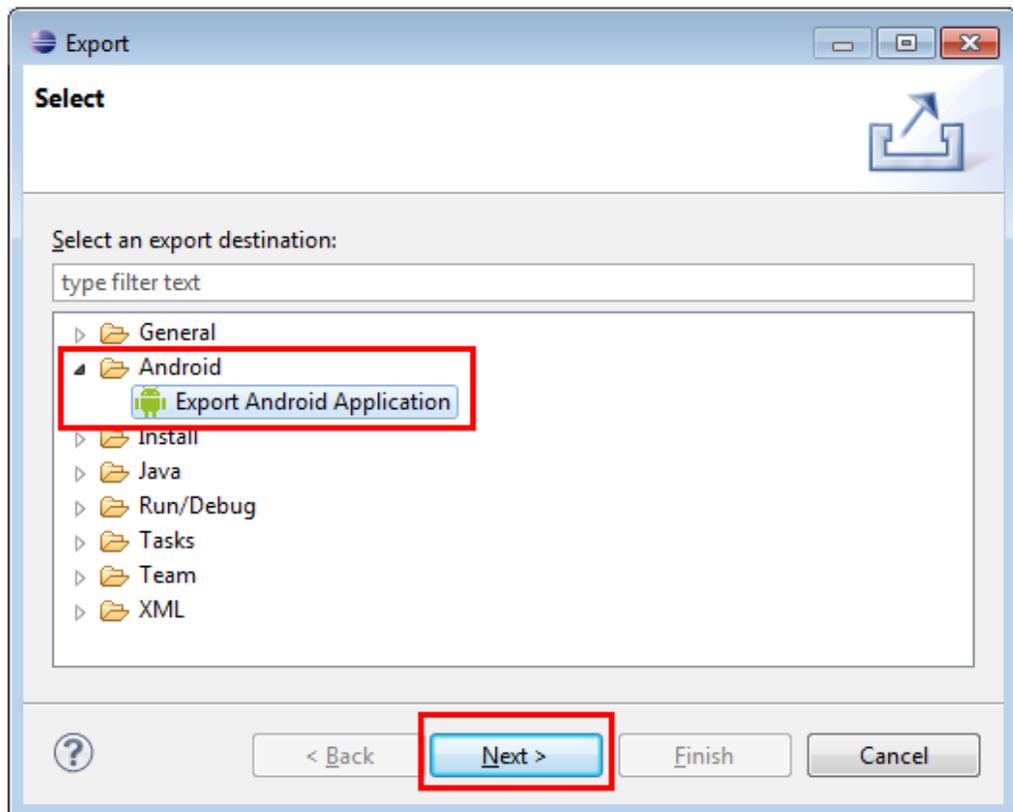
The Eclipse software (with the ADT plug-in installed) makes the process of digitally signing your Android applications very simple! You will end up with an “.apk” file that contains your entire application. The “.apk” will be securely signed with a unique key that you create.

The digital signature consists of a public and private key value. The public key value remains on the application. The private key value remains with the developer. When the user installs an update, the two keys are compared to make sure they match. All Android applications are capable of being self-signed, which means you can create your own digital signature.

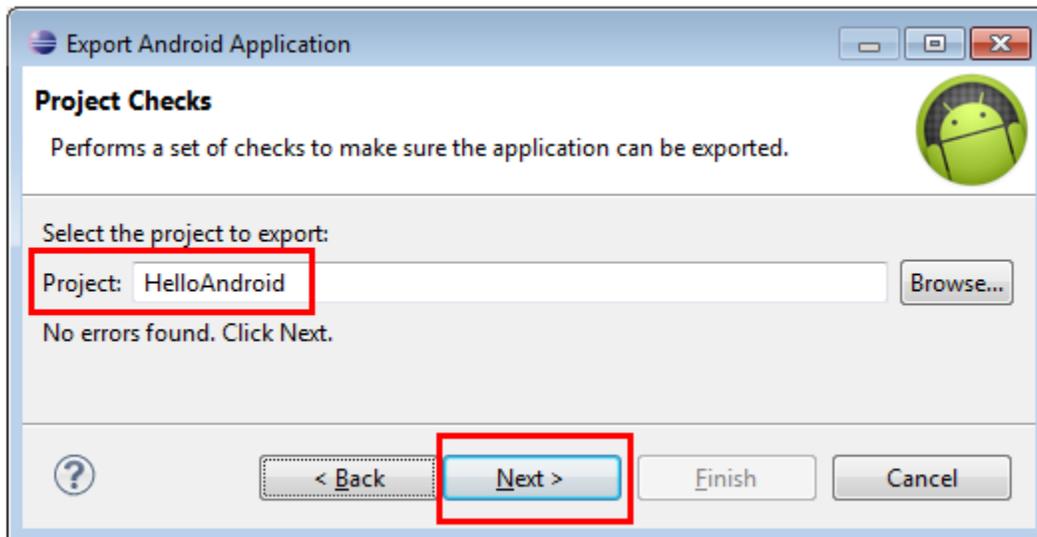
To start this process, open Eclipse and highlight your project in the Package Explorer. Then select “File → Export” from the Eclipse menu. You should see a screen like the one below:

When you expand the “Android” item, you will see the option “Export Android Application”. This is the option that will allow us to prepare our application for publishing.

Click on the “Next” button to begin this process.



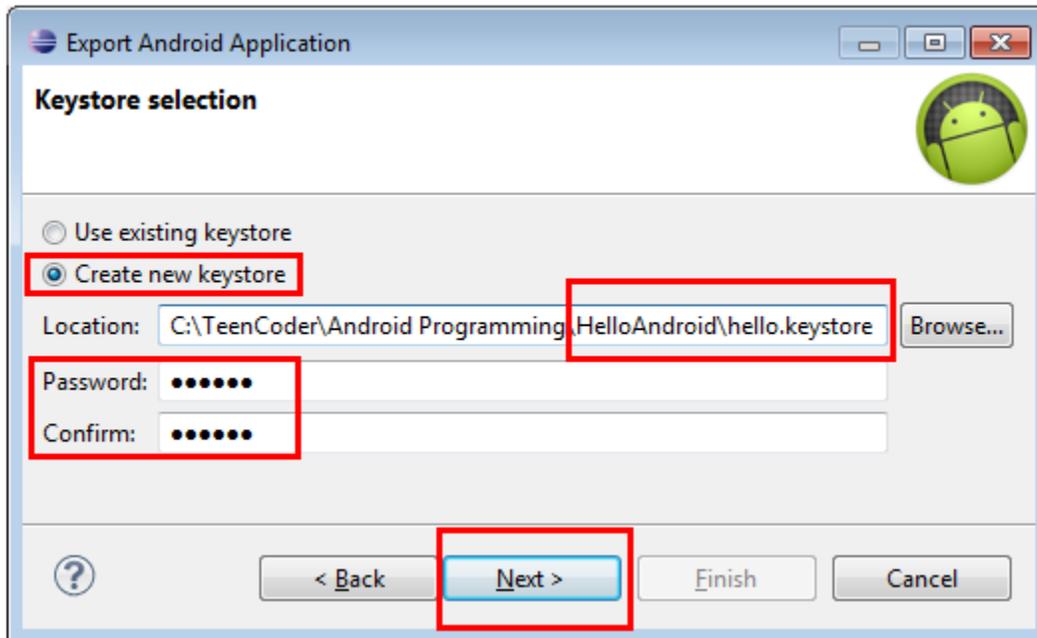
You should see the following screen appear:



On this screen, you can see that we are allowed to choose the project that we want to “Export”. The project that was highlighted in the Package Explorer should appear in the input box. However, if you needed to export a different project, you can find that project now by clicking on the “Browse” button. Once you have the correct project listed in the box, click on the “Next” button to continue.

## Creating Your Digital Signature

On this screen, Eclipse is giving us the opportunity to create a new *keystore*, which will contain your digital signature. Make sure you choose the “Create new keystore” radio button is selected and then take a look at the next few fields.

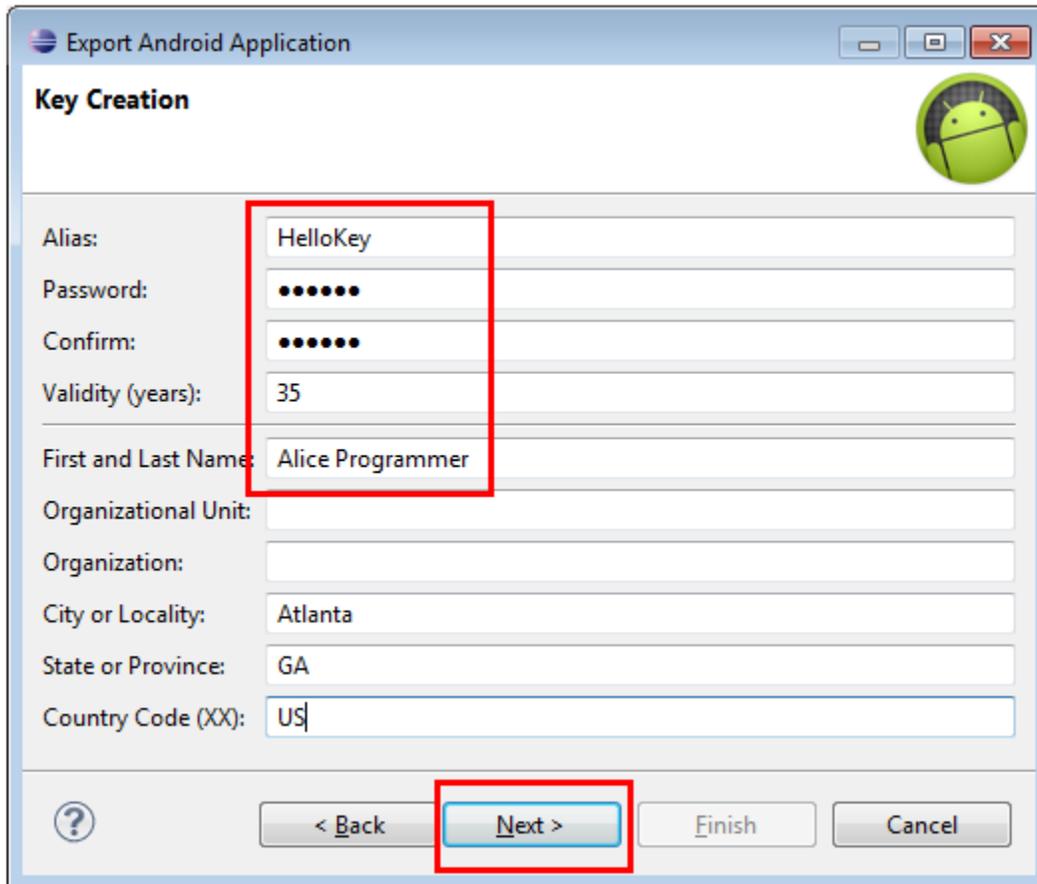


The first field asks you to choose a location and create a filename for your new keystore file. You can place this file anywhere you would like, although storing it in the project folder makes it easier to find later. You will also need to give your keystore file a name. Again, you can use any name you want, but a good format is to use your application name and then the extension “.keystore”.

The second and third fields will allow you to set an additional password for your keystore value. You can choose any value you want. Just remember your password in case you ever need it again!

Once you have filled in these fields, click on the “Next” button to continue.

On this screen, you will have to enter some more information for your digital signature. The “Alias” can be any text you want – we used our application name plus the word “Key” for this value.



The screenshot shows a dialog box titled "Export Android Application" with a sub-section "Key Creation". The dialog contains several input fields: "Alias" (containing "HelloKey"), "Password" (masked with dots), "Confirm" (masked with dots), "Validity (years)" (containing "35"), "First and Last Name" (containing "Alice Programmer"), "Organizational Unit", "Organization", "City or Locality" (containing "Atlanta"), "State or Province" (containing "GA"), and "Country Code (XX)" (containing "US"). At the bottom, there are four buttons: a help icon, "< Back", "Next >" (highlighted with a red box), "Finish", and "Cancel".

The password value can be different than the previous screen’s password, although you can keep these passwords the same.

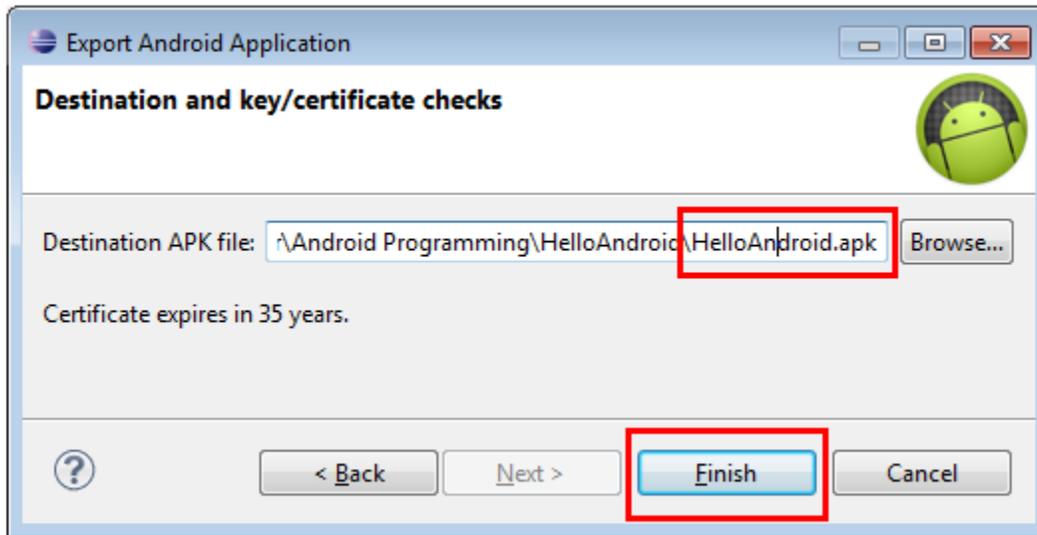
The “Validity” box will contain the number of years that this key value is valid. You can choose any value for this field, although the longer, the better!

In the second section on the screen, you are asked to fill-in some information about yourself or your company. You must fill-in at least one field in this section in order to be able to continue.

When you are ready, click the “Next” button to continue the process.

## Creating your Application .APK File

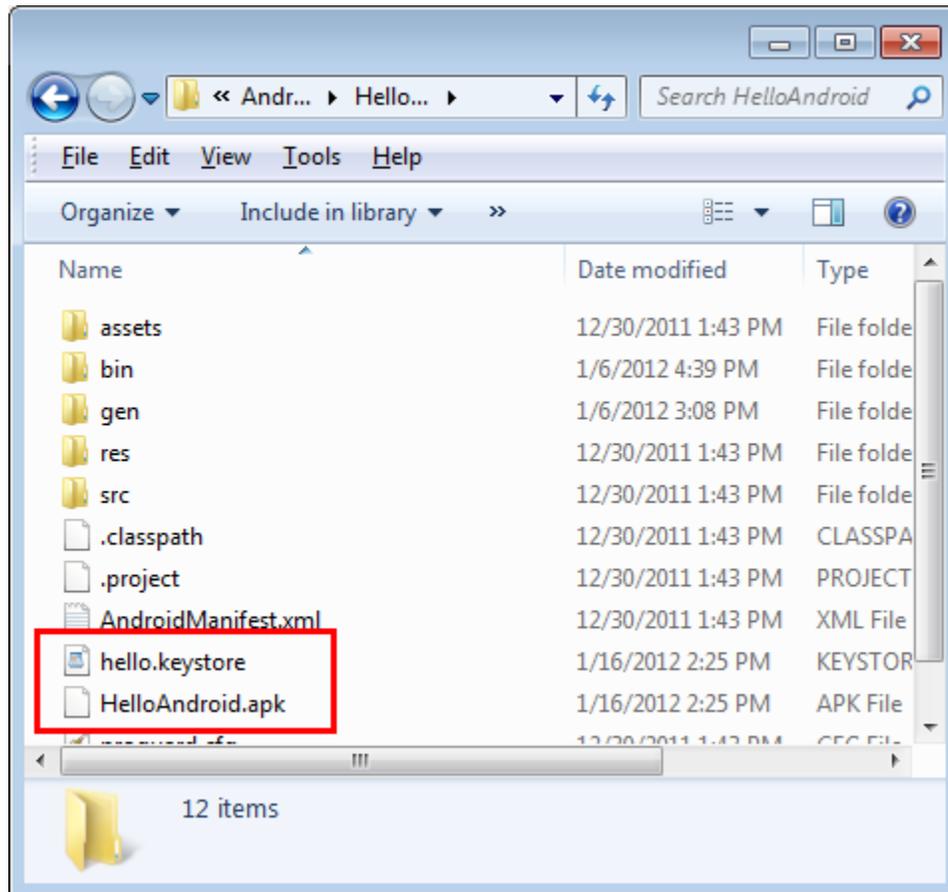
In this screen, you will choose a location for the final “apk” file for your application. Again, you can choose any output directory that makes sense for you, although to keep it simple, you might want to output the “apk” file to your project directory.



When you have chosen a location for your “apk” file, click on the “Finish” button to complete the process.

Once you click on the “Finish” button, you will not necessarily see any signs that your key creation and program signing worked properly. This is okay! In this case, no news is good news.

To verify that the process worked, navigate to the directory that you specified during the signing process and look for your keystore and apk files:



Here, you can see that we have created the “hello.keystore” and “HelloAndroid.apk” files successfully. Your application’s APK file is now ready for publishing!