

Getting started with Android

Before we begin, there is a prerequisite, which is to plug the Android device into your computer, and load the drivers for the OS. In writing this article, I was using Windows XP, 7 (64 bit) and 8 (64 bit) for the computer and OS. The Android devices I used were a Samsung Galaxy S III and a Nexus 7 tablet.

In all I tried 5 computers, of which on 2 could I correctly install the Samsung Galaxy S III and Nexus 7 tablet. I went on the Internet to find solutions, the most popular theory hat to do with the MTP driver (Media Transfer Protocol). However none of the solutions worked.

I also had an HP Slate 7 tablet, after searching the internet for a windows driver, I discovered, much to my chagrin, there is no driver for this tablet, so don't go there!

In conclusion, if you can't get your device's drivers to load, you will only be able to run in a simulated device mode.

Installation

There is a very good tutorial on installing Eclipse and the Android SDK on the internet. Here is the main link: <http://developer.android.com/training/basics/firstapp/index.html>. It walks you through building your first app and tells you what you need to install so you can build your first app.

Most of it is accurate but there are a few errors. However I do recommend you read the pages so that you get the most out of this but you can skip. Here is a synopsis with corrections.

Starting here: <http://developer.android.com/sdk/index.html>. It tells you to do these 3 things:

1. Download the Android SDK.
2. Install the ADT plugin for Eclipse (if you'll use the Eclipse IDE).
3. Download the latest SDK tools and platforms using the SDK Manager.

So, Step 1. Download the Android SDK. This includes Eclipse and Java all in one. Go to this link:

<http://developer.android.com/sdk/index.html>, and click on the button "Download the SDK ADT bundle for Windows", and follow the on screen prompts for T's & C's, 32 or 64 bit (the article suggests you use the 32 bit version even if you have a 64 bit IDE)

You will download a 'zip' file. Once downloaded, unpack it using the Windows Extract all feature or similar extraction tool. Copy the entire contents of the folder to some directory on your computer where you will launch eclipse. On my computer I placed it:

C:\adt-bundle-windows-x86-20130729\adt-bundle-windows-x86-20130729\eclipse.

Quoting the instructions, "That's it! The IDE is already loaded with the Android Developer Tools plugin and the SDK is ready to go."

However there is a cautionary note: Do not move any of the files or directories from the `adt-bundle-<os_platform>` directory. If you move the `eclipse` or `sdk` directory, ADT will not be able to locate the SDK and you'll need to manually update the ADT preferences.

Step 2, the ADT Plugin: <http://developer.android.com/sdk/installing/installing-adt.html>

The page title is a misnomer since it is called "Installing the Eclipse Plugin" but what you are really doing is installing the ADT plugin. The installation instructions are correct so I'll plagiarize here:

1. Start Eclipse, then select Help > Install New Software.
2. Click Add, in the top-right corner.
3. In the Add Repository dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*: <https://dl-ssl.google.com/android/eclipse/>
4. Click OK.

If you have trouble acquiring the plugin, try using "http" in the Location URL, instead of "https" (https is preferred for security reasons).

5. In the Available Software dialog, select the checkbox next to Developer Tools and click Next.
6. In the next window, you'll see a list of the tools to be downloaded. Click Next.
7. Read and accept the license agreements, then click Finish.

If you get a security warning saying that the authenticity or validity of the software can't be established, click OK.

8. When the installation completes, restart Eclipse.

The next section of that webpage, 'Configure the ADT Plugin' can be ignored since it is done automatically by the steps above. You won't even see the Welcome statement discussed there, but it does take you to the next and final step, adding platforms and packages.

Step 3, Adding platforms and packages: <http://developer.android.com/sdk/installing/adding-packages.html>

The instructions here are correct here as well, with one exception. You can also open the SDK Manager from within the Eclipse IDE. Just click on Window >> SDK Manager and continue on. Continuing with my plagiarizing:

The Android SDK separates tools, platforms, and other components into packages you can download using the Android SDK Manager. The original SDK package you've downloaded includes only the SDK

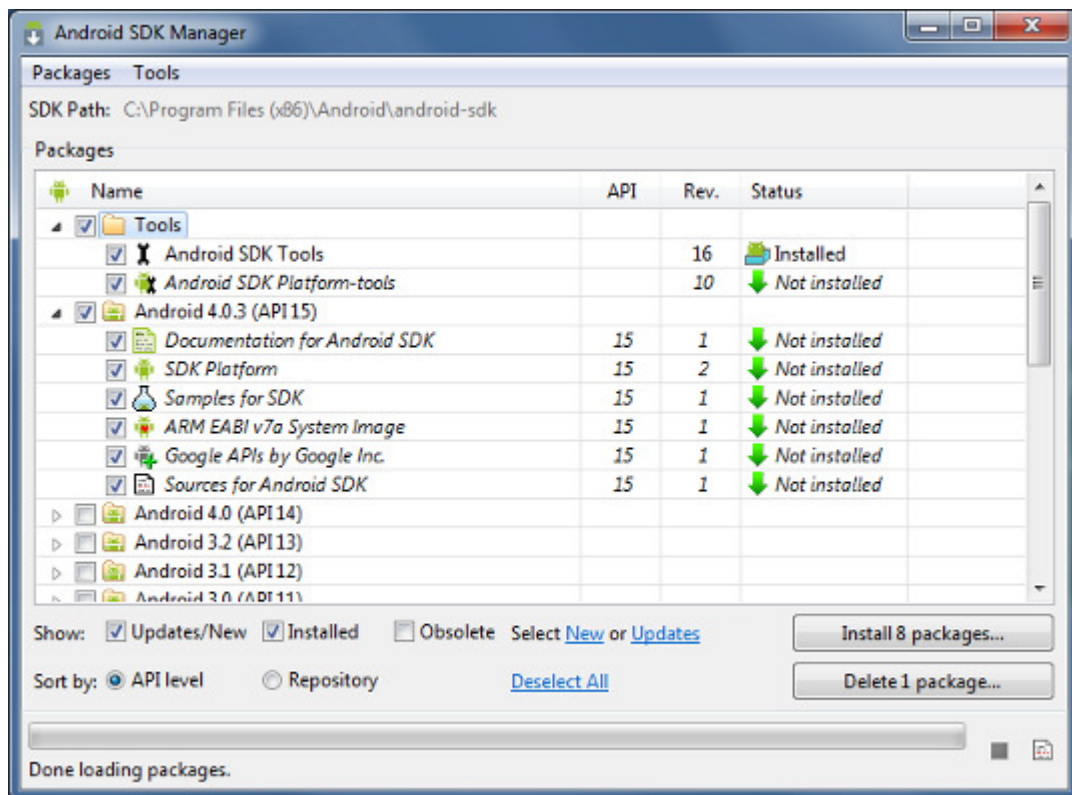
Tools. To develop an Android app, you also need to download at least one Android platform and the latest SDK Platform-tools.

1. Launch the SDK Manager.

If you've used the Windows installer to install the SDK tools, you should already have the Android SDK Manager open. Otherwise, you can launch the Android SDK Manager in one of the following ways:

- On Windows, double-click the SDK Manager.exe file at the root of the Android SDK directory.
 - On Mac or Linux, open a terminal and navigate to the tools/ directory in the Android SDK, then execute `android sdk`.
2. The SDK Manager shows all the SDK packages available for you to add to your Android SDK. As a minimum configuration for your SDK, we recommend you install the following:
 - The latest Tools packages (check the Tools folder).
 - The latest version of Android (check the first Android folder).
 - The Android Support Library (open the Extras folder and check Android Support Library).

Once you've chosen your packages, click Install. The Android SDK Manager installs the selected packages into your Android SDK environment.

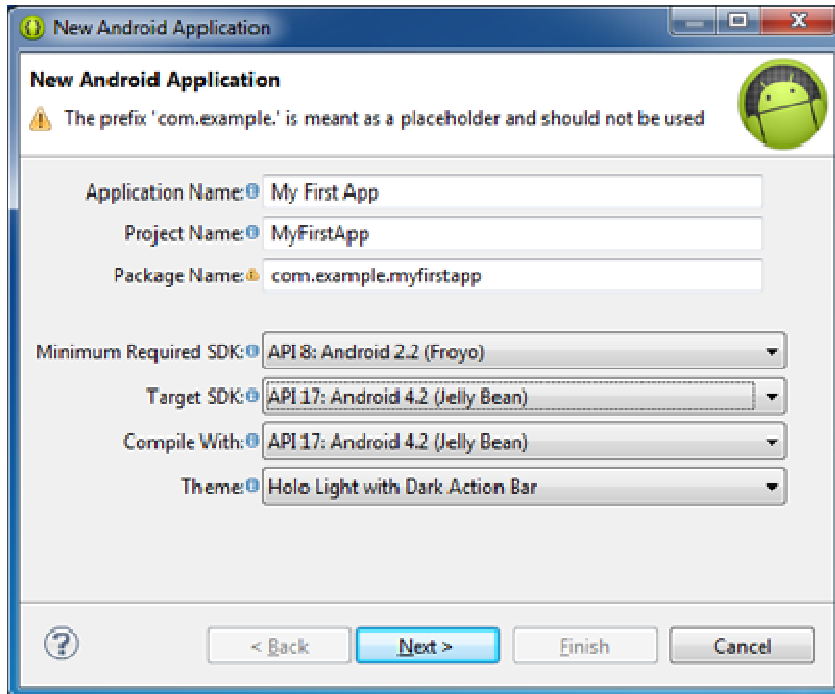


That's it. You can now move on to creating your first app. The lesson starts on this page:

<http://developer.android.com/training/basics/firstapp/creating-project.html>

The instructions here are pretty close, but as I have OCD tendencies, there are a few omissions I want to correct.

1. Launch Eclipse
2. File >> New >> Android Application Project. The following dialog box appears:



3. In the Application Name text box, enter "My First App". The Project name and Package name will auto populate. No need to change anything else, select **Next >**.
4. The next few screens are fine the way they are, so again, click **Next >**, **Next >** again, **Next >** one more time, then click on **Finish**.

That's it. You have now created your first Android App, the standard 'Hello World.'

Running your new app: <http://developer.android.com/training/basics/firstapp/running-app.html>

This page is pretty close to accurate; please read and follow the instructions here. You will find some valuable nuggets of info. Same page moves on to 'Run on a real device'. This section is very important. If you can't run this simple example now, you'll have a bunch of problems later.

Here it states for you to install your Androids OEM USB drivers. I suggested you do this at the very beginning because it is critical to you being able to complete this. The list is big, but not inclusive of all manufacturers. Remember what I said about the HP Slate 7.

One of the most important things to do here is enable USB debugging on your device. The one they don't mention, which I found you also must do, is go into your device's security settings and enable "Allow installation of apps from sources other than the play store". On the Galaxy phone and Nexus 7 tablet it is entitled "Unknown sources" it should be the same on other devices.

Though the web page goes into running on an emulator, which is beyond the scope of this article.

If you complete this, and saw "Hello World!" on your device, then congratulations! You are ready for more.

A few notes:

If you have a nexus tablet and having a problem getting it to connect to your PC, there are a couple things you can do to get past this.

1. Go into your settings >> About device, and tap 7 times on the android version or build number. This will enable developer options.
2. Connect your tablet to the PC, go to the device manager, find the bad entry for the tablet (or phone). Then go where you installed Eclipse IDE, for example: C:\adt-bundle-windows-x86-20130729\adt-bundle-windows-x86-20130729, click on folders SDK\ Extras\ Google\usb_driver. Copy this path. Back to the bad device manager entry. Double click on that device item, select update driver, and put this new path as the location to get the driver. That should work.

This may also work for the HP tablet. Found this on line:

Find the Google USB driver android_winusb.inf which is located in
sdk>\extras\google\usb_driver\.

Add below sections and save it,

```
; HP
%SingleAdbInterface% = USB_Install, USB\VID_03F0&PID_5E1D
%CompositeAdbInterface% = USB_Install, USB\VID_03F0&PID_5D1D&MI_01
%CompositeAdbInterface% = USB_Install, USB\VID_03F0&PID_601D&MI_01
```

Manually install the driver for the device, from Device Manager by specifying the folder where the above file is stored.

After all these done, restart ADB service to check the device (adb devices)

3. When you go to run the Hello World app, should you get the error, "Android Failed to install HelloWorld.apk on device," it may be because you have plugged in your device into a low powered USB port such as the ones on the front of your computer or a USB hub. Or you may be using an inferior USB cable, ie one that has thin conductors. Tables come with 'Beefy' cables for charging. I found I needed to use that cable to communicate with the PC as well.

