PAUL CRANE

ANDROID DEVELOPMENT SETUP ON WINDOWS

PAUL.CRANE@OTAGO.AC.NZ

First printing, May 2011

Introduction

This is a guide for setting up the Android Development environment on Windows. The process is outlined at Google's instructions[1] and also at Android Development[2] (this is catered for Mac OS X only, however).

By the end of this you will have Eclipse setup to develop on real Android devices or a virtual Android device.

[1] http://developer.android.com/sdk/installing.html

[2] http://learnaboutandroid.blogspot.com/2011/05/setting-up-your-androidjava-environment.html

Downloading

The first thing that we need to do is to download the following components:

- Java SDK
- Eclipse IDE
- Android SDK

The links for these appear in the side note. It's a good idea to start them all downloading at the beginning so as not to waste time.

Java SDK

Download the Java SDK from Oracle's website[3] and install.

[3] http://www.oracle.com/technetwork/java/javase/downloads/index.html

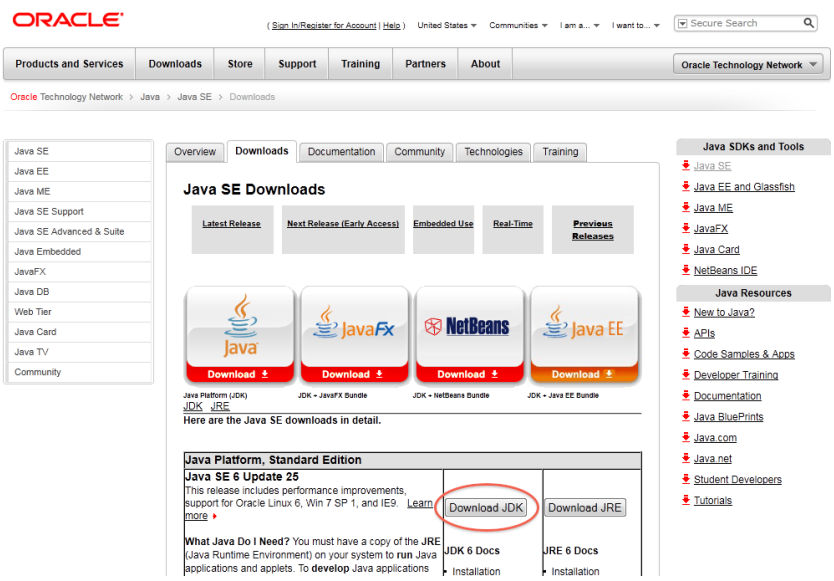

Figure 1: Download the Java SDK, select the highlighted option.

Eclipse IDE

You only need the basic version of Eclipse[4] (Google recommend the 'classic' version).

[4] http://www.eclipse.org/downloads

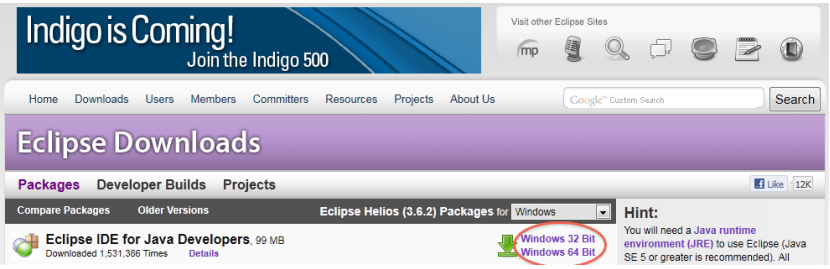

Figure 2: Download Eclipse, select the highlighted option.

You can just extract the contents to your `C:\Program Files(x86)\` and make a shortcut to `eclipse.exe` on your desktop.

Android SDK

Download the installer[5] and run it. It will install to `C:\Program Files(x86)\Android`. If you decided to change this location, make a note of it because we will need it later when we come to setup Eclipse.

[5] http://developer.android.com/sdk/index.html

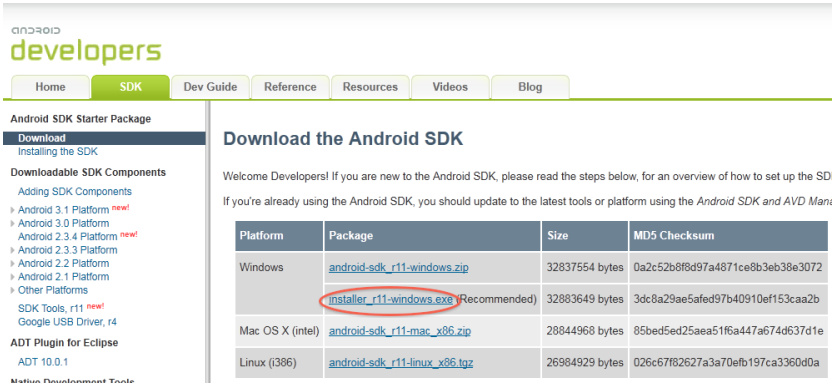

Figure 3: Download Android SDK, select the highlighted option.

Once the installer has finished it's work, let it start the **SDK Manager** for you. A window will pop up and ask which packages you want to install. According to Google's Device Chart 4, the most common devices are running 2.2 and 2.1. Select these from the list. Ensure that the *Google USB Driver package* is selected too 5.

Eclipse Setup

There still a little bit more work to do before being able to develop for the phones. We need to install the ADT plugin for Eclipse.

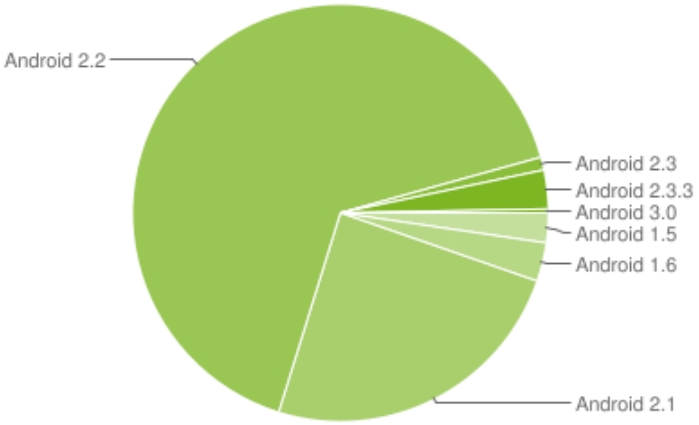

Figure 4: Distribution of Android handsets, retrieved May 28 2011 *Source:* `http://developer.android.com/resources/dashboard/platform-versions.html`

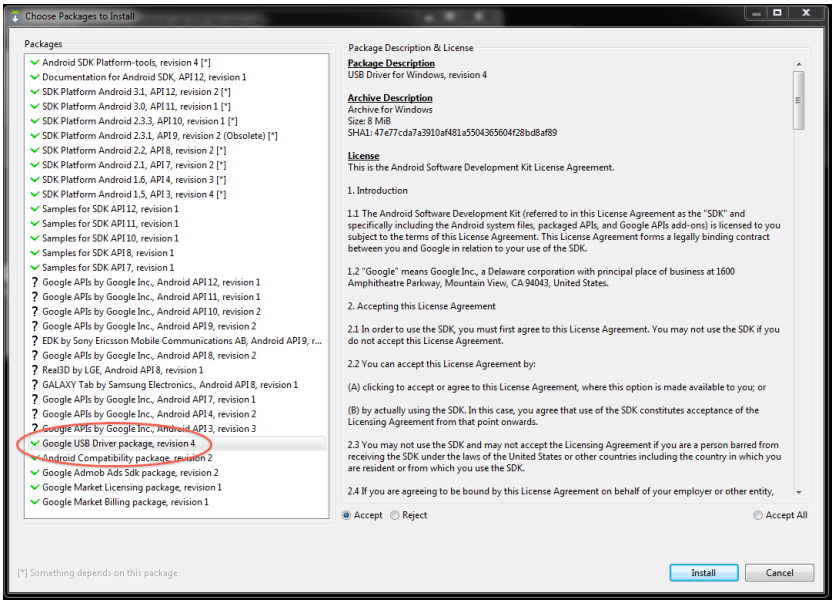

Figure 5: Android SDK package selection — ensure the *Google USB Driver package* is selected

Open Eclipse and select a workspace. This is where your source code will live.

Once you have selected a workspace the next screen you'll see is this:

Select the arrow icon on the right to go to the workbench. On the left of the workbench is where your projects end up, for the moment it'll be empty (you've not created anything yet!). The portion in the middle is where the code will appear.

To install the plugin, go to *Help* and select *Install New Software*. Click on the *Add* button. You will need to enter the information as in 8. Once you've added that information you will see the list populated with items that can be installed. Select the *Developer Tools* and proceed with the installation.

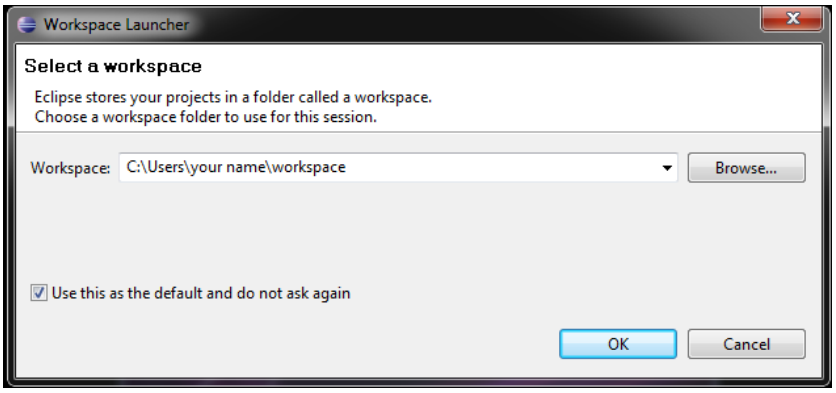

Figure 6: By default it'll substitute your name in there, you can select this workspace by default every time you open Eclipse

Figure 7: Eclipse's Opening Screen

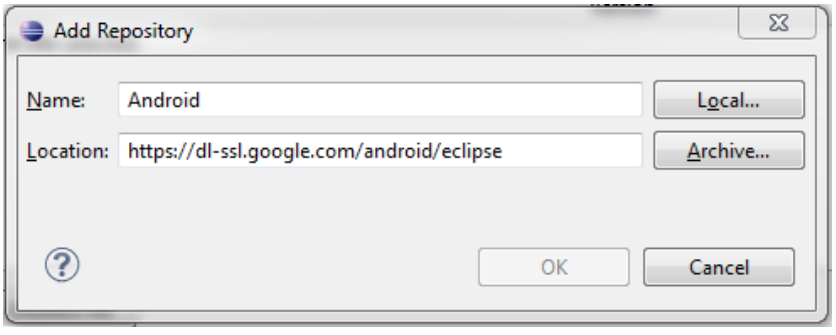

Figure 8: Adding a new software source to Eclipse

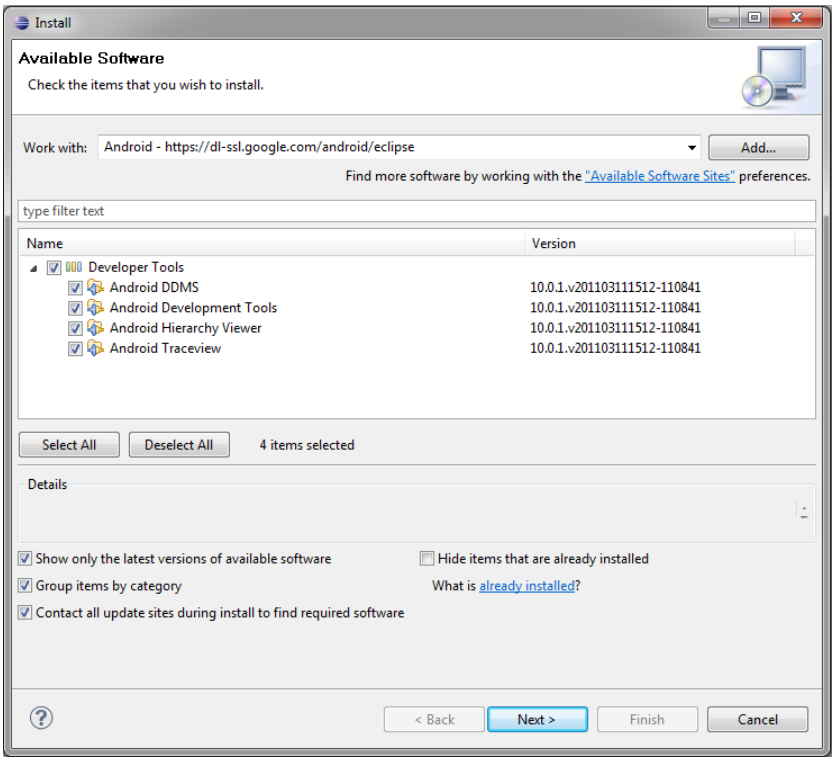

Figure 9: Proceeding with the installation

Eclipse will ask to restart after the plugins have been installed. Allow it to restart. When you open Eclipse you will see a new menu icon 10. This means that the plugin has installed successfully.

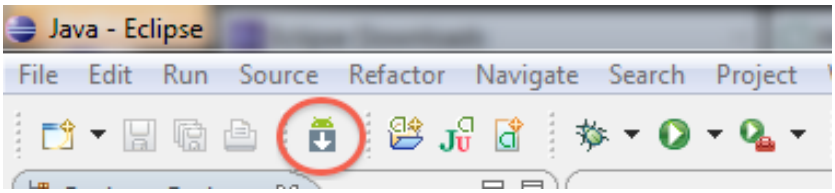

Figure 10: Plugin installation success!

The last thing that we need to do is to tell Eclipse where to find the SDK. To do this, click on *Preferences* and under *Android* there is a box there for the SDK Location. Click the *edit* button, and navigate to the path where the SDK was installed, by default it is `C:\Program Files(x86)\Android\android-sdk`.[6] Click *Apply*, you should see the list of available targets update.

[6] If you've changed this in a previous step, put the changed path here.

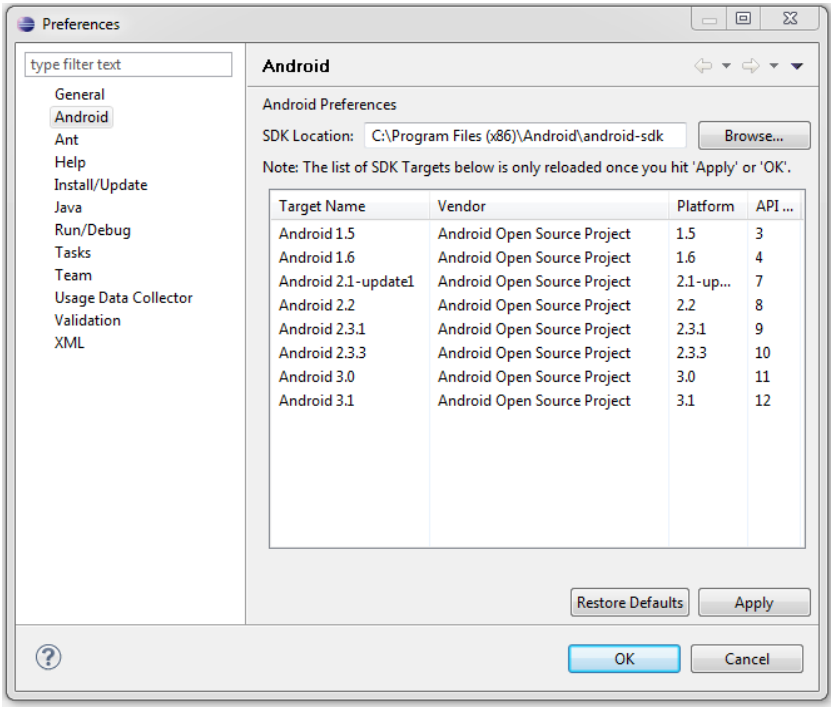

Figure 11: Locating the Android SDK for Eclipse

Developing

There are two paths from here. You can develop using an emulated device or with a real one. The problem with the emulation is that there isn't an easy way to test sensors and accelerometer based applications.

We'll first setup a virtual device then move on to the real device setup. Finally, we'll get a basic 'Hello World!' project started.

Virtual Devices

If you click on the icon in 10 you'll be presented with an AVD manager. Click on the *New* button on the right hand side, and create an AVD. When it comes to naming it's a good idea to name it something meaningful, here we use the version number of the target (so we know instantly which device we can deploy on).

Once you've created the device, you can start it up. You may have to wait.

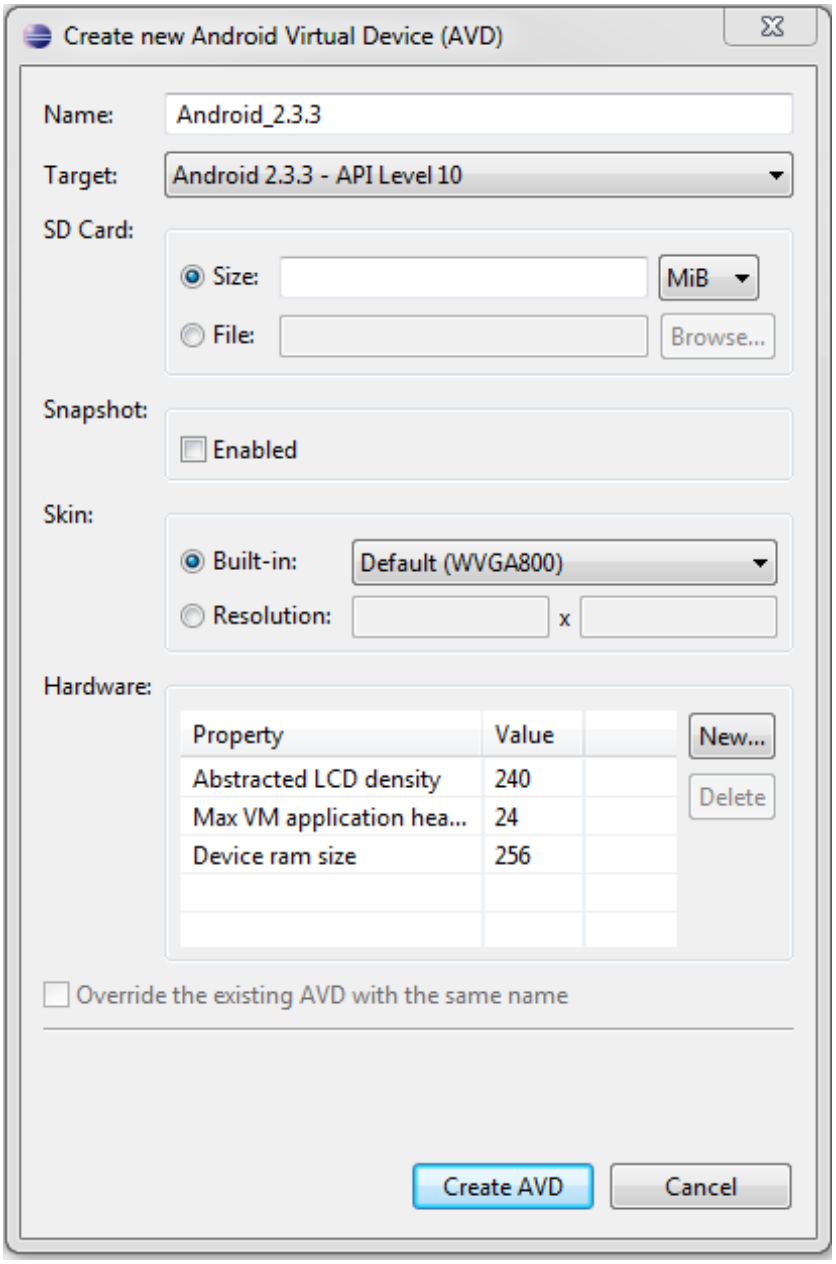

Figure 12: Creating a new Android Virtual Device

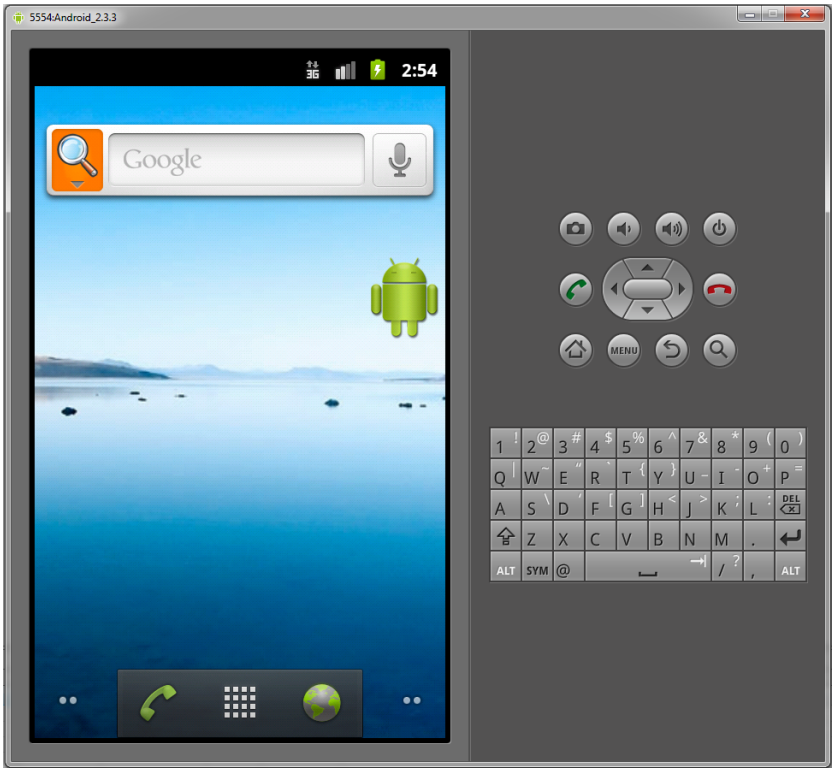

Figure 13: Running the Android Virtual Device

Real Devices

For developing on Windows with real devices you need to jump through another couple of hoops that are unnecessary for Mac and Linux platforms.

Plug your phone in via it's USB cable. Windows will try to find drivers for it. Depending on the phone it may succeed — for the Nexus S, it didn't. In order to install the drivers, you need to right click on *Computer* (in the *Start Menu*), and select *Manage*. Select the *Device Manager* from the next window.

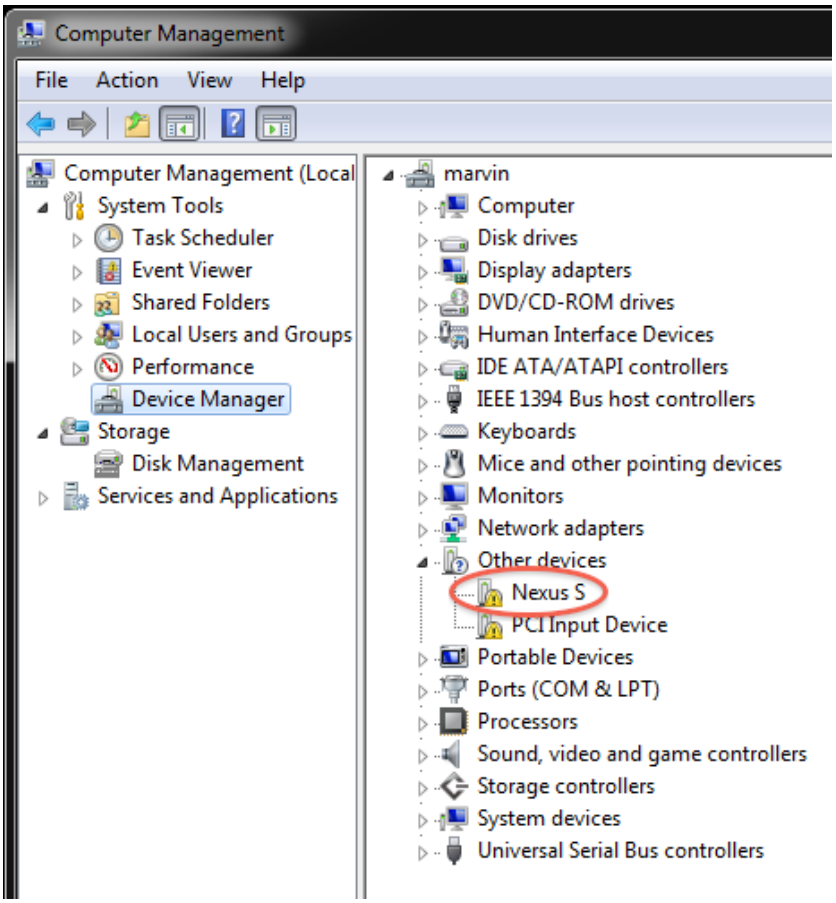

Figure 14: Device Manager with the driverless Android Phone

If you expand *Other Devices* you should see your phone listed there. Right click your phone, and select *Update Driver Software* and navigate to `C:\Program Files(x86)\Android\android-sdk\extras\google\usb_driver`. Allow the installation process to proceed.

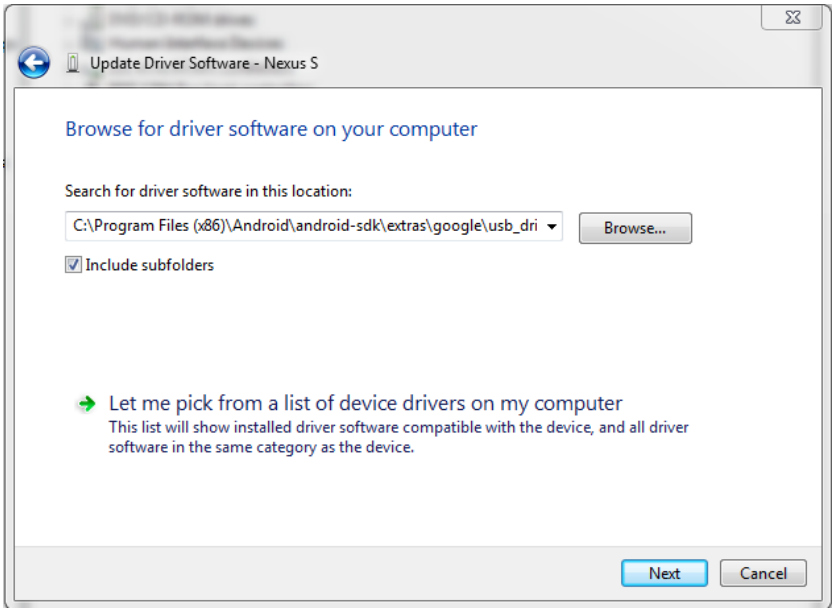

Figure 15: Browsing for the drivers for the phone

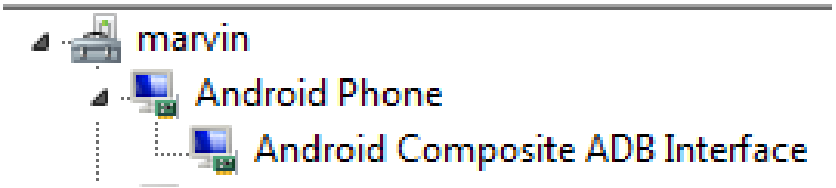

Figure 16: Successful installation of the USB driver

Hello World!

In Eclipse, start a new *Other Project*, and use the *Android Project* wizard. Fill out the fields according to 17 and it'll appear in the *Package Explorer*.

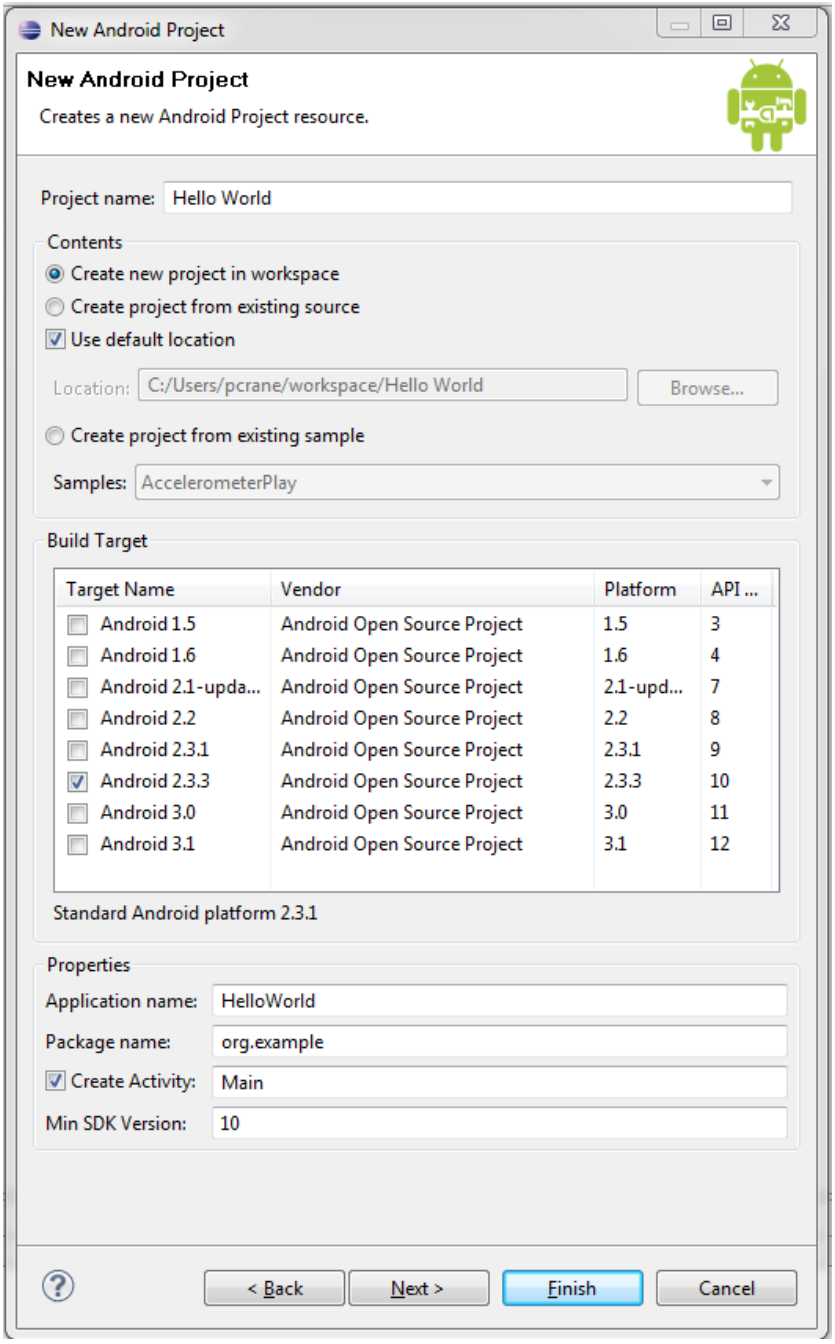

Figure 17: Successful installation of the USB driver

Expand it and you'll see all the different parts of the project. Click

on the *Run* button, and it should start in the device (if one is attached) or the previously created virtual device.

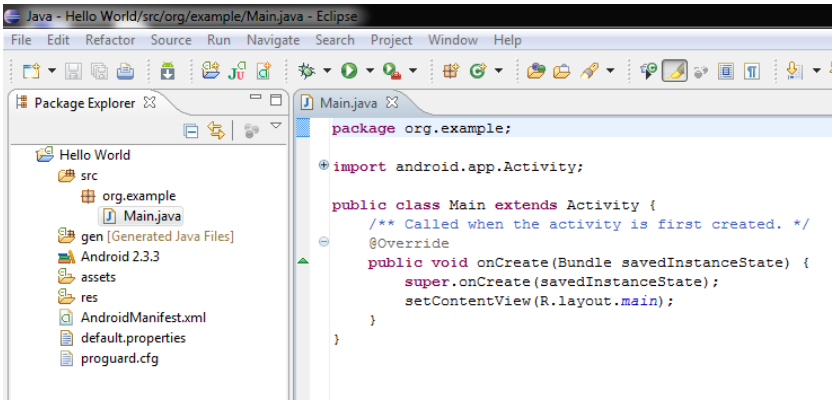

Figure 18: Default Hello World code

Congratulations, you've now got a working development environment!

DDMS

A key feature is the ability to view what's happening on the device (real or virtual) at any given time. Eclipse has *perspectives* that enable developers to group a series of panes together (by default you're in the 'Java' perspective). You can change perspectives by using the menu on the top-right (highlighted in 19). Select DDMS from the list.

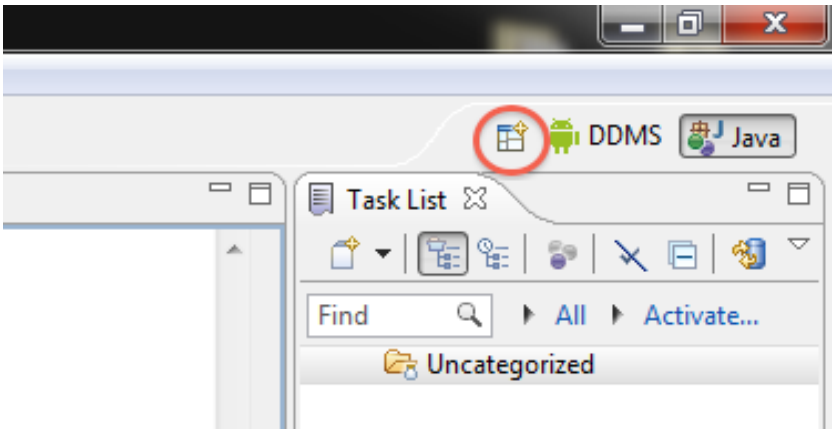

Figure 19: Opening the DDMS perspective. In this example, it's already been added to the menu bar

You'll see the tools needed to help debug your applications.

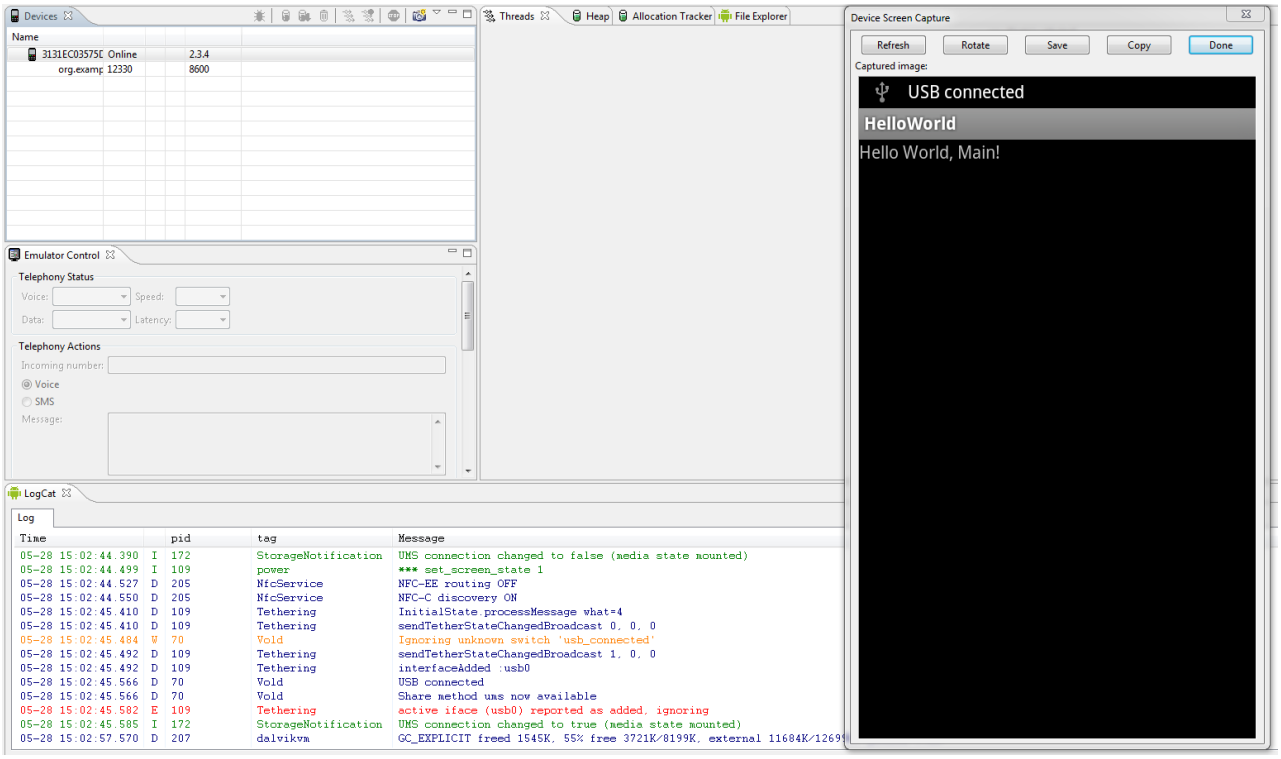

Figure 20: Showing DDMS perspective as well as a screen capture from a real device