

IEMS 5722  
Mobile Network Programming and Distributed Server Architecture  
2014-2015 Semester 2

**Assignment 0: Android Development Basics**

Due Date: 22<sup>nd</sup> January, 2014 (Thursday)

**Notes:**

- i.) You are strongly advised to go through all the steps in this assignment in order to get familiar with the Android development environment
- ii.) Pay attention to Section 4, where you are required to perform some tasks
- iii.) See the instructions at the end of this assignment, follow them to submit your files for marking
- iv.) Late submissions will receive 30% mark penalty

**1. Aim**

To learn the fundamentals of Android development from project creation to installation on a physical device

**2. Objectives**

- Set up the Android development environment
- Create a Hello World Android application
- Understand the Android project structure
- How to run an app in the emulator
- How to deploy to app to a physical device

**3. Instructions**

**3.1 Setting up the development environment**

Please ensure you have at least **JDK 6** installed before continuing. Note that JDK and JRE is different. JDK is required for development. You can get the latest version from <http://www.oracle.com/technetwork/java/javase/downloads/index.html>

**3.1.1 Eclipse ADT Bundle**

The bundle includes everything you need to start Android app development and is the recommended option for this course.

Download the ADT bundle from <https://developer.android.com/sdk/index.html>

Once you have downloaded the package, unzip it to your development location (such as C:\Development\EclipseADT)

When you run it the first time, Eclipse will ask you to specify a workspace. This is where your project's files will be saved. You can choose a default location to use or change it later if you want.

### 3.1.2. Eclipse Android Plug-in

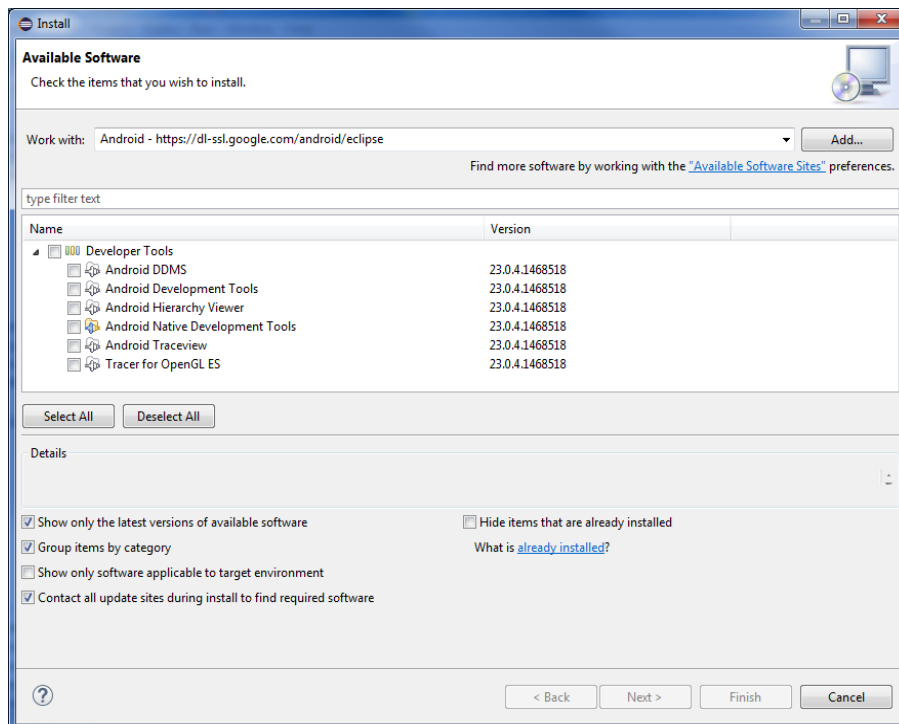
If you have an existing installation of Eclipse, you can download the ADT via the **“Install New Software”** menu.

Go to **Help > Install New Software** to open the Install window. Click the Add button and enter the following:

**Name** : Android Development Tools (Can be anything, but choose something easy to remember)

**Location** : <https://dl-ssl.google.com/android/eclipse>

The window should show a list of available software as seen below.

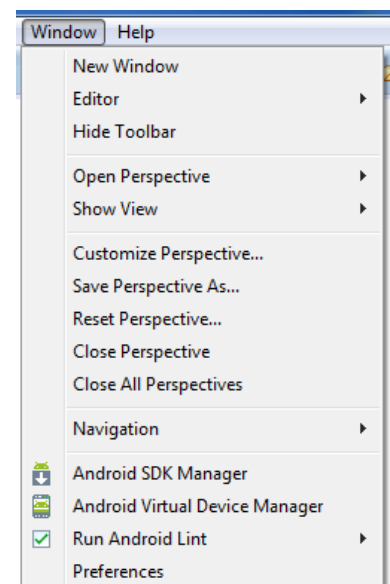


The important options to select are the **Android DDMS** and the **Android Development Tools**. Select those and follow the prompts to complete the installation. You may need to restart Eclipse to see the changes.

### 3.1.3 Installing the SDK (Android ADT and Eclipse plug-in)

In the Eclipse Java perspective, several Android options should now be available under the Window menu (right).

Click on Android SDK Manager to show the different API available. By default, the Android ADT includes the current version, but you will need to download other versions if you want to build your app against older Android versions. Targeting earlier versions let you reach a larger number of users. You will also need to use the SDK manager to download later versions of the API when they are released in the future.



For this course, we will use Android 4.0.3 (API 15).

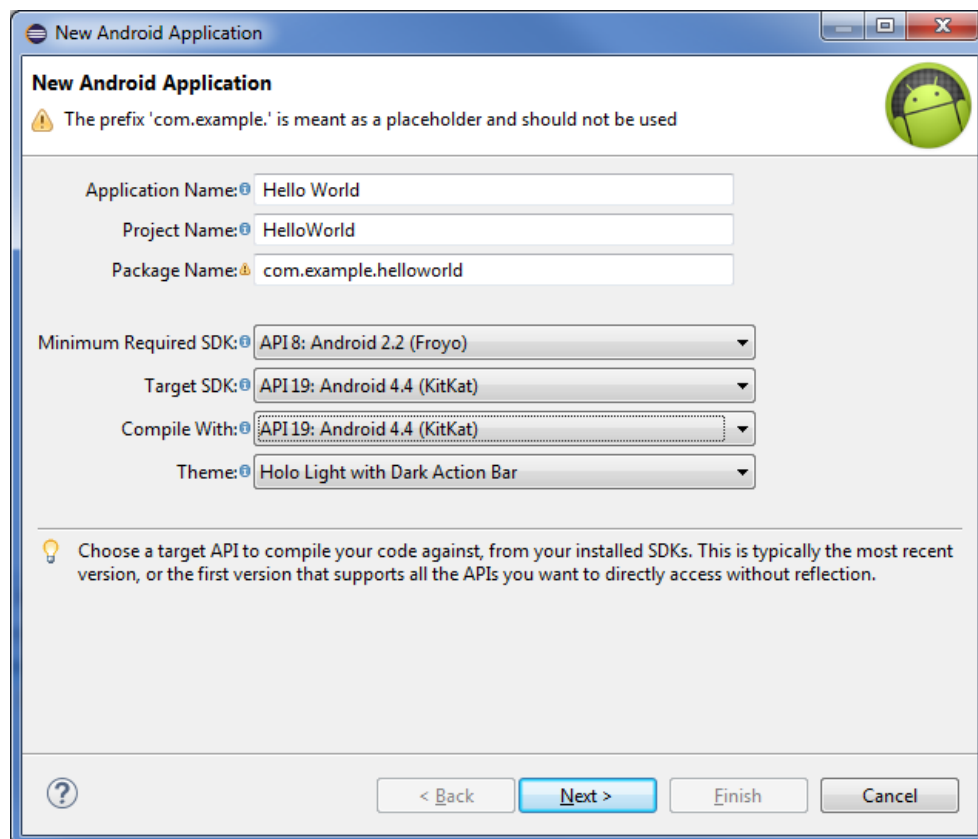
Download the SDK Platform for that version.

### 3.2. Creating your first app

To create an Android project, go to **File > New > New Android Application**

The New Android Application dialog appears. In the Application Name field, enter “Hello World”. This is the name that shows up in the **Android application launcher**.

Project Name and Package Name should auto complete for you, following the Java package naming rules as seen in Figure 3.1.

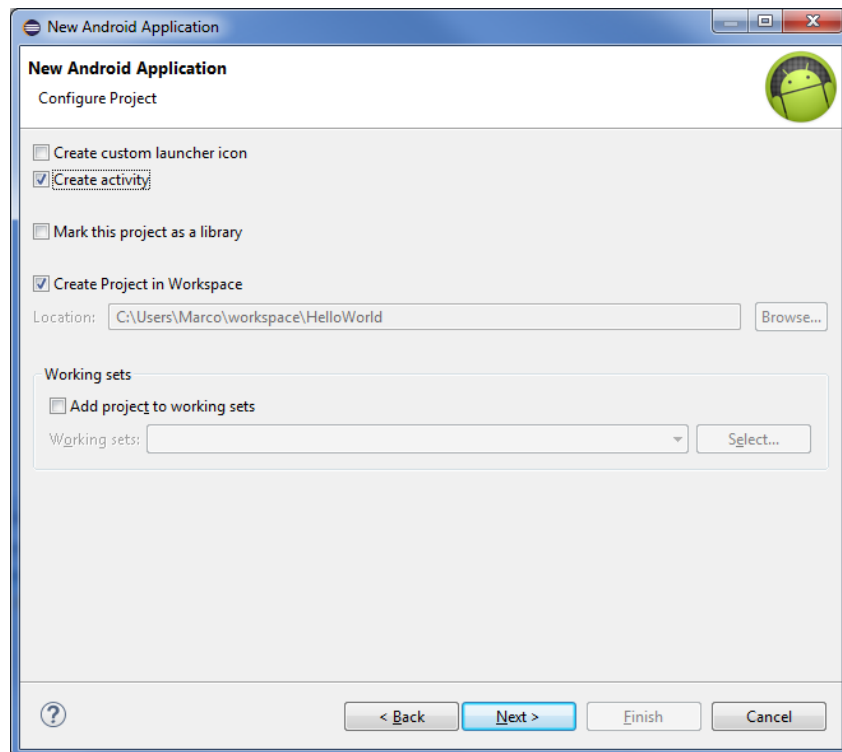


**Figure 3.1** - New Android Application

You can also set the Android versions to support here. The **Minimum Required SDK** is the lowest version of Android that your app supports. Lower versions target more devices, but less SDK features will be available.

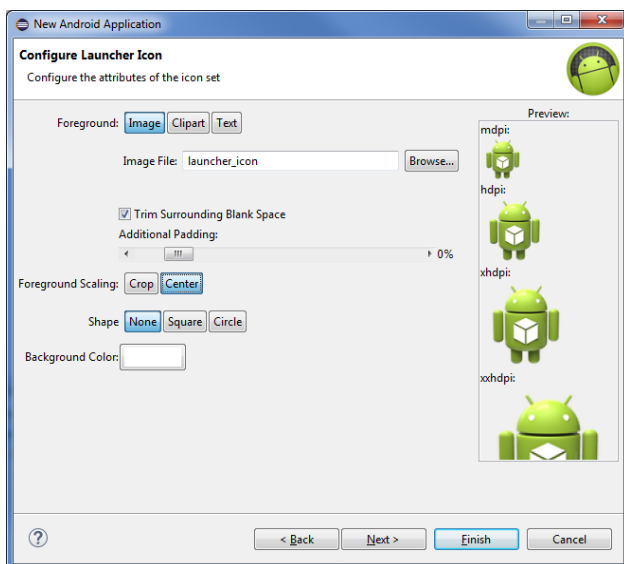
The **Target SDK** is the highest version of Android that your app is known to work with. The **Compile With** is the SDK version to use to build your app. You can set both Target SDK and Compile With to the latest version that you have downloaded. Set minimum to SDK 14 and target SDK to 19 in this example.

When you click next, it will bring up more project set up options. Checking the Create custom launcher icon will let you customise the **launcher icon** (optional). Check the **Create activity** and **Create Project in Workspace** options. This will create your project in your workspace with an empty activity. An example of the dialog is seen in Figure 3.2.

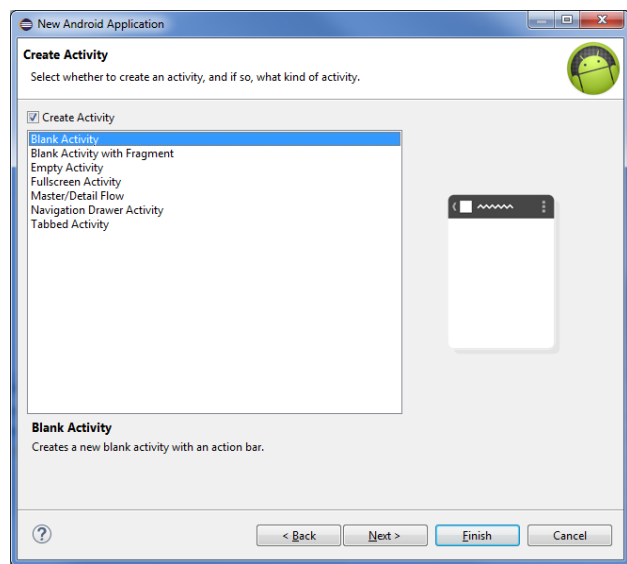


**Figure 3.2 - Initial app options**

If you selected the custom icon option, the next screen will let you choose your icon image, and its display settings as seen in Figure 3.3. You can set this up later in your project as well.



**Figure 3.3 - App icon options**



**Figure 3.4 - Activity design options**

Click Next to proceed to the activity options screen shown in Figure 3.4.

This screen shows some activity design options, such as having navigation menus or full screen displays. For this project, we only need a blank activity, but you are encouraged to try out the other ones to see how they work.

Click next to set up the activity. The blank activity has two options - the “**Activity Name**” and the “**Layout Name**” as seen in Figure 3.5. This screen defines your initial activity - when your app is launched, this activity and layout will be the entry point into your app. You can leave these as default for now.

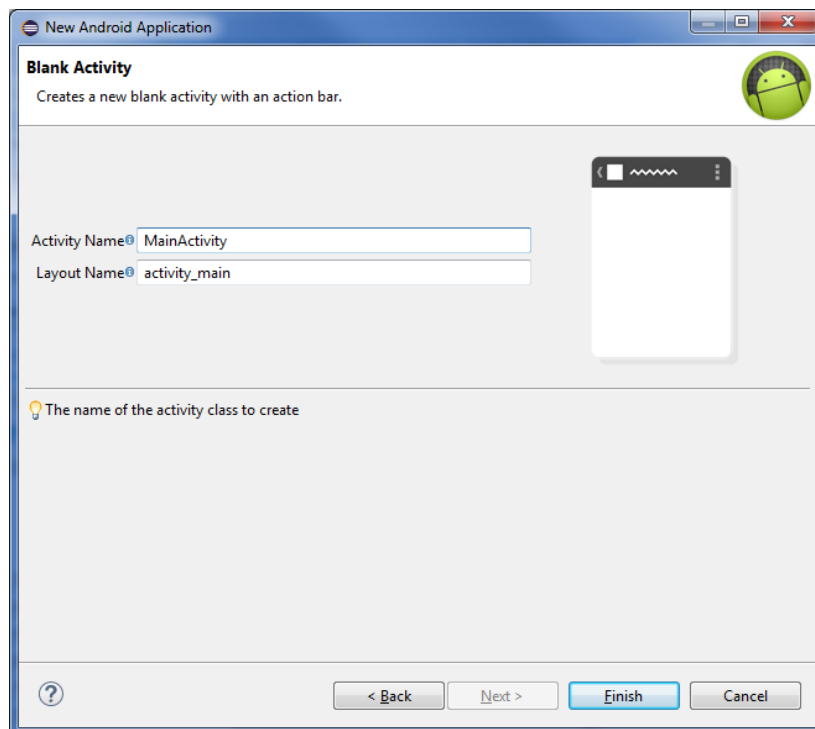


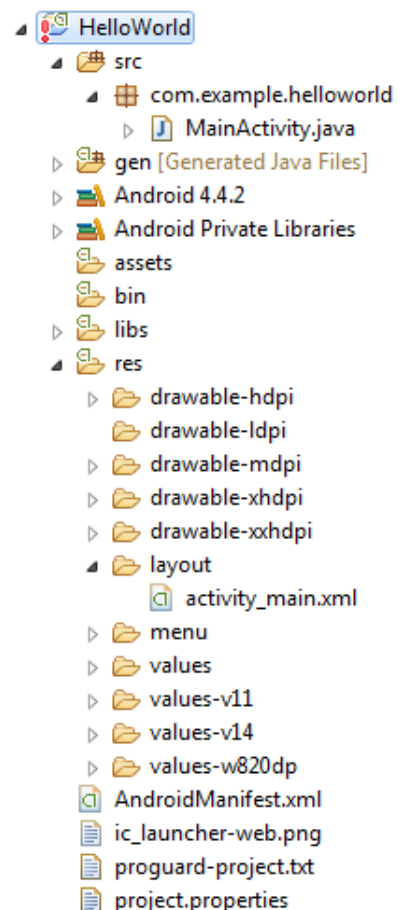
Figure 3.5 - Activity options

Click Finish to create the project and activity. You should be back in Eclipse with a new HelloWorld project in the **Package Explorer** panel.

To see what Eclipse has done, explore the **project tree** as seen in Figure 3.6 on the right.

It has created a HelloWorld project, using the project settings we chose in Figure 3.1. The **src** folder contains the Java files. The first package is **com.example.helloworld**. This is also specified from Figure 3.1.

The **MainActivity.java** is created inside the package. Likewise, the **activity\_main.xml** file has been created in the **res/layout** folder. Both of these were specified from the Activity options from Figure 3.5.

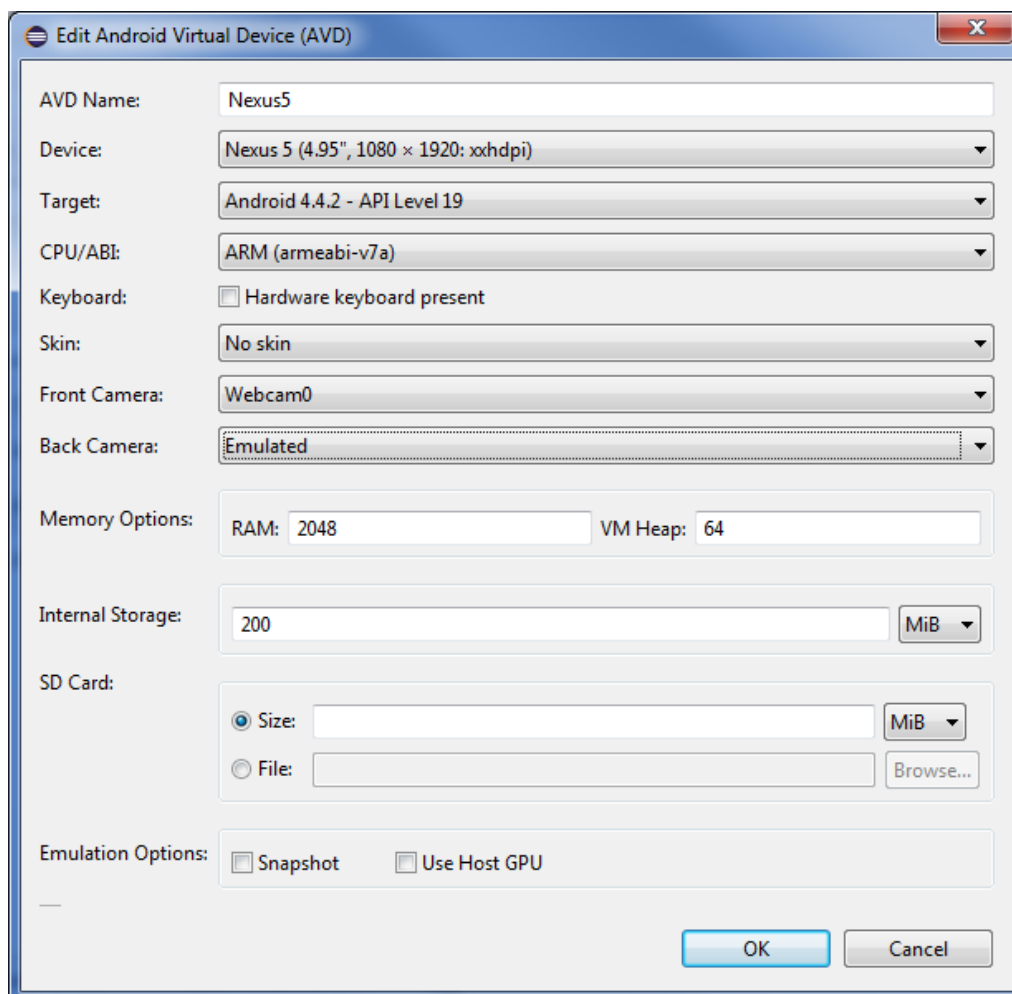


### 3.3 Running your app in the emulator

#### 3.3.1 Create an emulator profile

In Eclipse, go to **Window > Android Virtual Device (AVD) Manager**. From here you can create, edit and delete your emulator profiles. Click on the Create button to define a new emulator profile.

Emulator profiles allow you to test your app with different settings so you can see how your app behaves even if you don't have a particular device physically. A sample AVD profile is seen in Figure 3.6 which is sufficient for this exercise. If you want to simulate a particular device, you will need to look up the particular devices profile.



**Figure 3.6 - Sample Android Device Profile**

#### 3.3.2 Starting the emulator

In the **AVD Manager window**, choose the emulator you wish to run and click the Start button. This will bring up the Launch Options window, where you can scale the emulator display size. You may need to experiment with this to fit your display best.

When you click Launch again, the emulator will begin. This may be quite slow to start, but you will not need to close the emulator until you are finished.

### 3.3.3 Deploying the app

In Eclipse, right click your HelloWorld project and choose **Run As > Android Application**. It may ask you where do you want to deploy your app to if you have more than one emulator running or if your phone is also plugged in. Choose the emulator.

If you modify your app, you can deploy it again to the emulator without having to restart the emulator. Just choose **Run As > Android Application** again.

## 3.4 Running your app on a physical device

### 3.4.1 Setting up your computer

You will need an Android device and a USB cable to connect your device to your computer.

Install **Google USB Driver** from the **Android SDK Manager**. This will be at the near the bottom of the list, under the **Extras** folder. The figure shown below has the Google USB driver checked. Note that some Android devices do not need the Google USB driver for development. The Galaxy Nexus for example, uses the OEM driver. Consult your manufacturer documentation if in doubt.

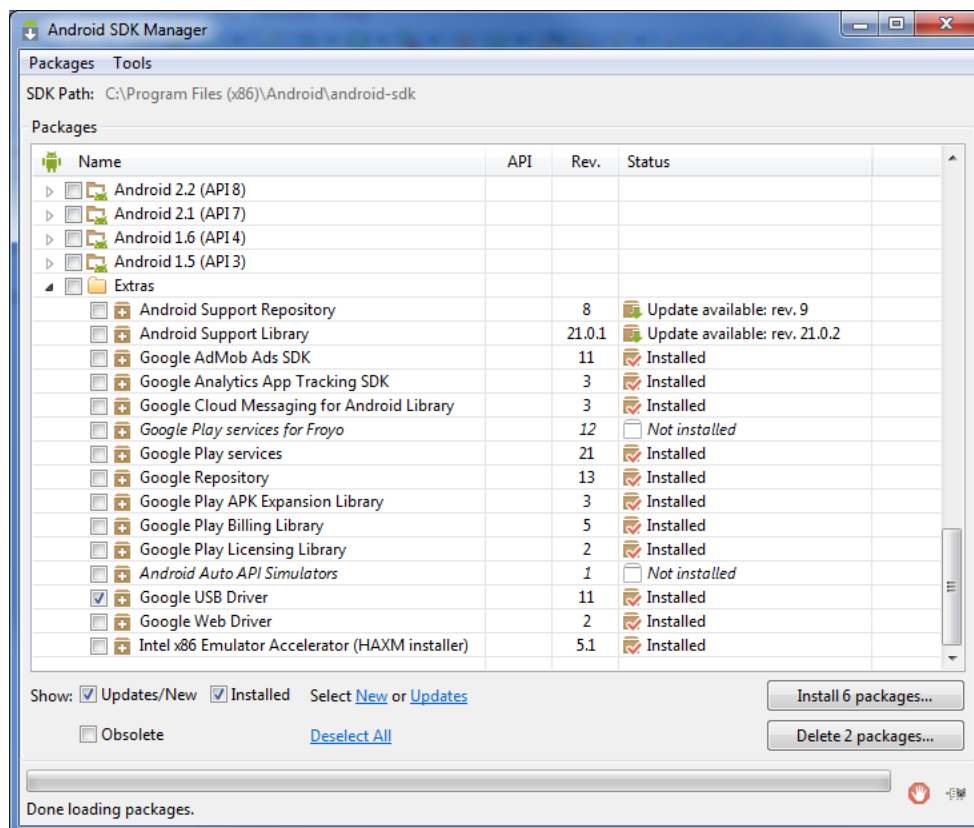


Figure 3.7 - Android SDK Manager showing the USB driver

### 3.4.2 Setting up your Android device

Devices running Android 4.2+ have developer mode disabled by default. To enable it on your device, go to **Settings > About Phone > Build Number**. On some devices this may be under **Settings > About Phone > Software Information > Build Number**.

Tap on Build Number **7 times**. There should be a pop up notification saying “you are x steps away from being a developer” which counts down after two taps.

If you were successful, the **Developer Options** menu should be visible under **Settings**.

### 3.4.3 Deploying your app

Connect your device to your laptop using the USB cable. On your device, you may need to switch to **MTP mode**.

In Eclipse, right click on your project and select **Run As > Android Application**. A window prompt will appear asking you where you want to deploy it to. Your physical device should appear in the list. Select your device then click deploy.

If this is the first time you have deployed to your phone from this computer, your phone will ask you if you want to allow apps to be installed from this source. You can also save this source so it won't ask you again next time. Accept the connection to continue installation.

The app will run when automatically when it has been installed.

## 4. Tasks

### 4.1 Modifying your app

#### 4.1.1 Change the text

You will notice that the app shows “Hello World” already. Look through the app files Eclipse generated and find where and how the text has been displayed.

Then add in another line of text, showing your name and student number. **DO NOT** hardcode your text in the Java source file, use the *string resources*. As a challenge, try to put your text in the center of the screen.

Hint - Start from MainActivity.java and look at what layout it references. Then look in the layout and look at what string resource it uses.

#### 4.1.2 Changing the logo

The app has launch icon by default. This is stored in the **res/drawable** folder, which is where graphics such as icons and images can be stored.

There are several drawable folders corresponding to different resolution devices: Low DPI, Mid DPI, High DPI and Extra High DPI. Android chooses the files most appropriate to the device that is running the app.

If the file is missing from a particular folder, it will choose the image from the next best available resolution. For example, a High resolution device will try to source its images from the **res/drawable-hdpi** folder. If it cannot find a particular image, it will look in the **res/drawable-xhdpi** folder and scale down the resolution to fit. If not successful, it will then look in the **res/drawable-mdpi** folder and scale up the resolution. This is not particularly beautiful so you should ensure that you provide a low/med/high/extra high resolution image for your app.



Change your apps logo to be the **cuhk-logo.gif** but do not rename the file. For this exercise, you do not need to create logos of different resolutions.

Hint - What other file is used by Android to determine what goes in your app?

## 5. Submission

To submission your assignment, create a folder with a name in the following format:  
<your\_student\_id>\_assgn0

Copy the follow materials into the folder you created:

- The **src** folder (including all Java source code files)
- The **res** folder (including all the sub-folders and files)
- The **AndroidManifest.xml** file

Compress the folder into a .zip file, and submit it in the CUHK eLearning System online:

<https://elearn.cuhk.edu.hk/>

## 6. Resources

- Google Android Training  
<https://developer.android.com/training/index.html>
- Android Style Resources  
<https://developer.android.com/design/downloads/index.html>
- Android API Reference  
<http://developer.android.com/reference/packages.html>
- Where to go if you have a question (apart from Google):  
<http://stackoverflow.com/questions/tagged/android>