Android Give It A Go

A little something to get you going

Tuesday 30th October 2012 @ 2pm in Kilburn 1.4

Installation Guide

Before you attend the first Man-UP Android event it is important that you set up your laptop so that you're ready to jump straight into development. This document will detail installing the Android SDK with Eclipse.

If you need any help with these instructions then post in the Man-Up Facebook group:

https://facebook.com/groups/manchester.ultimate.programming/

Installing the Java Development Kit

As Android apps are developed in Java, you'll need to install the latest JDK before you can start using the Android SDK. To download the JDK, visit:

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

If you run a 64 bit version of Windows, make sure to download the Windows x64 version.

Installing the Android SDK

To download the SDK simply visit http://developer.android.com/sdk/index.html. It's recommended that you install the SDK in an easy to reach location (i.e. C:\android-sdk or ~/android-sdk). This may save you some time in the future!

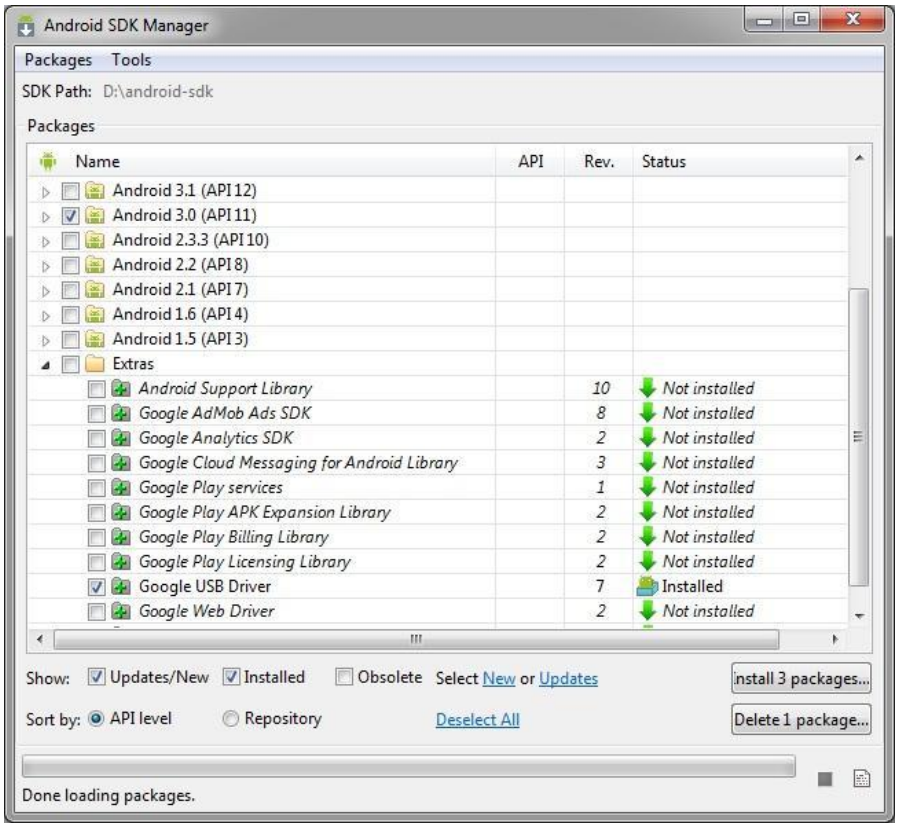
On OS X and Linux you should make sure that the SDK is in your path. To do this add lines such as these to your *.bashrc* on Linux or *.profile* on OS X:

```
export PATH=$PATH:~/android-sdk/tools/
export PATH=$PATH:~/android-sdk/platform-tools/
```

Note: The exact lines you add in depend on where you extracted the SDK.

To check you've installed it correctly open a new terminal and run android which should open the SDK manager.

Once you've installed the SDK, load it up by running the *exe* or android in the terminal and install the latest version of the Android platform, tools, and (if you're on Windows) the USB driver located in *Extras*.



Installing Eclipse

Eclipse is the de facto development environment for Android. It's completely free and can be downloaded from:

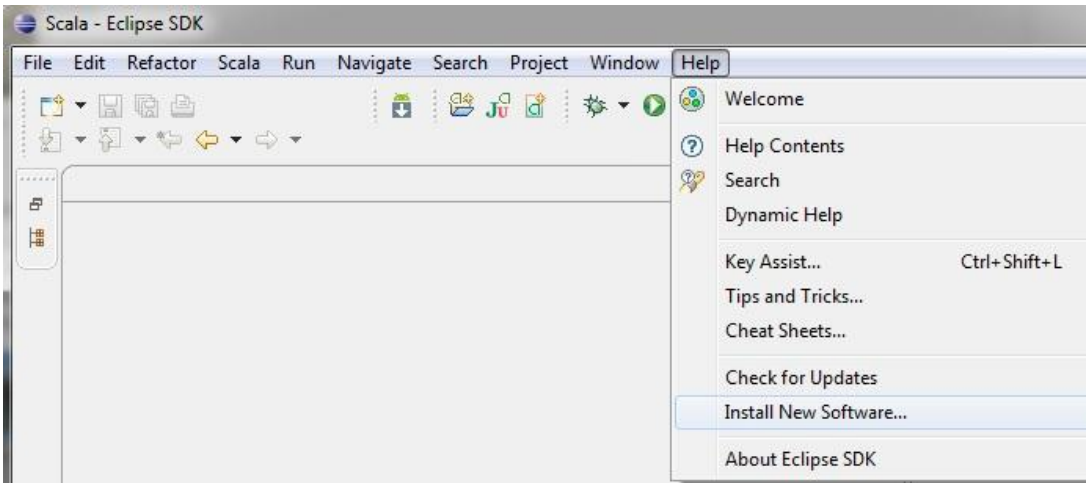
http://www.eclipse.org/downloads/packages/eclipse-ide-java-ee-developers/junosr1

Note: it is possible to use any text editor you like and the command line if you prefer. For starting out it's best to take a look at Eclipse though

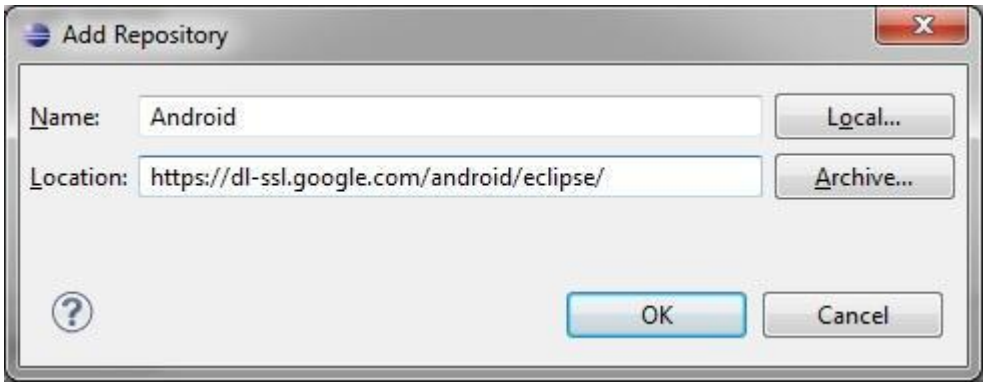
Installing the Eclipse Android Development Tools

The ADT includes a variety of widgets and development tools which will help streamline your development using the Eclipse IDE. These can be downloaded straight from Eclipse:

1. Click "Help" then "Install New Software"

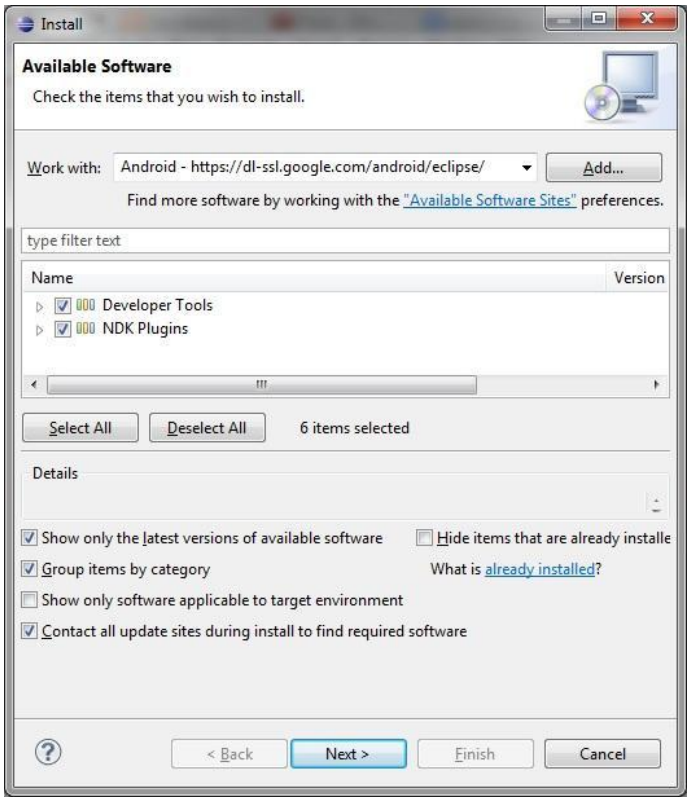


2. Click "Add..." then enter "Android" into the 'Name' field and *https://dl-ssl.google.com/android/eclipse/* into the 'Location' field



Note: If for some reason you're unable to download the ADT, we'll have an archive copy available for you at the meeting

3. Select both packages, click Next, then follow the instructions



Once you've installed the ADT, you'll need to restart Eclipse for the changes to take effect.

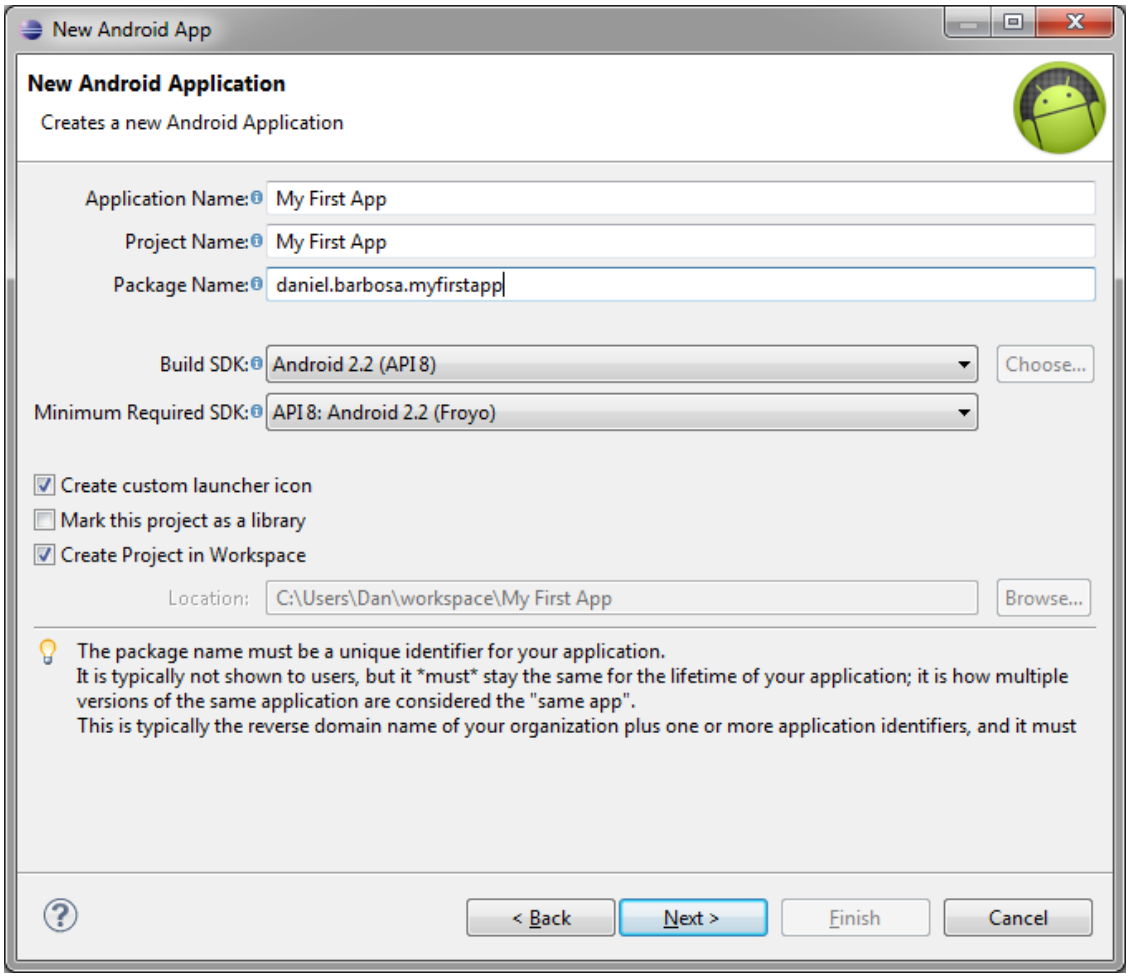
Creating an Android Project

To start a new Android project, simply follow File->New->Project then select "Android Application Project" within the Android folder. This option will automatically generate some barebones code for you to work with.

Note: You can achieve the same thing on the command line by running `android create project` *with all the required parameters*

You'll be asked to specify a few details regarding your project. Application Name and Project Name are self explanatory, but for Package Name you'll need to specify a Java-style package name. For commercial projects this is normally your company's reversed domain name followed by the project name. For example *com.danielbarbosa.myfirstapp*

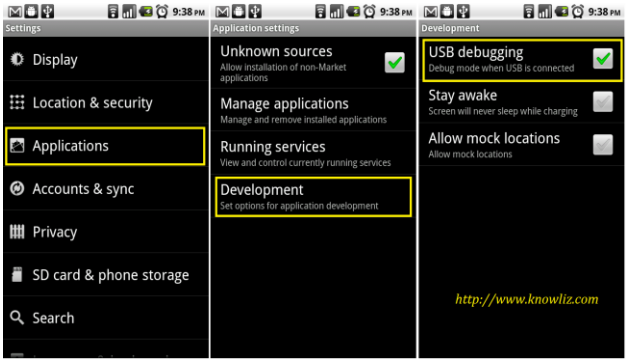
For the Build SDK, set this to the latest Android version. You should set the Minimum Required SDK to an older version if you want your app to be compatible with older versions of Android.



For the purpose of this basic introduction, you can leave the values unchanged for the rest of the setup process.

Connecting to your Android Device

In order for you to test your applications on your Android device, you will need to enable USB debugging. This can be done in the settings menu



Note: In newer versions of android the USB debugging setting is not in Application but in a new top level setting section called development

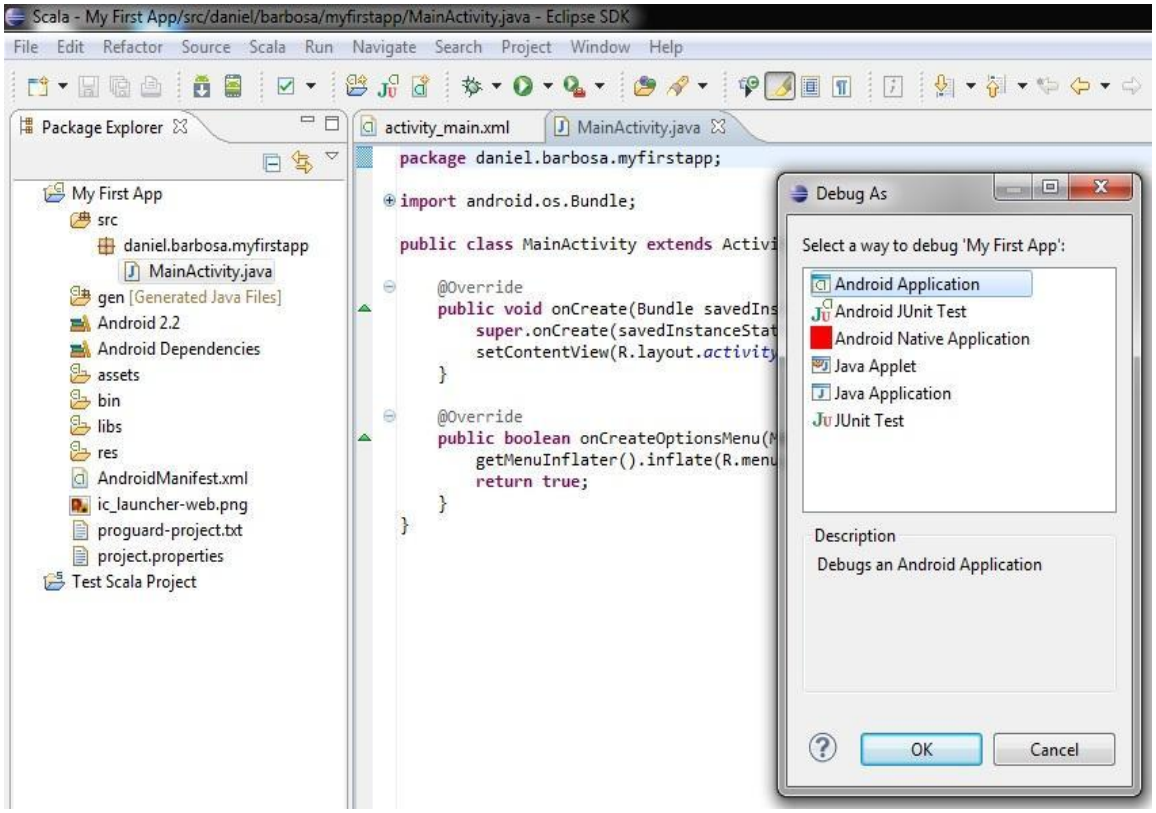
On OS X you can now just connect your device with a USB cable and it will work.

On Linux you can connect it but you may have some problems with USB permissions. Take a look at *Problem 5* in the next section for help on how to fix that.

On Windows you can connect your device to your laptop with your USB cable and, in theory, the necessary drivers should automatically install. If not, you may need to visit your device manufacturer's website for the relevant USB drivers.

Running your application

You should now be able to run your basic app on your phone. This can be done in Eclipse by hitting F11 and selecting 'Android Application' in the 'Debug as' menu.



You're now ready to start your app development!

Cheat Sheet

You can use this cheat sheet to help you out when you run into some of the common problems in Android development.

Problem 1 - Cannot build

- `Project 'Project Name' is missing required source folder: 'gen'`
- `The project could not be built until build path errors are resolved`
- `Unable to open class file R.java.`

Solution

- Go to the project menu and select **Project → Clean**.

Problem 2 - Activity not found

- `android.content.ActivityNotFoundException: Unable to find explicit activity class; have you declared this activity in your AndroidManifest.xml?`

Solution

- Check if you declared your *Activity* in the *AndroidManifest.xml* file

Problem 3 - No devices

- List of devices is empty (`adb devices`)

Solution

- The communication with the emulator/device is handled by the Android Debug Bridge (adb)
- *The long one:* Eclipse allows to reset the adb. Select the DDMS perspective via *Window → Open Perspective → Other → DDMS*. Select the "*Reset adb*" in the Device View.
- *The short one:* In terminal type adb kill-server and then adb start-server

Problem 4 - adb not found

- `adb: command not found`

Solution

- Make sure the path to the ADT is included in your PATH variable. You can do that by concatenating it to the existing one (files: *.profile* for OS X, *.bash_profile* or *.bashrc* for linux):
  `export PATH=$PATH:/home/<username>/android-sdk/tools/`
  `export PATH=$PATH:/home/<username>/android-sdk/platform-tools/`
- *Note:* The absolute path depends on the location where you saved your SDK in

Problem 5 - USB permissions

- `List of devices attached`
  `???????????? no permissions`

Solution

- The solution of this problem is platform dependent:
  - *Linux:* You need to add a udev rules file that contains a USB configuration for each type of device you want to use for development. Follow the instructions in step three here:
    - http://developer.android.com/tools/device.html
  - *Windows:* You need to install a USB driver for adb. See the OEM USB Drivers document:
    - http://developer.android.com/tools/extras/oem-usb.html
    - Hopefully you will not get this problem on windows, as the **windows-SDK** from the USB memory sticks contain the driver
  - *OS X:* It just works. You should never have this problem ;-)

Problem 6 - R file problems

- `main cannot be resolved or is not a field`
- `id cannot be resolved or is not a field`
- `R.layout.main cannot be resolved or is not a field`

Solution

- Eclipse is not good at handling errors with layout files. Make sure to import your *R* file import `your.application.packagename.R` rather than import `android.R`

Problem 7 - Emulator won't start

- Emulator does not start

Solution

- Make sure that the *android-sdk version* is in a path without any spaces in the path name.

As you can expect, these are not all the problems you might encounter while you develop your application, but some of the most common ones. Ask someone if you need some help.

Eclipse Keyboard Shortcuts

The following table contains some of the keyboard shortcuts, used in Eclipse to speed up your work.

Action	**Keyboard Shortcut**
Create a new file in the current package	*Alt+Shift+N*
Organize the import statements	*Ctrl+Shift+O*
Format your code	*Ctrl+Shift+F*
Auto-complete the word / show list of suggestions	*Ctrl + Space*
Run your app	*Ctrl+F11*
Rename an object	*Alt+Shift+R (you should select it first)*
Search through Java files	*Ctrl+H*
Navigate among the tabs	*Alt+left-arrow , Alt+right-arrow*

Helpful links

- http://stackoverflow.com/ - Why reinventing the wheel again?
- http://developer.android.com/training/index.html - Training, Advanced Training
- http://developer.android.com/guide/components/index.html - API Guides
- http://developer.android.com/tools/index.html - Android Developer Tools (ADT)
- http://developer.android.com/design/index.html - Design the UI
- http://developer.android.com/distribute/index.html - Publish, Distribute, Monetize