Steps for setting up the Android development environment

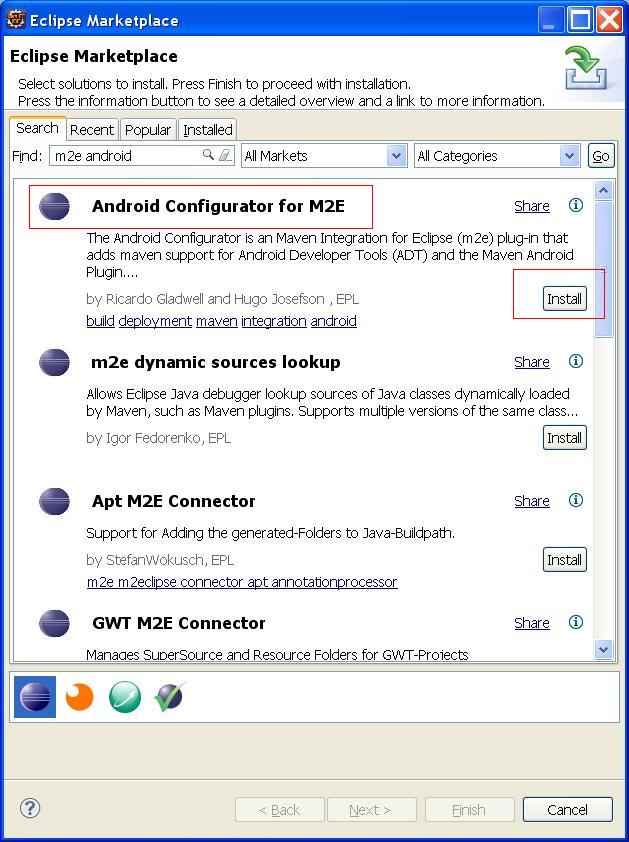
**Install Eclipse and EGit**

1. Download the latest eclipse and install it
2. Run Eclipse and go to Help🡪Eclipse Marketplace…
3. Enter “Egit” into the search and hit enter
4. Install the Egit plugin into eclipse and restart eclipse.



**Install the Maven Android plugin**

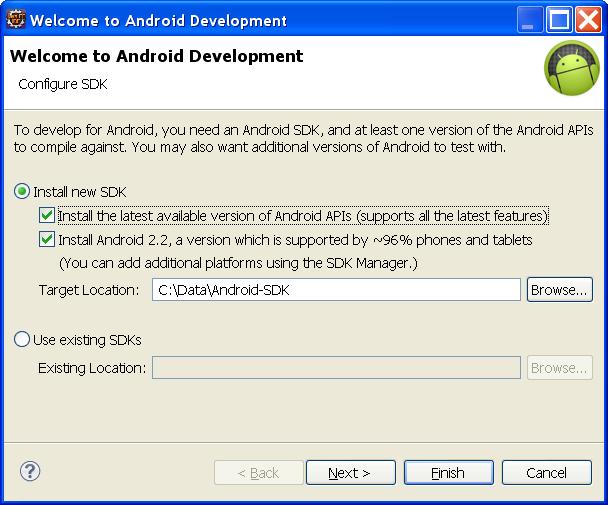
1. Run Eclipse and go to Help🡪Eclipse Marketplace…
2. Enter “m2e android” into the search and hit enter
3. Install *Android Configurator for M2E.*
4. Restart Eclipse when finished.



**Finish Setup for Android Configurator and install Android SDKs**

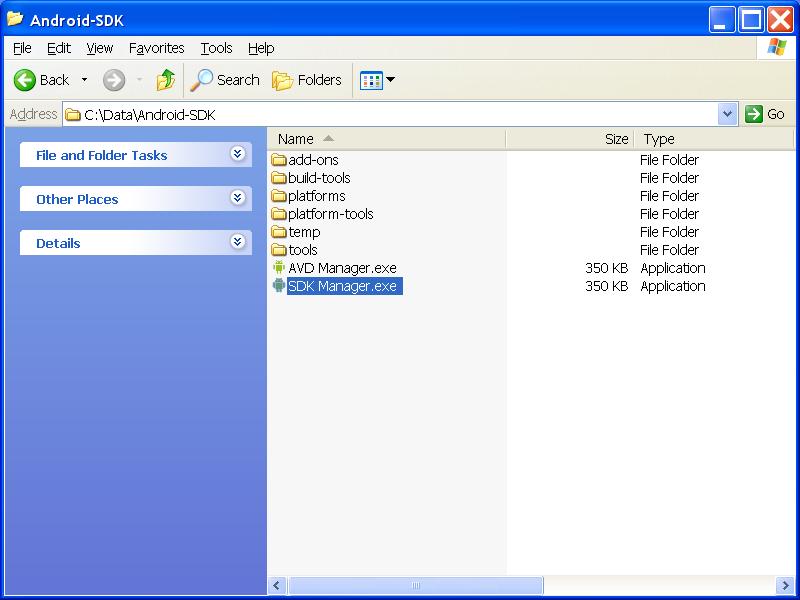
The Android Configurator for M2E eclipse plugin will allow you to select or install the Android SDKs you need on your system in order to develop this app. The next step is to allow this plugin to install the Android SDKs for you.

1. After Eclipse restarted it will prompt you for some initial preference that has something to do with the SDK location (can’t remember), just close the first small popup window.
2. Next the Android Development wizard will show. Fill it in as shown in the screen shot below. (You can use whatever Target Location you prefer)



This might take a little while to download.

After you finish this wizard you will have the Android SDK installed and you should open the Target Location where the SDK is installed because you need to use the SDK Manager to install more functionality in order for the project to build properly. If the SDK Manager does not open properly try adding the JAVA\_HOME environmental variable and point it to your installed jdk.



Inside of the SDK manager select the following two checkboxes for Android 4.3 (API 18), and Android SDK build tools 18.1.1. Click the Install packages button and install all of them. This will also take a while to download and install.

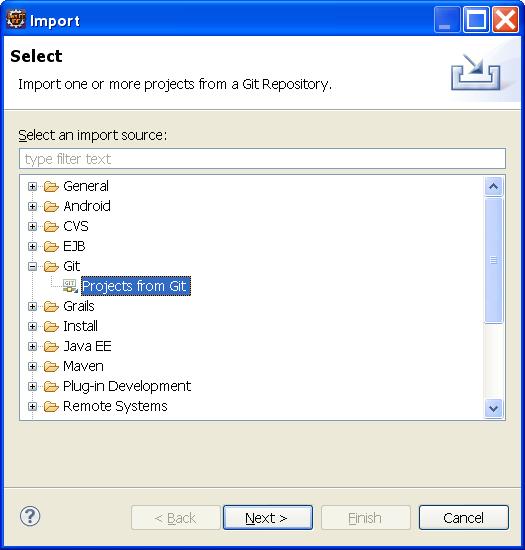
**Create Environmental Variable for Android VM Data**

In order to run the Android VM via eclipse, an environmental variable must be created for storing the data related to the VM.

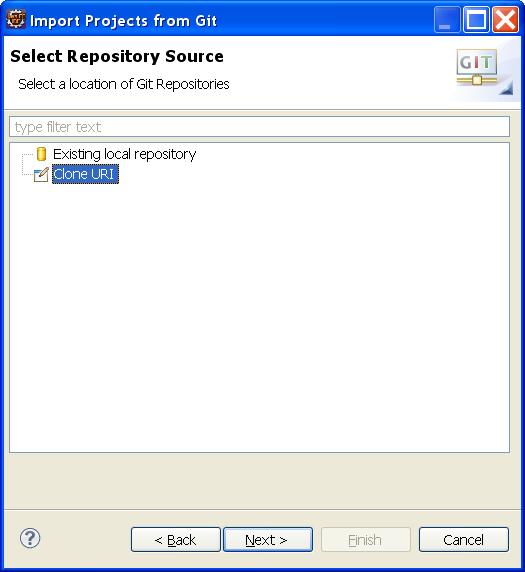
1. Create the environmental variable named: ANDROID\_SDK\_HOME
2. Give it the value C:\Data\Android-SDK (or whatever you want).
3. Click ok to all windows and then make sure to restart eclipse.

**Loading the remote project into your local environment**

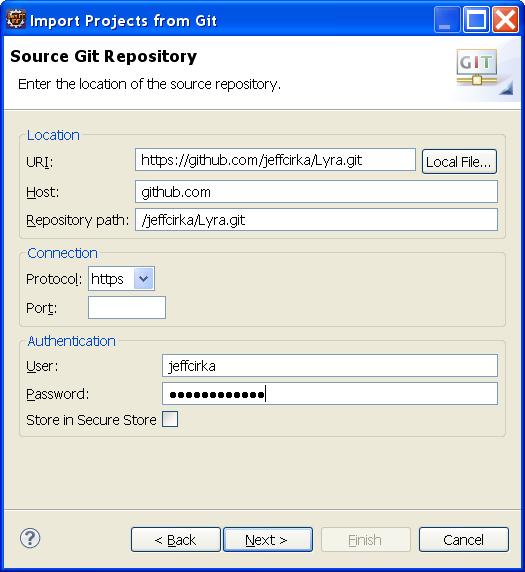
1. Inside of eclipse, right-click the empty area inside of the left pane (where projects usually reside) and select “Import”
2. Expand Git and select Projects from Git



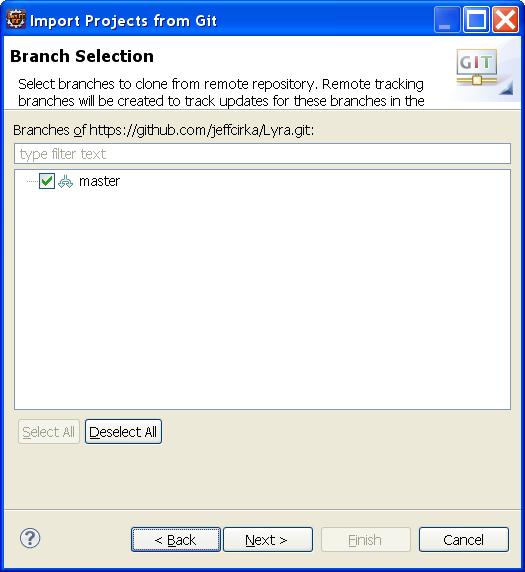
Click next



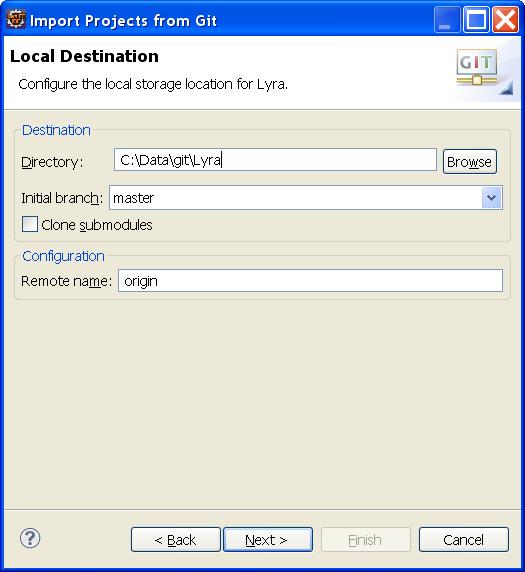
Select Clone URI and then click Next.



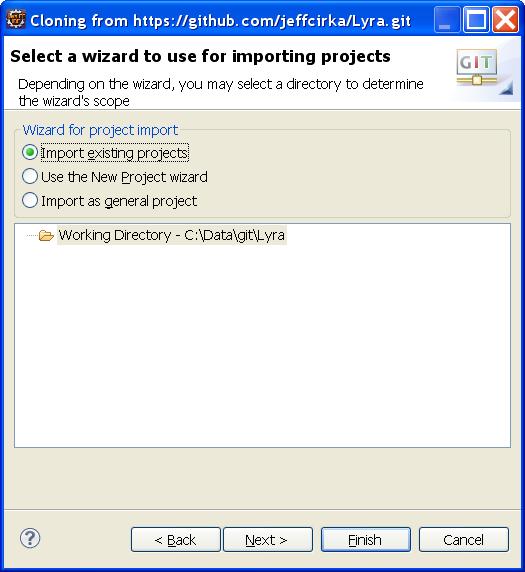
Fill in the URI of the remote Git Repository: https://github.com/jeffcirka/Lyra.git and enter your login information. Click Next.



If all goes well you should see master, leave it checked and click next.



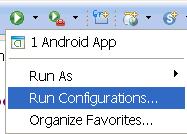
Fill in your Destination directory. This is where the project code will be stored locally on your machine from this point forward. You might prefer a path along the lines of: “…/workspaces/myworkspace/Lyra” Instead of the path shown in the image.



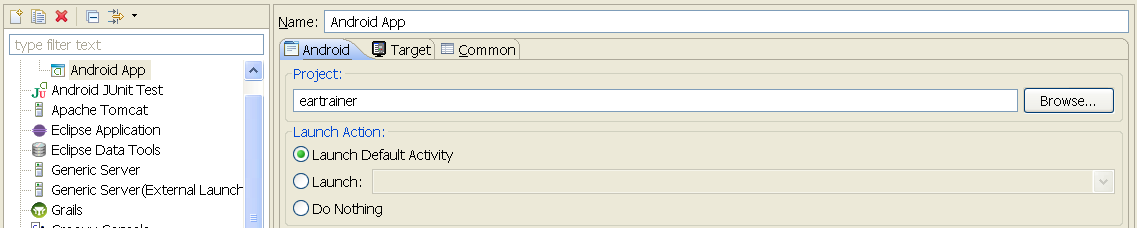
Leave the Import existing projects as shown above. Click Finish!

**Running the app for the first time**

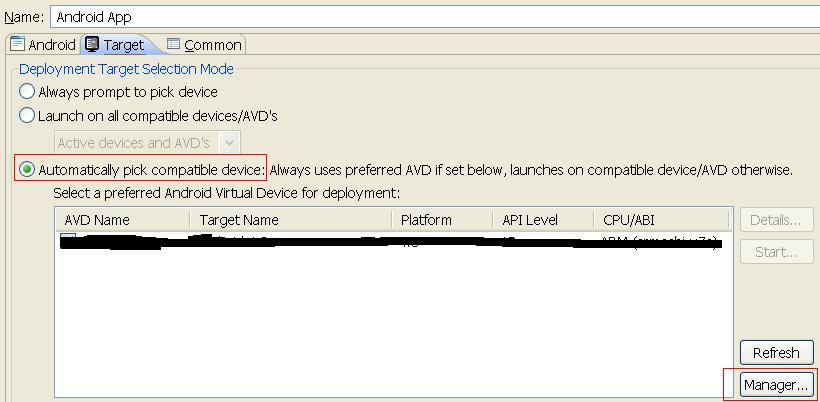
1. Open the java file: HelloAndroidActivity.java located in the src/main/java folder of the eartrainer project (it is in the com.lyra.eartrainer package)
2. Normally you would click the Play button to run the java file as a java application. However, we need to create a runtime configuration for this application. Click the Drop-down for the Play button in the toolbar and then click Run Configurations



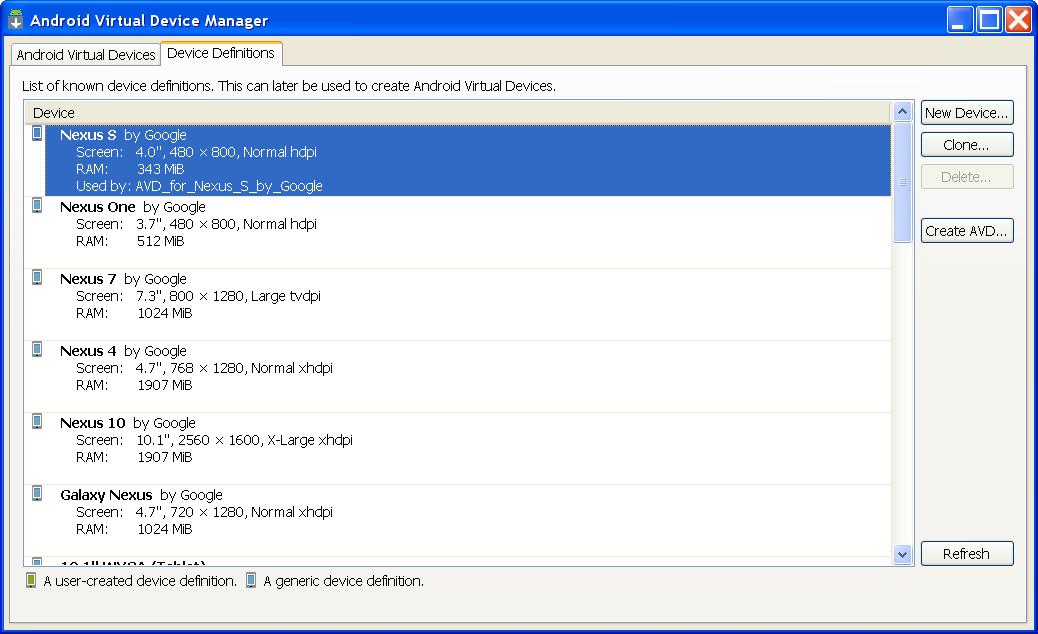
1. Name it “Android App” or whatever you want.
2. Click browse and select the eartrainer project for the “Project” field.



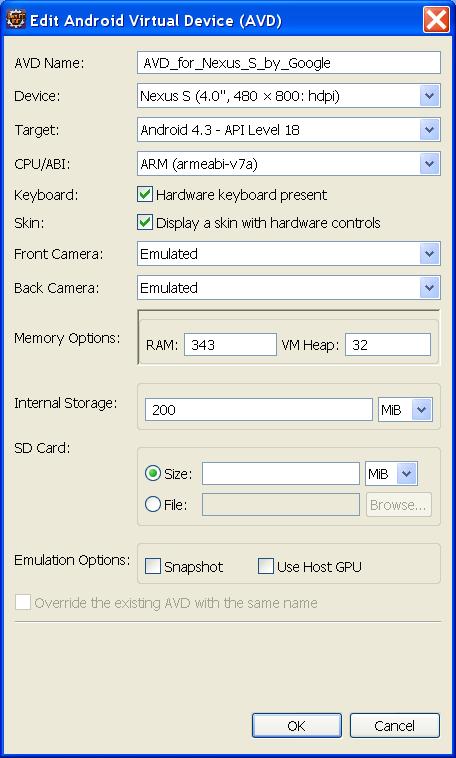
1. Select the “Target” tab and select the bottom radio option for “Automatically pick compatible device …”



1. Click Manage.
2. Select the Device Definitions tab and select the top device in the list (nexus s)

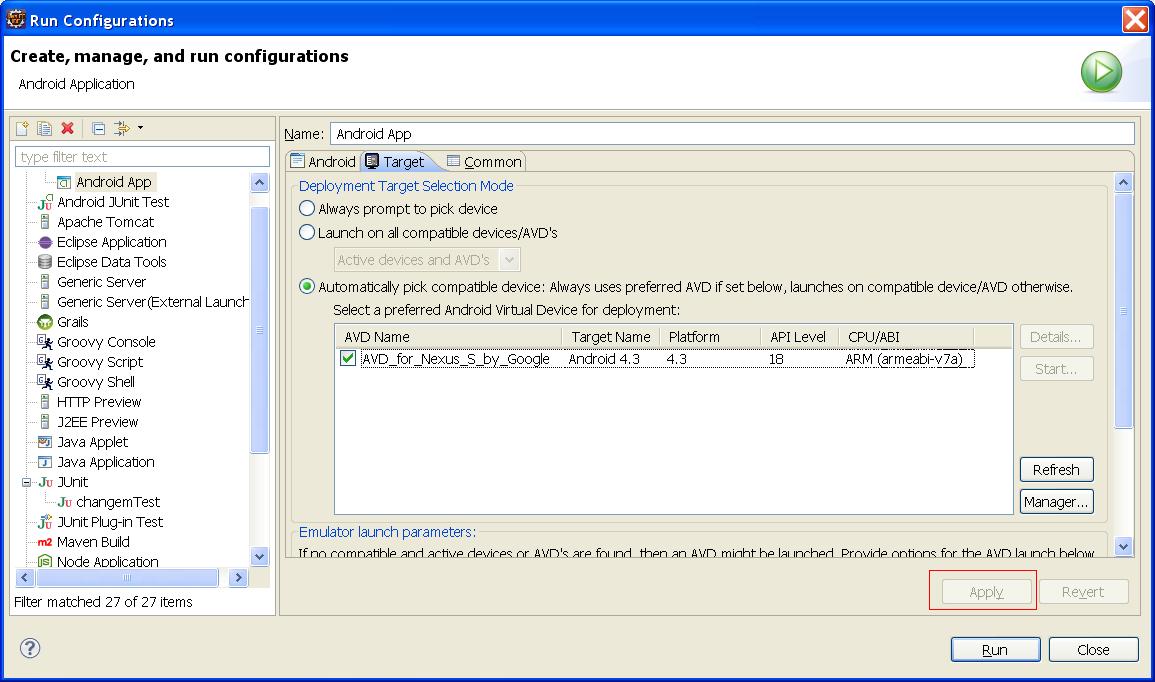


1. Click Create AVD
2. Fill it in as shown below

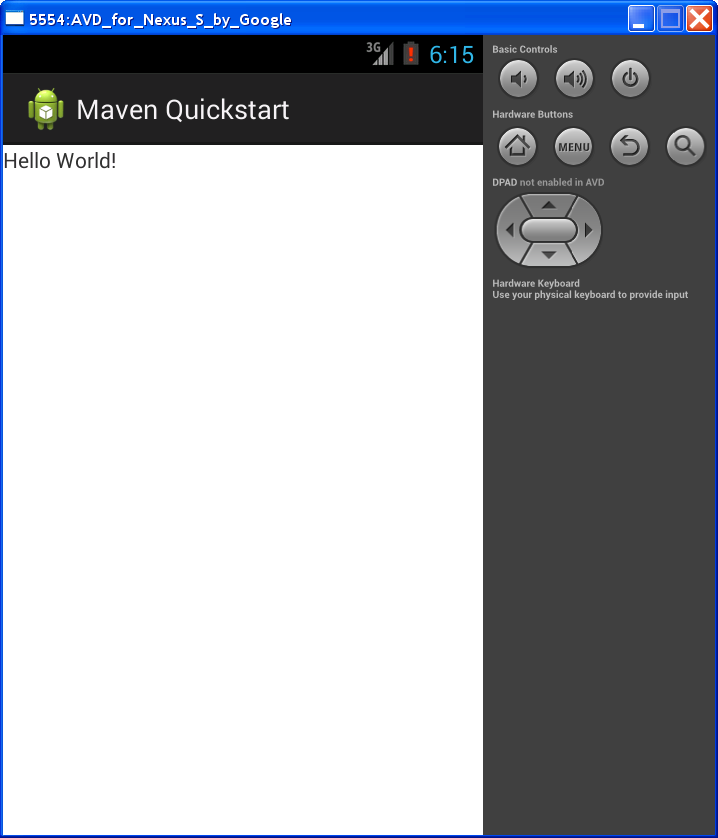


*Note: Make sure armeabi-v7a was selected for CPU/ABI. If you couldn’t select it then you need to re-open your Android SDK Manager and download the ARM EABI v7a under Android 4.3 (API 18).*

1. For faster VM speeds, the RAM and Internal Storage settings can be increased to 768 in this VM.
2. Click the “OK” button
3. Select the device and click Start if you’d like to test it.
4. Make sure that the device is checked in the list. Click Apply.



1. You should be able to run this application simply by clicking the play button in the toolbar.



Assuming everything worked fine then you should see a window like this. It will first need to boot the android VM which takes a really long time. Eventually it will deploy the apk to the VM and then automatically run the app on the VM as shown above. You might need to click somewhere in the window or on the menu button in order to get it to show after it first loads.