**MIND STORM SOFTWARE PVT LTD**

**Hands-On Exercise – ex01 - HelloWorld.doc**

**Objective**Write our first Android application. This exercise is meant to write your first application and in the process get familiar with the Eclipse based tools for writing Android applications. We will also enhance the application with a small feature.

**Assumptions**

* Development Environment for Android (Java SDK, Eclipse, Android SDK) has been setup successfully.
* You are familiar with using Eclipse.
* Android SDK 2.3.3 or 4.x is available and Android Virtual Devices are already created.

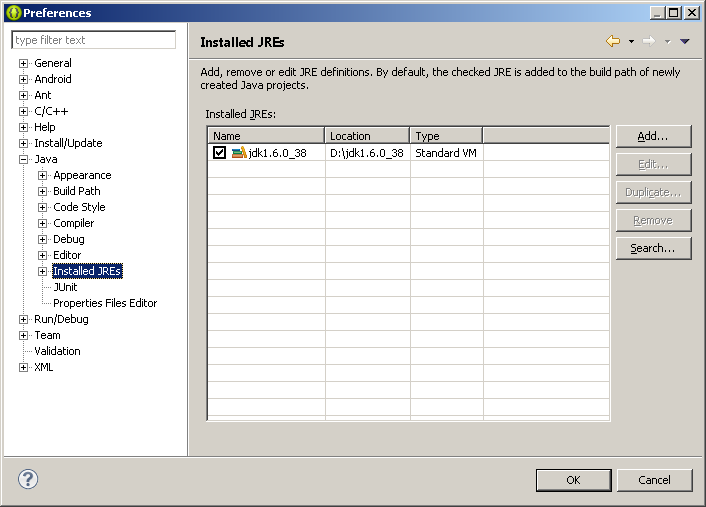
**If you have not setup the environment, please refer to Android Dev Environment Setup.docx for the instructions.**

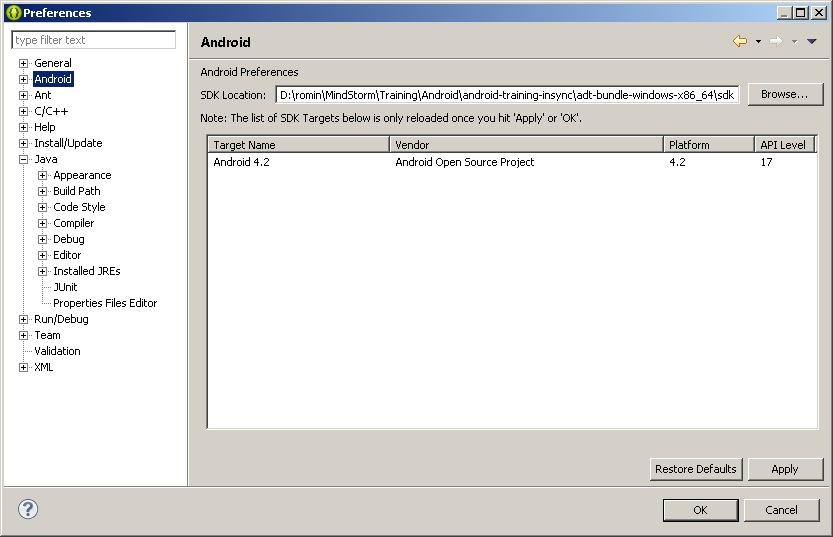
**Do not proceed without completing that.**

**Step by Step Instructions**

Let us do a few checks first to ensure that your environment is setup and running.

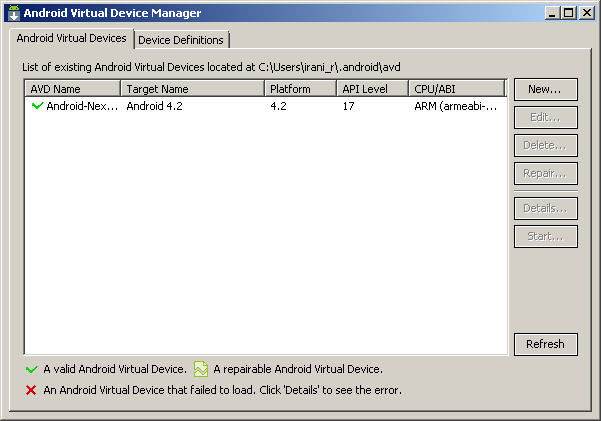
1. Launch Eclipse
2. Check if Java SDK is setup in Eclipse. Click on **Window 🡪 Preferences.** Then go to **Java 🡪 Installed JREs** as shown below. Ensure that a Java environment is selected.



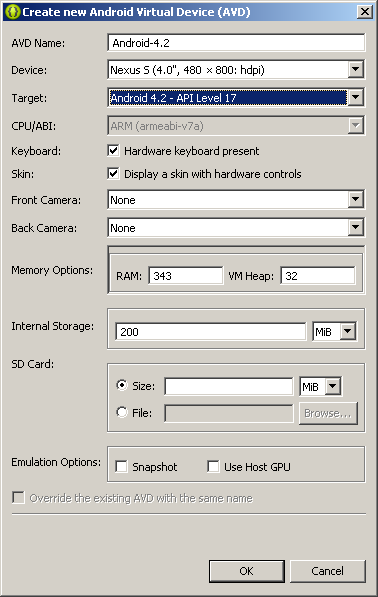
1. Check if Android SDK is setup in Eclipse. Click on **Window 🡪 Preferences.** Go to **Android** as shown below and ensure that the Android SDK versions are loaded as shown below. You may not have all the Android SDK Versions but at least you should have Android 4.2. 

**Create the AVD** if not created and Start it.

1. Click on **Window 🡪 Android Virtual Device Manager**. This will bring up the window as shown below:

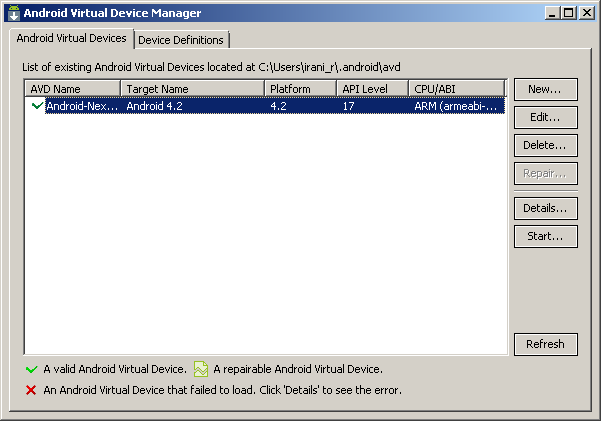


1. If there are no AVDs, create one for the appropriate Android version that you want to target. This can be done by clicking on **New** and filling up the details as shown below:

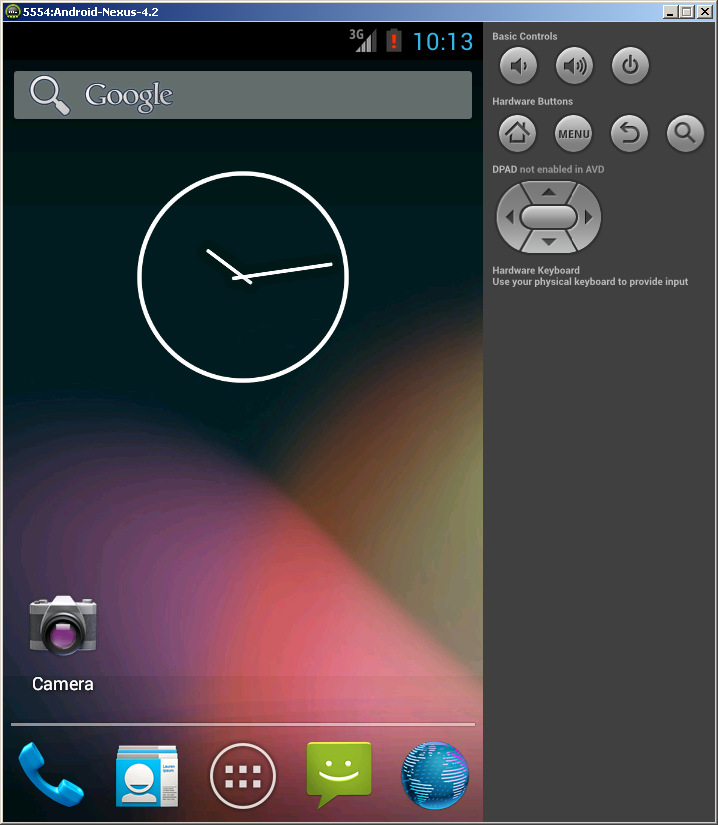


Click on **OK** to create the Android Virtual Device. It will now appear in the list of AVDs in the Android SDK and AVD Manager.

Select one of them and click on **Start**. Adjust the Screen size if needed:

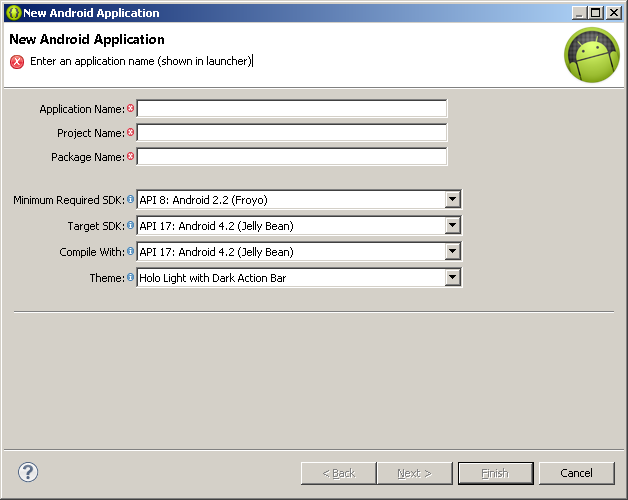


Once the AVD is started, unlock the screen and you should see a screen that looks like this:



**Create the Project:**

1. In Eclipse, click on **File 🡪 New 🡪 Android Application Project**.
2. This will bring up the New Project dialog as shown below:

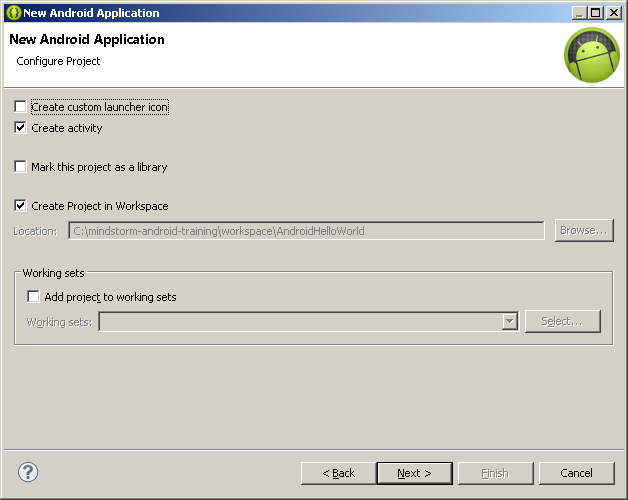


You will need to fill out the following entries:

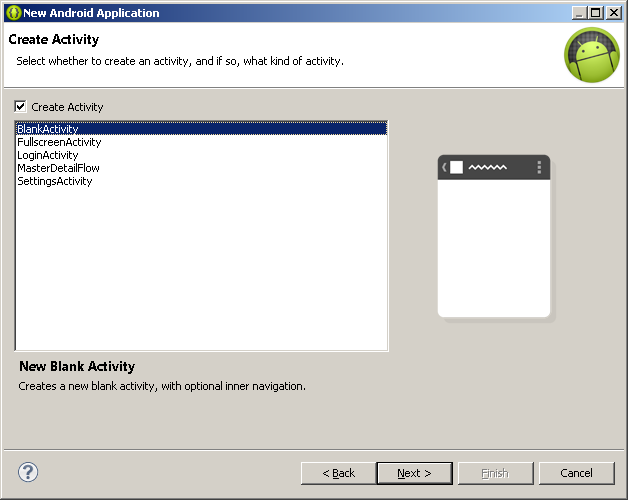
* Application Name: Please type in **AndroidHelloWorld**
* Project Name: This will get automatically filled up as the same like Application Name, leave it as is.
* Package Name: This will also get automatically generated. Change it to **com.mindstormsoftware.firstapp**

Leave all other settings as is and click on the **Next**  button.

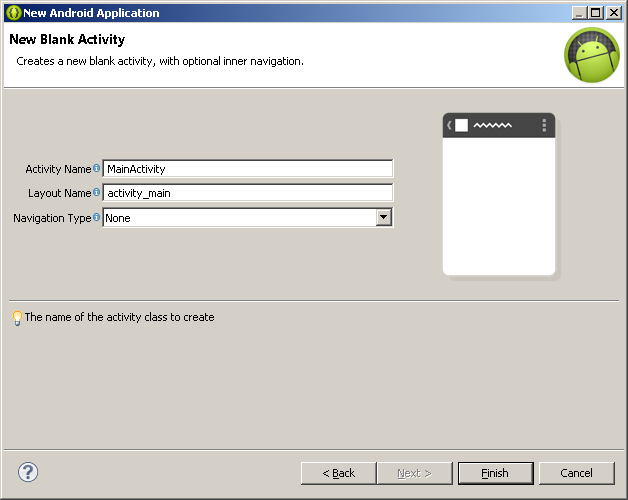
1. This will bring up the next step as shown below:



**Deselect** the **Create custom launcher icon** as shown above and click on **Next**. This will bring up the next screen as shown below:

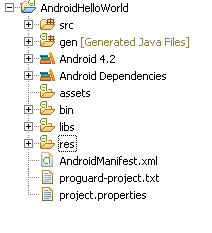


Go with the default option of creating a **BlankActivity** for now. Click on the **Next** button. This will bring up a screen where you need to provide the activity (main screen) details. Leave the entries at their default value.



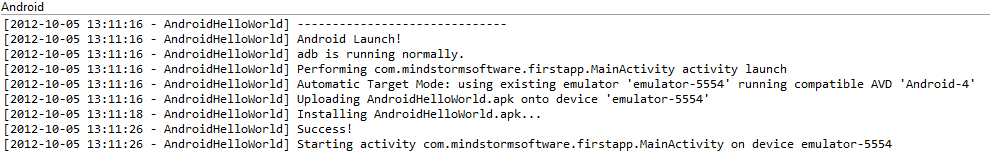
Click on **Finish**.

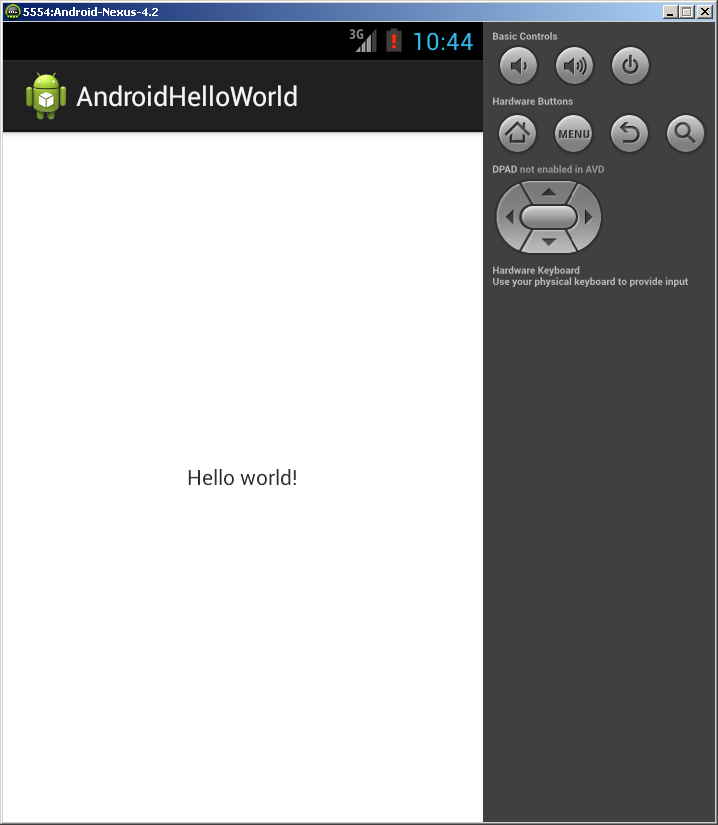
This will generate your Project. Look at the Project in the Project Explorer as shown below:



**Run the Application**

1. **Right click** the Project in the Project Explorer.
2. Select **Run As 🡪 Android Application**
3. Assuming that the Android Virtual Device for Android 4.x is running, the console will display the messages as shown below where it will install the App on the Emulator and launch the Activity.

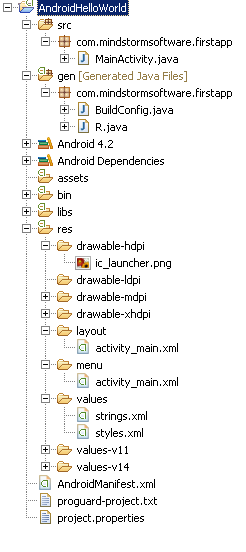


1. Switch to the **Emulator** and you should see your Application running now 

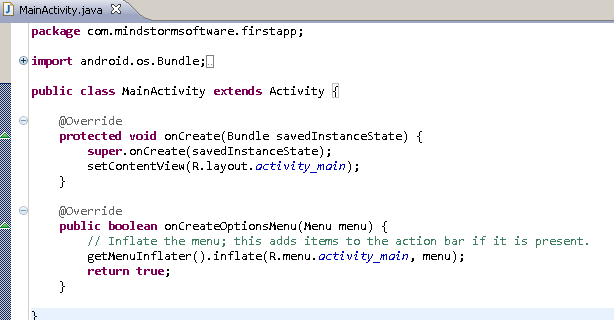
**Code Walkthrough**

Take a look at the code that is generated. The next few screens show the different parts of the project that you should focus on.

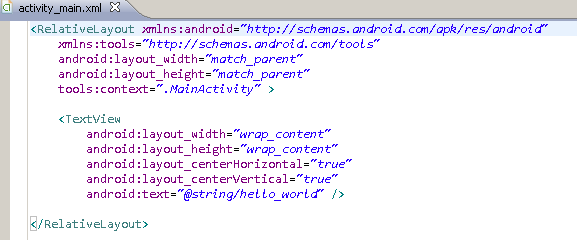
**Project Directory**



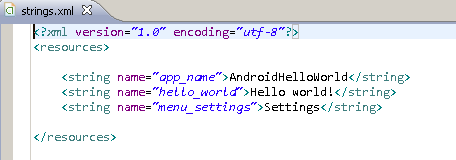
**Main Activity – MainActivity.java**



**Main Layout File – res/layout/activity\_main.xml**

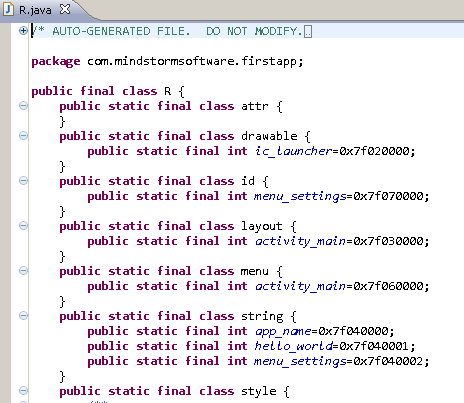


**All resource Strings – res/values/strings.xml**

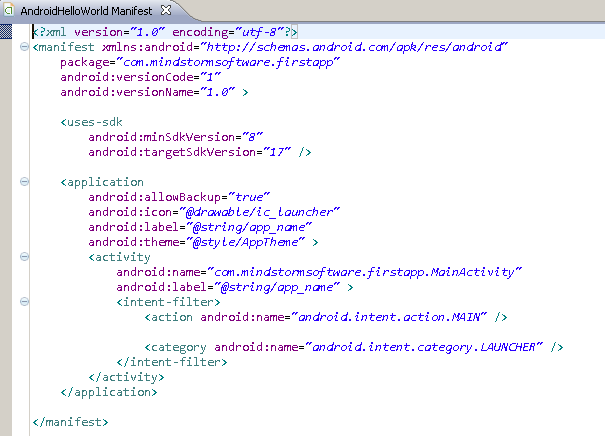


**The R file.**

All the resources are weaved into this file and can be referenced from code.



**Android Manifest file – AndroidManifest.xml**



**Try out the following**

1. We shall add a Button to the main activity and which on click will display a message to the user

**Steps**

1. Go to **res/layout/activity\_main.xml** and add the highlighted piece of text:

<RelativeLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

xmlns:tools=*"http://schemas.android.com/tools"*

android:layout\_width=*"match\_parent"*

android:layout\_height=*"match\_parent"*

tools:context=*".MainActivity"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:layout\_centerHorizontal=*"true"*

android:layout\_centerVertical=*"true"*

android:text=*"@string/hello\_world"* />

<Button

android:id=*"@+id/btnClickMe"*

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"Click Me"* />

</RelativeLayout>

1. Go to **src --> com.mindstormsoftware.firstapp** and **AndroidHelloWorldActivity.java** and enter the following piece of code as shown below:

package com.mindstormsoftware.firstapp;

import android.os.Bundle;

import android.app.Activity;

import android.view.Menu;

import android.view.View;

import android.view.View.OnClickListener;

import android.widget.Button;

import android.widget.Toast;

public class MainActivity extends Activity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

//Normal Button

Button btn1 = (Button)findViewById(R.id.btnClickMe);

btn1.setOnClickListener(new OnClickListener() {

public void onClick(View v) {

showToast("Button clicked");

}

});

}

@Override

public boolean onCreateOptionsMenu(Menu menu) {

// Inflate the menu; this adds items to the action bar if it is present.

getMenuInflater().inflate(R.menu.activity\_main, menu);

return true;

}

private void showToast(String msg) {

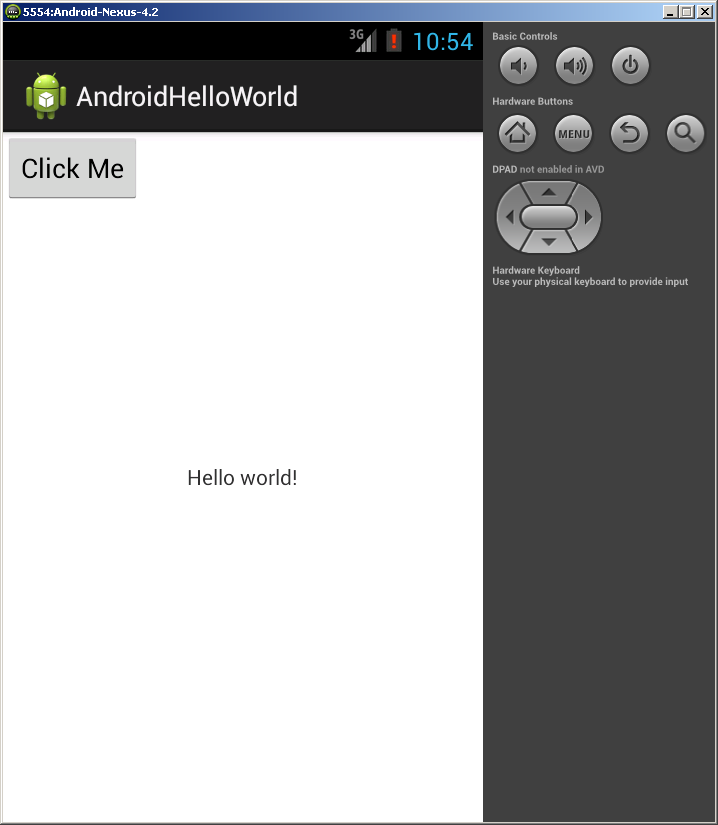
Toast.makeText(this, msg, Toast.LENGTH\_SHORT).show();

}

}

Resolve the package imports in Eclipse and save all your files.

1. Run the Project again. **Right-click Project** and select **Run As 🡪 Android** Application.
2. This should display the screen as shown below:



1. Clicking on the button will result in the message being shown in the Toast.