Android Development Tutorial

Nikhil Yadav
CSE40816/60816 - Pervasive Health
Fall 2011

Outline

- Setting up the Android Development Environment (Windows)
- Starters Android “Hello World” empty project
- Android project components
- Example: Tabbed application project
- Accessing Sensors on an Android Device: GPS, accelerometers and the Google API
- Database connections – Local SQLite and remote access

Android Development Environment

- Download the latest JDK (Java distribution)

- Download the Eclipse IDE (e.g. Galileo) from: www.eclipse.org/downloads/

- Install the Android SDK starter package from: http://developer.android.com/sdk/index.html

- In Eclipse, install the ADT (Android Developer Tools) plugin: http://developer.android.com/sdk/eclipse-adt.html

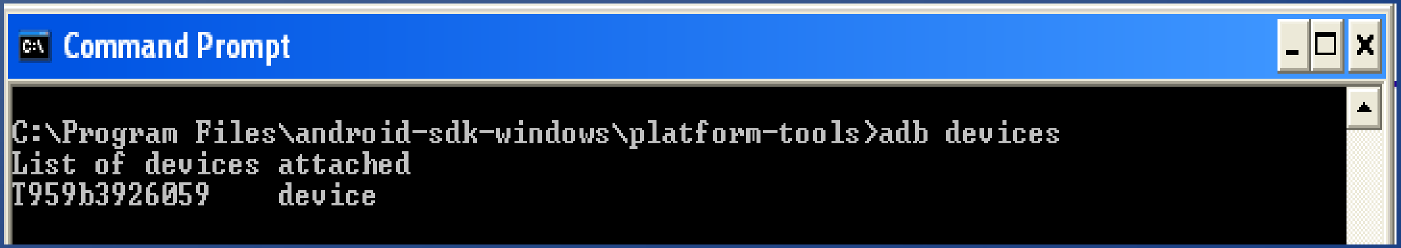
- Instructions on setting up the SDK and development Environment can be found on: http://developer.android.com/sdk/installing.html

The Android SDK starter package

- Not the full development environment, includes the core SDK tools

- Keep track of where you have downloaded this it is required for the ADT plugin setup, e.g. C:\Program Files\android-sdk-windows\platform-tools>

- Useful for “adb” commands



```
C:\Program Files\android-sdk-windows\platform-tools>adb devices
List of devices attached
T959b3926059    device
```

The ADT plugin

- Gives access to Android development tools from within the Eclipse IDE

- Automates the process of building a new Android project by setting up all the basic files needed for development

- Allows code signing of your app so it can be distributed

Installing the ADT plugin

(Link: http://developer.android.com/sdk/eclipse-adt.html#installing)

To simplify ADT setup, it is recommend installing the Android SDK prior to installing ADT

.**Eclipse 3.5 (Galileo) and 3.6 (Helios)**

1.Start Eclipse, then select Help>Install New Software....
2.Click Add, in the top-right corner.
3.In the *Add Repository* dialog that appears, enter "ADT Plugin" for the *Name* and the following URL for the *Location*:
https://dl-ssl.google.com/android/eclipse/
Note: If you have troubles try using "http" instead of "https"Click OK.
4.In the *Available Software* dialog, select the checkbox next to *Developer Tools* and click Next.
5.In the next window, you'll see a list of the tools to be downloaded. Click Next.
6.Read and accept the license agreements, then click Finish.

7.When the installation completes, restart Eclipse.

Configuring the ADT plugin

1. Select **Window**>**Preferences...**to open the Preferences panel (Mac OS X:**Eclipse**>**Preferences**).

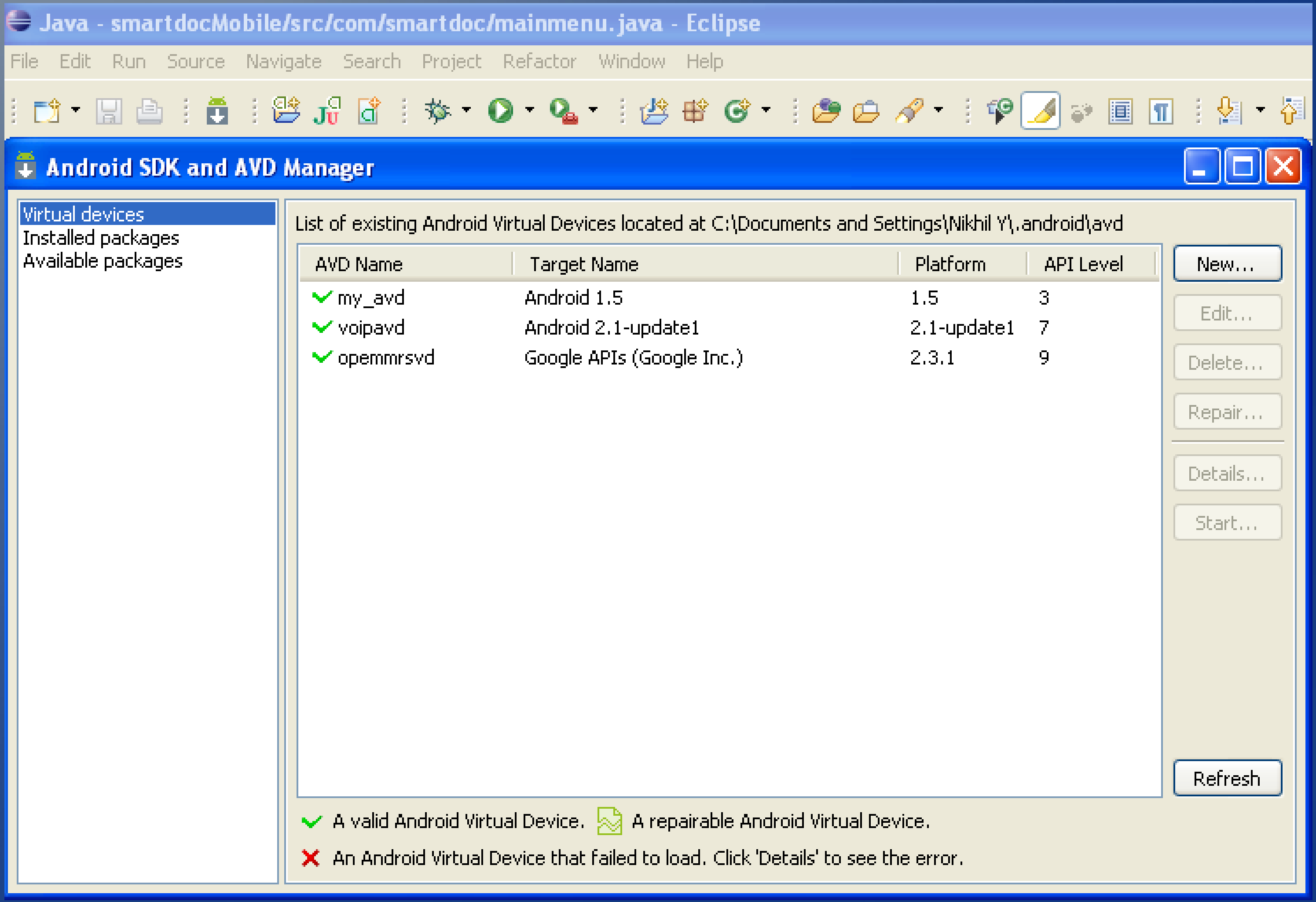
2. Select **Android** from the left panel.

3. For the *SDK Location* in the main panel, click **Browse...**and locate your downloaded SDK directory (e.g. C:\Program Files\android-sdk-windows\)

4. Click **Apply**, then **OK**

Android SDK and AVD manager

- Add your virtual development devices using this
- AVD is a target device simulator: no sensors



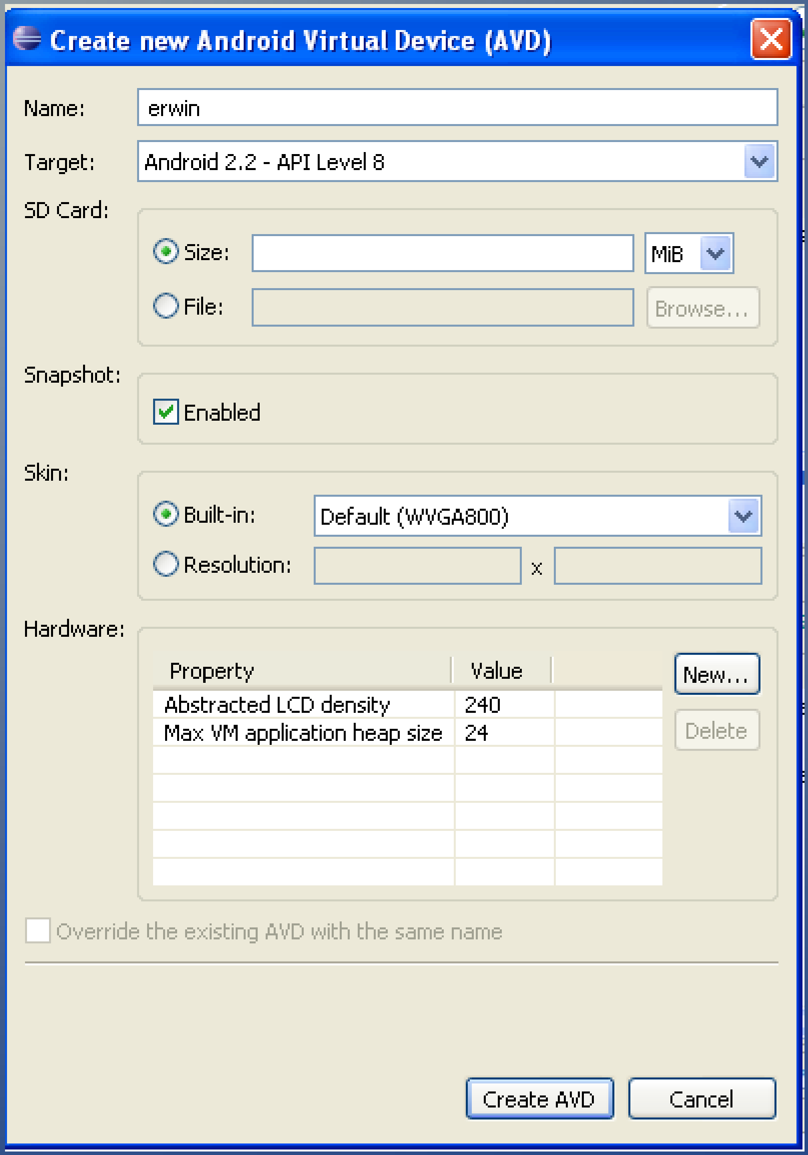
Java - smartdocMobile/src/com/smartdoc/mainmenu.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
Android SDK and AVD Manager
Virtual devices
Installed packages
Available packages
New...
Edit...
Delete...
Repair...
Details...
Start...
Refresh

List of existing Android Virtual Devices located at C:\Documents and Settings\Nikhil Y\.android\avd

AVD Name	Target Name	Platform	API Level
✓ my_avd	Android 1.5	1.5	3
✓ voipavd	Android 2.1-update1	2.1-update1	7
✓ opemmrsvd	Google APIs (Google Inc.)	2.3.1	9

✓ A valid Android Virtual Device. A repairable Android Virtual Device.

✗ An Android Virtual Device that failed to load. Click 'Details' to see the error.



Create new Android Virtual Device (AVD)

Name: erwin

Target: Android 2.2 - API Level 8

SD Card:

● Size: MiB

○ File: Browse...

Snapshot:

☑ Enabled

Skin:

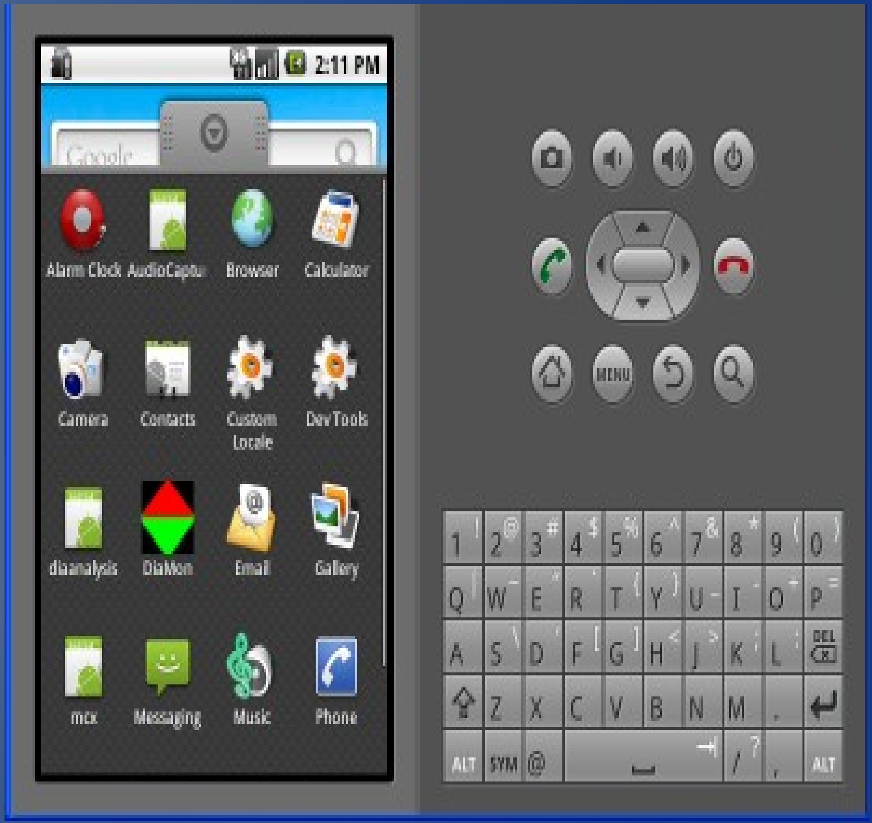
● Built-in: Default (WVGA800)

○ Resolution: x

Hardware:

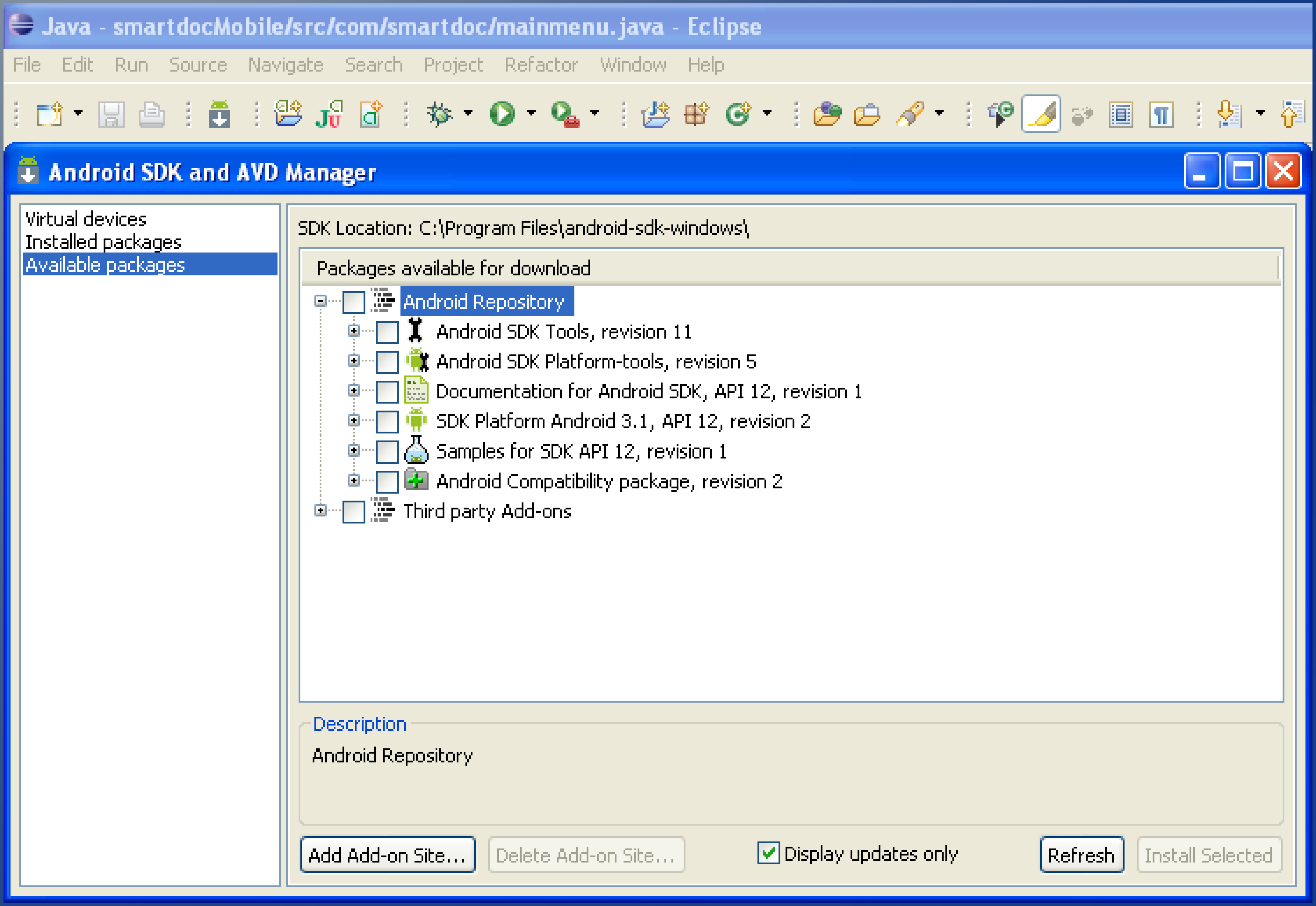
Property	Value
Abstracted LCD density	240
Max VM application heap size	24

□ Override the existing AVD with the same name



Android SDK and AVD Manager

- Used for updating Android SDK tools etc.



Java - smartdocMobile/src/com/smartdoc/mainmenu.java - Eclipse
File Edit Run Source Navigate Search Project Refactor Window Help
Android SDK and AVD Manager
Virtual devices
Installed packages
Available packages

SDK Location: C:\Program Files\android-sdk-windows\

Packages available for download

- □ Android Repository
  - □ Android SDK Tools, revision 11
  - □ Android SDK Platform-tools, revision 5
  - □ Documentation for Android SDK, API 12, revision 1
  - □ SDK Platform Android 3.1, API 12, revision 2
  - □ Samples for SDK API 12, revision 1
  - □ Android Compatibility package, revision 2
- □ Third party Add-ons

Description

Android Repository

Pointers

Update the system's PATH variable to recognize two folders inside your **android-sdk-winwows**. The first is: **tools** and the second is **platform-tools**.

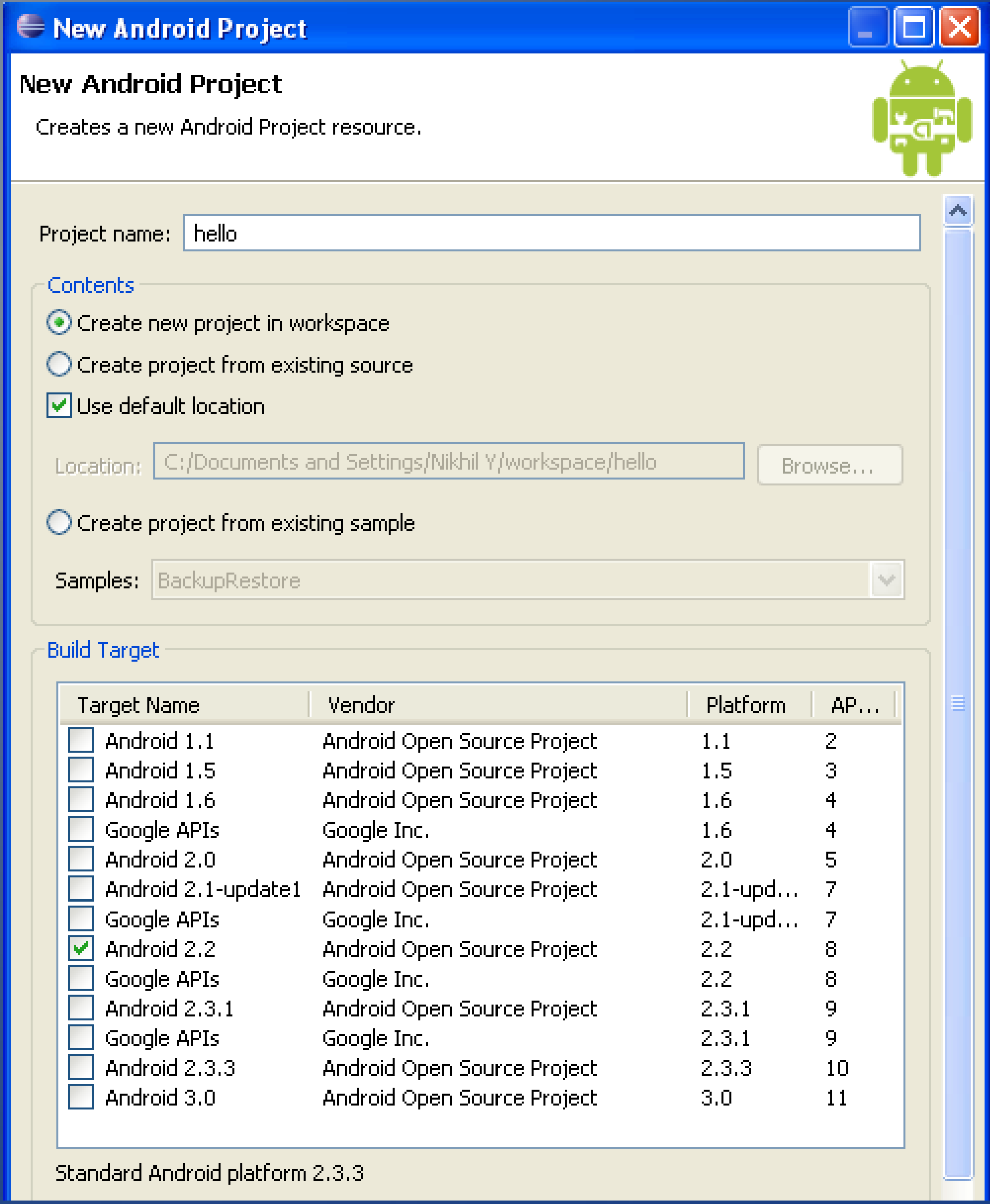
1. Windows > Start > Control Panel > System > Advanced > Environment Variables > System Variables > PATH > Edit
2. Add references to the sub-folders mentioned above. In this example: c:\android-sdk-windows\tools;C:\android-sdk-windows\platform-tools;
3. OK

Make sure Java and Eclipse Environments and path variables are setup correctly:

Some interesting tutorials on Android projects can be found on:

http://www.hometutorials.com/google-android.html

Starters: “Hello World”



New Android Project

Creates a new Android Project resource.

Project name: hello

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: C:/Documents and Settings/Nikhil Y/workspace/hello

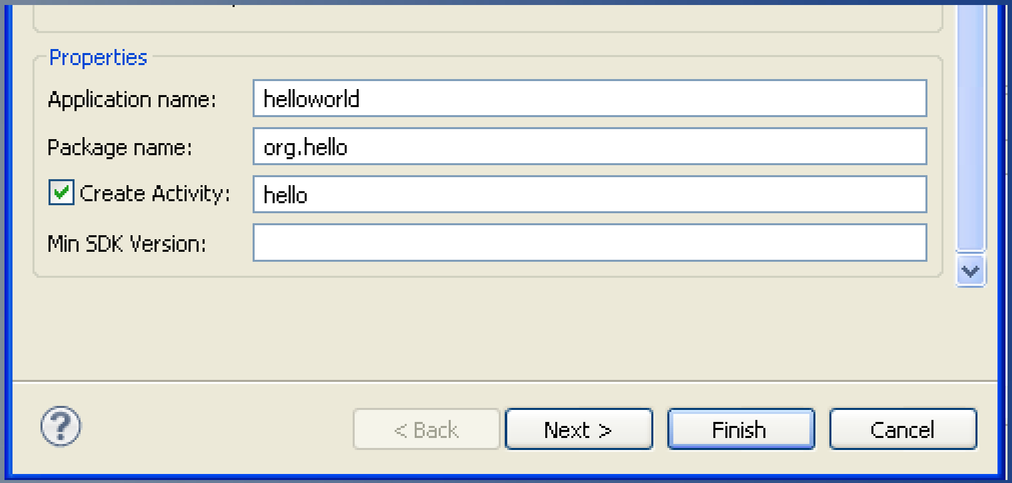
○ Create project from existing sample

Samples: BackupRestore

Build Target

	Target Name	Vendor	Platform	AP...
□	Android 1.1	Android Open Source Project	1.1	2
□	Android 1.5	Android Open Source Project	1.5	3
□	Android 1.6	Android Open Source Project	1.6	4
□	Google APIs	Google Inc.	1.6	4
□	Android 2.0	Android Open Source Project	2.0	5
□	Android 2.1-update1	Android Open Source Project	2.1-upd...	7
□	Google APIs	Google Inc.	2.1-upd...	7
☑	Android 2.2	Android Open Source Project	2.2	8
□	Google APIs	Google Inc.	2.2	8
□	Android 2.3.1	Android Open Source Project	2.3.1	9
□	Google APIs	Google Inc.	2.3.1	9
□	Android 2.3.3	Android Open Source Project	2.3.3	10
□	Android 3.0	Android Open Source Project	3.0	11

Standard Android platform 2.3.3



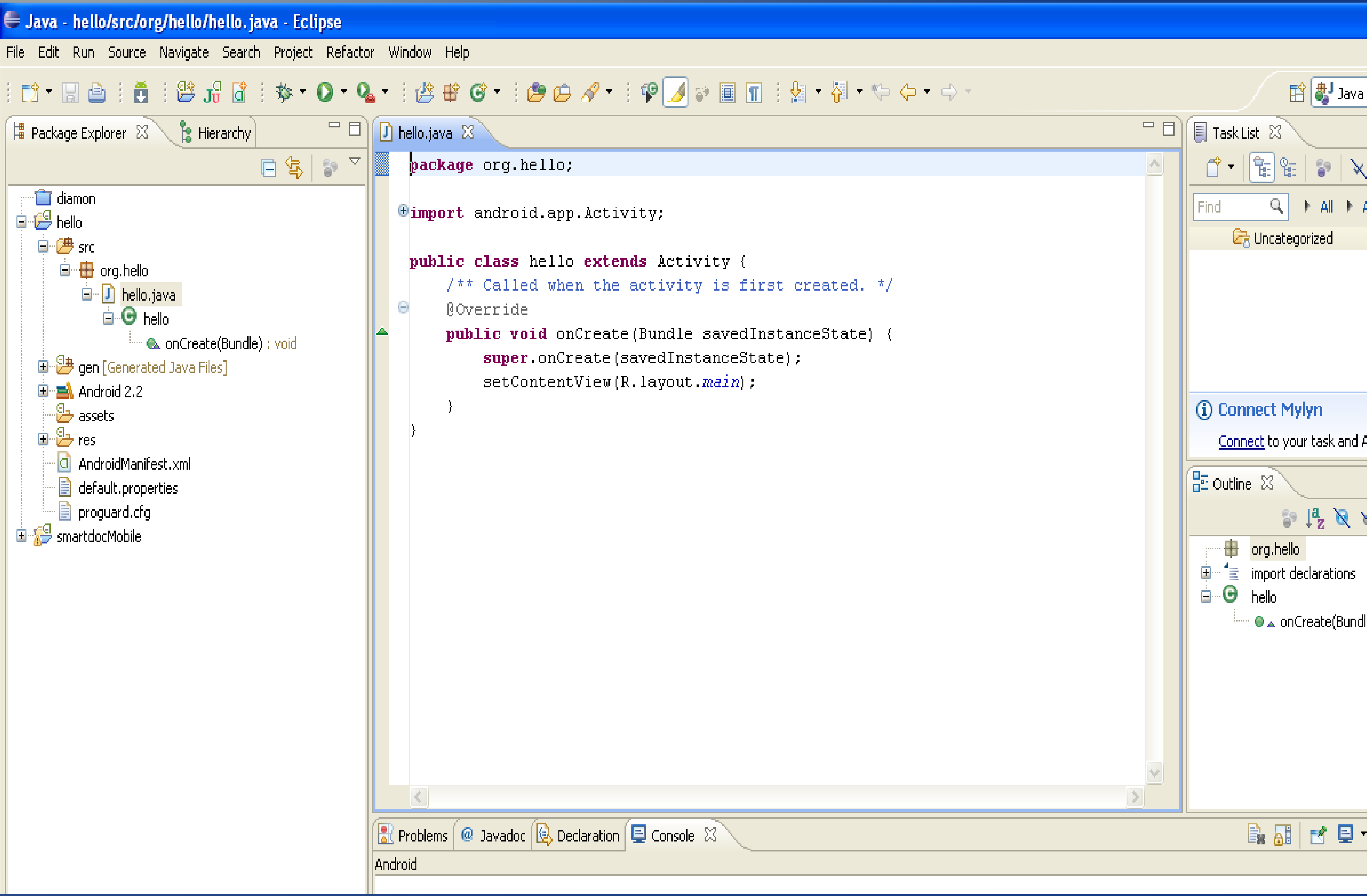
Properties

Application name: helloworld

Package name: org.hello

☑ Create Activity: hello

Min SDK Version:



```java
package org.hello;

import android.app.Activity;

public class hello extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Android Project Components

Once you complete the New Project Wizard, ADT creates the following folders and files in your new project:

•**src/** Includes your stub Activity Java file. All other Java files for your application go here.

•***<Android Version>/*** (e.g., Android 1.5/) Includes the android.jar file that your application will build against.

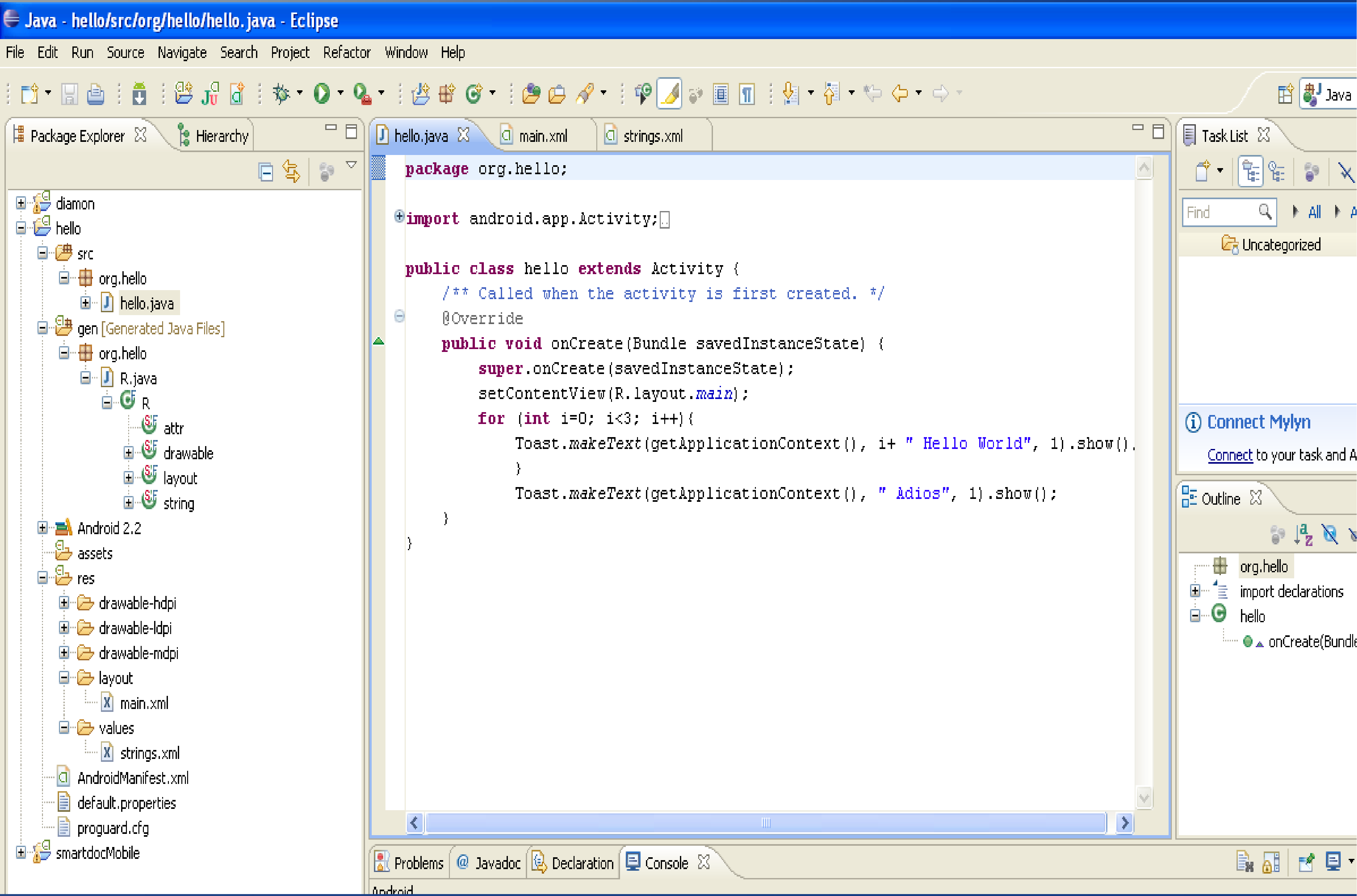
•**gen/** This contains the Java files generated by ADT, such as your R.java file and interfaces created from AIDL files.

•**assets/**This is empty. You can use it to store raw asset files.

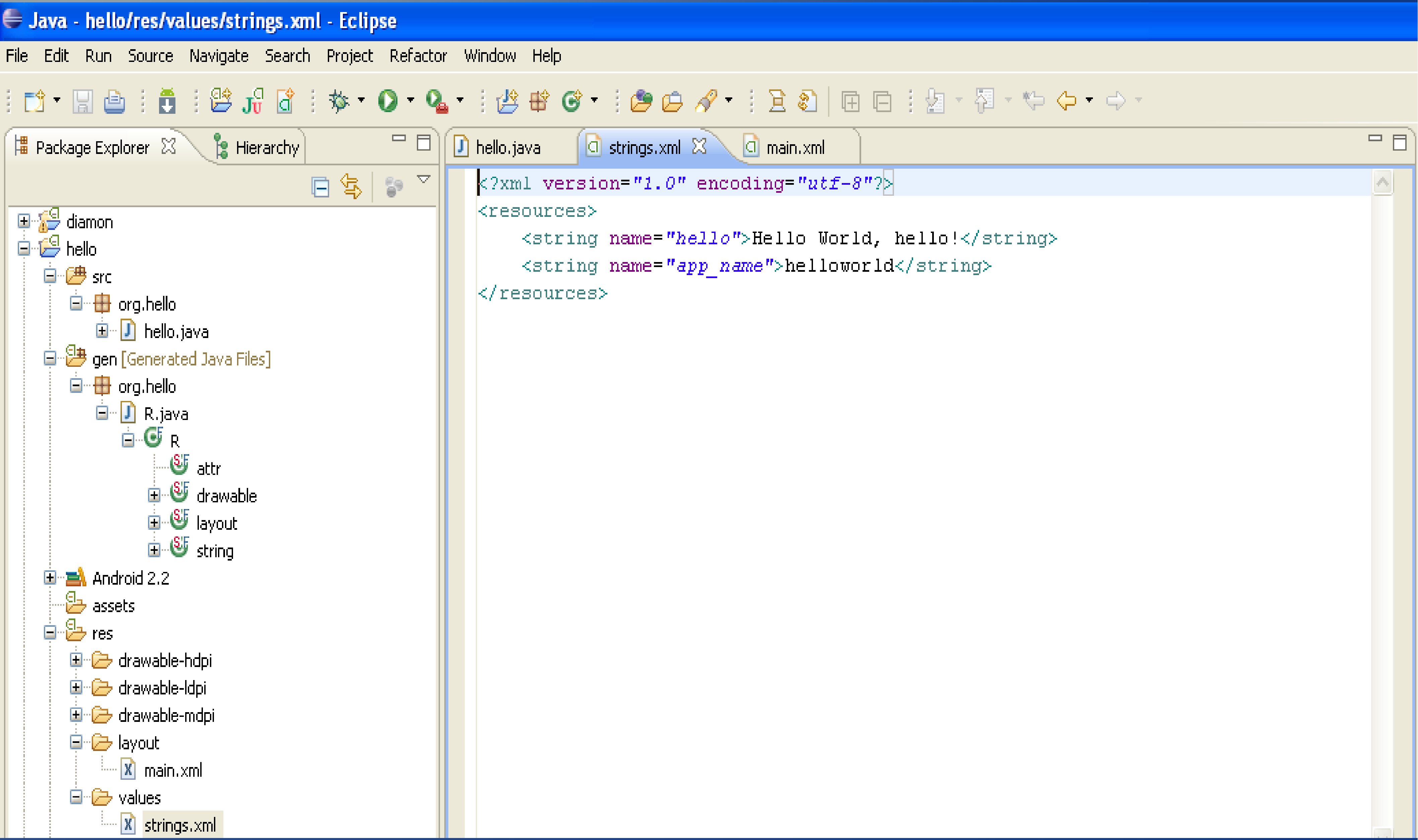
•**res/**A folder for your application resources, such as *drawable* files, *layout* files, *string* values, etc.

•**AndroidManifest.xml** The Android Manifest for your project.

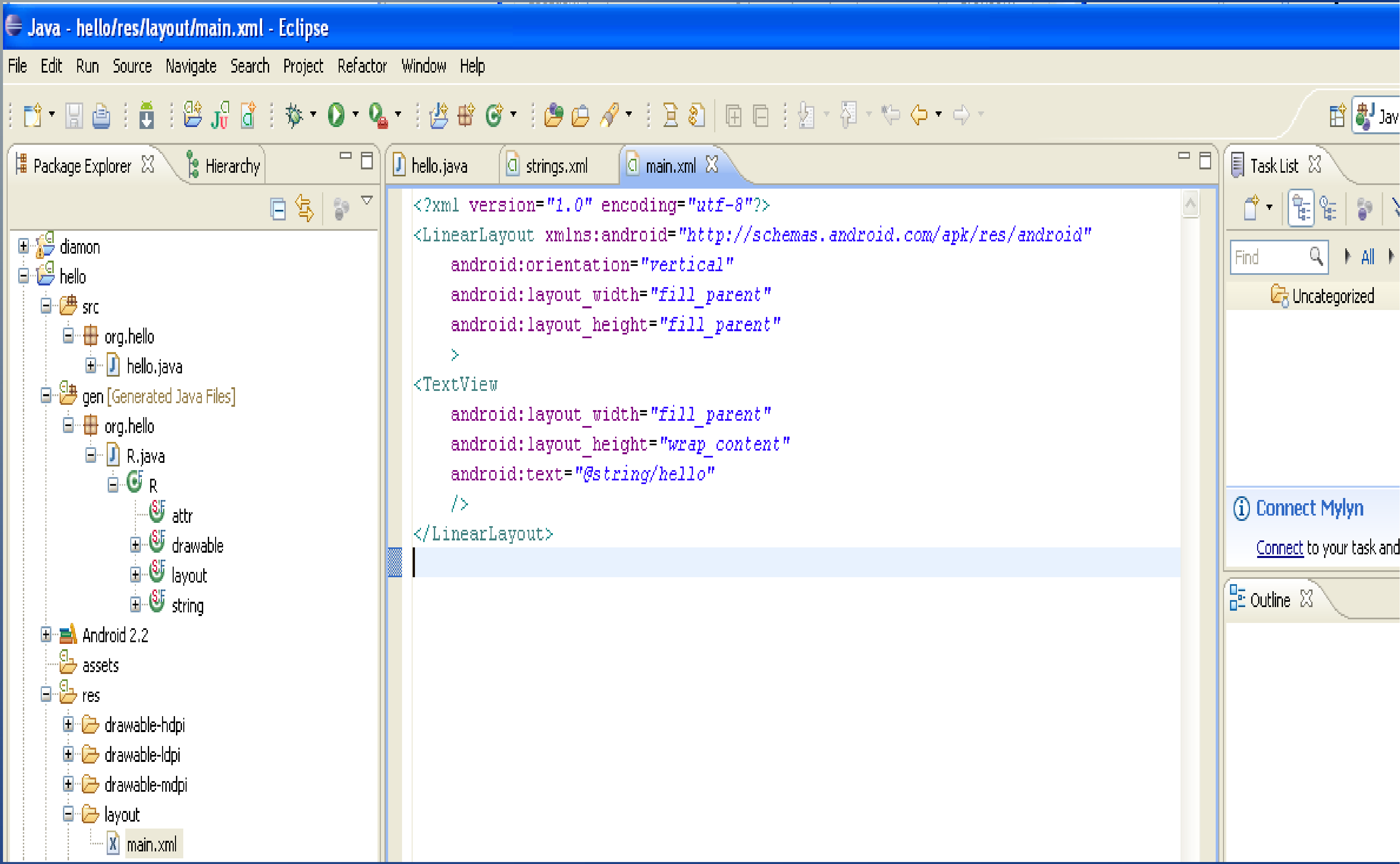
•**default.properties**This file contains project settings, such as the build target.



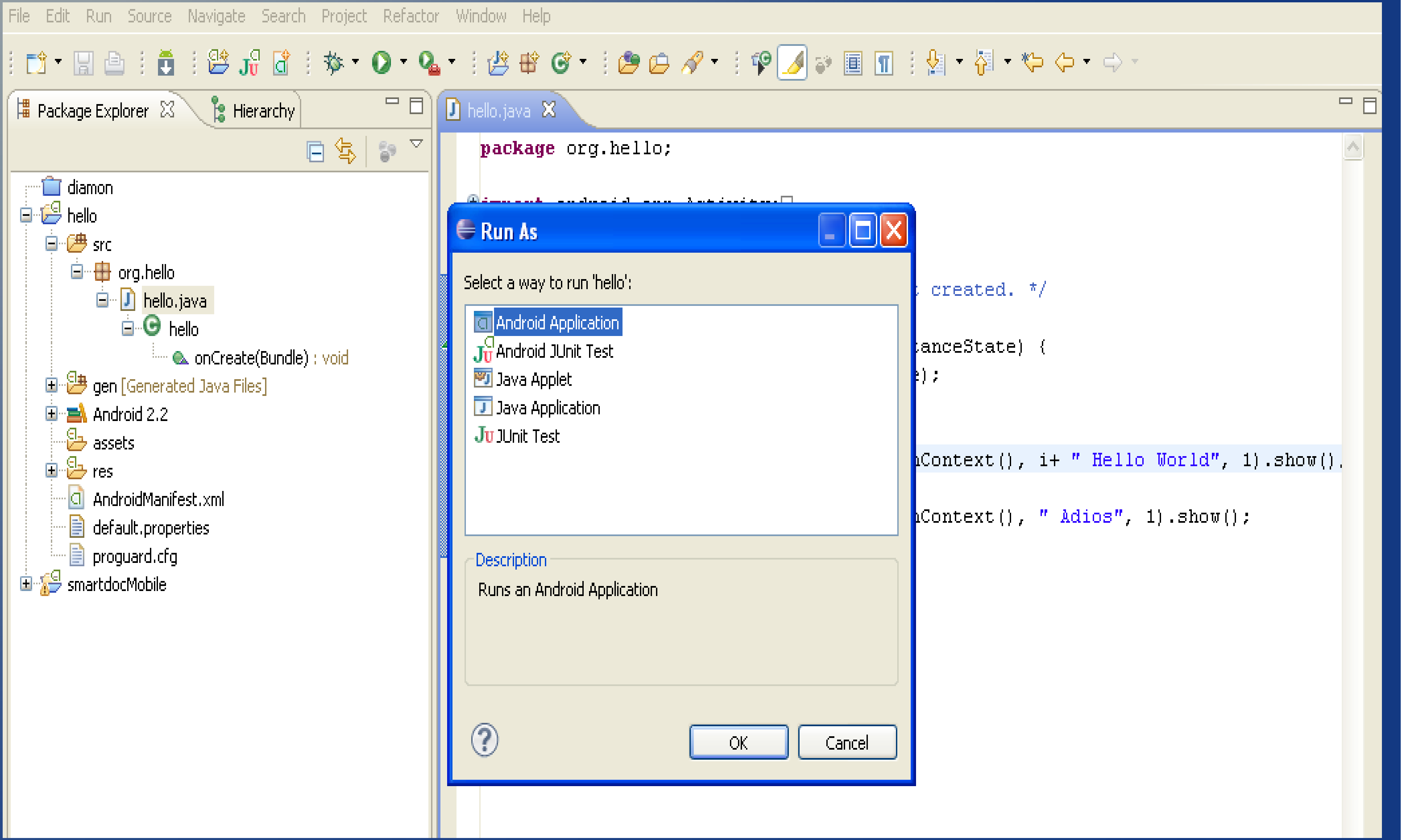
```
package org.hello;

import android.app.Activity;

public class hello extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        for (int i=0; i<3; i++){
            Toast.makeText(getApplicationContext(), i+ " Hello World", 1).show().
            }
            Toast.makeText(getApplicationContext(), " Adios", 1).show();
    }
}
```



```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Hello World, hello!</string>
    <string name="app_name">helloworld</string>
</resources>
```



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
</LinearLayout>
```



```
package org.hello;

import android.app.Activity;
```

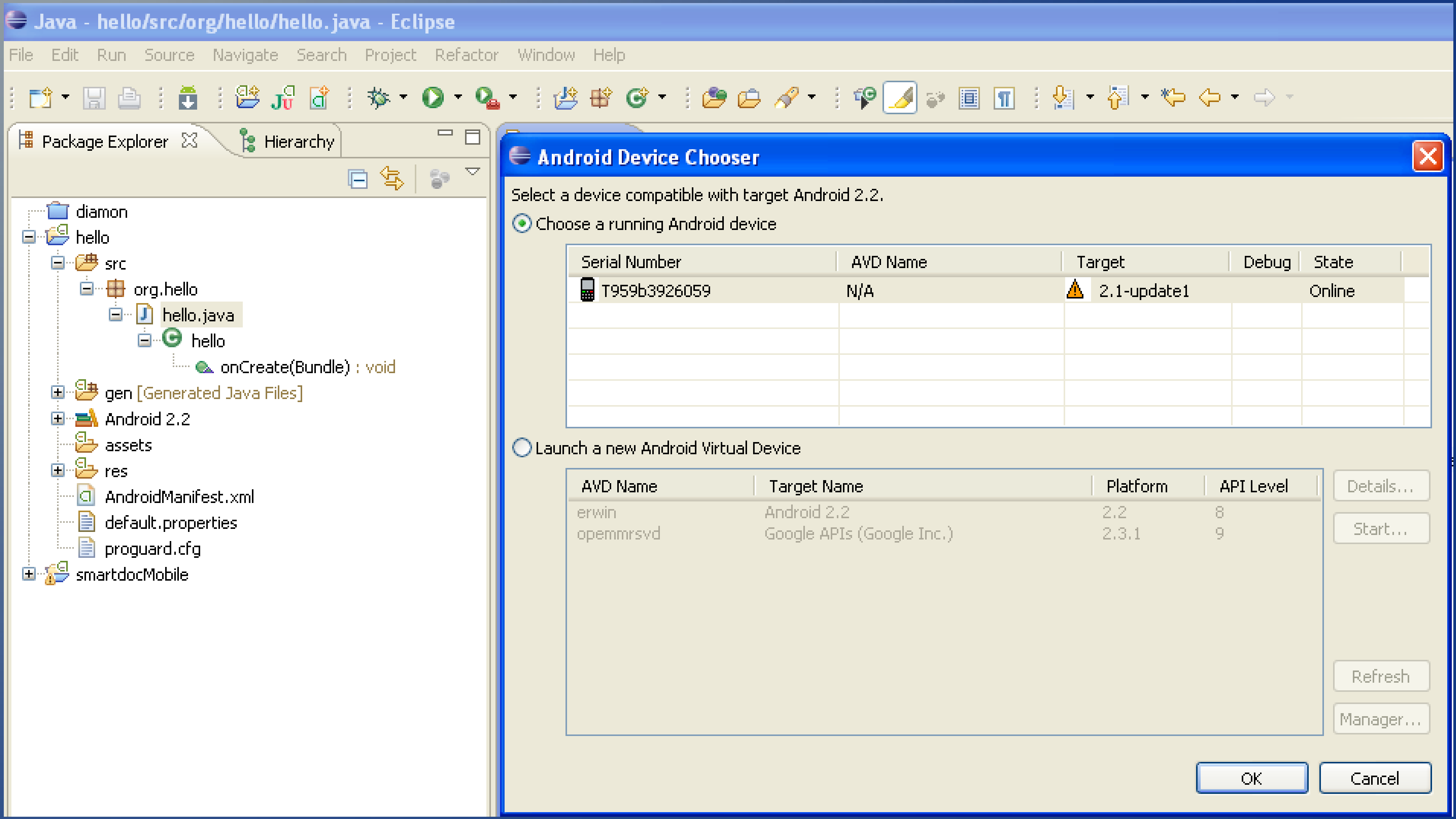
```
 created. */
anceState) {
e);
nContext(), i+ " Hello World", 1).show().
nContext(), " Adios", 1).show();
```

Run As

Select a way to run 'hello':

Description

Runs an Android Application



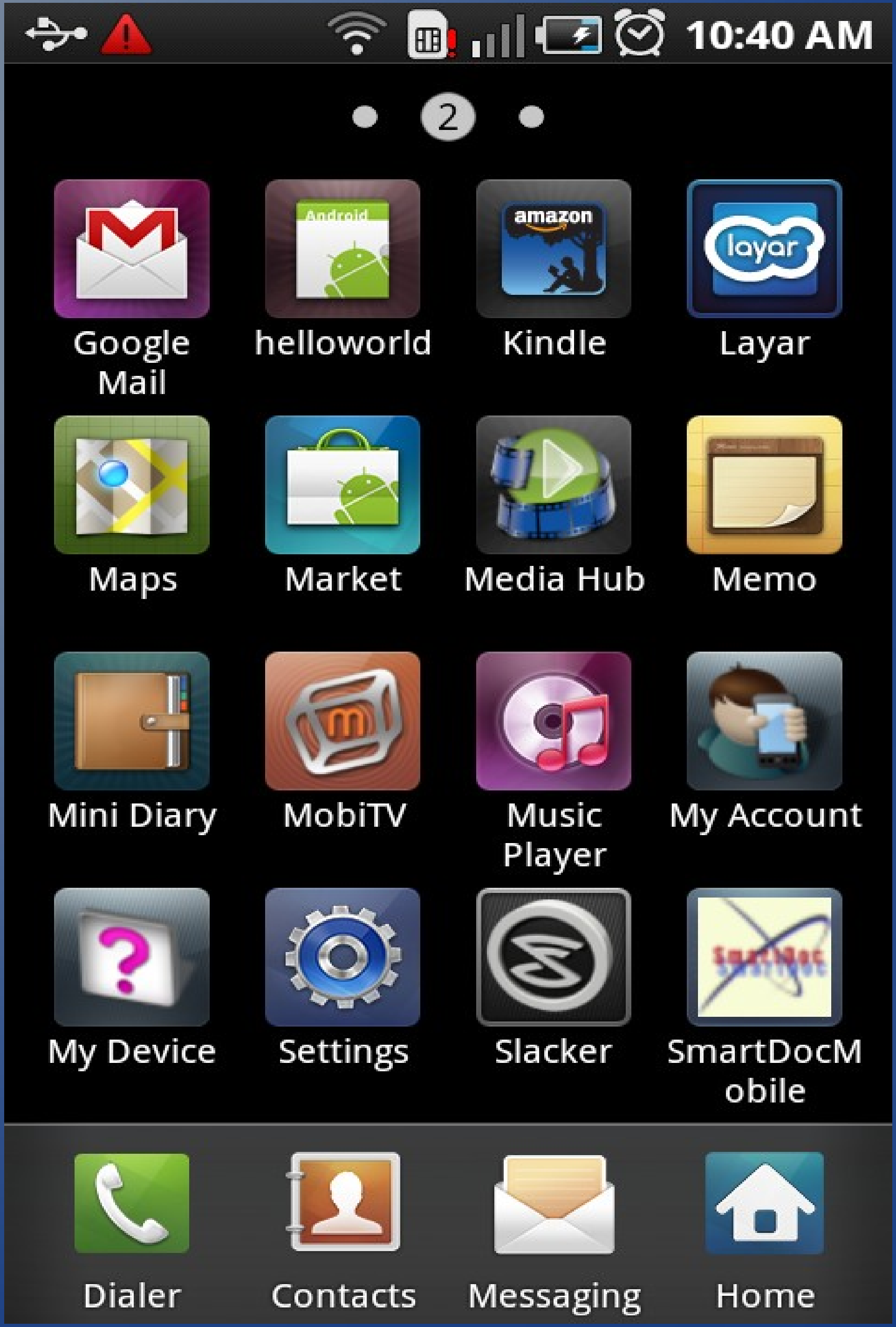
Select a device compatible with target Android 2.2.

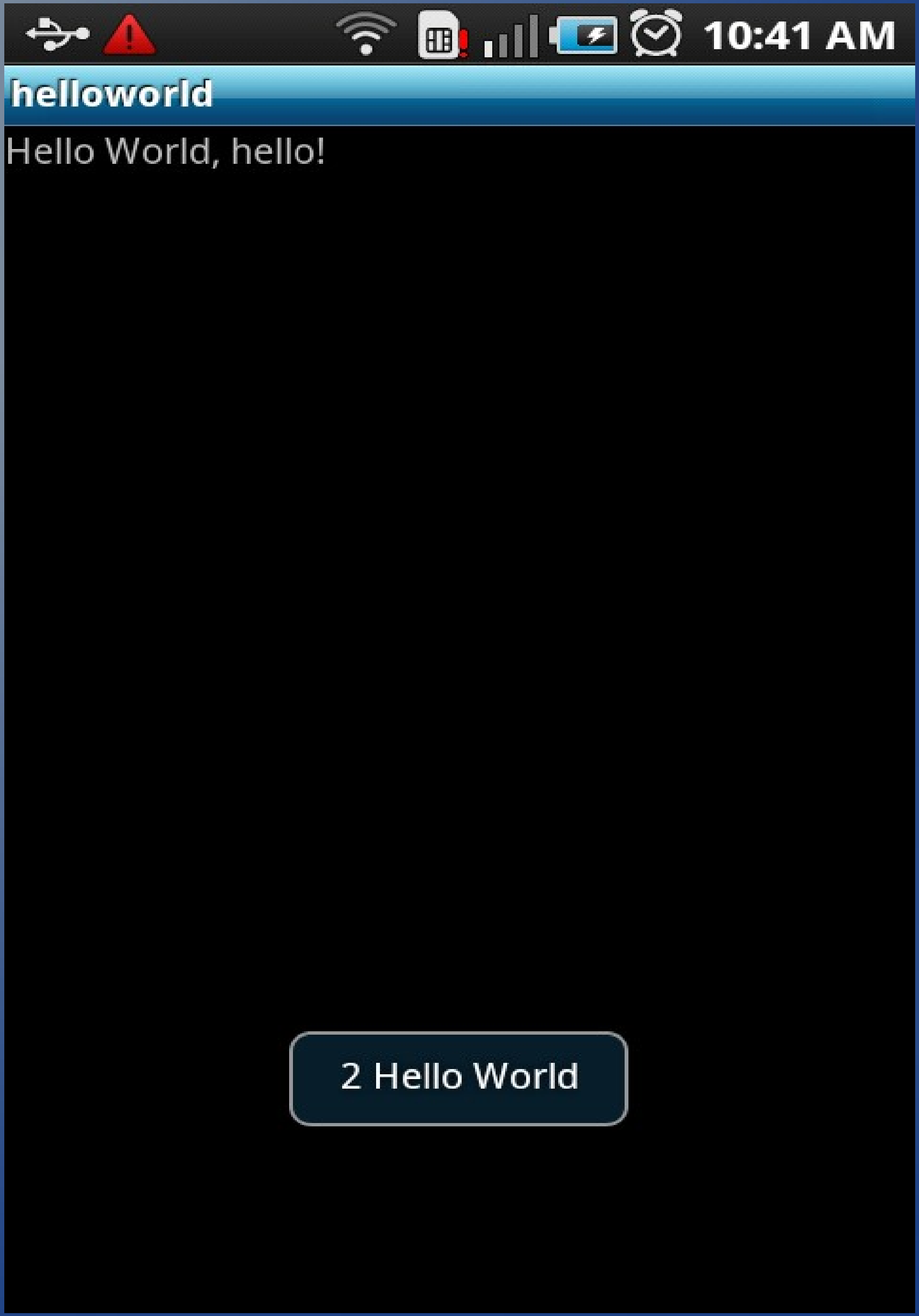
● Choose a running Android device

Serial Number	AVD Name	Target	Debug	State
T959b3926059	N/A	⚠ 2.1-update1		Online

○ Launch a new Android Virtual Device

AVD Name	Target Name	Platform	API Level
erwin	Android 2.2	2.2	8
opemmrsvd	Google APIs (Google Inc.)	2.3.1	9


10:40 AM
2
Google Mail
helloworld
Kindle
Layar
Maps
Market
Media Hub
Memo
Mini Diary
MobiTV
Music Player
My Account
My Device
Settings
Slacker
SmartDocMobile
Dialer
Contacts
Messaging
Home

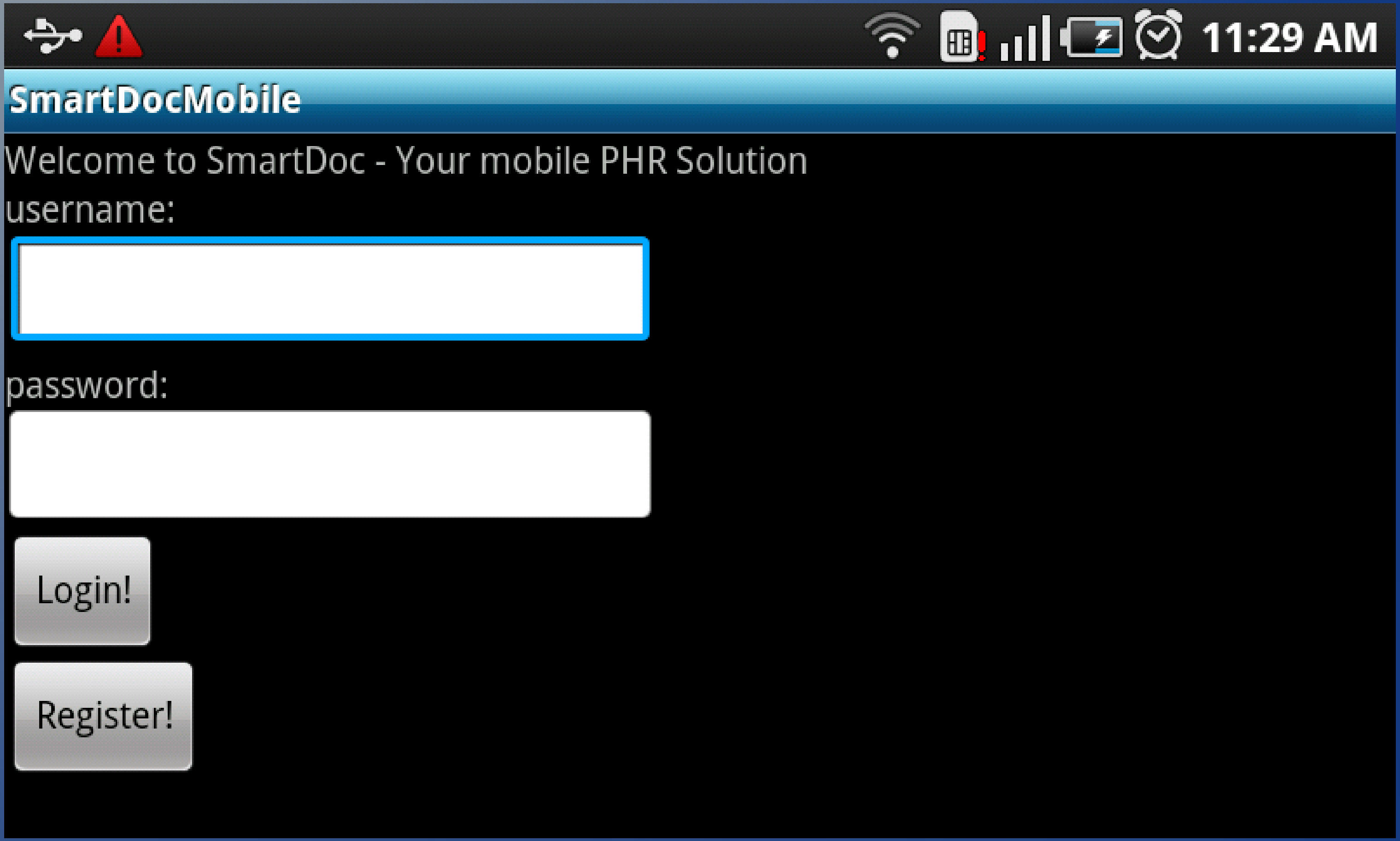


helloworld

Hello World, hello!

2 Hello World

A Tabbed application

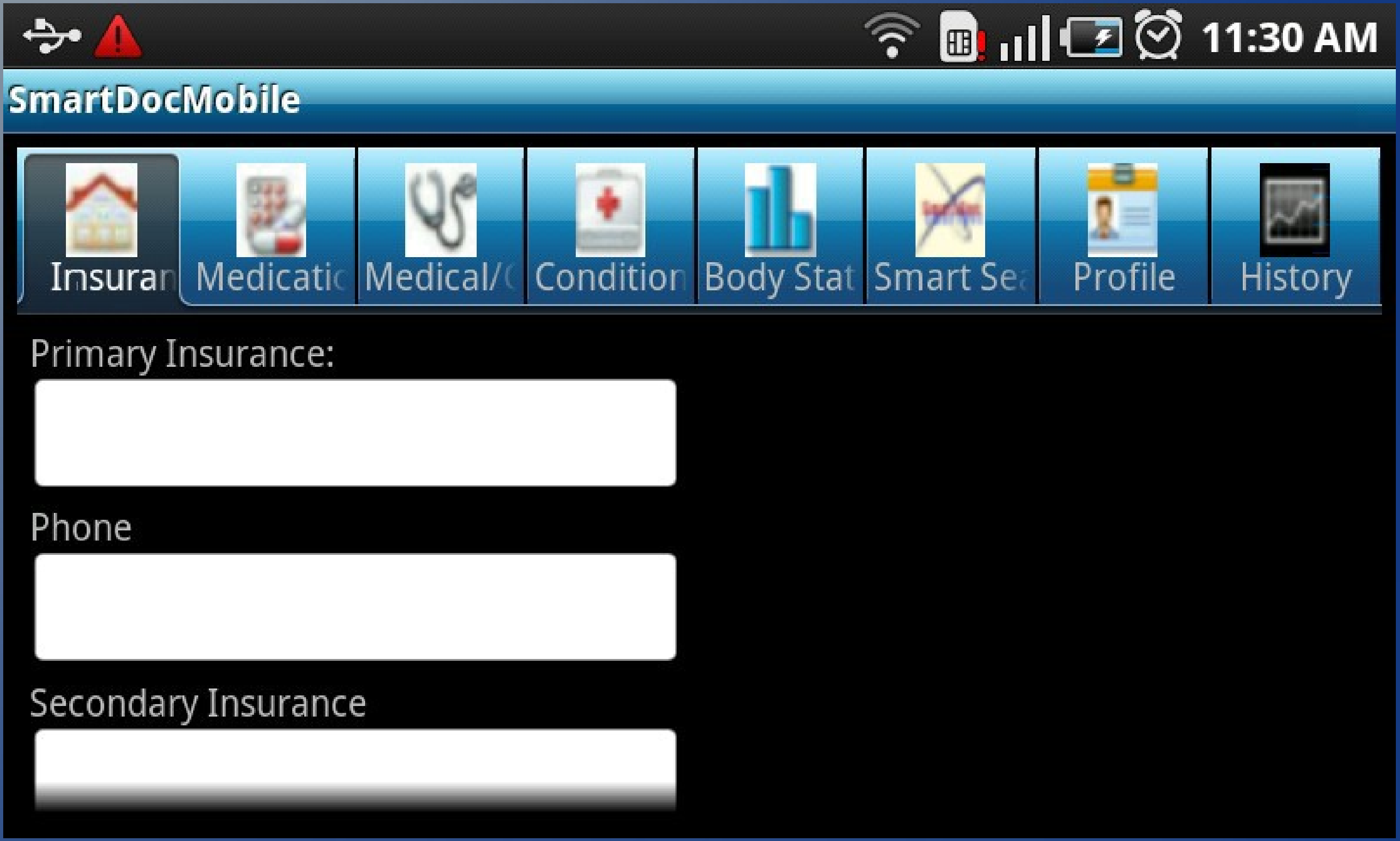


SmartDocMobile

Welcome to SmartDoc - Your mobile PHR Solution

username:

password:

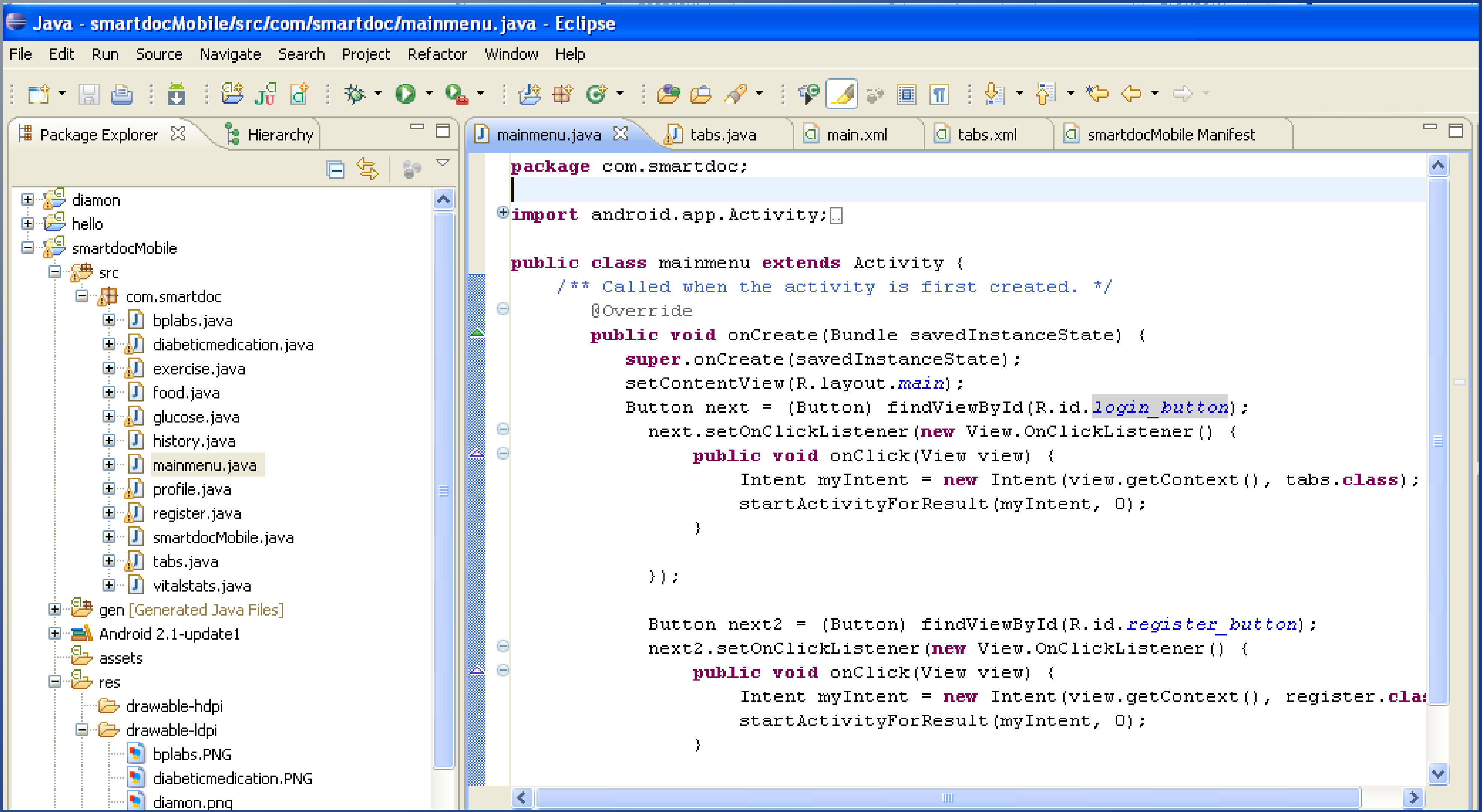


Primary Insurance:

Phone

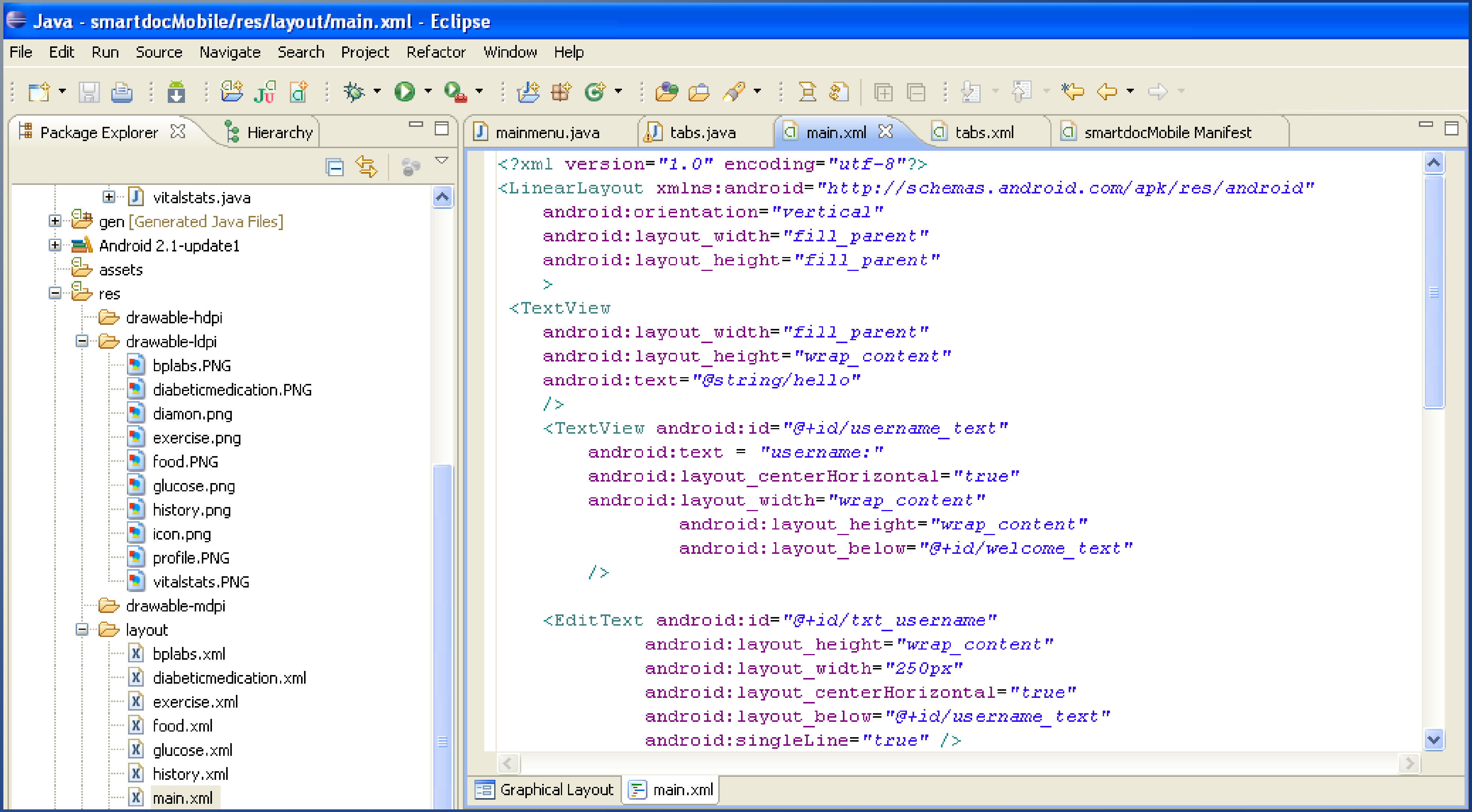
Secondary Insurance

mainmenu.java



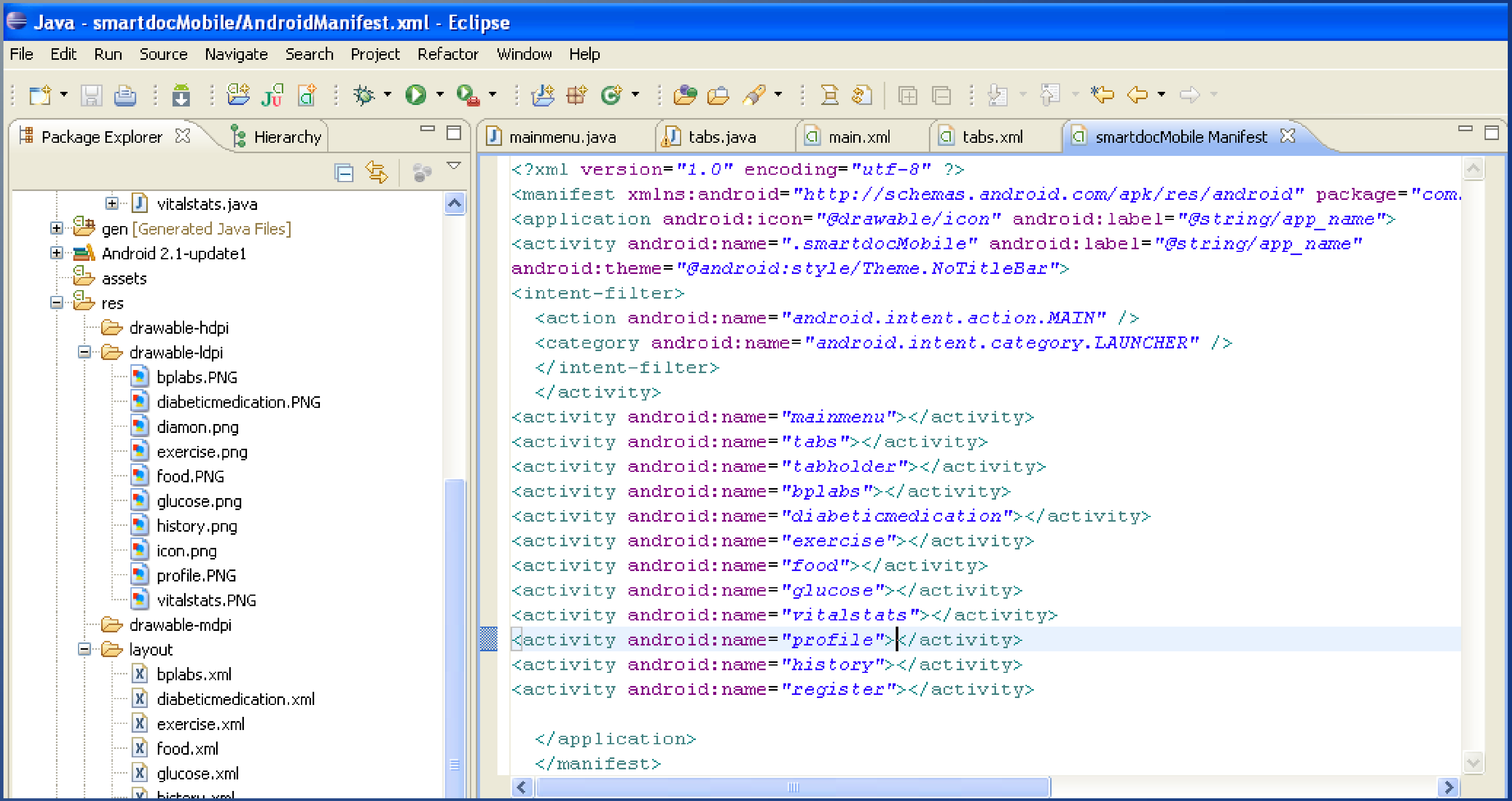
```
package com.smartdoc;

import android.app.Activity;

public class mainmenu extends Activity {
    /** Called when the activity is first created. */
       @Override
       public void onCreate(Bundle savedInstanceState) {
          super.onCreate(savedInstanceState);
          setContentView(R.layout.main);
          Button next = (Button) findViewById(R.id.login_button);
            next.setOnClickListener(new View.OnClickListener() {
                public void onClick(View view) {
                    Intent myIntent = new Intent(view.getContext(), tabs.class);
                    startActivityForResult(myIntent, 0);
                }

            });

            Button next2 = (Button) findViewById(R.id.register_button);
            next2.setOnClickListener(new View.OnClickListener() {
                public void onClick(View view) {
                    Intent myIntent = new Intent(view.getContext(), register.cla
                    startActivityForResult(myIntent, 0);
                }
```

main.xml



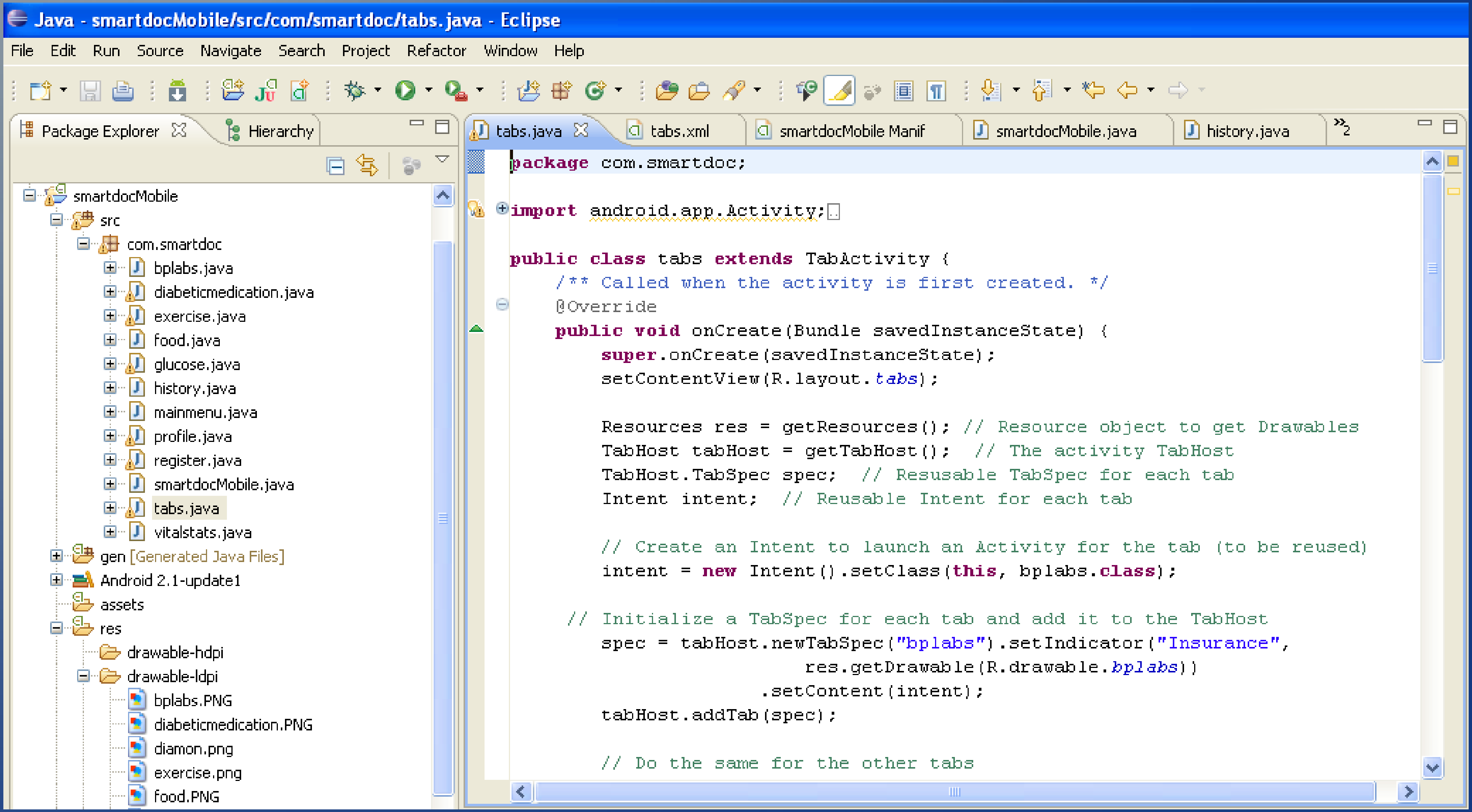
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
 <TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    />
    <TextView android:id="@+id/username_text"
        android:text = "username:"
        android:layout_centerHorizontal="true"
        android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:layout_below="@+id/welcome_text"
        />

    <EditText android:id="@+id/txt_username"
             android:layout_height="wrap_content"
             android:layout_width="250px"
             android:layout_centerHorizontal="true"
             android:layout_below="@+id/username_text"
             android:singleLine="true" />
```

Android Manifest.xml



```xml
<?xml version="1.0" encoding="utf-8" ?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.
<application android:icon="@drawable/icon" android:label="@string/app_name">
<activity android:name=".smartdocMobile" android:label="@string/app_name"
android:theme="@android:style/Theme.NoTitleBar">
<intent-filter>
  <action android:name="android.intent.action.MAIN" />
  <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
  </activity>
<activity android:name="mainmenu"></activity>
<activity android:name="tabs"></activity>
<activity android:name="tabholder"></activity>
<activity android:name="bplabs"></activity>
<activity android:name="diabeticmedication"></activity>
<activity android:name="exercise"></activity>
<activity android:name="food"></activity>
<activity android:name="glucose"></activity>
<activity android:name="vitalstats"></activity>
<activity android:name="profile"></activity>
<activity android:name="history"></activity>
<activity android:name="register"></activity>

  </application>
  </manifest>
```

tabs.java



```java
package com.smartdoc;

import android.app.Activity;

public class tabs extends TabActivity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.tabs);

        Resources res = getResources(); // Resource object to get Drawables
        TabHost tabHost = getTabHost();  // The activity TabHost
        TabHost.TabSpec spec;  // Resusable TabSpec for each tab
        Intent intent;  // Reusable Intent for each tab

        // Create an Intent to launch an Activity for the tab (to be reused)
        intent = new Intent().setClass(this, bplabs.class);

      // Initialize a TabSpec for each tab and add it to the TabHost
        spec = tabHost.newTabSpec("bplabs").setIndicator("Insurance",
                          res.getDrawable(R.drawable.bplabs))
                      .setContent(intent);
        tabHost.addTab(spec);

        // Do the same for the other tabs
```

Accessing Sensors

There are various sensors on the Android phones, e.g. GPS, camera, accelerometer

- Cannot be tested on Simulator
- Tutorial and sample source code to use GPS and accelerometer sensors available soon
- Using the Wi-Fi and Bluetooth Interfaces instructions on:

http://www.tutorialforandroid.com/2009/10/turn-off-turn-on-wifi-in-android-using.html
http://developer.android.com/guide/topics/wireless/bluetooth.html

GPS Sensor Access

- To use GPS functionality in your Android application, you'll need to add the ACCESS_FINE_LOCATION permission to the AndroidManifest.xml file.

```
 <uses-permission
android:name="android.permission.ACCESS_FINE_LOCATION" />
```

- In Android, location-based services are provided by the LocationManager class located in the android.location package.

- Using the LocationManager class, your application can obtain periodic updates of the device's geographical locations as well as fire an intent when it enters the proximity of a certain location.

Accelerometer and Access to other Sensors

- An instance of the SensorManager is required in order to retrieve informations about the supported sensors.

- No permission is required to access the sensor service. It is then possible to retrieve the list of available sensors of a certain type.

- For an accelerometer sensor, the type to use is given by the Sensor.TYPE_ACCELEROMETER constant.

- If at least one Sensor exists, it is possible to register a SensorEventListener for a Sensor of the list.

- It is possible to specify the delivering rate for sensor events. Specified rate must be one of :

SensorManager.SENSOR_DELAY_FASTEST : as fast as possible

SensorManager.SENSOR_DELAY_GAME : rate suitable for game

SensorManager.SENSOR_DELAY_NORMAL : normal rate

SensorManager.SENSOR_DELAY_UI : rate suitable for UI Thread

Android and Databases

- Access to remote database e.g. MySQL can occur using a PHP script hosted on the server performing the query and JSON formatting for data exchange for reference check out the example at:

  http://www.helloandroid.com/tutorials/connecting-mysql-database

- Local light weight database system.

  http://developer.android.com/reference/android/database/sqlite/SQLiteDatabase.html

Questions?

http://developer.android.com/index.html