Introduction to Android Programming

Rajiv Ratn Shah
(rajiv@comp.nus.edu.sg)

August, 27

CS5248 Fall 2014

*Based on slides from Dr. Beomjoo Seo

Contents

- Introduction
- Installation
- Example by a sample app, “Hello, CS5248”
  - Create an Android Project
  - Building and Running
  - Debugging
- General Topics
  - Fundamental Components
  - Activity, Intent
  - UI : Layout, Menu
  - Thread
  - Misc: Media Recorder, HTTP Post, MP4Parser
- Sample Application

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

Application Fundamental

- Android apps are written in the Java
- The Android OS is a multi-user Linux system
- Android app lives in its own security sandbox
  - System assigns each app a unique Linux user ID
  - Each process has its own virtual machine
  - Every application runs in its own Linux process.
- The Android system implements the ***principle of least privilege***
- How can an app share data with other apps?

http://developer.android.com/guide/components/fundamentals.html

Software Downloads

- Java Compiler
  - Latest JDK (JRE alone will not work)
    - Java SE 8u20 or Java SE 7u67
- Android Developer Tools (ADT)
  - Download ADT Bundle form http://developer.android.com/sdk/index.html
  - SDK for existing IDE

Environment Setup

- Three basic steps to get an application running on an Android emulator ADT (or on device)
  - Step 1: Configure the SDK
  - Step 2: Define emulator type and device
  - Step 3: Build and run your first Android App

Step-1: Configure the SDK

- Unzip ADT Bundle to a known location and give some meaningful name (say ADT)
  - **Install the Eclipse ADT Bundle**
    - Unpack the ZIP file (named adt-bundle-<os_platform>.zip) to ADT folder
    - Open the ADT/eclipse/ directory and launch **eclipse**.
  - That's it!

Caution: Do not move any of the files (*i.e.* eclipse/ or sdk/) or directories from the ADT directory. If you do so, you'll need to manually update the ADT preferences.

http://developer.android.com/sdk/installing/adding-packages.html

Configure the SDK for Existing IDE

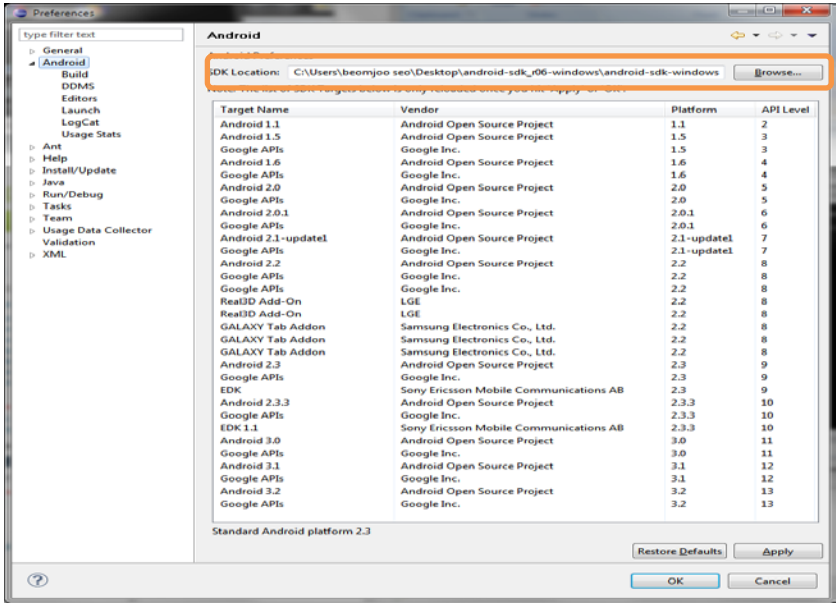
- Install Eclipse IDE
  - Eclipse 3.7.2 or above
    - Eclipse JDT plugin
  - Run Android SDK starter package
  - Install ADT plugin
    - Eclipse > Help > Install New Software … -> Add
      - https://dl-ssl.google.com/android/eclipse/
    - Configure ADT plugin

http://developer.android.com/sdk/installing/installing-adt.html

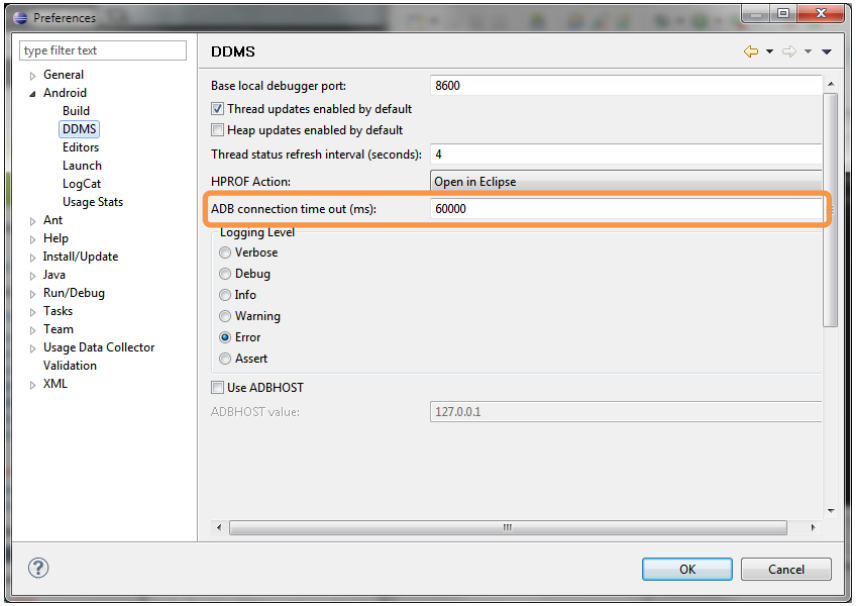
Eclipse ADT Plugin Configuration

At **Windows > Preferences > Android**

Specify Android SDK location

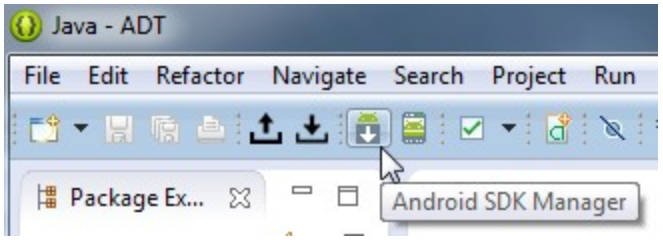


Increase ADT connection time out

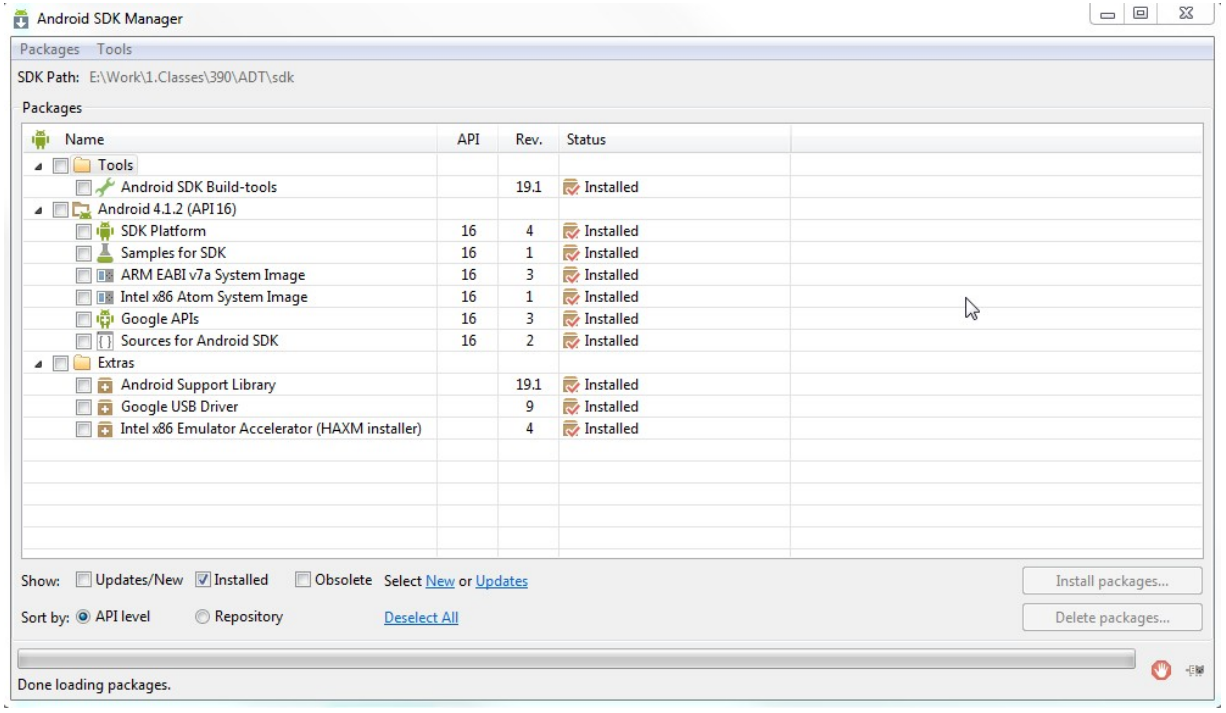


Install SDK Packages

- Start your Android SDK manager
  - ADT/SDK Manager.exe or



- Installed packages



Install SDK Packages

- You should install the packages in two steps:
  - Select packages in "**Tools**" only (Android SDK Tools, Android SDK Platform-tools, Android SDK Build-tools). Download and install these.
  - After successful installation of packages in "**Tools**", quit and re-start Eclipse to continue with the rest of the installations.

Step-2: Android Virtual Device (AVD)

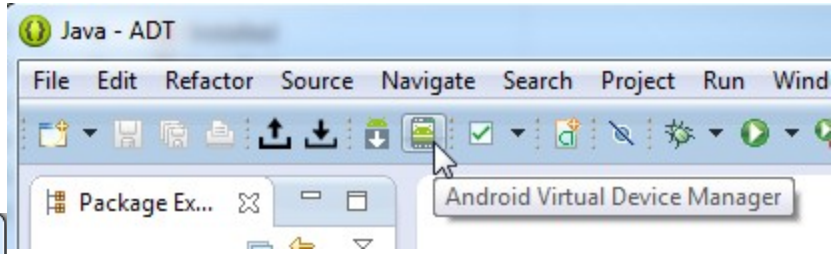
- Define and Start your Android Emulator
  - Define a *virtual device type*
  - Define an instance of *virtual device* based on the type we defined
  - Start our instance of virtual device.

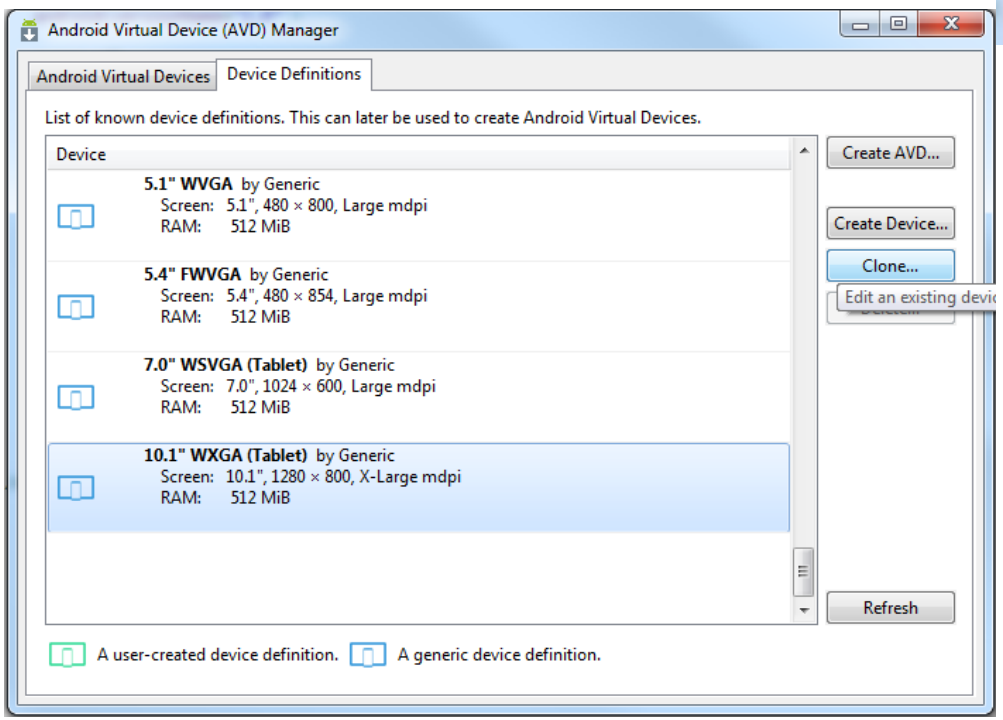
Android Virtual Device (AVD)

- Define Android Virtual Device type:
  - Start the *AVD Manager*

Android Virtual Device (AVD)

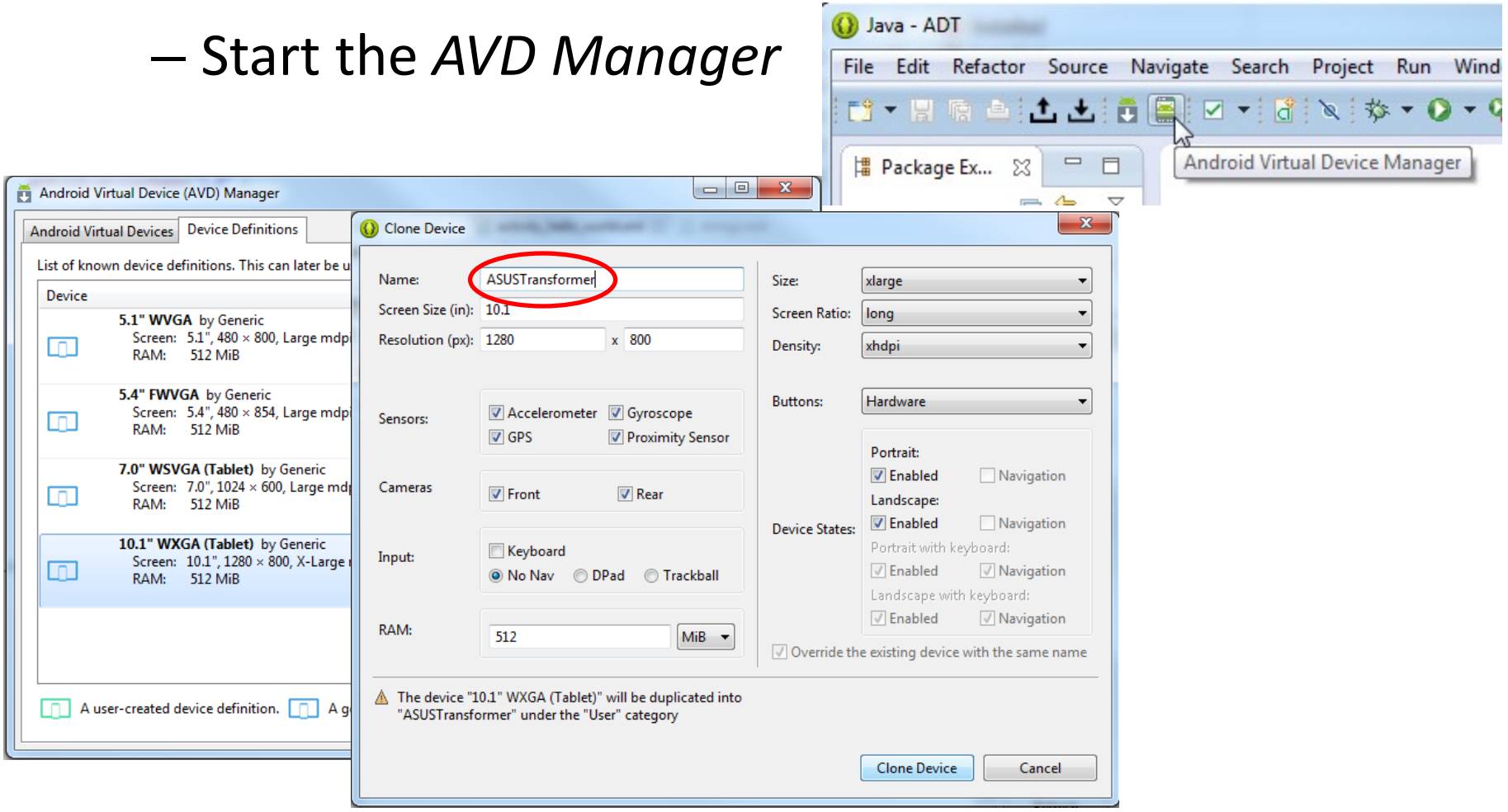
- Define Android Virtual Device type:
  - Start the *AVD Manager*





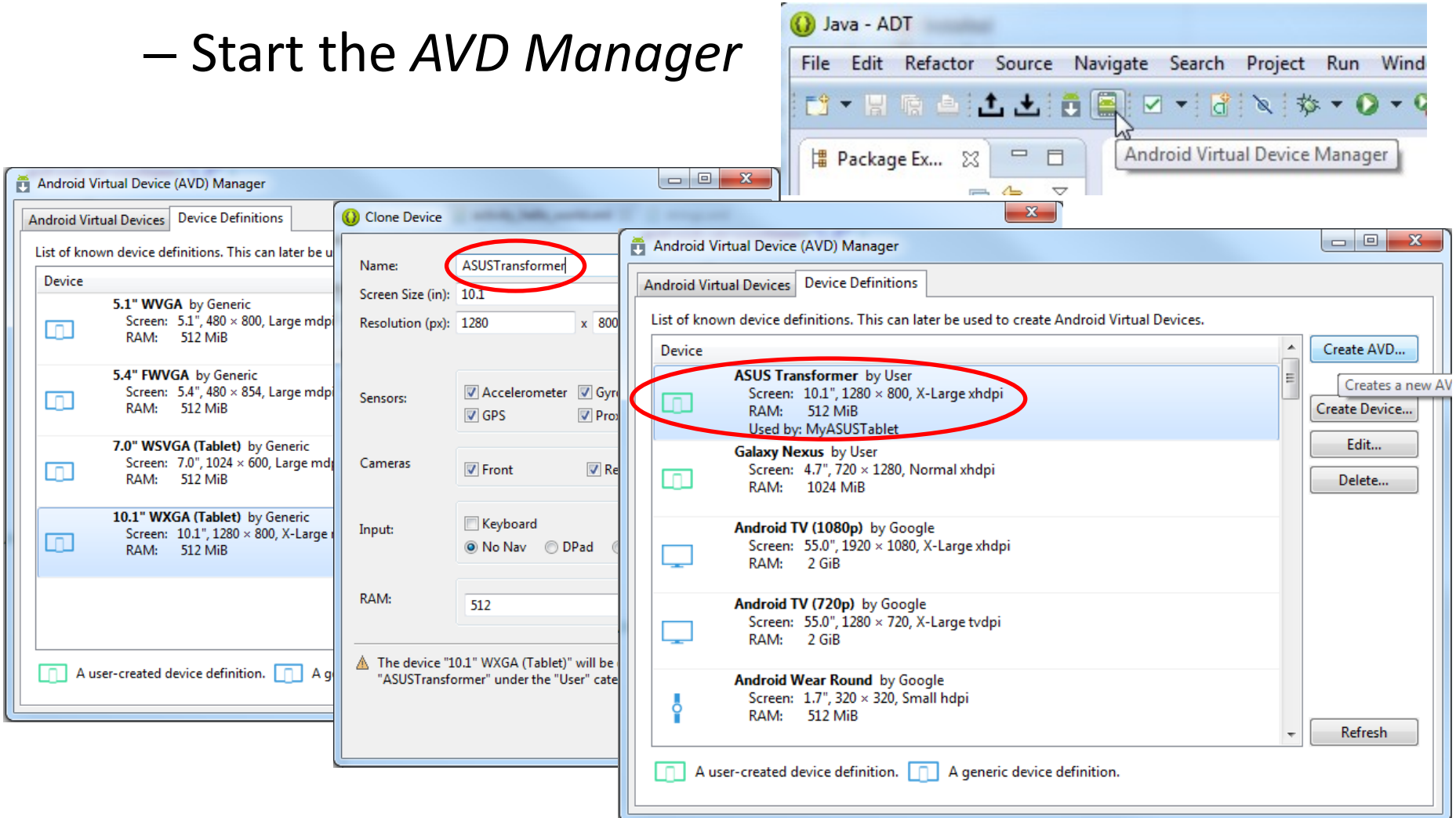
Android Virtual Device (AVD)

- Define Android Virtual Device type:
  - Start the *AVD Manager*

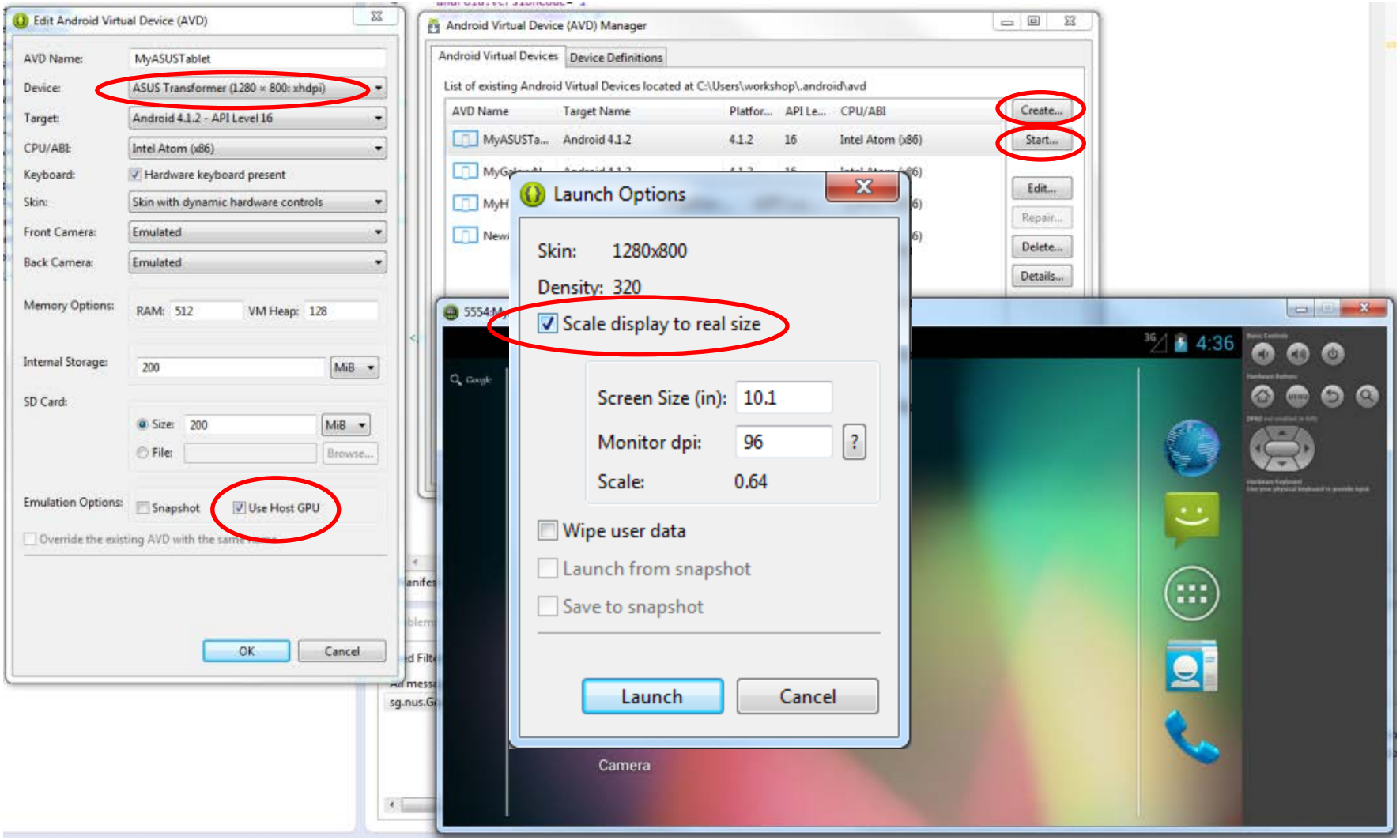


Android Virtual Device (AVD)

- Define Android Virtual Device type:
  - Start the *AVD Manager*



Android Virtual Device (AVD)

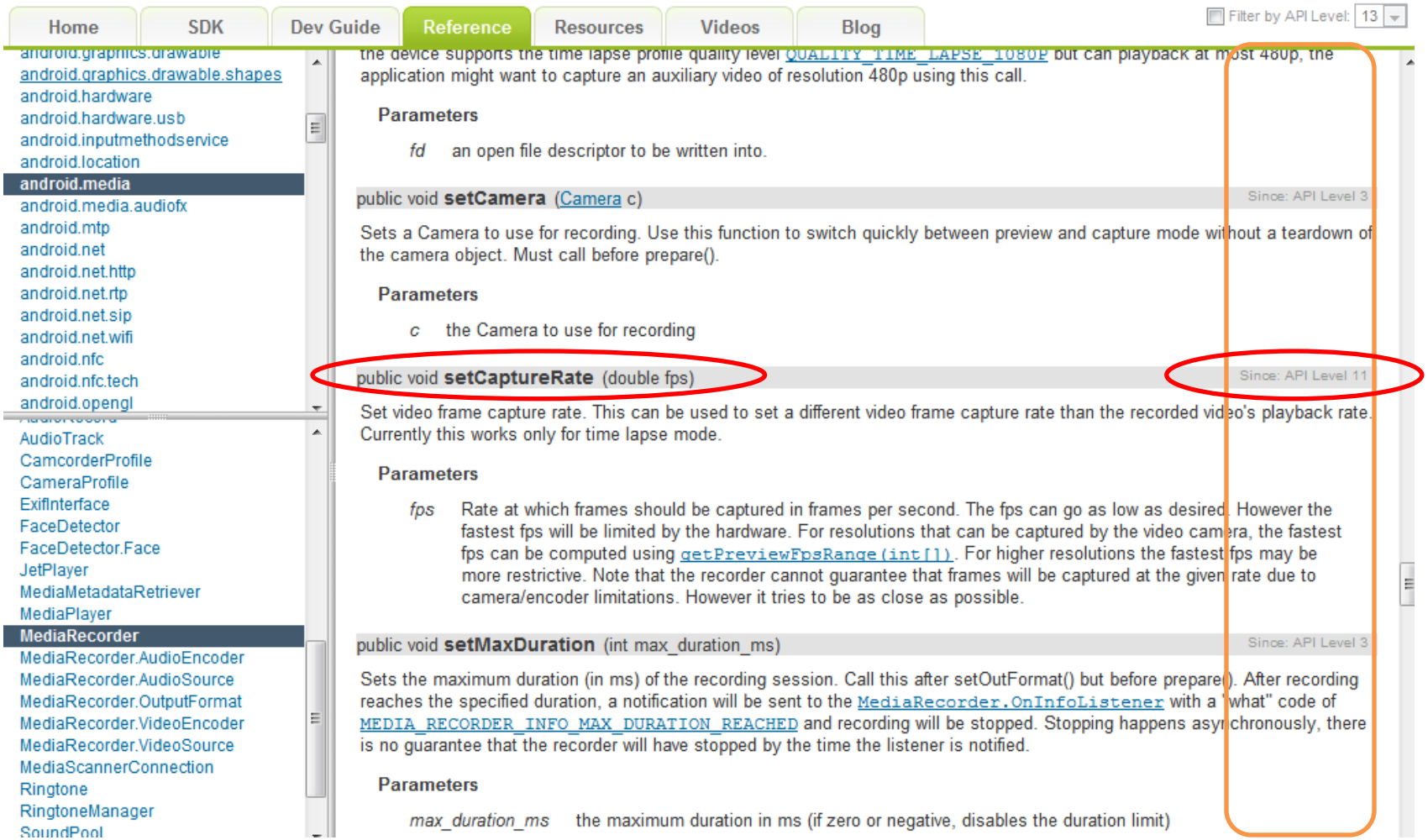


SDK Version Mapping Convention

Platform Version	API Level	VERSION_CODE	Notes
Android 4.4	19	KITKAT	Platform Highlights
Android 4.3	18	JELLY_BEAN_MR2	Platform Highlights
Android 4.2, 4.2.2	17	JELLY_BEAN_MR1	Platform Highlights
Android 4.1, 4.1.1	16	JELLY_BEAN	Platform Highlights
Android 4.0.3, 4.0.4	15	ICE_CREAM_SANDWICH_MR1	Platform Highlights
Android 4.0, 4.0.1, 4.0.2	14	ICE_CREAM_SANDWICH	
Android 3.2	13	HONEYCOMB_MR2	
Android 3.1.x	12	HONEYCOMB_MR1	Platform Highlights
Android 3.0.x	11	HONEYCOMB	Platform Highlights
Android 2.3.4 Android 2.3.3	10	GINGERBREAD_MR1	Platform Highlights
Android 2.3.2 Android 2.3.1 Android 2.3	9	GINGERBREAD	
Android 2.2.x	8	FROYO	Platform Highlights
Android 2.1.x	7	ECLAIR_MR1	Platform Highlights
Android 2.0.1	6	ECLAIR_0_1	
Android 2.0	5	ECLAIR	
Android 1.6	4	DONUT	Platform Highlights
Android 1.5	3	CUPCAKE	Platform Highlights
Android 1.1	2	BASE_1_1	
Android 1.0	1	BASE	

http://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels

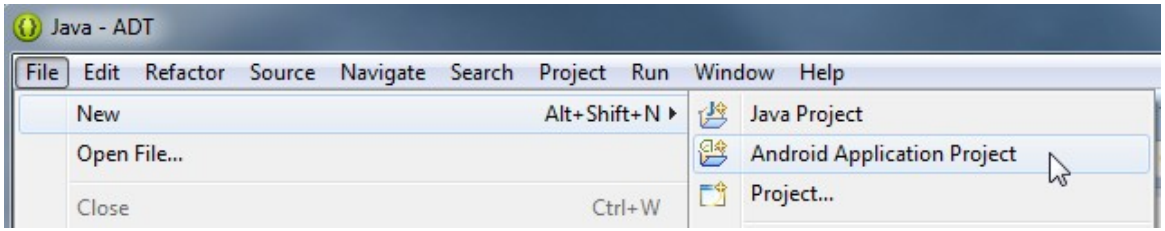
API Level

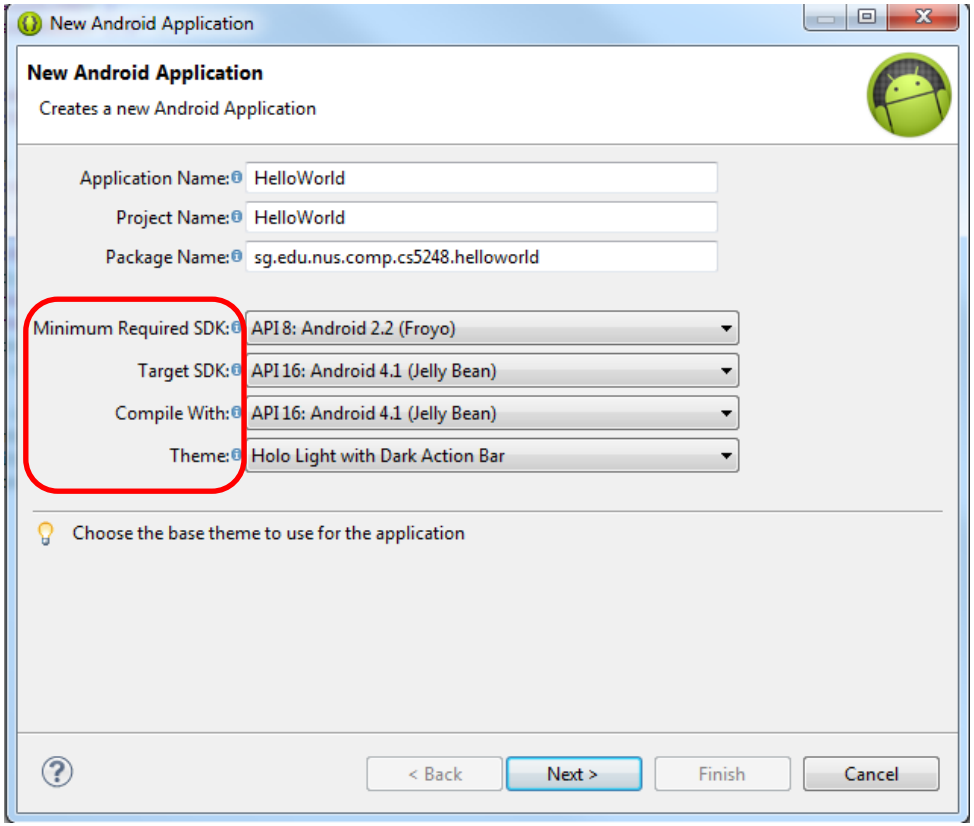


API Level of built-in methods indicates that it work only with android version with API level equal or above that value. E.g. 13 here

Step-3: Creating a Sample App

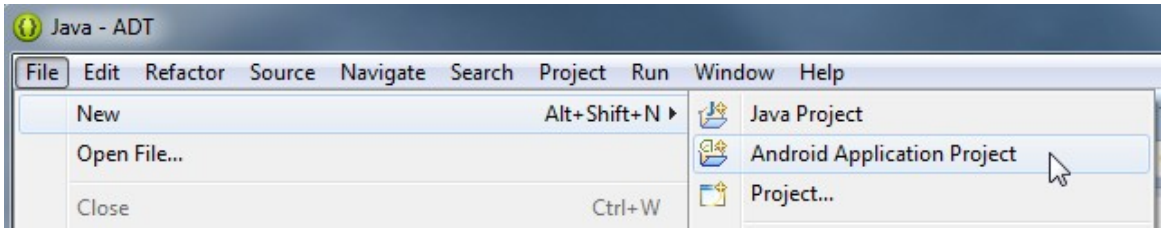
- "Hello, CS5248"

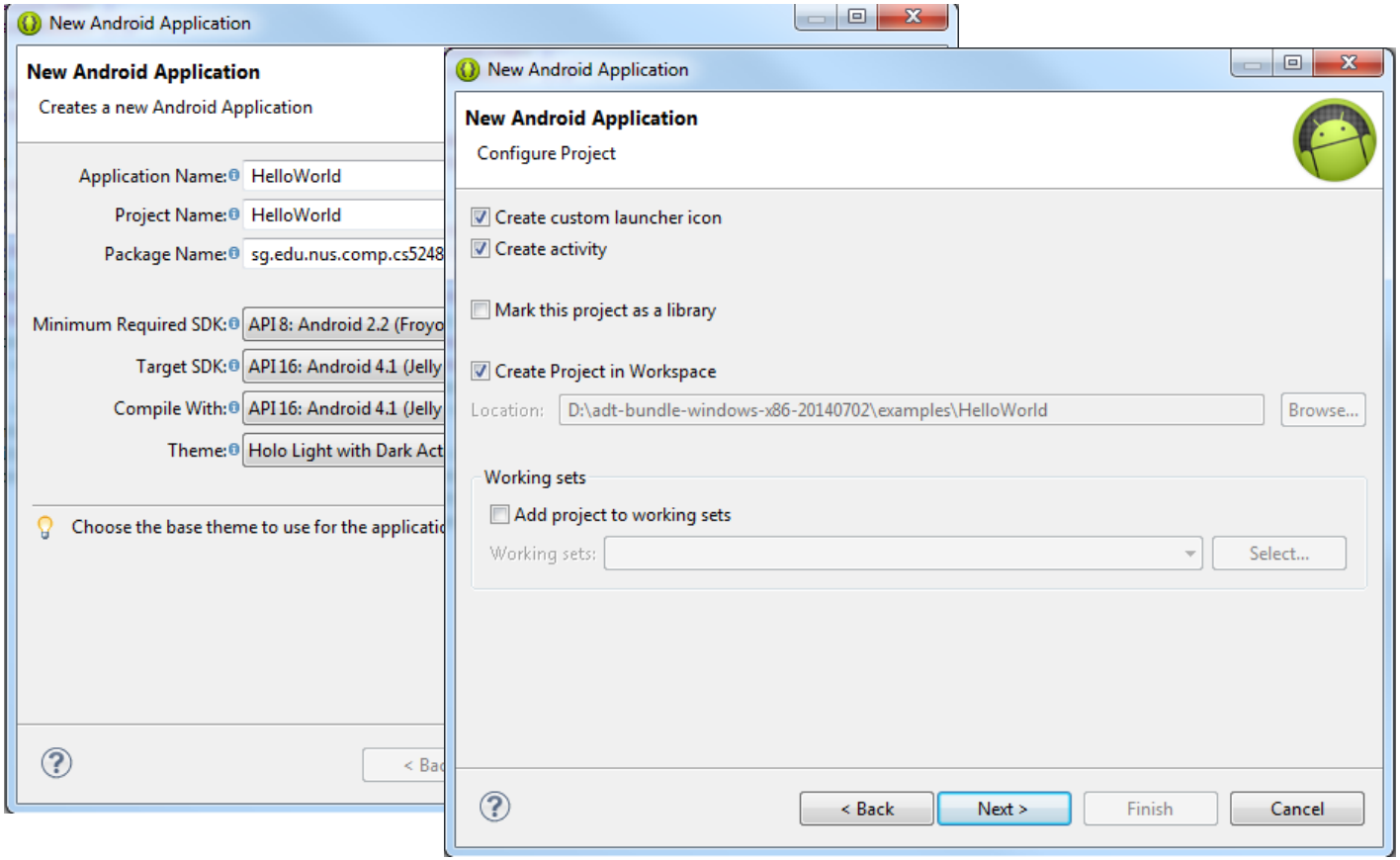




Step-3: Creating a Sample App

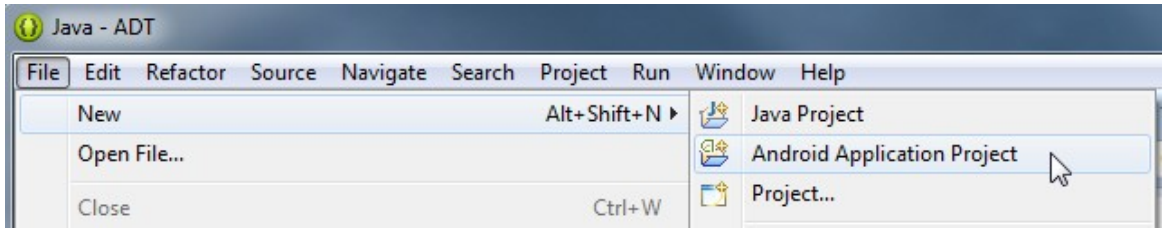
- "Hello, CS5248"

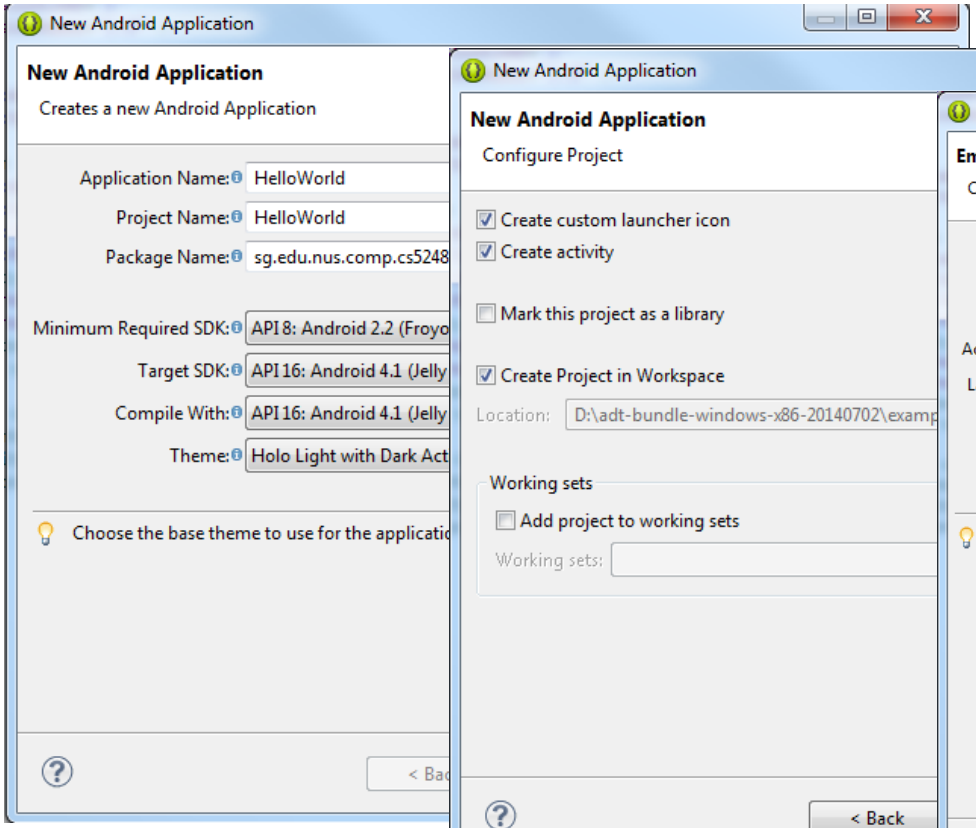




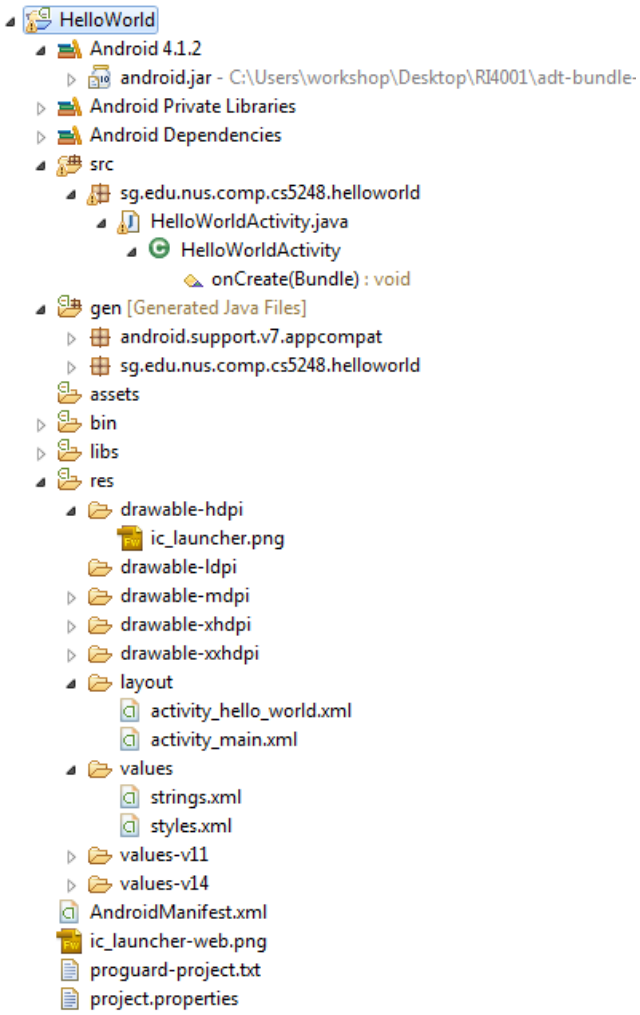
Step-3: Creating a Sample App

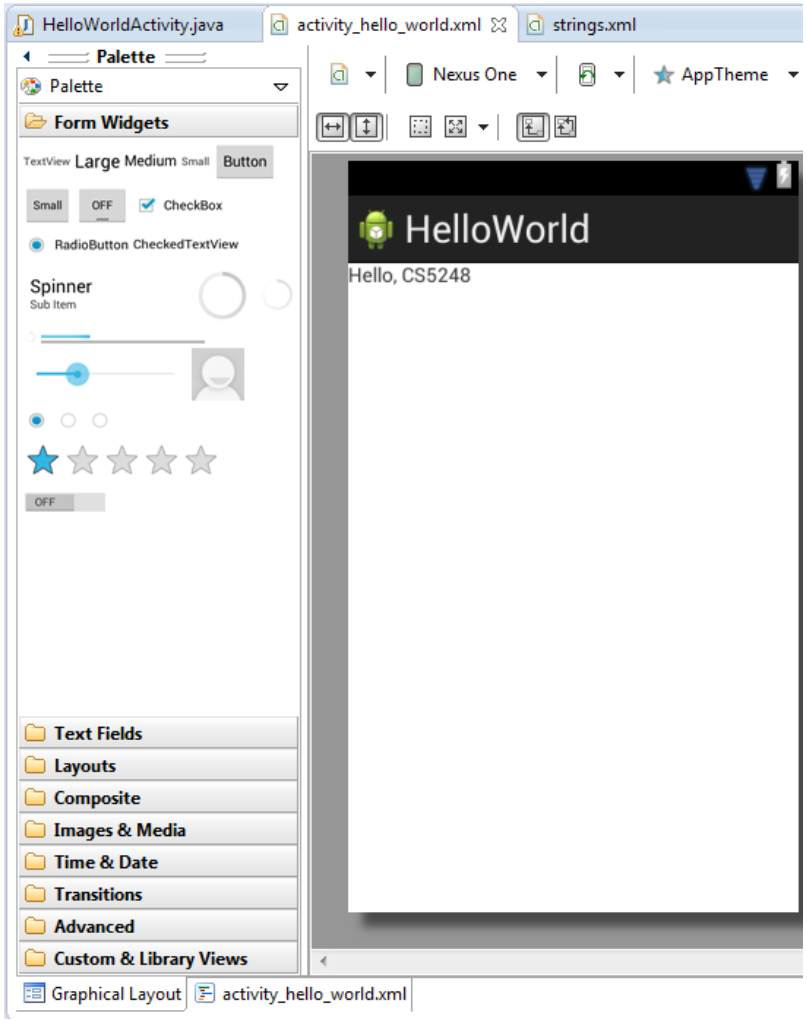
- "Hello, CS5248"





Eclipse > New > Android Project

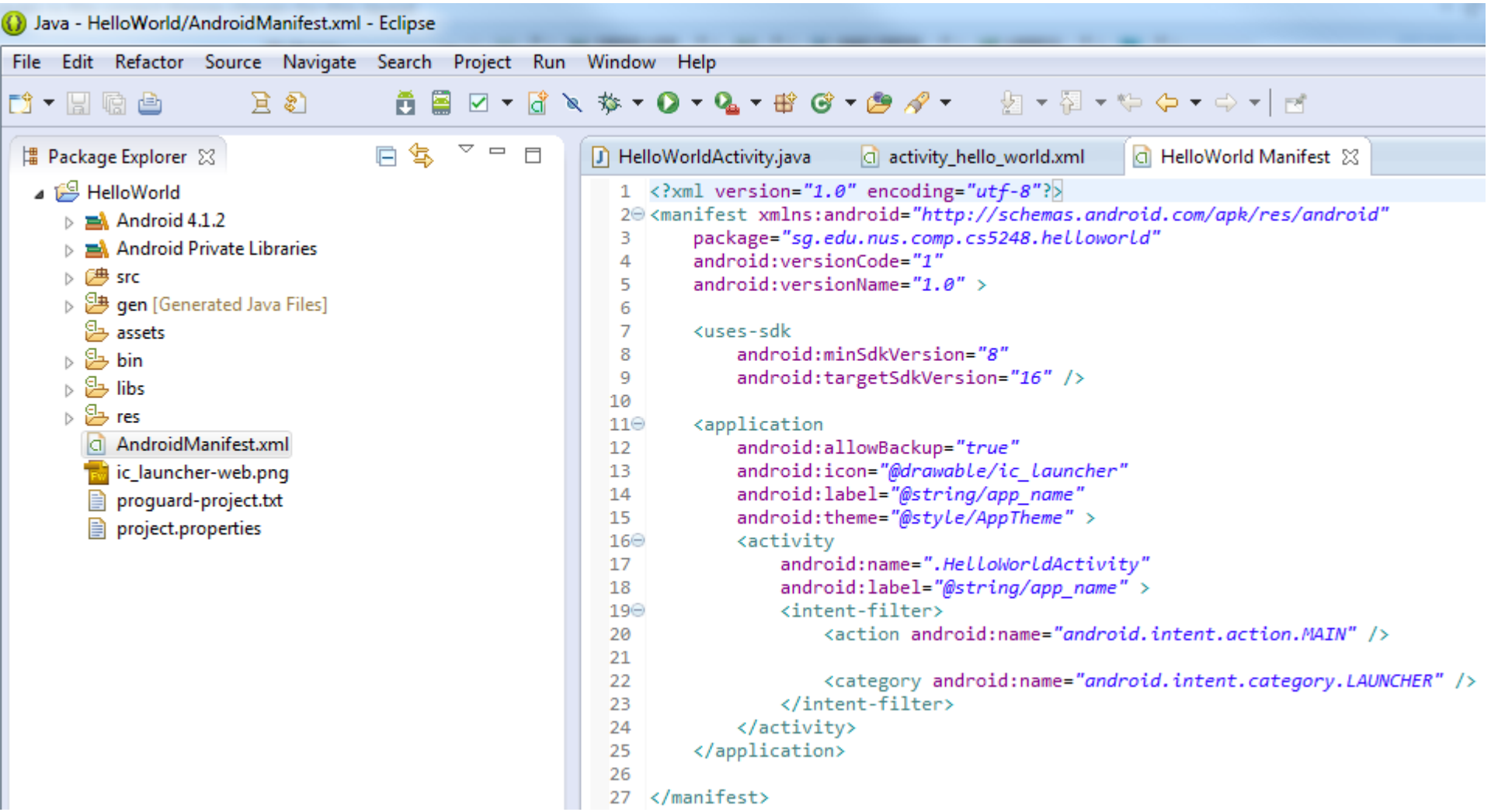




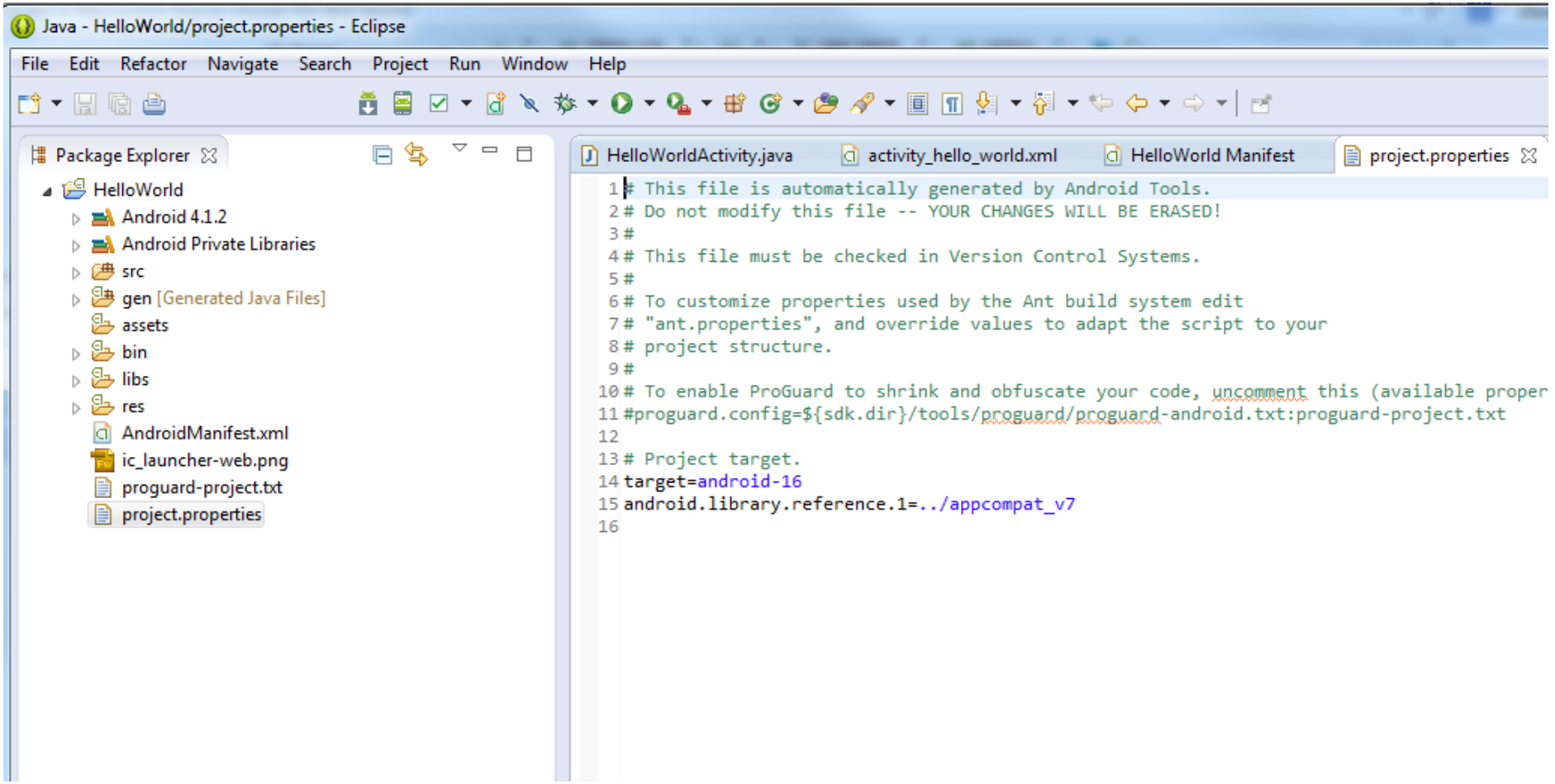
The Manifest File

- Identify any user permissions the application requires
- Declare the minimum API Level required by the application
- Declare hardware and software features used or required by the application
- API libraries the application needs to be linked

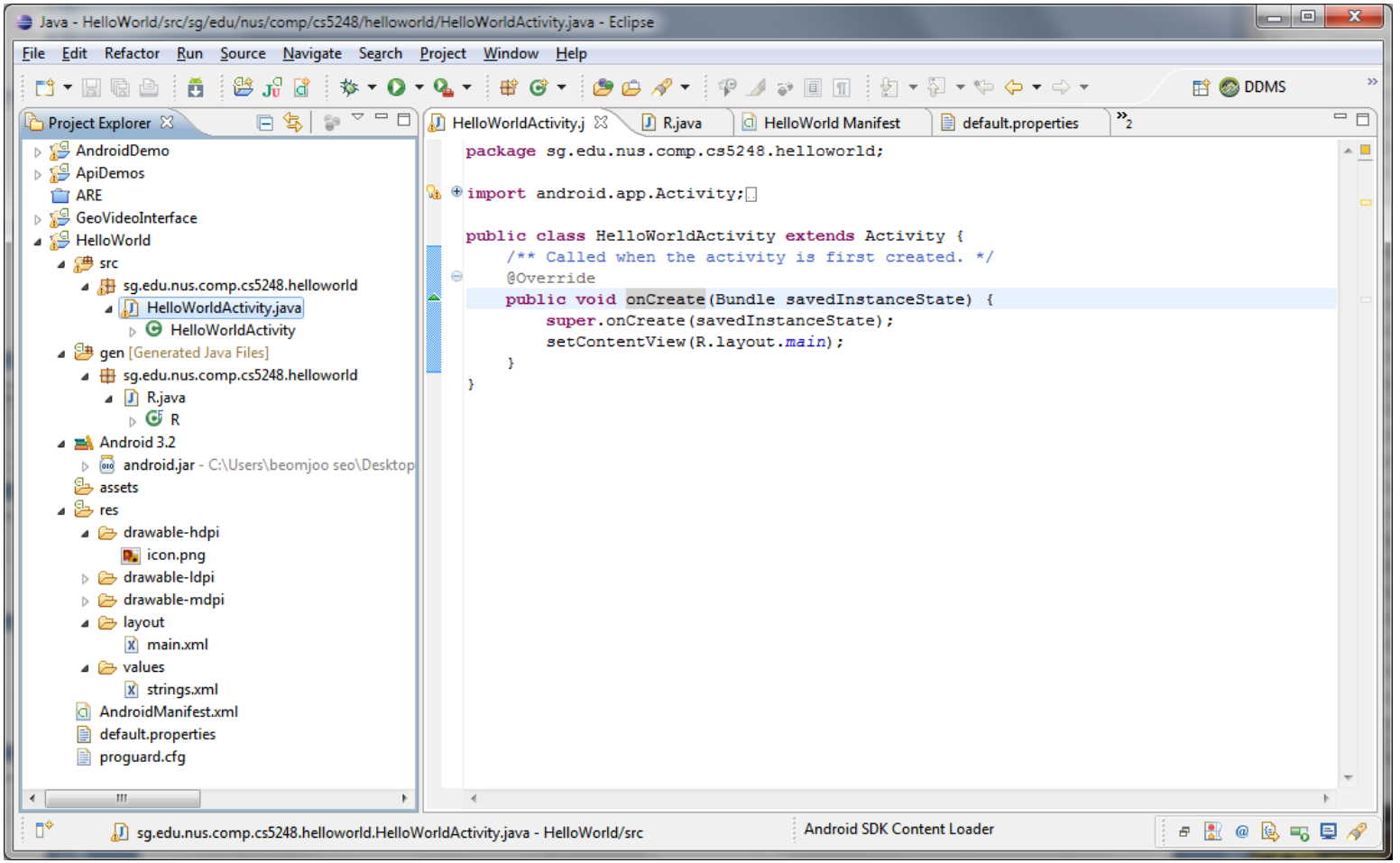
The Manifest file



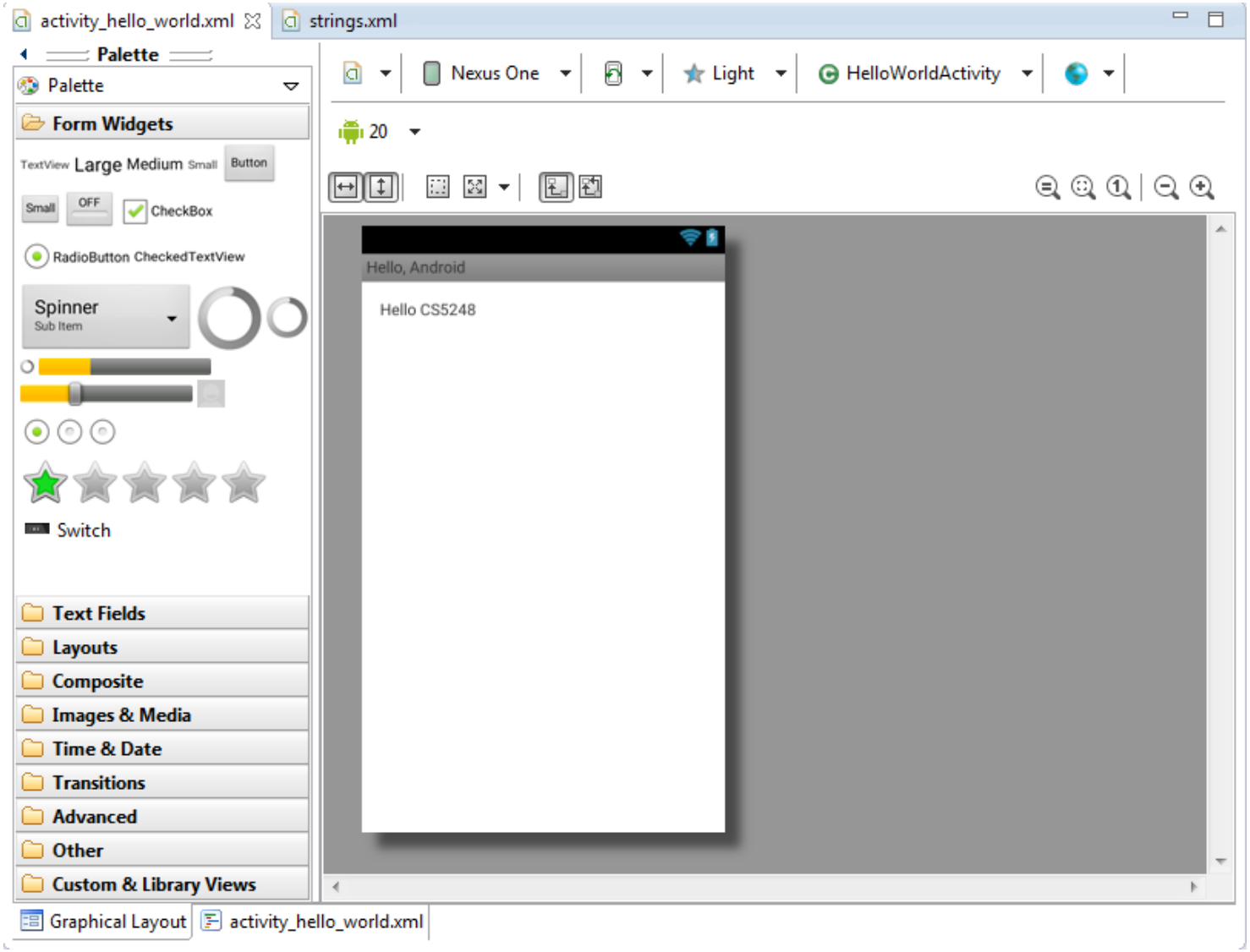
Default.properties



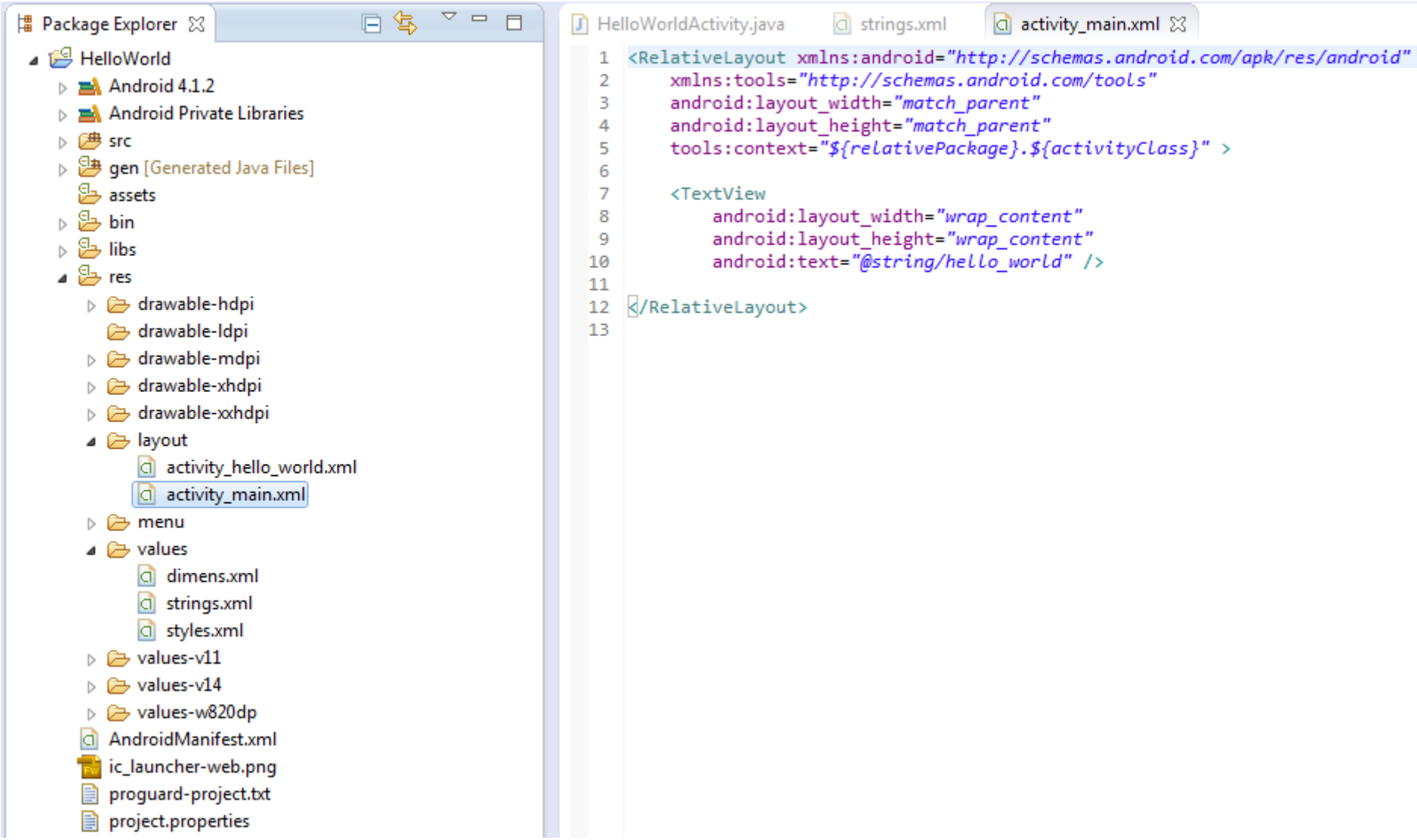
Main Activity Class



Graphical Layout Editor



XML Layout File



Value Resources

activity_main.xml | MainActivity.java | strings.xml | arrays.xml | colors.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

<color name="mango">#FFB272</color>
<color name="light">#FFE4A4</color>
<color name="dark_green_text">#CC6600</color>
<color name="chocklate">#CC6600</color>

</resources>
```

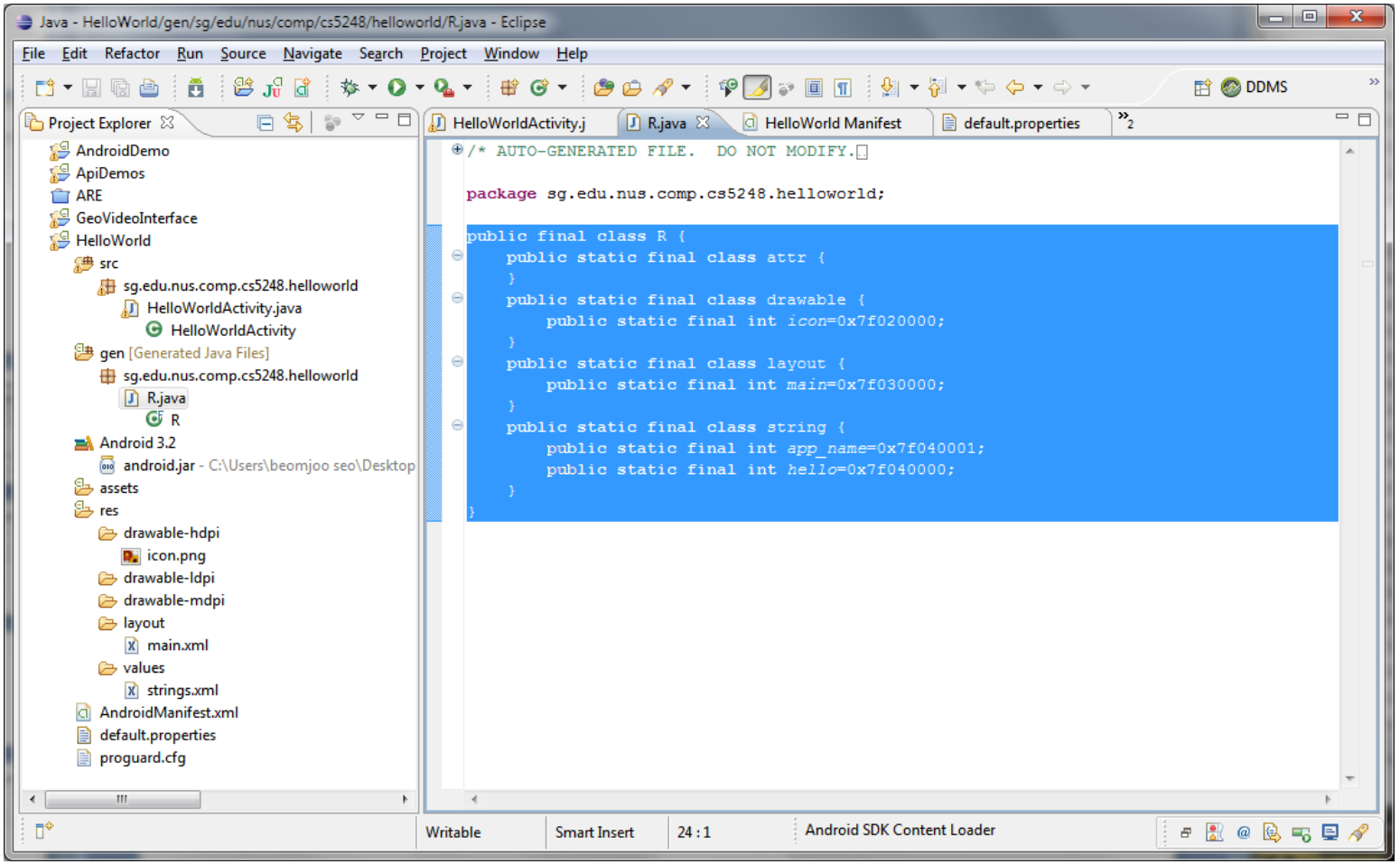
HelloWorldActivity.java | strings.xml | activity_main.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">Hello, Android</string>
    <string name="hello_world">Hello CS5248</string>

</resources>

```

activity_main.xml | MainActivity.java | strings.xml | arrays.xml

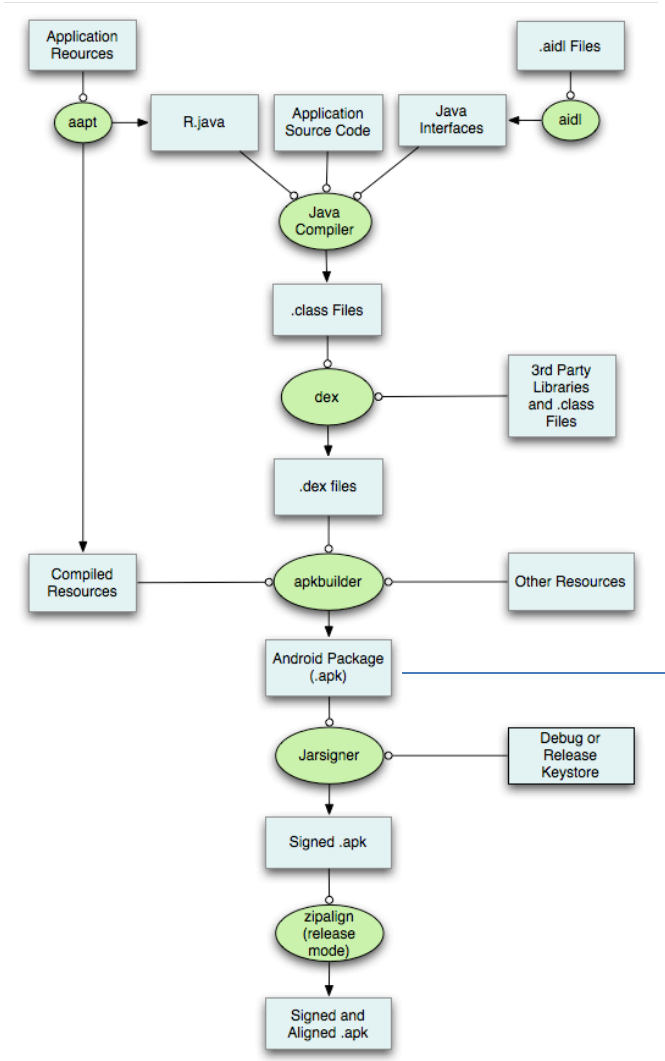
```xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string-array name="pref_sampling_rate_entries">
        <item>Fast</item>
        <item>Normal (default)</item>
        <item>Slow</item>
    </string-array>
    <string-array name="pref_sampling_rate_values">
        <item>1</item>
        <item>2</item>
        <item>3</item>
    </string-array>
    <string-array name="pref_segment_duration_entries">
        <item>3 seconds (Default)</item>
        <item>5 seconds </item>
        <item>10 seconds</item>
    </string-array>
    <string-array name="pref_segment_duration_values">
        <item>3</item>
        <item>5</item>
        <item>10</item>
    </string-array>
</resources>
```

R.java


Java - HelloWorld/gen/sg/edu/nus/comp/cs5248/helloworld/R.java - Eclipse
File Edit Refactor Run Source Navigate Search Project Window Help
DDMS
Project Explorer
AndroidDemo
ApiDemos
ARE
GeoVideoInterface
HelloWorld
src
sg.edu.nus.comp.cs5248.helloworld
HelloWorldActivity.java
HelloWorldActivity
gen [Generated Java Files]
sg.edu.nus.comp.cs5248.helloworld
R.java
R
Android 3.2
android.jar - C:\Users\beomjoo seo\Desktop
assets
res
drawable-hdpi
icon.png
drawable-ldpi
drawable-mdpi
layout
main.xml
values
strings.xml
AndroidManifest.xml
default.properties
proguard.cfg
HelloWorldActivity.j
R.java
HelloWorld Manifest
default.properties
/* AUTO-GENERATED FILE. DO NOT MODIFY.
package sg.edu.nus.comp.cs5248.helloworld;
public final class R {
public static final class attr {
}
public static final class drawable {
public static final int icon=0x7f020000;
}
public static final class layout {
public static final int main=0x7f030000;
}
public static final class string {
public static final int app_name=0x7f040001;
public static final int hello=0x7f040000;
}
}
Writable
Smart Insert
24 : 1
Android SDK Content Loader

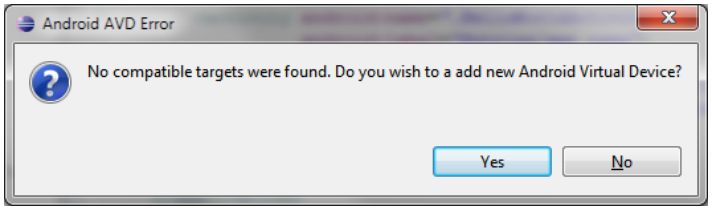
Building App

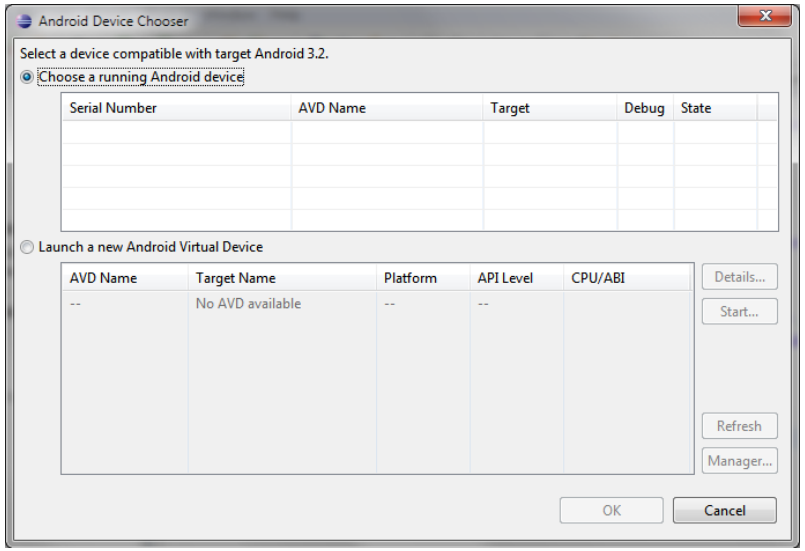
Building Process



Running

No Android Device ?

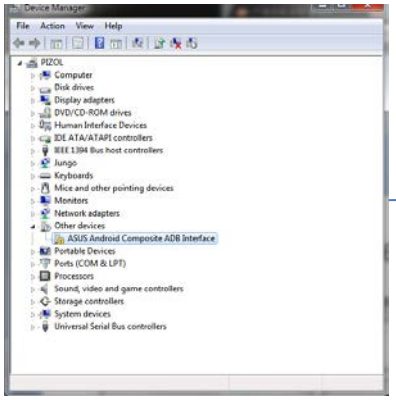




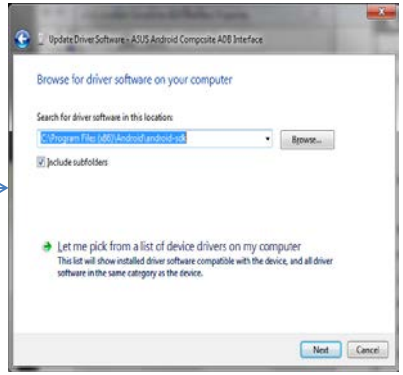
Install USB Driver !!!

USB Driver Installation

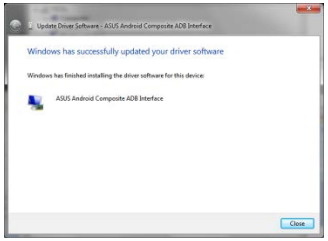
1. Install OEM USB Driver from ASUS website.
   http://support.asus.com/download/ and search by Transformer
2. Update Driver Software from Device Manager



Update Driver software

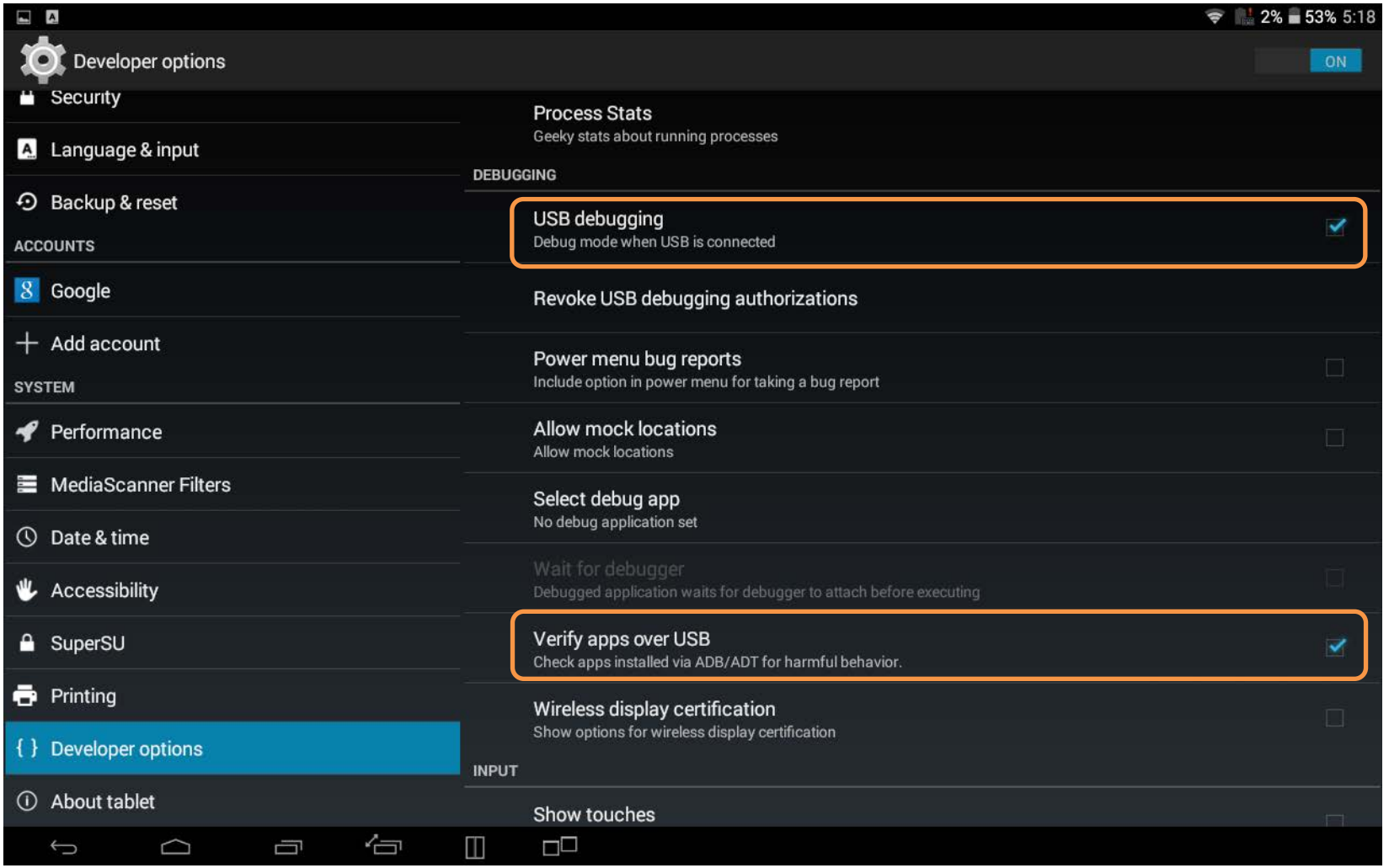


3. Locate USB Driver folder



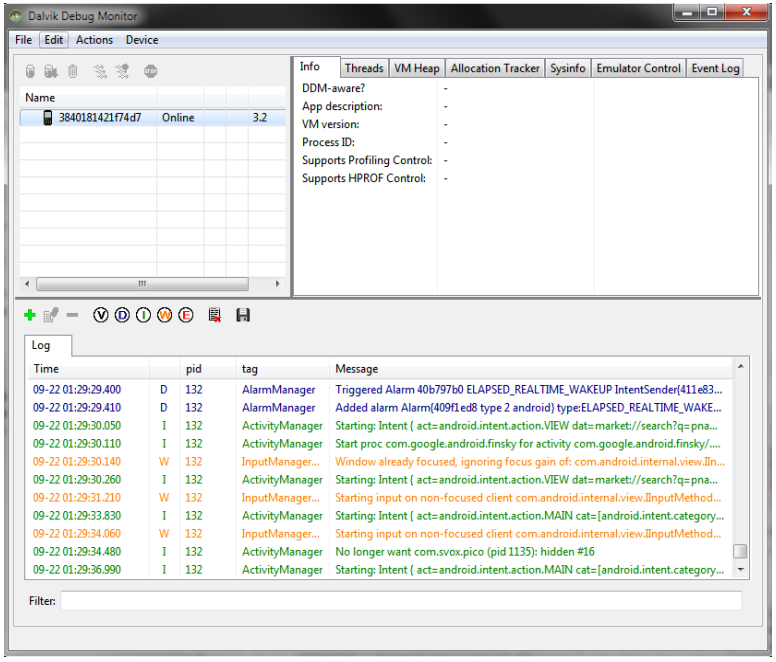
4. Enable USB Debugging at Transformer

Enabling USB debugging

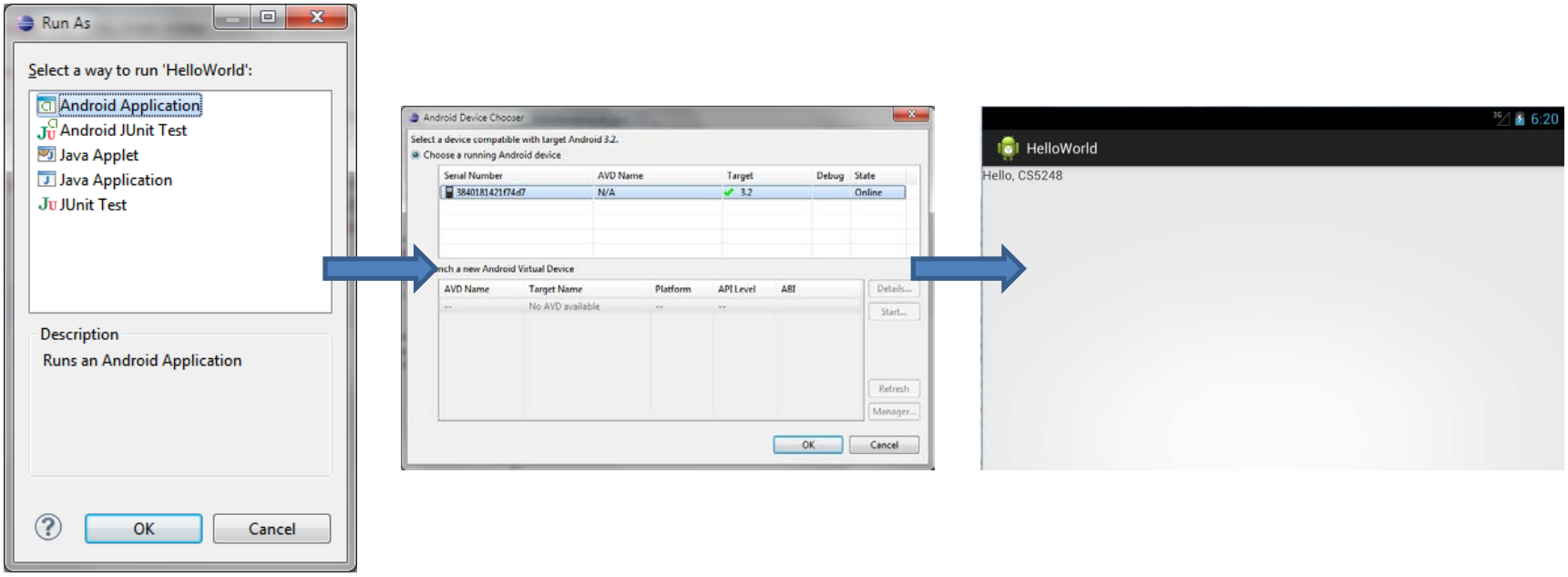


After Successful USB Driver Installation

<sdk>/tools/ddms.bat



Running Sample App.



Sample Hello World Code

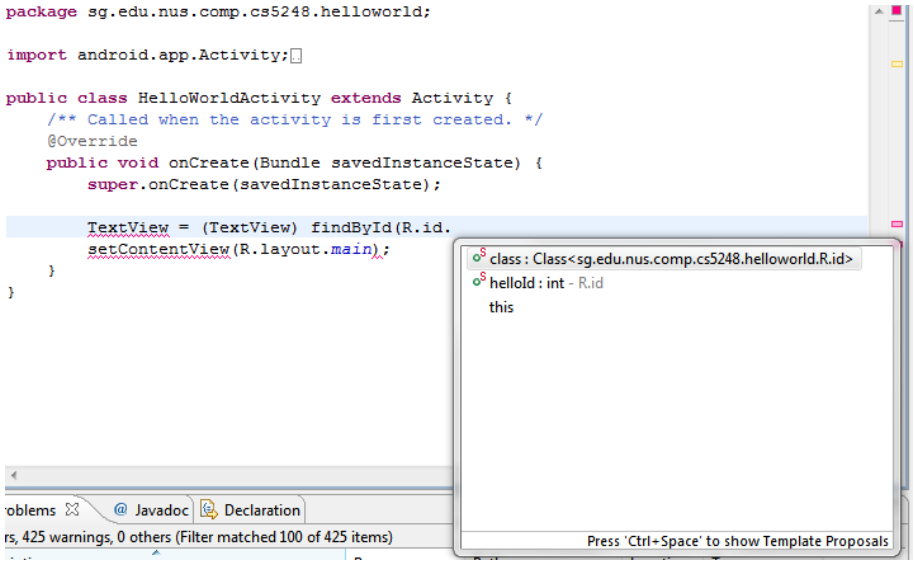
```
package sg.edu.nus.comp.cs5248.helloworld;

import android.app.Activity;

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

Editing Sample Code

```
<?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"
    android:weightSum="1">
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="@string/hello"
    android:id="@+id/helloId"/>
</LinearLayout>
```

Add id to main.xml

"@+id/helloId"



Edit HelloWorldActivity.java

Important points

- UI Element can have an Id
- Variables in our code *link* to UI elements
- update UI element content from our program code whenever we like

BE PATIENT Since Eclipse may not be responsive for a long time !!!

A Sample Code

```
package sg.edu.nus.comp.cs5248.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;


public class HelloWorldActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        TextView tv = (TextView) findViewById(R.id.helloId);
        tv.setText("hello, CS5248");
        setContentView(R.layout.main);
    }
}
```

2 ← TextView tv = (TextView) findViewById(R.id.helloId);

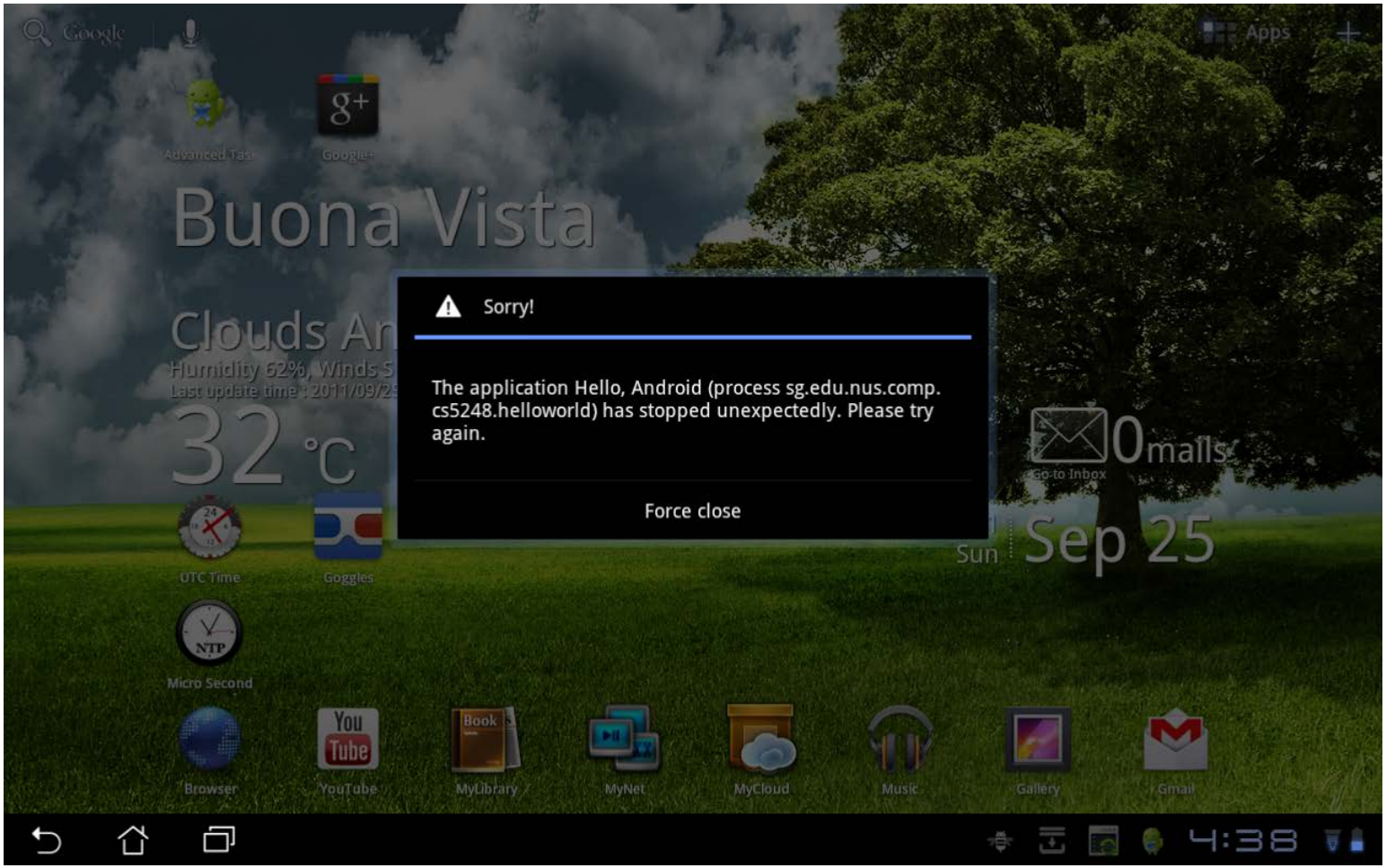
1 → R.id.helloId

3 ← tv.setText("hello, CS5248");

Important points

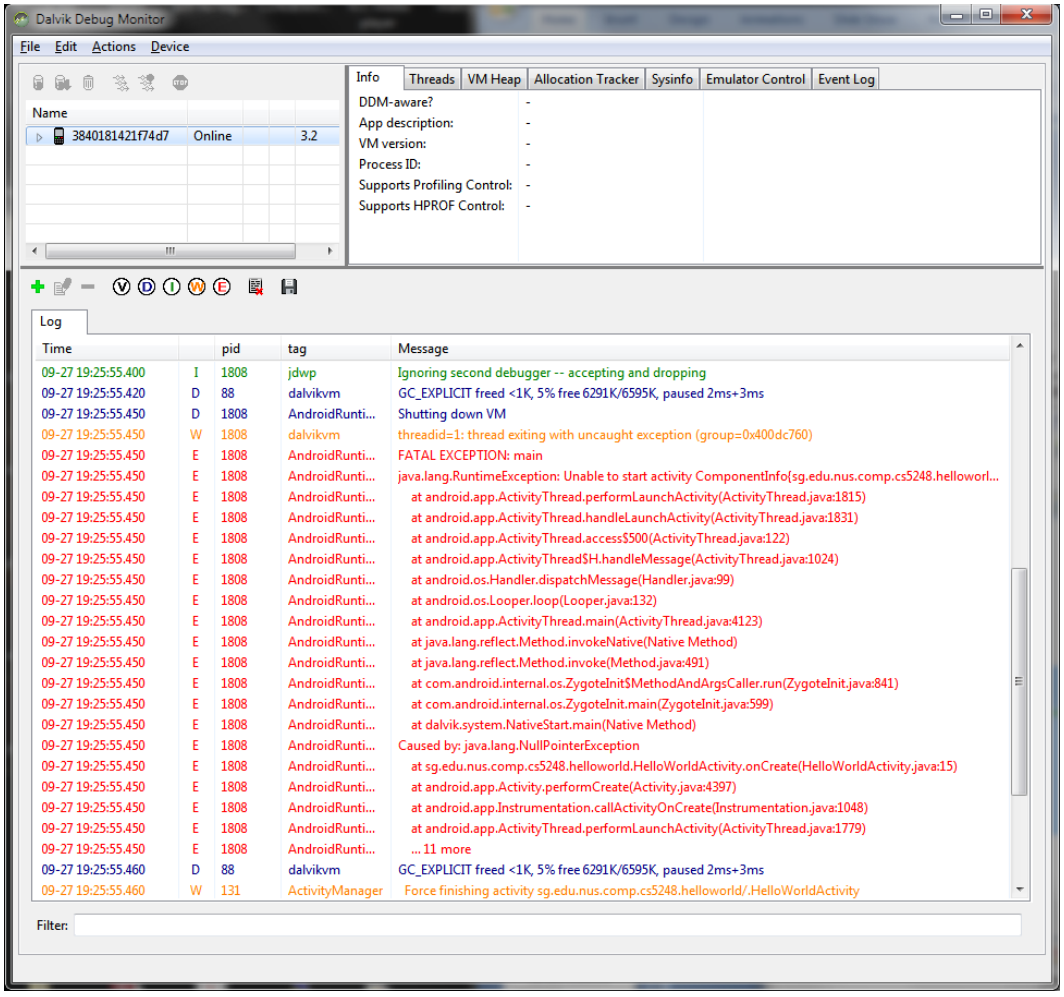
1. UI Element can have an Id
2. Variables in our code *link* to UI elements
3. Update UI element content from our program code whenever we like

App Failure


Google
Apps
Buona Vista
Sorry!
The application Hello, Android (process sg.edu.nus.comp.cs5248.helloworld) has stopped unexpectedly. Please try again.
Force close
32 °C
0 mails
Go to Inbox
Sun
Sep 25
UTC Time
Goggles
Micro Second
Browser
YouTube
MyLibrary
MyNet
MyCloud
Music
Gallery
Gmail
4:38

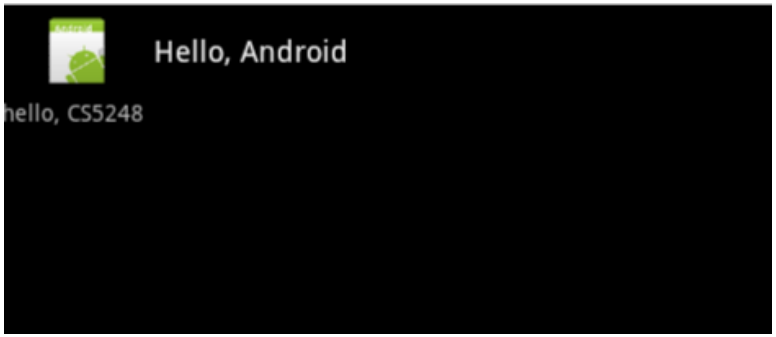
Let’s Debug the code

Always, Look at DDMS !!!



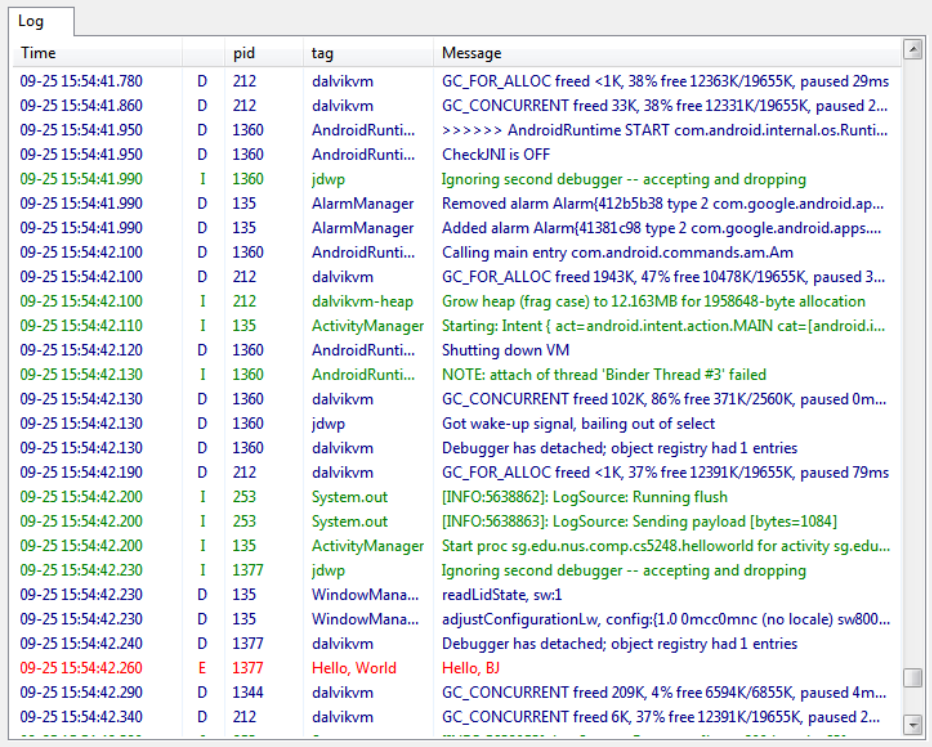
Correction

```
package sg.edu.nus.comp.cs5248.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;


public class HelloWorldActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        TextView tv = (TextView) findViewById(R.id.helloId);
        tv.setText("hello, CS5248");
    }
}
```



Debugging via Dalvik Debug Monitor Service (DDMS)

```
Log.e("Hello, World", "Hello, BJ");
```



Log

Time		pid	tag	Message
09-25 15:54:41.780	D	212	dalvikvm	GC_FOR_ALLOC freed <1K, 38% free 12363K/19655K, paused 29ms
09-25 15:54:41.860	D	212	dalvikvm	GC_CONCURRENT freed 33K, 38% free 12331K/19655K, paused 2...
09-25 15:54:41.950	D	1360	AndroidRunti...	>>>>>> AndroidRuntime START com.android.internal.os.Runti...
09-25 15:54:41.950	D	1360	AndroidRunti...	CheckJNI is OFF
09-25 15:54:41.990	I	1360	jdwp	Ignoring second debugger -- accepting and dropping
09-25 15:54:41.990	D	135	AlarmManager	Removed alarm Alarm{412b5b38 type 2 com.google.android.ap...
09-25 15:54:41.990	D	135	AlarmManager	Added alarm Alarm{41381c98 type 2 com.google.android.apps....
09-25 15:54:42.100	D	1360	AndroidRunti...	Calling main entry com.android.commands.am.Am
09-25 15:54:42.100	D	212	dalvikvm	GC_FOR_ALLOC freed 1943K, 47% free 10478K/19655K, paused 3...
09-25 15:54:42.100	I	212	dalvikvm-heap	Grow heap (frag case) to 12.163MB for 1958648-byte allocation
09-25 15:54:42.110	I	135	ActivityManager	Starting: Intent { act=android.intent.action.MAIN cat=[android.i...
09-25 15:54:42.120	D	1360	AndroidRunti...	Shutting down VM
09-25 15:54:42.130	I	1360	AndroidRunti...	NOTE: attach of thread 'Binder Thread #3' failed
09-25 15:54:42.130	D	1360	dalvikvm	GC_CONCURRENT freed 102K, 86% free 371K/2560K, paused 0m...
09-25 15:54:42.130	D	1360	jdwp	Got wake-up signal, bailing out of select
09-25 15:54:42.130	D	1360	dalvikvm	Debugger has detached; object registry had 1 entries
09-25 15:54:42.190	D	212	dalvikvm	GC_FOR_ALLOC freed <1K, 37% free 12391K/19655K, paused 79ms
09-25 15:54:42.200	I	253	System.out	[INFO:5638862]: LogSource: Running flush
09-25 15:54:42.200	I	253	System.out	[INFO:5638863]: LogSource: Sending payload [bytes=1084]
09-25 15:54:42.200	I	135	ActivityManager	Start proc sg.edu.nus.comp.cs5248.helloworld for activity sg.edu...
09-25 15:54:42.230	I	1377	jdwp	Ignoring second debugger -- accepting and dropping
09-25 15:54:42.230	D	135	WindowMana...	readLidState, sw:1
09-25 15:54:42.230	D	135	WindowMana...	adjustConfigurationLw, config:{1.0 0mcc0mnc (no locale) sw800...
09-25 15:54:42.240	D	1377	dalvikvm	Debugger has detached; object registry had 1 entries
09-25 15:54:42.260	E	1377	Hello, World	Hello, BJ
09-25 15:54:42.290	D	1344	dalvikvm	GC_CONCURRENT freed 209K, 4% free 6594K/6855K, paused 4m...
09-25 15:54:42.340	D	212	dalvikvm	GC_CONCURRENT freed 6K, 37% free 12391K/19655K, paused 2...

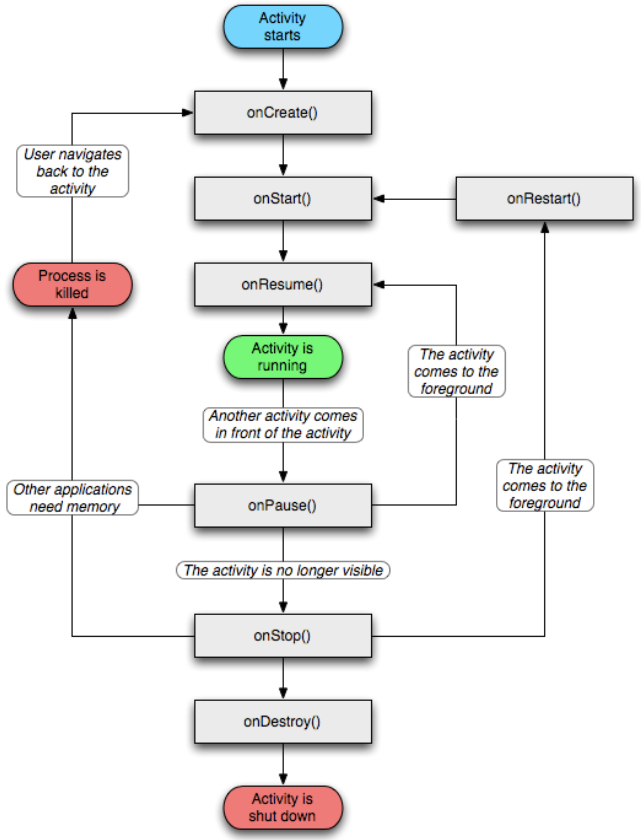
Now, everything is ready

Application Components

- Android application
  - .apk : android package
- Four Application Components
  - Activity
  - Service
  - Content Provider
  - Broadcast Receiver
- Communication among components except Content Provider
  - Intent

Activity 1

- Activity Lifecycle



- Implement Lifecycle Callbacks

```
public class ExampleActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        // The activity is being created.
    }
    @Override
    protected void onStart() {
        super.onStart();
        // The activity is about to become visible.
    }
    @Override
    protected void onResume() {
        super.onResume();
        // The activity has become visible (it is now "resumed").
    }
    @Override
    protected void onPause() {
        super.onPause();
        // Another activity is taking focus (this activity is about
    }
    @Override
    protected void onStop() {
        super.onStop();
        // The activity is no longer visible (it is now "stopped")
    }
    @Override
    protected void onDestroy() {
        super.onDestroy();
        // The activity is about to be destroyed.
    }
}
```

Activity 2

- Hierarchical View Architecture
  - ViewGroup (Layout)
    - View
    - View
    - ViewGroup (Layout)
      - View
      - View
      - …
    - View
    - ViewGroup
      - …

- Declare Activity in the manifest

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

Activity 3

- Start an Activity

```
Intent intent = new Intent(this, SignInActivity.class);
startActivity(intent);
```

- Start an Activity for a Result
  - Caller Activity

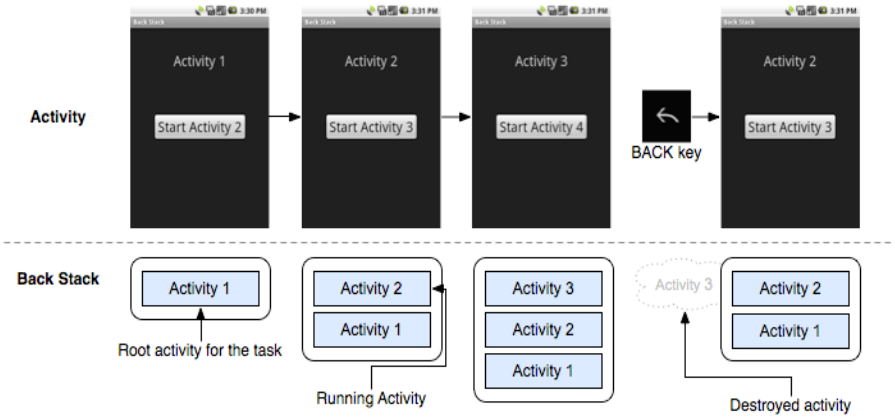
```
private void pickContact() {
    // Create an intent to "pick" a contact, as defined by
    Intent intent = new Intent(Intent.ACTION_PICK, Contacts
    startActivityForResult(intent, PICK_CONTACT_REQUEST);
}

@Override
protected void onActivityResult(int requestCode, int result
    // If the request went well (OK) and the request was PI
    if (resultCode == Activity.RESULT_OK && requestCode ==
        // Perform a query to the contact's content provide
        Cursor cursor = getContentResolver().query(data.get
        new String[] {Contacts.DISPLAY_NAME}, null, null, n
        if (cursor.moveToFirst()) { // True if the cursor i
            int columnIndex = cursor.getColumnIndex(Contact
            String name = cursor.getString(columnIndex);
            // Do something with the selected contact's nam
        }
    }
}
```

  - Callee Activity

```
setResult(RESULT_OK, new Intent().putExtra("key", "value"));
```

- finish

- Activities and Stack



Services

- A service is a component that runs in the bg to perform long-running operations
- A service does not provide a user interface
- Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it
- A service is implemented as a subclass of Service
- E.g a service might play music in the background while the user is in a different app

Broadcast Receivers

- A broadcast receiver is a component that responds to system-wide broadcast announcements
- Apps can also initiate broadcasts
- Although broadcast receivers don't display a UI, they may create a status bar notification to alert the user when a broadcast event occurs
- A broadcast receiver is implemented as a subclass of BroadcastReceiver and each broadcast is delivered as an Intent object.

Content Providers

- A content provider manages a shared set of app data
- Through the content provider, other apps can query or even modify the data
- Any app with the proper permissions can query part of the content provider
- A content provider is implemented as a subclass of ContentProvider and must implement a standard set of APIs that enable other apps to perform transactions.

Intent

- An intent is an abstract description of an operation to be performed
  - It can be used with startActivity to launch an Activity
  - broadcastIntent to send it to any interested BroadcastReceiver components
  - startService(Intent) or bindService(Intent, ServiceConnection, int) to communicate with a background Service

Activating Components

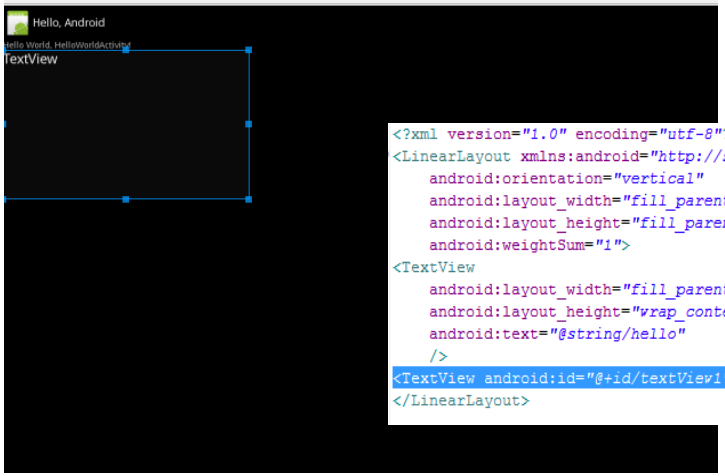
- Passing an Intent to startActivity() or startActivityForResult()
- passing an Intent to startService(). Or bind to the service by passing an Intent to bindService().
- Passing an Intent to methods like sendBroadcast(), sendOrderedBroadcast(), or sendStickyBroadcast().
- perform a query to a content provider by calling query() on a ContentResolver.

UI – Declaring Layout

- Initiated when called setContentView() on onCreate()

```
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.main_layout);
}
```

- Use Visual Layout Editor for initial layout design.
- Edit XML file extensively.



wrap_content or fill_parent

UI – Creating Menu

- Menu Types
  - Options Menu
    - Appears when a user touches MENU button.
  - Context Menu
  - Submenu
- How to
  - Specify menu items in a XML resource menu
  - Inflate a Menu Resource
  - Respond to user Action

At res/menu/game_menu.xml

```xml
<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
    <item android:id="@+id/new_game"
          android:icon="@drawable/ic_new_game"
          android:title="@string/new_game" />
    <item android:id="@+id/help"
          android:icon="@drawable/ic_help"
          android:title="@string/help" />
</menu>
```

inflate

```java
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.game_menu, menu);
    return true;
}
```

callback

```java
@Override
public boolean onOptionsItemSelected(MenuItem item) {
    // Handle item selection
    switch (item.getItemId()) {
    case R.id.new_game:
        newGame();
        return true;
    case R.id.help:
        showHelp();
        return true;
    default:
        return super.onOptionsItemSelected(item);
    }
}
```

UI – Handling UI Events

- onClick, onLongClick, onKey, onTouch, …

```
// Create an anonymous implementation of OnClickListener
private OnClickListener mCorkyListener = new OnClickListener() {
    public void onClick(View v) {
      // do something when the button is clicked
    }
};

protected void onCreate(Bundle savedValues) {
    ...
    // Capture our button from layout
    Button button = (Button)findViewById(R.id.corky);
    // Register the onClick listener with the implementation above
    button.setOnClickListener(mCorkyListener);
    ...
}
```

```
public class ExampleActivity extends Activity implements OnClickListener {
    protected void onCreate(Bundle savedValues) {
        ...
        Button button = (Button)findViewById(R.id.corky);
        button.setOnClickListener(this);
    }

    // Implement the OnClickListener callback
    public void onClick(View v) {
      // do something when the button is clicked
    }
    ...
}
```

Thread

- UI Thread
  - “Main” thread per application responsible for interacting with UI components.
- “Application Not Responding” problem
  - If UI thread is blocked more than several seconds, ANR dialog appears.
  - Do not block the UI thread
  - Do not access UI components outside UI thread.

```
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            Bitmap b = loadImageFromNetwork("http://example.com/ir
            mImageView.setImageBitmap(b);
        }
    }).start();
}
```

modify

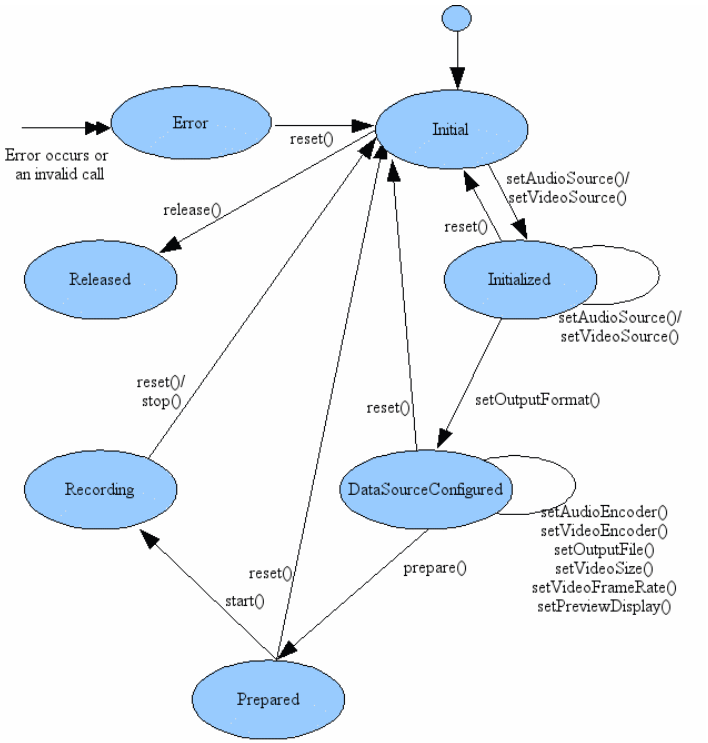
```
public void onClick(View v) {
    new Thread(new Runnable() {
        public void run() {
            final Bitmap bitmap = loadImageFromNetwork("http:
            mImageView.post(new Runnable() {
                public void run() {
                    mImageView.setImageBitmap(bitmap);
                }
            });
        }
    }).start();
}
```

Use AsyncTask, Instead

```
public void onClick(View v) {
    new DownloadImageTask().execute("http://example.com/image.png");
}

private class DownloadImageTask extends AsyncTask<String, Void, Bitmap> {
    /** The system calls this to perform work in a worker thread and
      * delivers it the parameters given to AsyncTask.execute() */
    protected Bitmap doInBackground(String... urls) {
        return loadImageFromNetwork(urls[0]);
    }

    /** The system calls this to perform work in the UI thread and delivers
      * the result from doInBackground() */
    protected void onPostExecute(Bitmap result) {
        mImageView.setImageBitmap(result);
    }
}
```

Misc - Media Recorder

- Modify CameraPreview to see the video during recording.



MediaRecorder state diagram

Misc - HTTP Post

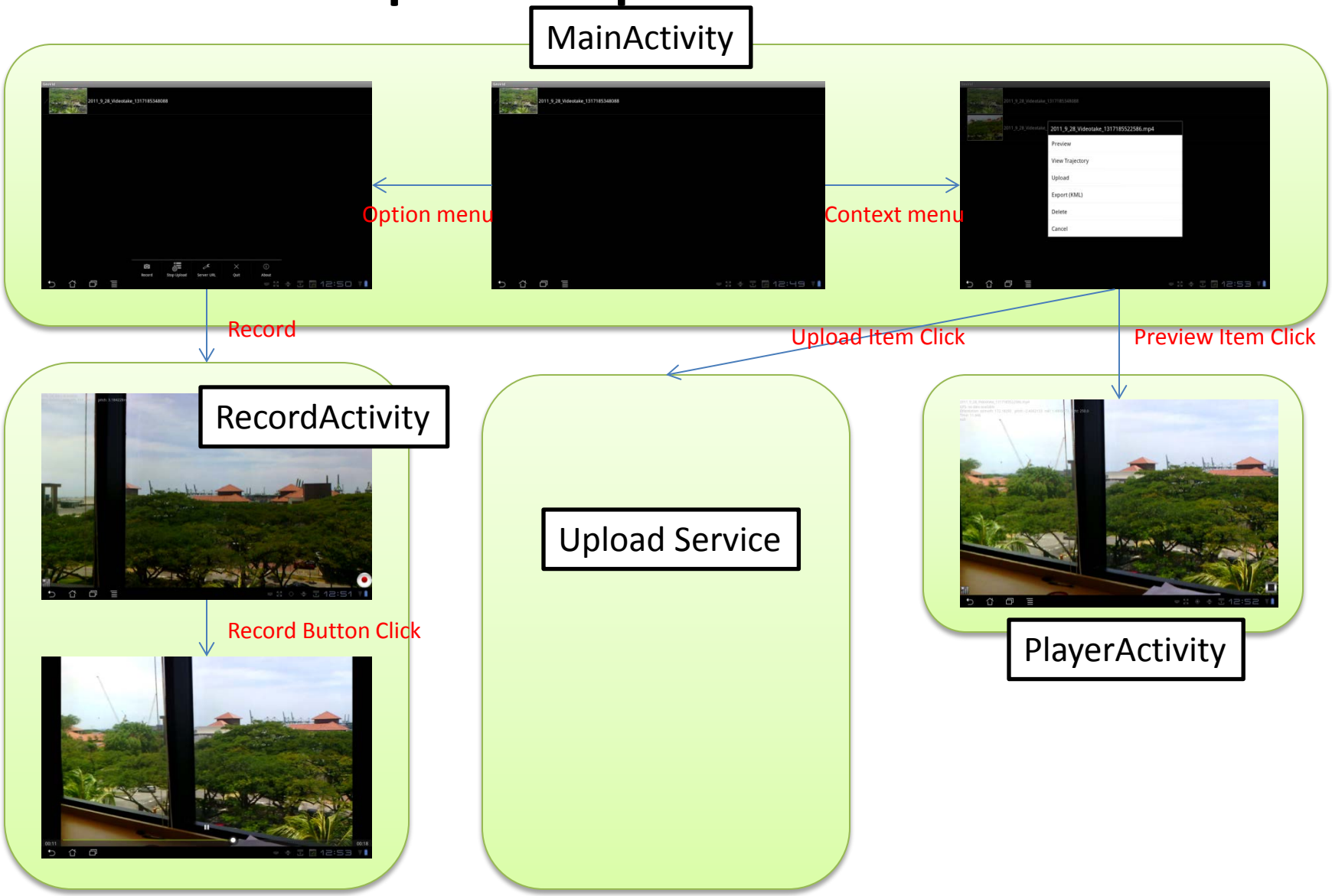
- Use HttpClient and HttpPost.
- Use “multi-part/form-data” to encapsulate segment.
- Do not excessively use memory.

Misc – Permission Issue

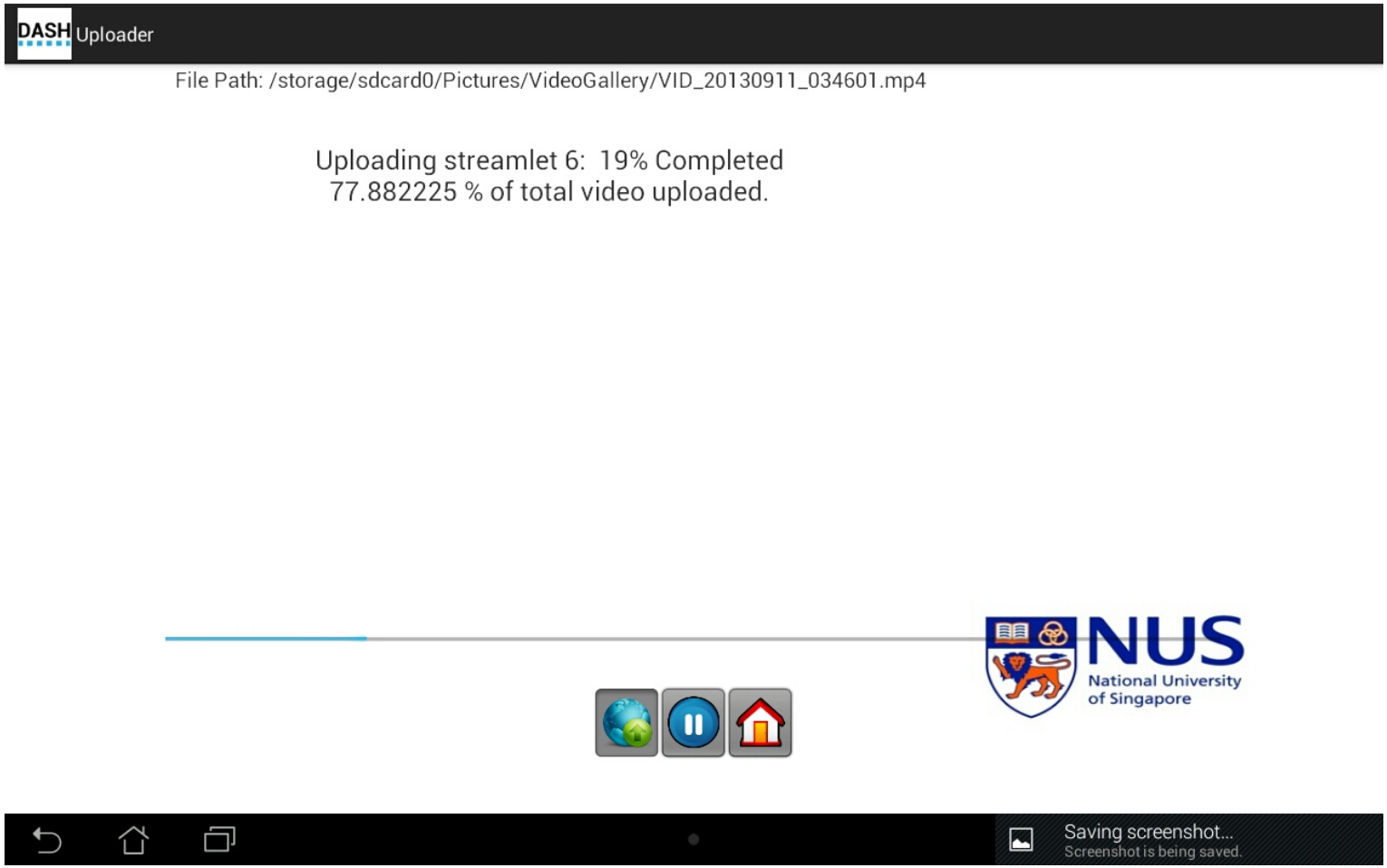
- Add following permissions to the manifest file.

```
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_MOCK_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.WAKE_LOCK" />

<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
```

Sample Implementation



Upload Service



Misc - Integration with MP4Parser

- Adding a jar file
  - Create its jar file.
  - Add jar file to the app.
    - Build Path > Configure Build Path > Libraries > Add JARs.
- If you include source files directly,
  - Put Isoparser-default.properties to assets folder
  - Change getResourceAsStream("isoparser-default.properties") in PropertyBoxParserImpl to "/assets/isoparser-default.properties".

https://github.com/sannies/mp4parser
https://code.google.com/p/mp4parser/

Misc - Adapter

- An Adapter object acts as a bridge between an AdapterView and the underlying data for that view (i.e. ListView, GridView, Gallery, etc.).
- An AdapterView is a view whose children are determined by an Adapter.

http://developer.android.com/reference/android/widget/Adapter.html
http://developer.android.com/reference/android/widget/AdapterView.html

Misc - Fragments

- A Fragment represents a behavior or a portion of user interface in an Activity.
- A fragment must always be embedded in an activity and the fragment's lifecycle is directly affected by the host activity's lifecycle.

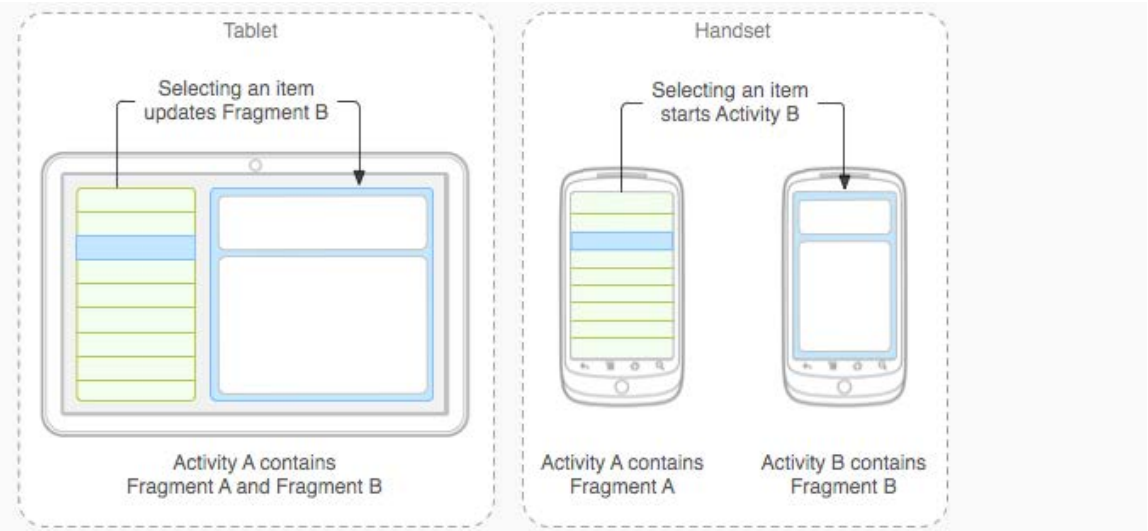
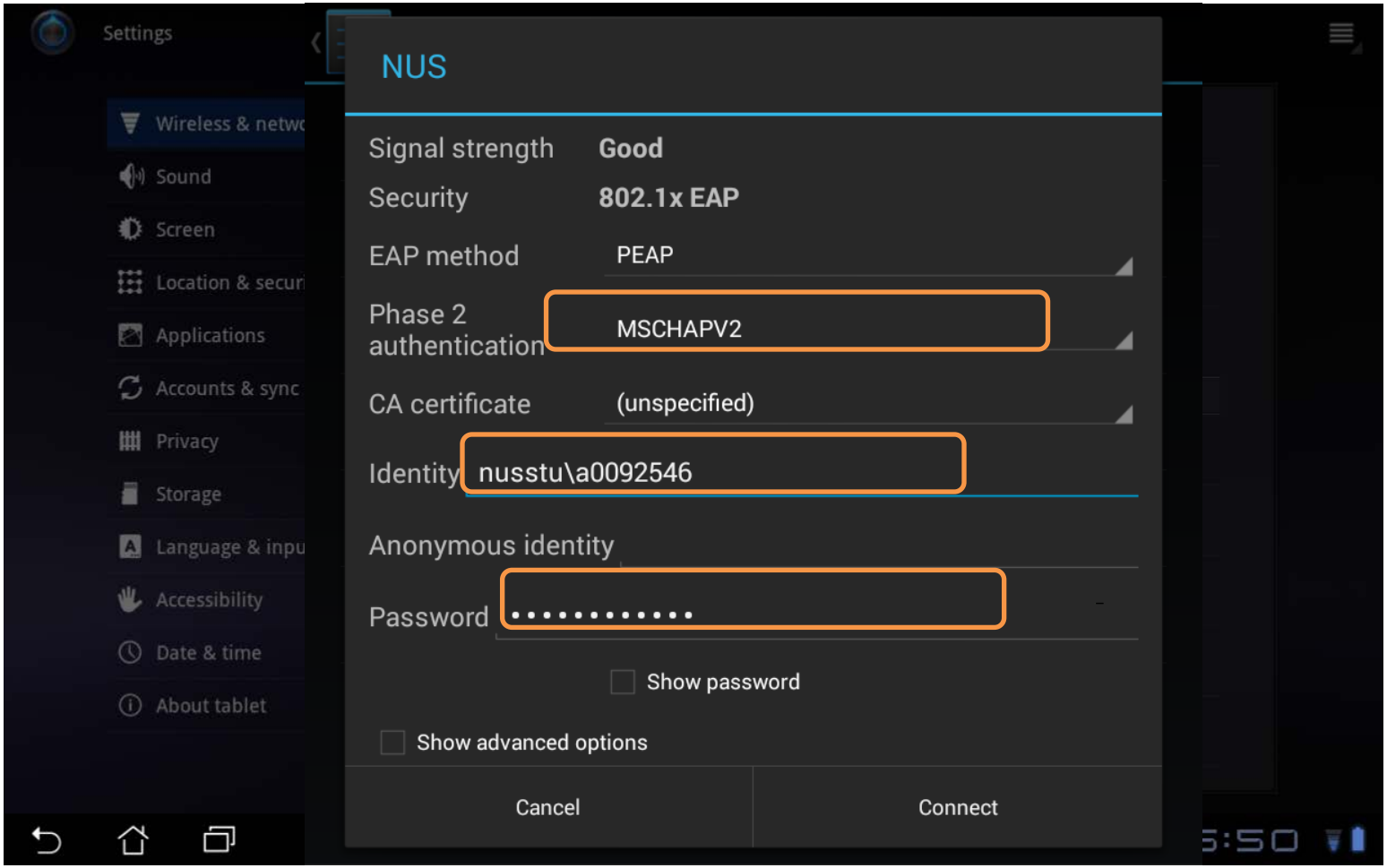


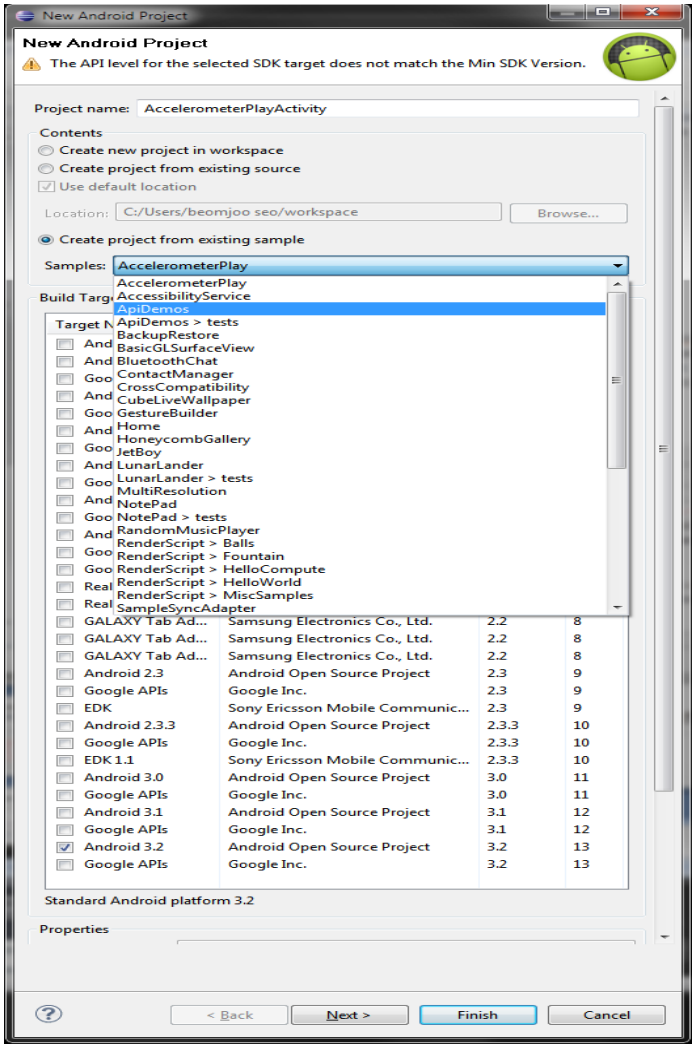
Figure 1. An example of how two UI modules defined by fragments can be combined into one activity for a tablet design, but separated for a handset design.

http://developer.android.com/guide/components/fragments.html

Misc - Connecting to NUS Wifi



Final Comment: Use ApiDemo !!!



General Reference

- How to work with Android GUI layouts https://www.youtube.com/watch?v=xn9KYnwIoBE
- Building and running your app http://developer.android.com/tools/building/building-eclipse.html
- Deployment of the **.apk file** http://stackoverflow.com/questions/3480201/how-do-you-install-an-apk-file-in-the-android-emulator
- Find the hidden developer option http://www.cnet.com/how-to/restore-the-developer-options-menu-in-android-4-2/
- Rotation lock/unlock http://www.howtogeek.com/howto/26715/how-to-make-your-android-phone-stop-rotating-the-screen-when-you%E2%80%99re-reading-sideways/
- Video tutorials http://www.youtube.com/watch?v=5RHtKIo_KDI
- Load/Open an existing package (e.g., the provided ClassExamples) into your eclipse
  - File --> Import ... --> General-Tab --> Existing Projects into Workspace (and click **Next**)
  - Select root Directory: click **Browse:** select the folder the include the *AndroidManifest.xml* or *project.properties* files.
  - Click finish