# Android Development Tutorial

Human-Computer Interaction (COMP 4020)
Winter 2014

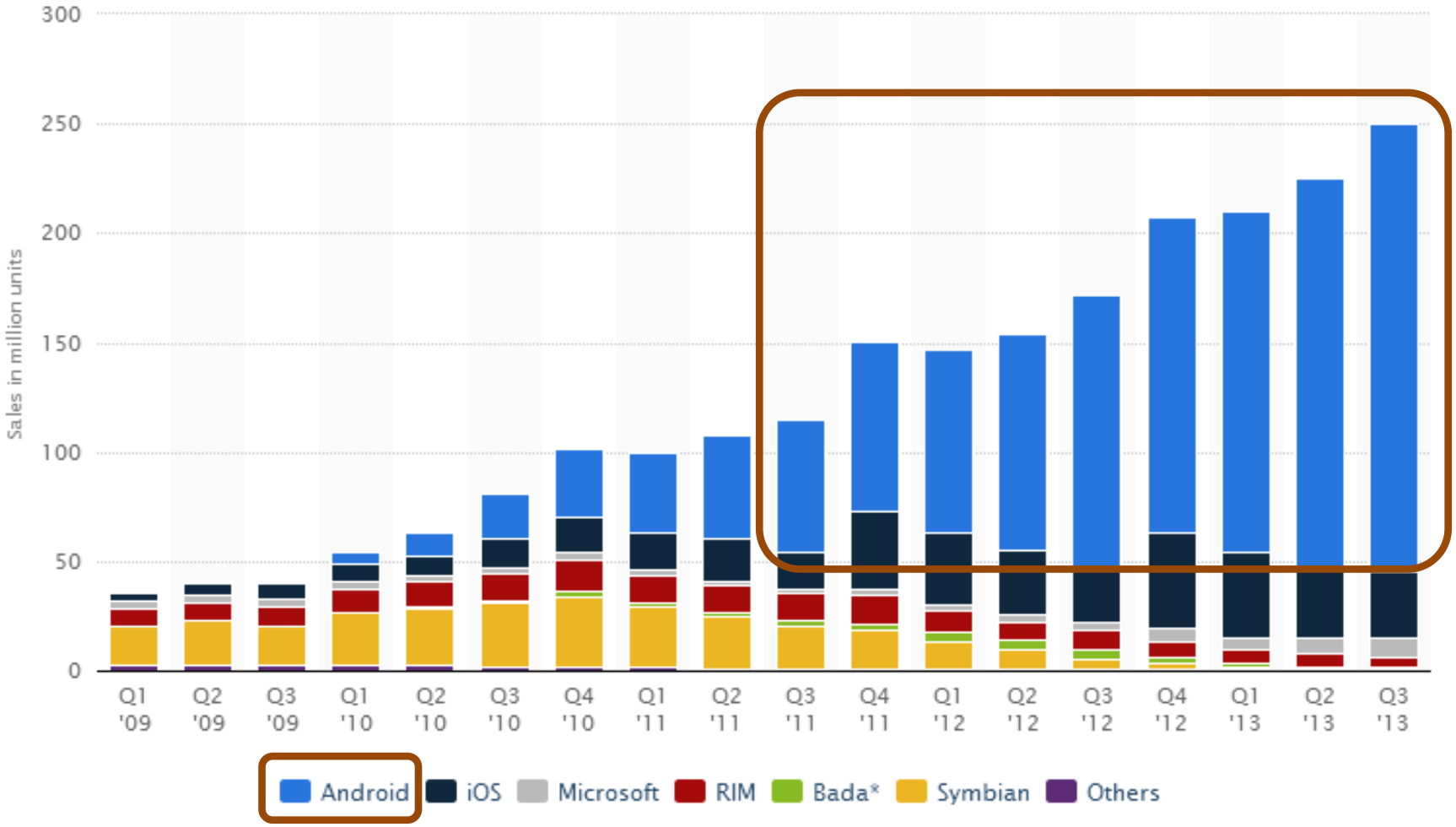
# Today: Android Tutorial

1. Installation Issues
2. Folder Structure
3. Core Components
4. Sample Applications

# Mobile Phone OS

- Symbian
- iOS
- BlackBerry
- Windows Phone
- Android
- ….

# World-Wide Smartphone Sales (In Million Units)



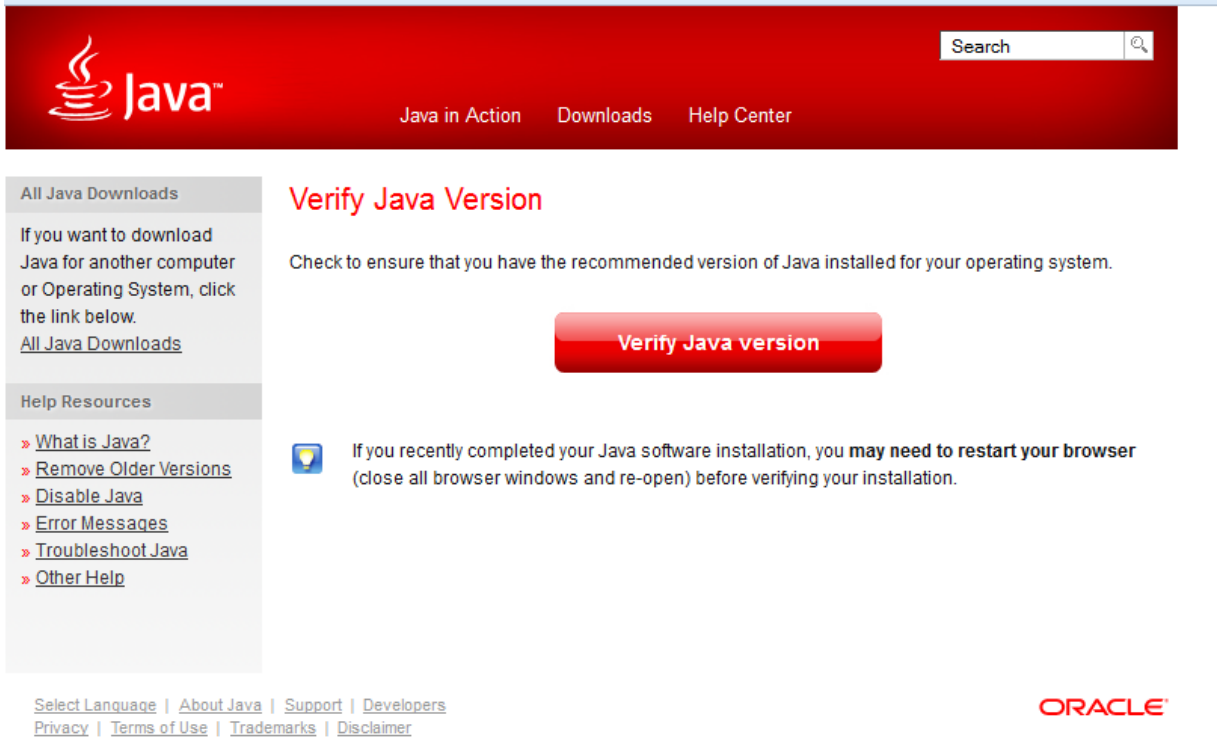

Source: Smartphone Sales, by operating system, Q1 2009-Q3 2013 http://www.statista.com/statistics/266219/global-smartphone-sales-since-1st-quarter-2009-by-operating-system/

# Installation Requirements

1. Java Runtime Environment (JRE)
2. Java Development Kit (JDK)
3. Android Developer Tools Bundle (ADT Bundle)
   - Eclipse + ADT plugin
   - Android SDK Tools
   - Android Platform-tools
   - The latest Android platform
   - The latest Android system image for the emulator

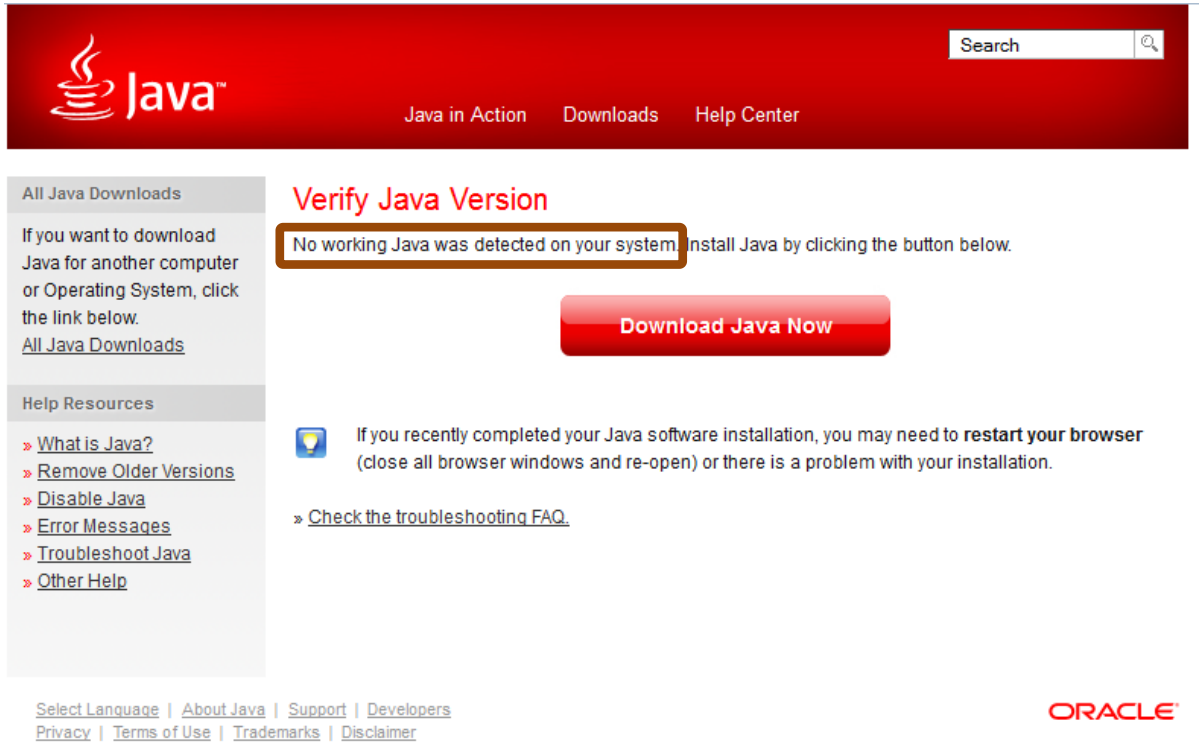
# Verify Java Version

http://www.java.com/en/download/installed.jsp



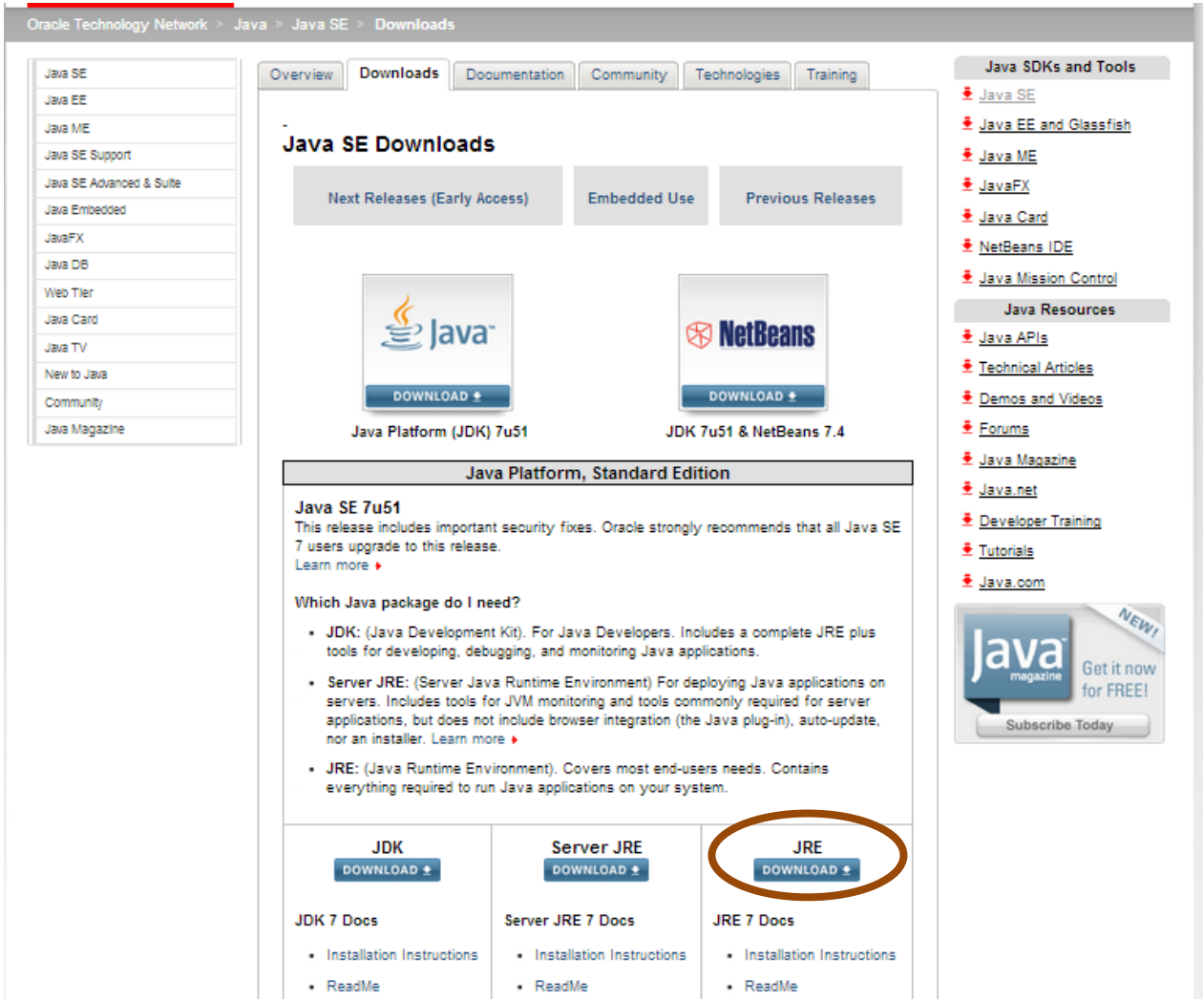
# Verify Java Version

http://www.java.com/en/download/installed.jsp

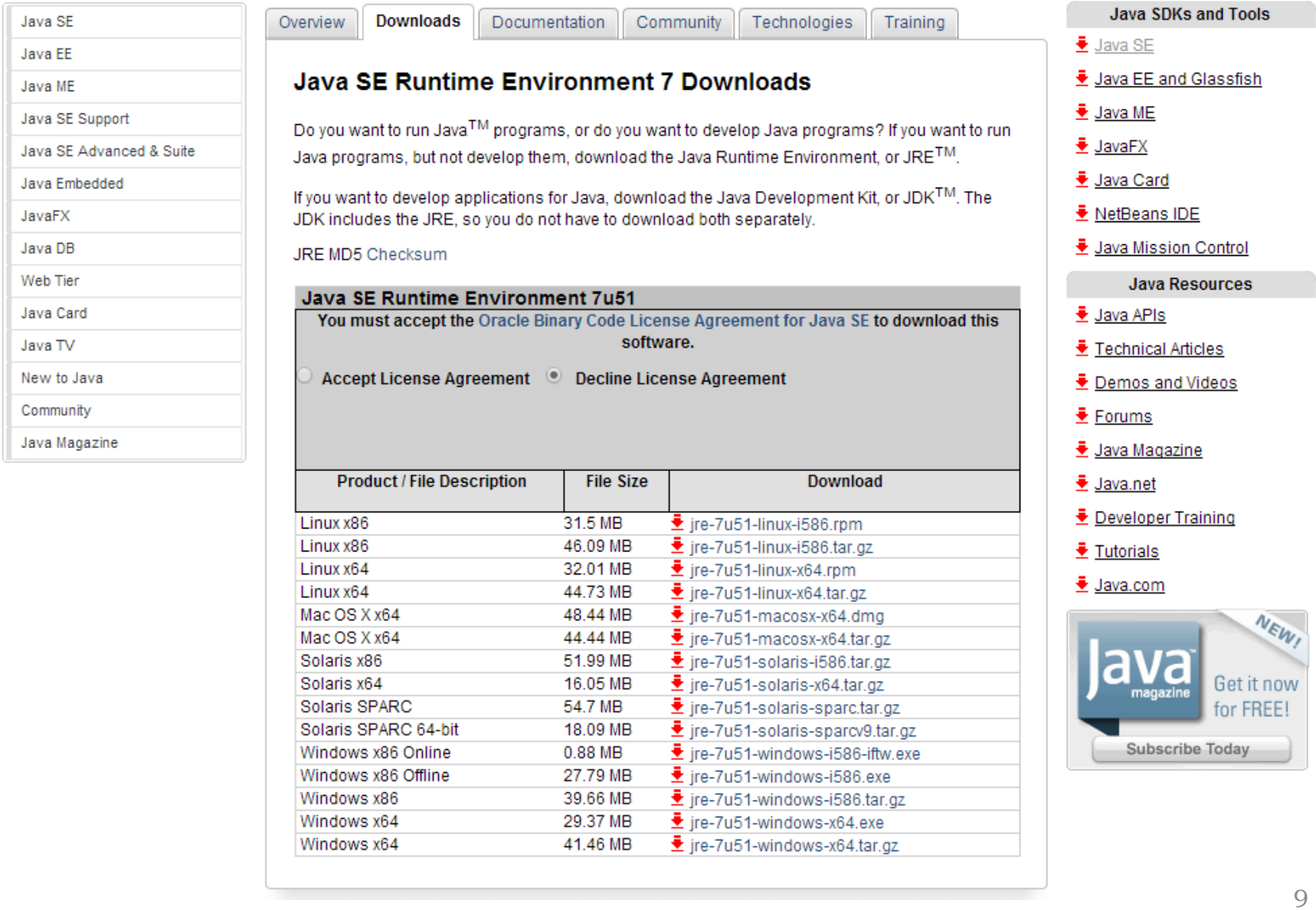


# Install JRE

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



# Install JRE



- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- JavaFX
- Java DB
- Web Tier
- Java Card
- Java TV
- New to Java
- Community
- Java Magazine

Overview | **Downloads** | Documentation | Community | Technologies | Training

## Java SE Runtime Environment 7 Downloads

Do you want to run Java™ programs, or do you want to develop Java programs? If you want to run Java programs, but not develop them, download the Java Runtime Environment, or JRE™.

If you want to develop applications for Java, download the Java Development Kit, or JDK™. The JDK includes the JRE, so you do not have to download both separately.

JRE MD5 Checksum

**Java SE Runtime Environment 7u51**

**You must accept the Oracle Binary Code License Agreement for Java SE to download this software.**

○ **Accept License Agreement** ◉ **Decline License Agreement**

| Product / File Description | File Size | Download |
|---|---|---|
| Linux x86 | 31.5 MB | jre-7u51-linux-i586.rpm |
| Linux x86 | 46.09 MB | jre-7u51-linux-i586.tar.gz |
| Linux x64 | 32.01 MB | jre-7u51-linux-x64.rpm |
| Linux x64 | 44.73 MB | jre-7u51-linux-x64.tar.gz |
| Mac OS X x64 | 48.44 MB | jre-7u51-macosx-x64.dmg |
| Mac OS X x64 | 44.44 MB | jre-7u51-macosx-x64.tar.gz |
| Solaris x86 | 51.99 MB | jre-7u51-solaris-i586.tar.gz |
| Solaris x64 | 16.05 MB | jre-7u51-solaris-x64.tar.gz |
| Solaris SPARC | 54.7 MB | jre-7u51-solaris-sparc.tar.gz |
| Solaris SPARC 64-bit | 18.09 MB | jre-7u51-solaris-sparcv9.tar.gz |
| Windows x86 Online | 0.88 MB | jre-7u51-windows-i586-iftw.exe |
| Windows x86 Offline | 27.79 MB | jre-7u51-windows-i586.exe |
| Windows x86 | 39.66 MB | jre-7u51-windows-i586.tar.gz |
| Windows x64 | 29.37 MB | jre-7u51-windows-x64.exe |
| Windows x64 | 41.46 MB | jre-7u51-windows-x64.tar.gz |

**Java SDKs and Tools**

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE
- Java Mission Control

**Java Resources**

- Java APIs
- Technical Articles
- Demos and Videos
- Forums
- Java Magazine
- Java.net
- Developer Training
- Tutorials
- Java.com

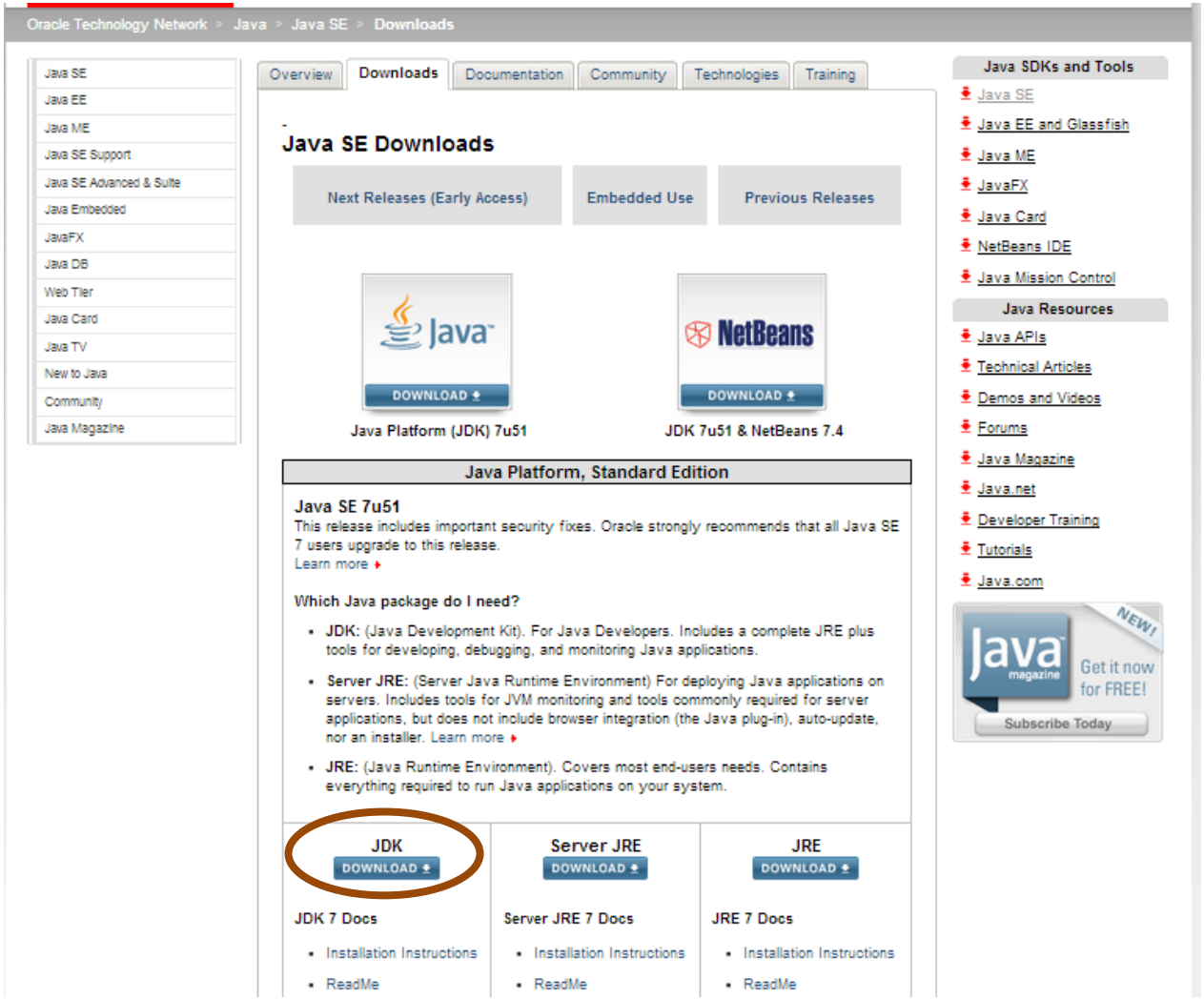
NEW! Java magazine Get it now for FREE! Subscribe Today

# Installation Requirements

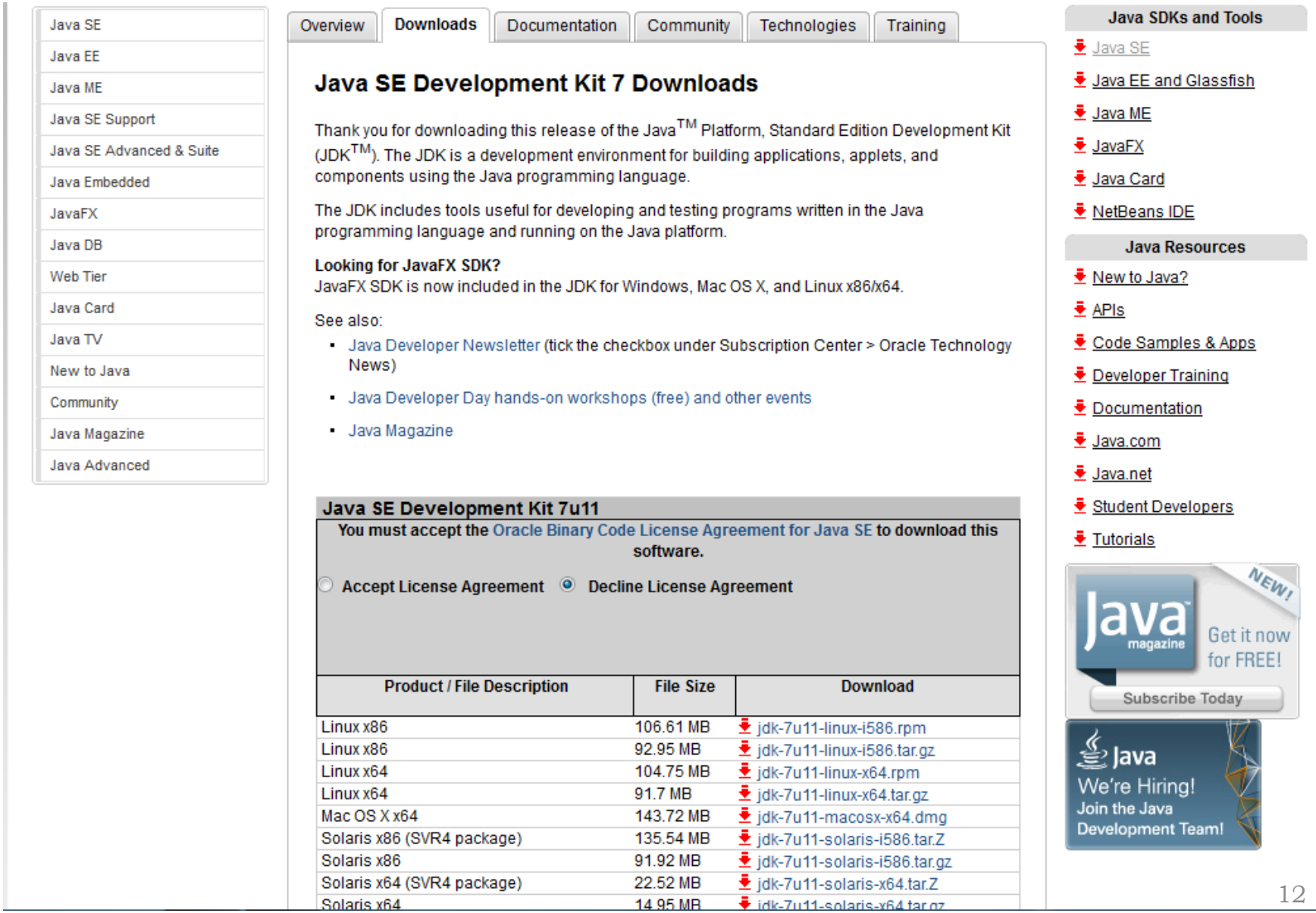
1. Java Runtime Environment (JRE) √
2. Java Development Kit (JDK)
3. Android Developer Tools Bundle (ADT Bundle)
   - Eclipse + ADT plugin
   - Android SDK Tools
   - Android Platform-tools
   - The latest Android platform
   - The latest Android system image for the emulator

# Install JDK

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



# Install JDK



- Java SE
- Java EE
- Java ME
- Java SE Support
- Java SE Advanced & Suite
- Java Embedded
- JavaFX
- Java DB
- Web Tier
- Java Card
- Java TV
- New to Java
- Community
- Java Magazine
- Java Advanced

Overview | Downloads | Documentation | Community | Technologies | Training

## Java SE Development Kit 7 Downloads

Thank you for downloading this release of the Java™ Platform, Standard Edition Development Kit (JDK™). The JDK is a development environment for building applications, applets, and components using the Java programming language.

The JDK includes tools useful for developing and testing programs written in the Java programming language and running on the Java platform.

**Looking for JavaFX SDK?**
JavaFX SDK is now included in the JDK for Windows, Mac OS X, and Linux x86/x64.

See also:

- Java Developer Newsletter (tick the checkbox under Subscription Center > Oracle Technology News)
- Java Developer Day hands-on workshops (free) and other events
- Java Magazine

**Java SE Development Kit 7u11**

You must accept the Oracle Binary Code License Agreement for Java SE to download this software.

Accept License Agreement  Decline License Agreement

| Product / File Description | File Size | Download |
|---|---|---|
| Linux x86 | 106.61 MB | jdk-7u11-linux-i586.rpm |
| Linux x86 | 92.95 MB | jdk-7u11-linux-i586.tar.gz |
| Linux x64 | 104.75 MB | jdk-7u11-linux-x64.rpm |
| Linux x64 | 91.7 MB | jdk-7u11-linux-x64.tar.gz |
| Mac OS X x64 | 143.72 MB | jdk-7u11-macosx-x64.dmg |
| Solaris x86 (SVR4 package) | 135.54 MB | jdk-7u11-solaris-i586.tar.Z |
| Solaris x86 | 91.92 MB | jdk-7u11-solaris-i586.tar.gz |
| Solaris x64 (SVR4 package) | 22.52 MB | jdk-7u11-solaris-x64.tar.Z |
| Solaris x64 | 14.95 MB | jdk-7u11-solaris-x64.tar.gz |

**Java SDKs and Tools**

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE

**Java Resources**

- New to Java?
- APIs
- Code Samples & Apps
- Developer Training
- Documentation
- Java.com
- Java.net
- Student Developers
- Tutorials

NEW! Java magazine Get it now for FREE! Subscribe Today

Java We're Hiring! Join the Java Development Team!

# Installation Requirements

1. Java Runtime Environment (JRE) √
2. Java Development Kit (JDK) √
3. Android Developer Tools Bundle (ADT Bundle)
   - Eclipse + ADT plugin
   - Android SDK Tools
   - Android Platform-tools
   - The latest Android platform
   - The latest Android system image for the emulator

# Install ADT Bundle

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

Developers | Design | Develop | Distribute

Training | API Guides | Reference | Tools | Google Services

- Developer Tools
- Download
  - Setting Up the ADT Bundle
  - Setting Up an Existing IDE
  - Exploring the SDK
  - Download the NDK
- Workflow
- Tools Help
- Revisions
- Extras
- Samples
- ADK

## Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

Download the SDK
ADT Bundle for Windows

With a single download, the ADT Bundle includes everything you need to begin developing apps:

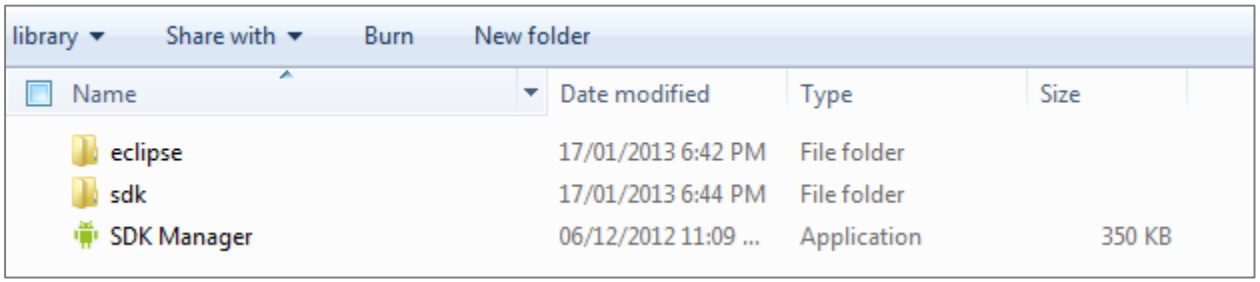
- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

If you prefer to use an existing version of Eclipse or another IDE, you can instead take a more customized approach to installing the Android SDK. See the following instructions.

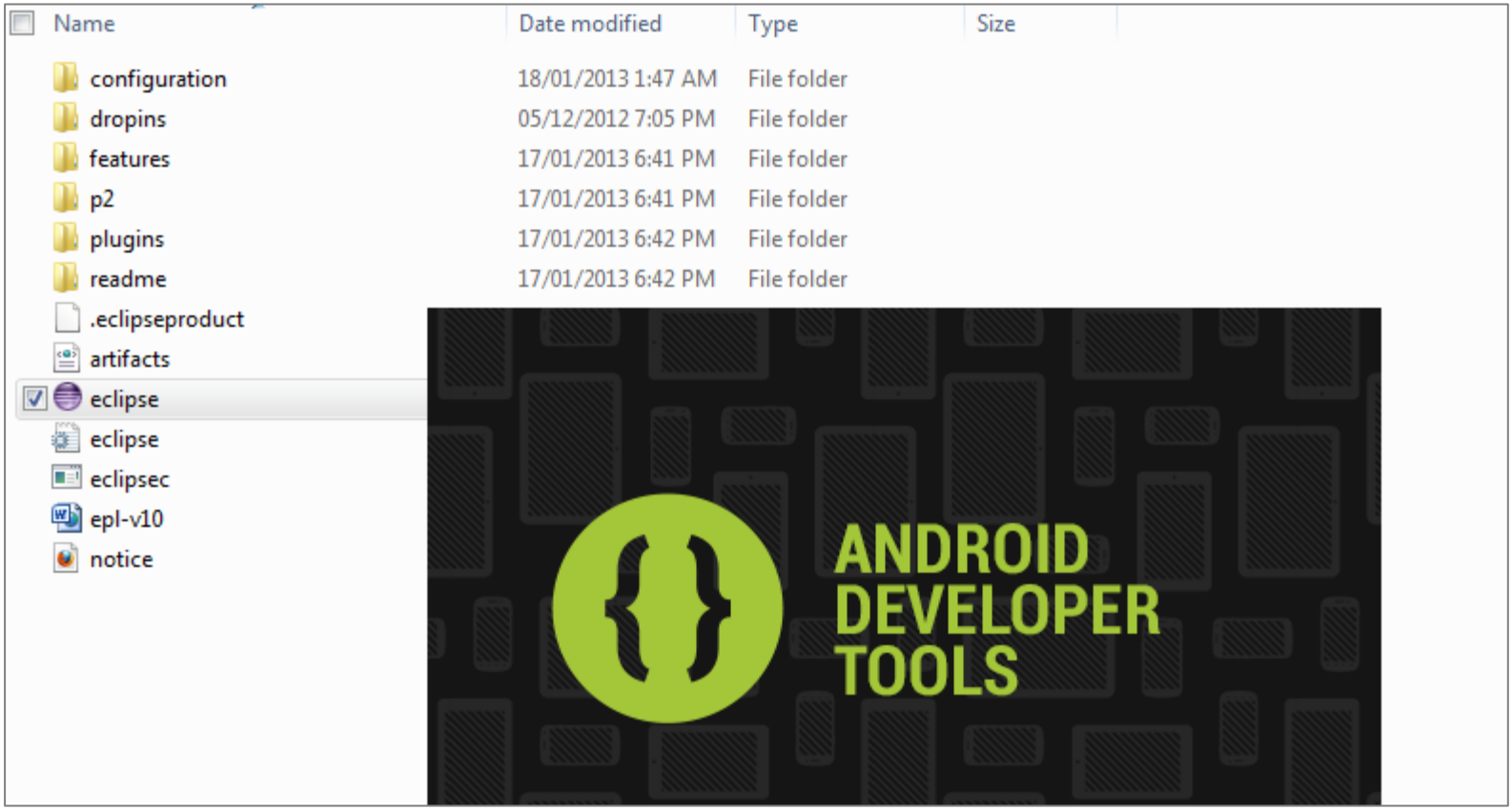
USE AN EXISTING IDE

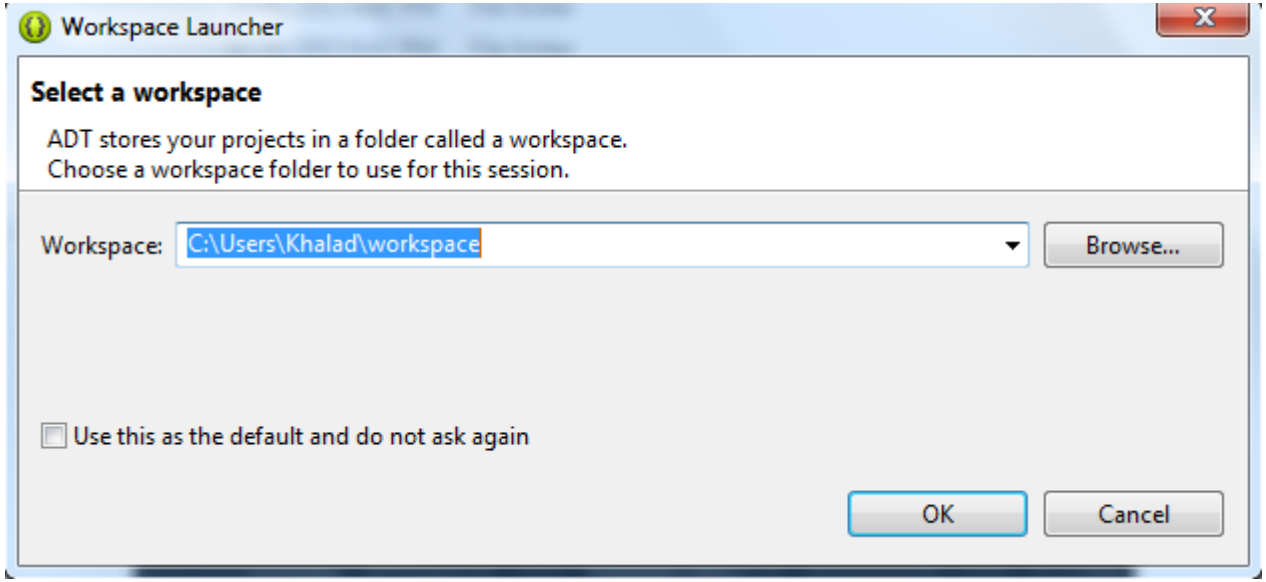
# Eclipse

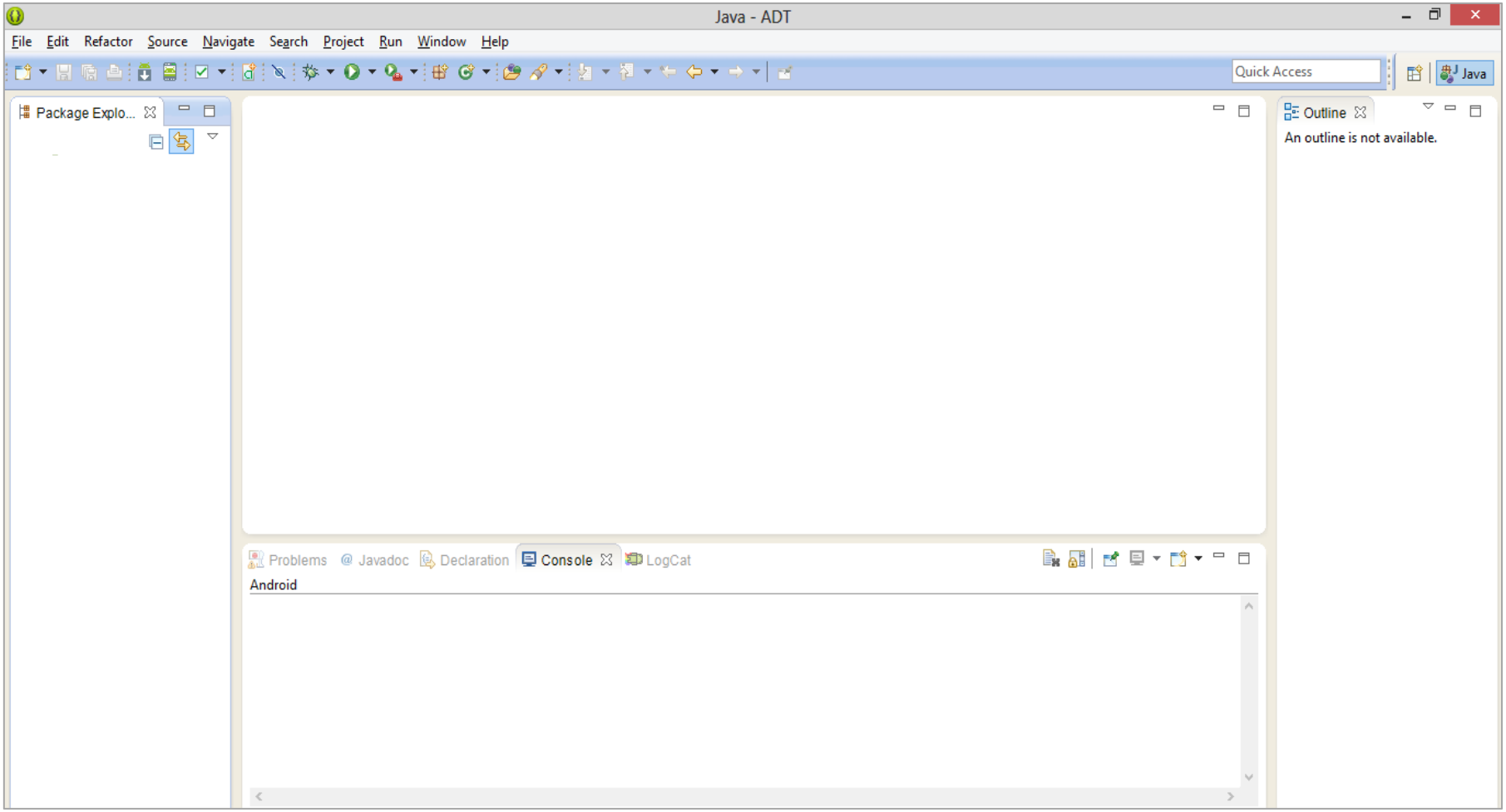
1


library
Share with
Burn
New folder
Name
Date modified
Type
Size
eclipse
17/01/2013 6:42 PM
File folder
sdk
17/01/2013 6:44 PM
File folder
SDK Manager
06/12/2012 11:09 ...
Application
350 KB

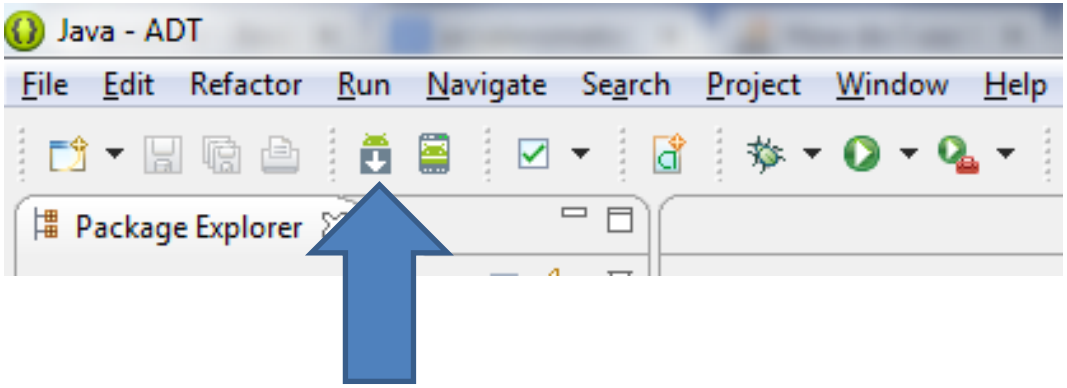
2


Name
Date modified
Type
Size
configuration
18/01/2013 1:47 AM
File folder
dropins
05/12/2012 7:05 PM
File folder
features
17/01/2013 6:41 PM
File folder
p2
17/01/2013 6:41 PM
File folder
plugins
17/01/2013 6:42 PM
File folder
readme
17/01/2013 6:42 PM
File folder
.eclipseproduct
artifacts
eclipse
eclipse
eclipsec
epl-v10
notice
ANDROID
DEVELOPER
TOOLS

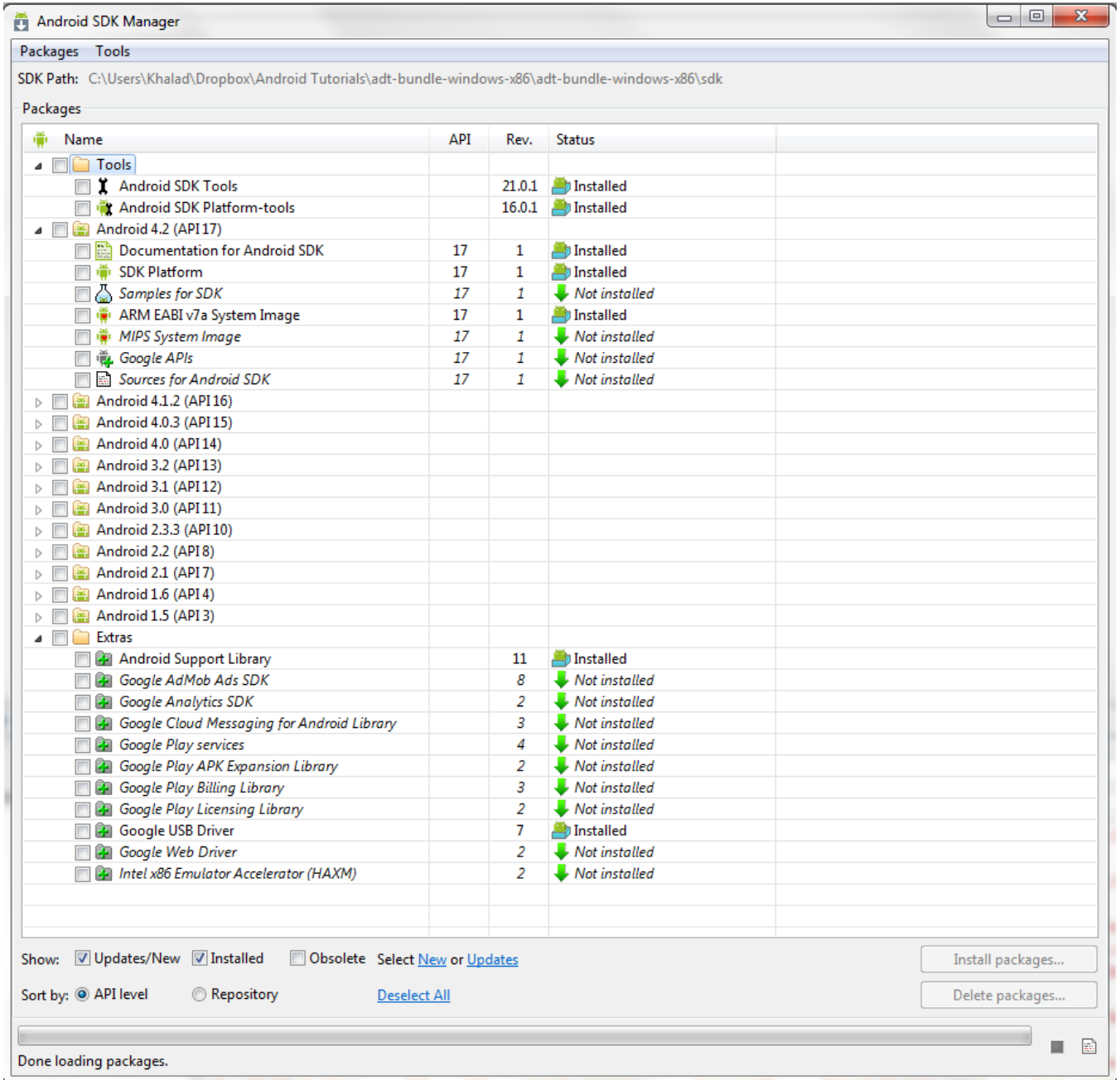

Workspace Launcher
Select a workspace
ADT stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.
Workspace:
C:\Users\Khalad\workspace
Browse...
Use this as the default and do not ask again
OK
Cancel


Java - ADT
File Edit Refactor Source Navigate Search Project Run Window Help
Quick Access
Java
Package Explo...
Outline
An outline is not available.
Problems Javadoc Declaration Console LogCat
Android

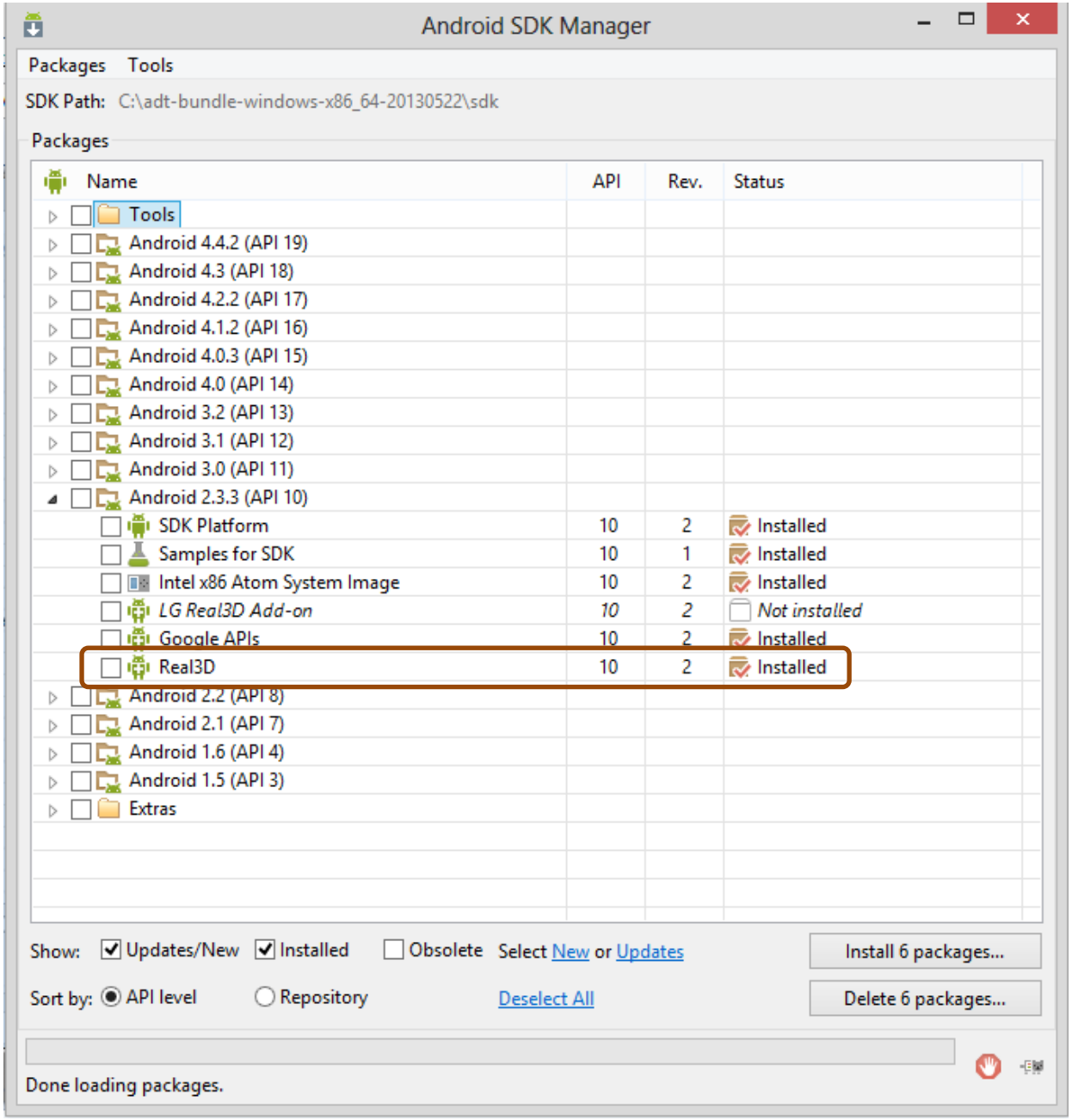
# Android SDK Manager



# Android SDK Manager

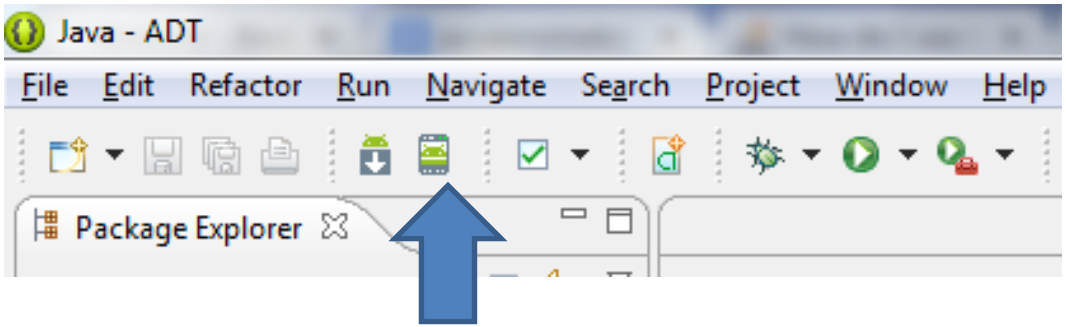


# Android SDK Manager

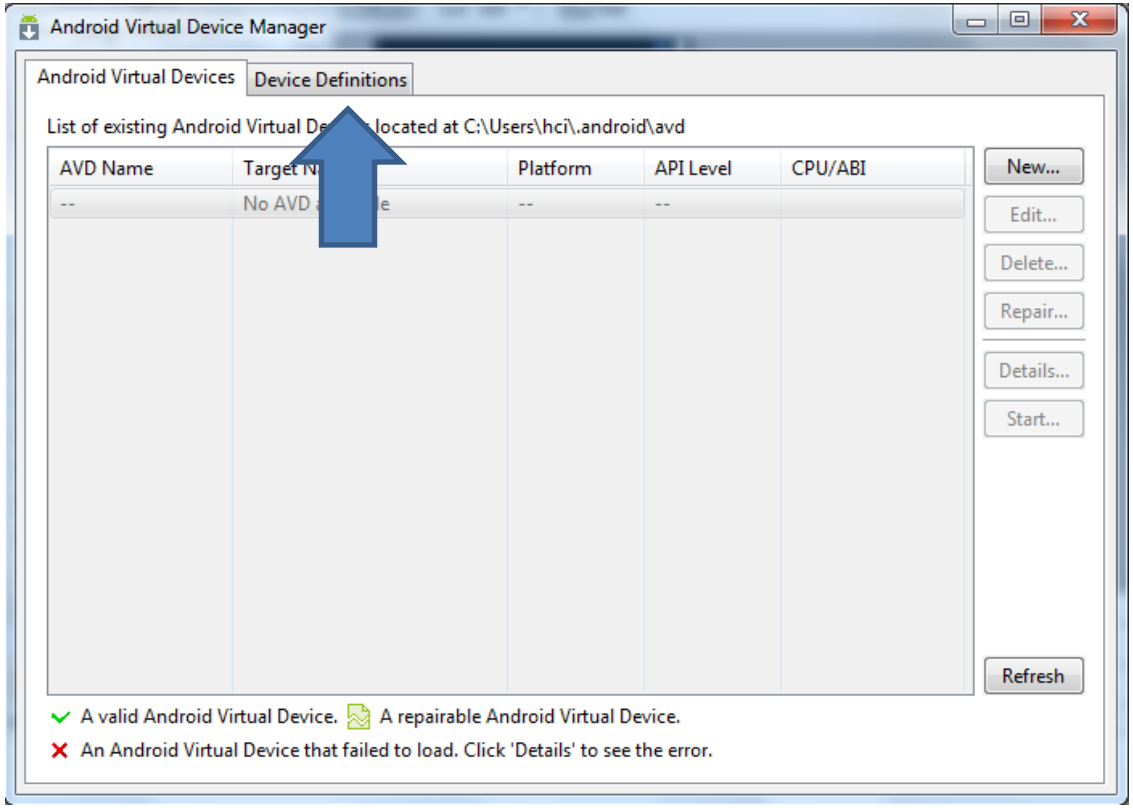


# Android Virtual Device Manager

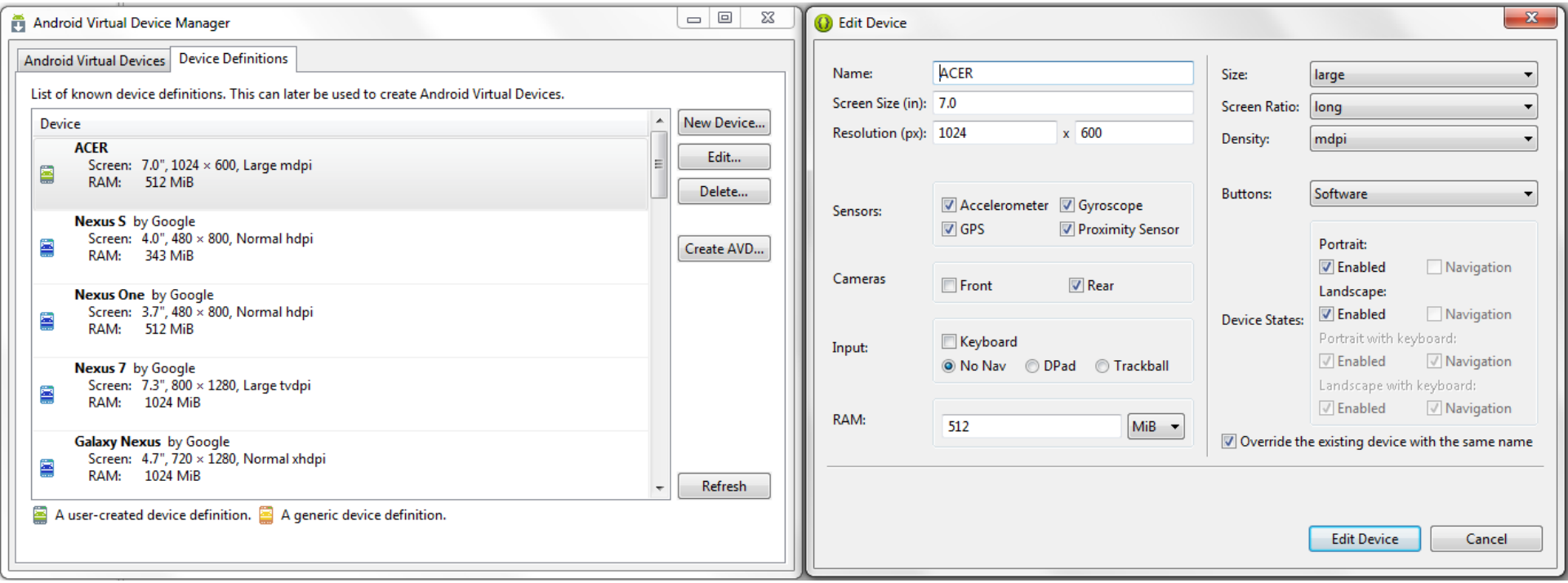
1



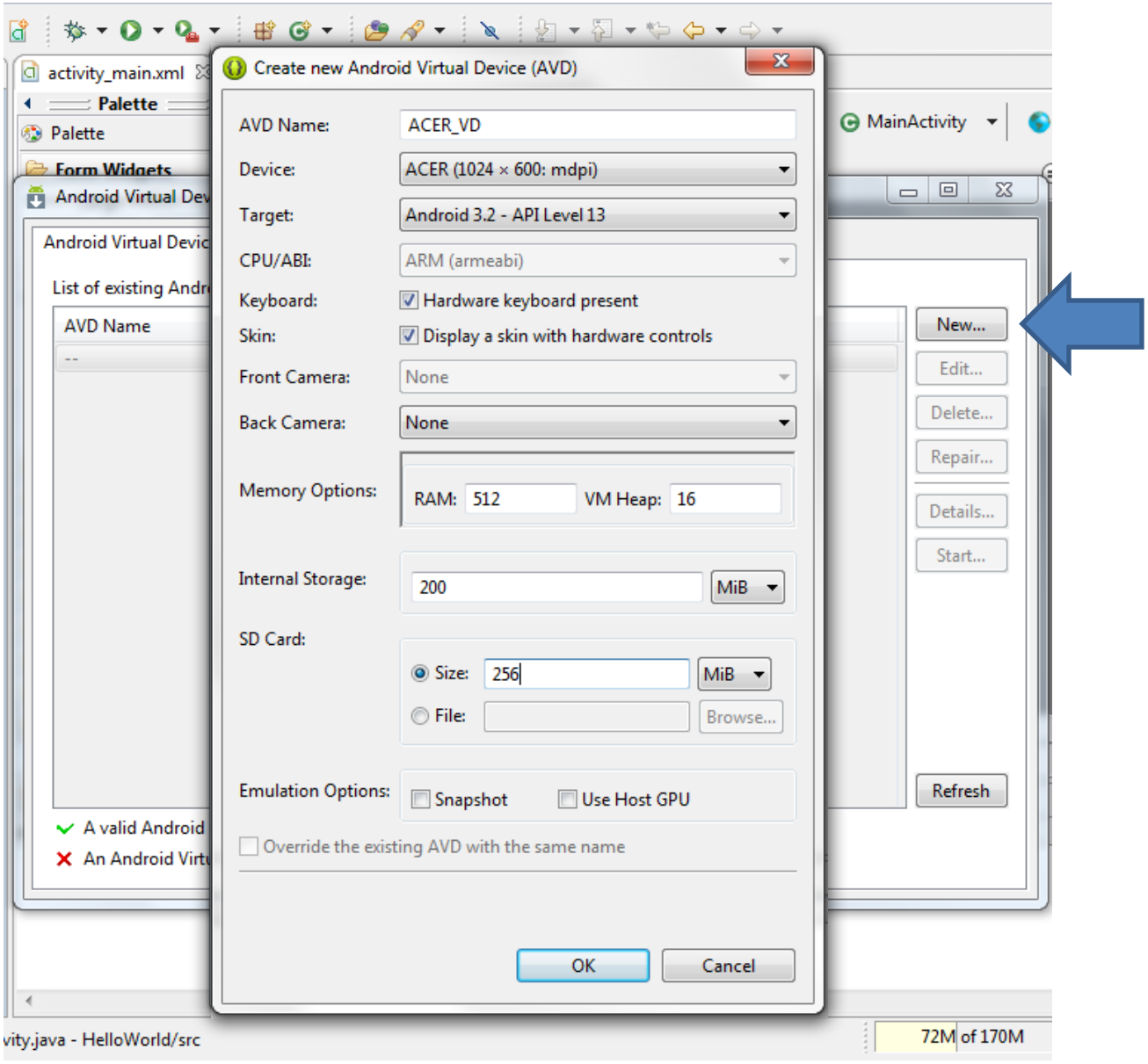

2



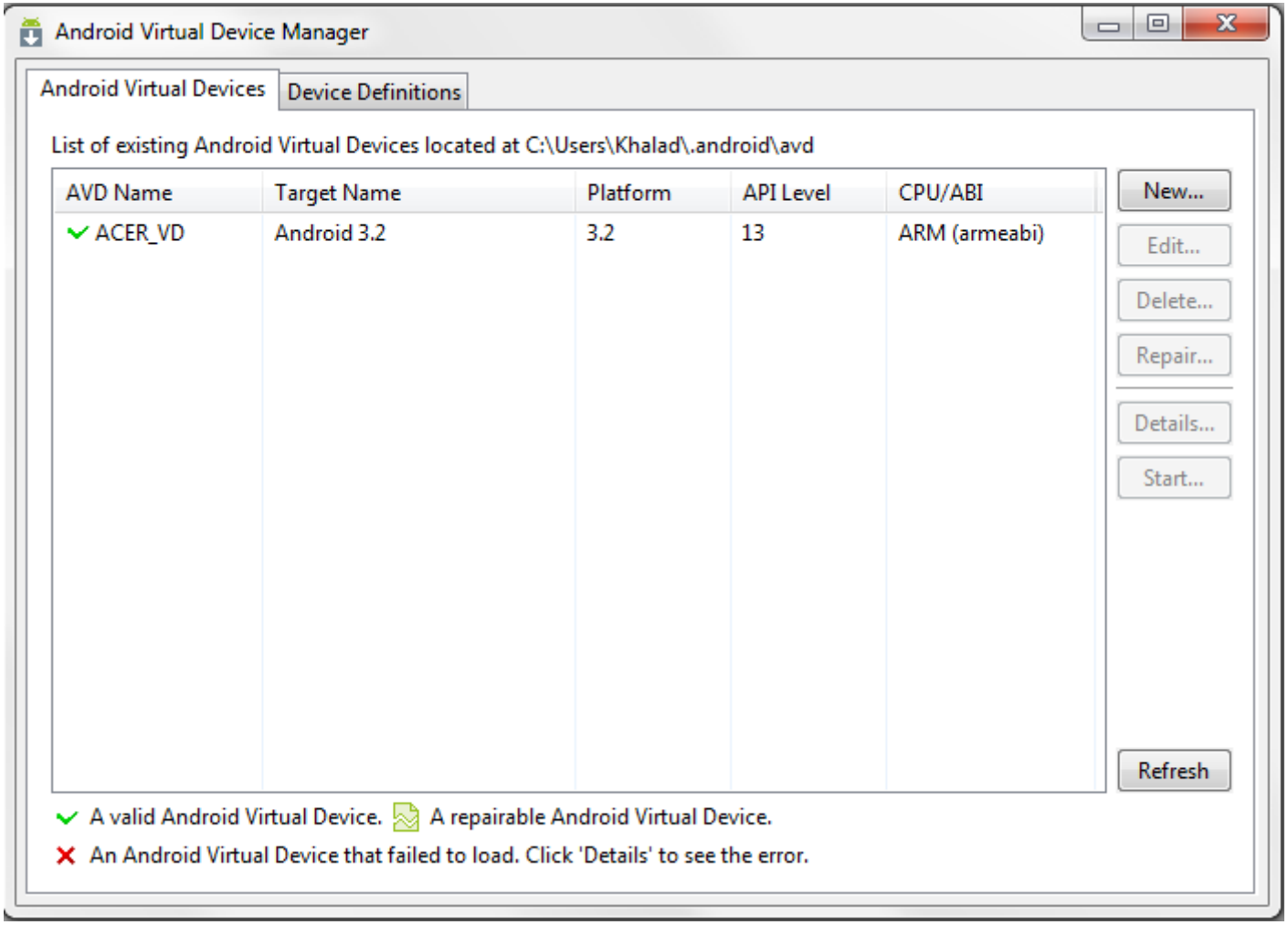
# Android Virtual Device Manager



# Android Virtual Device Manager



# Android Virtual Device Manager



# Enabling USB Debugging on device

Utilized to copy data between your device and computer.

For Android 3.2 or below:
Settings > Applications > Development > USB debugging

For Others:
Settings > Developer Options > USB debugging.

# Installation Requirements

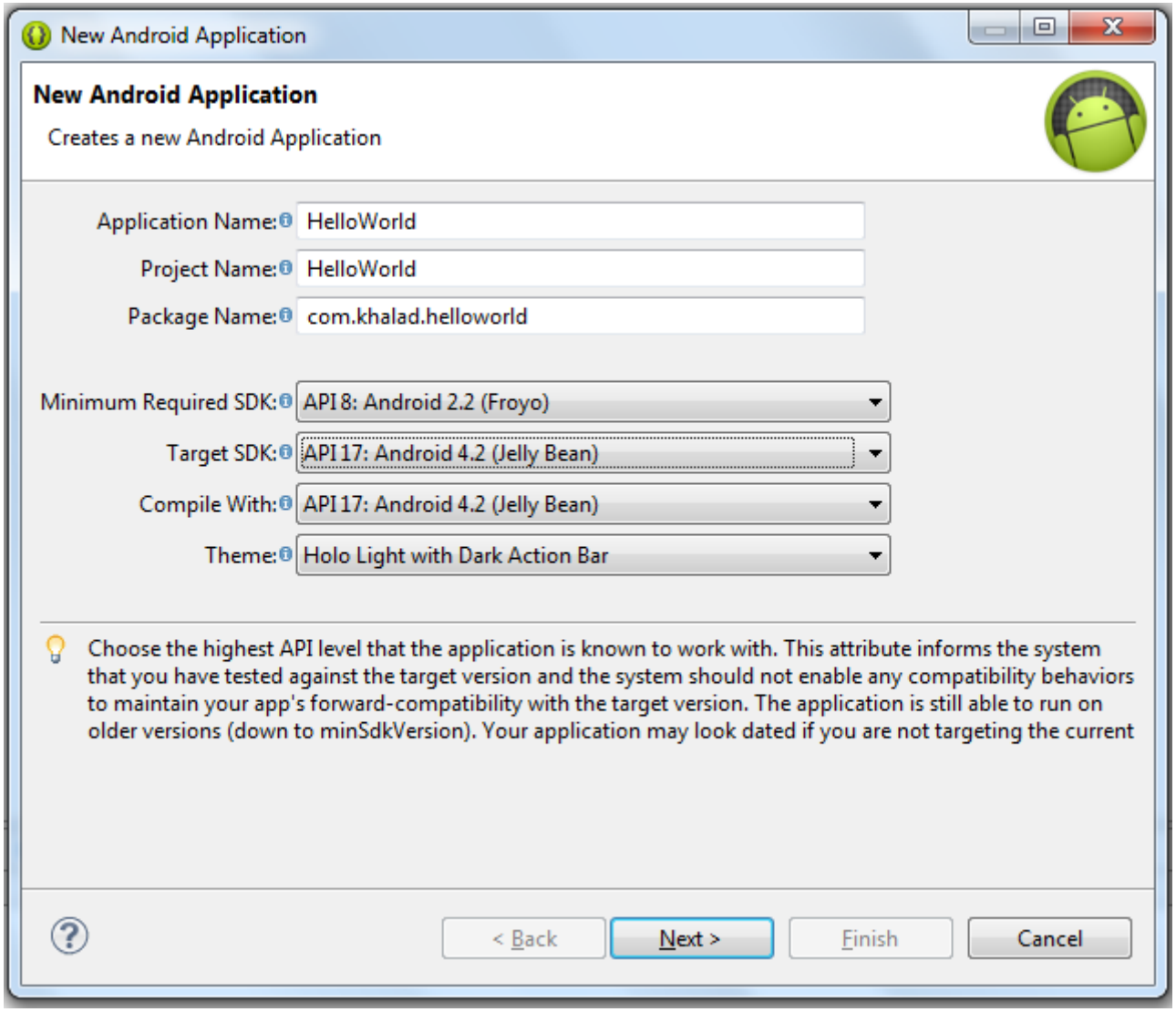
① Java Runtime Environment (JRE) √

② Java Development Kit (JDK) √

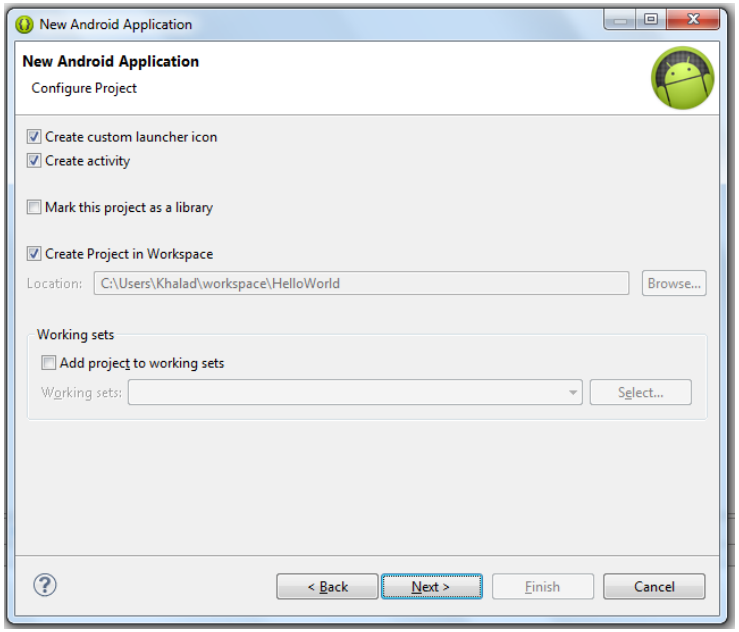
③ Android Developer Tools Bundle (ADT Bundle) √

- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

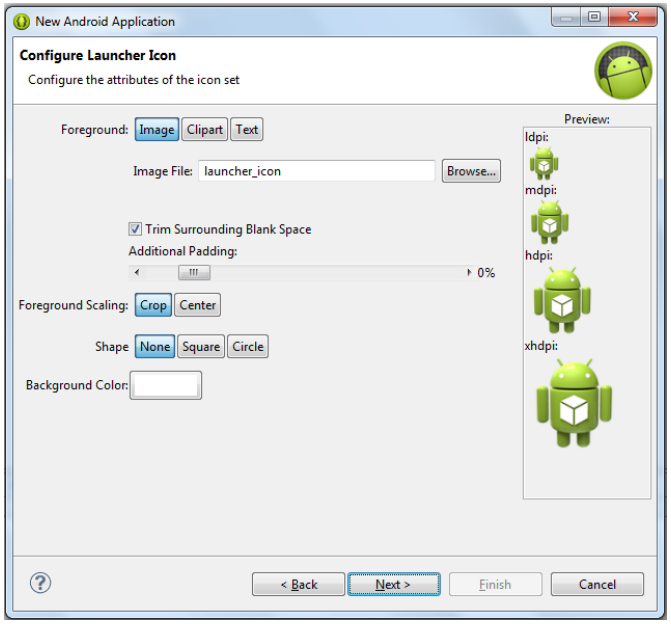
# File -> New -> Android Application Project



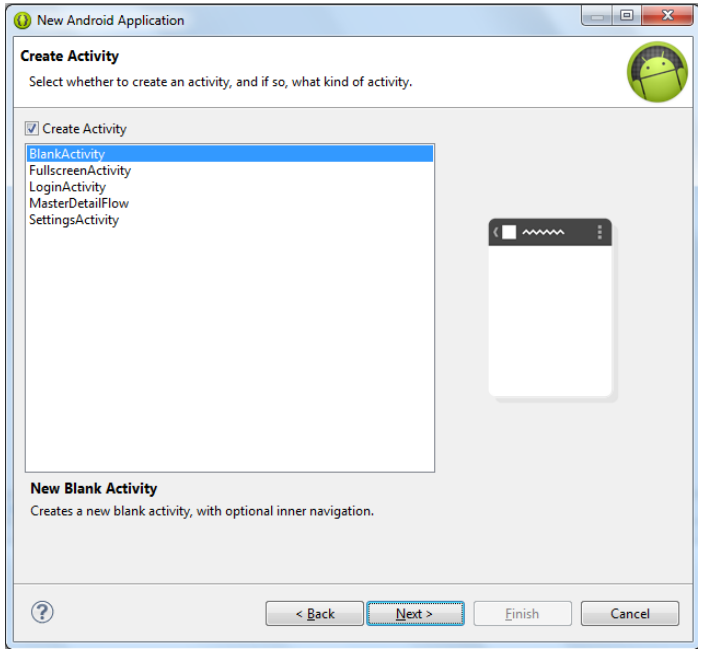
1


New Android Application
New Android Application
Configure Project
Create custom launcher icon
Create activity
Mark this project as a library
Create Project in Workspace
Location: C:\Users\Khalad\workspace\HelloWorld
Browse...
Working sets
Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

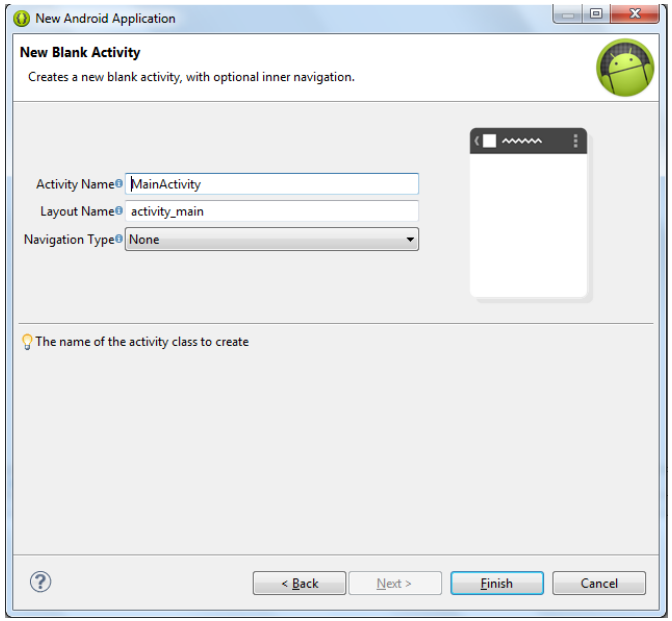
2

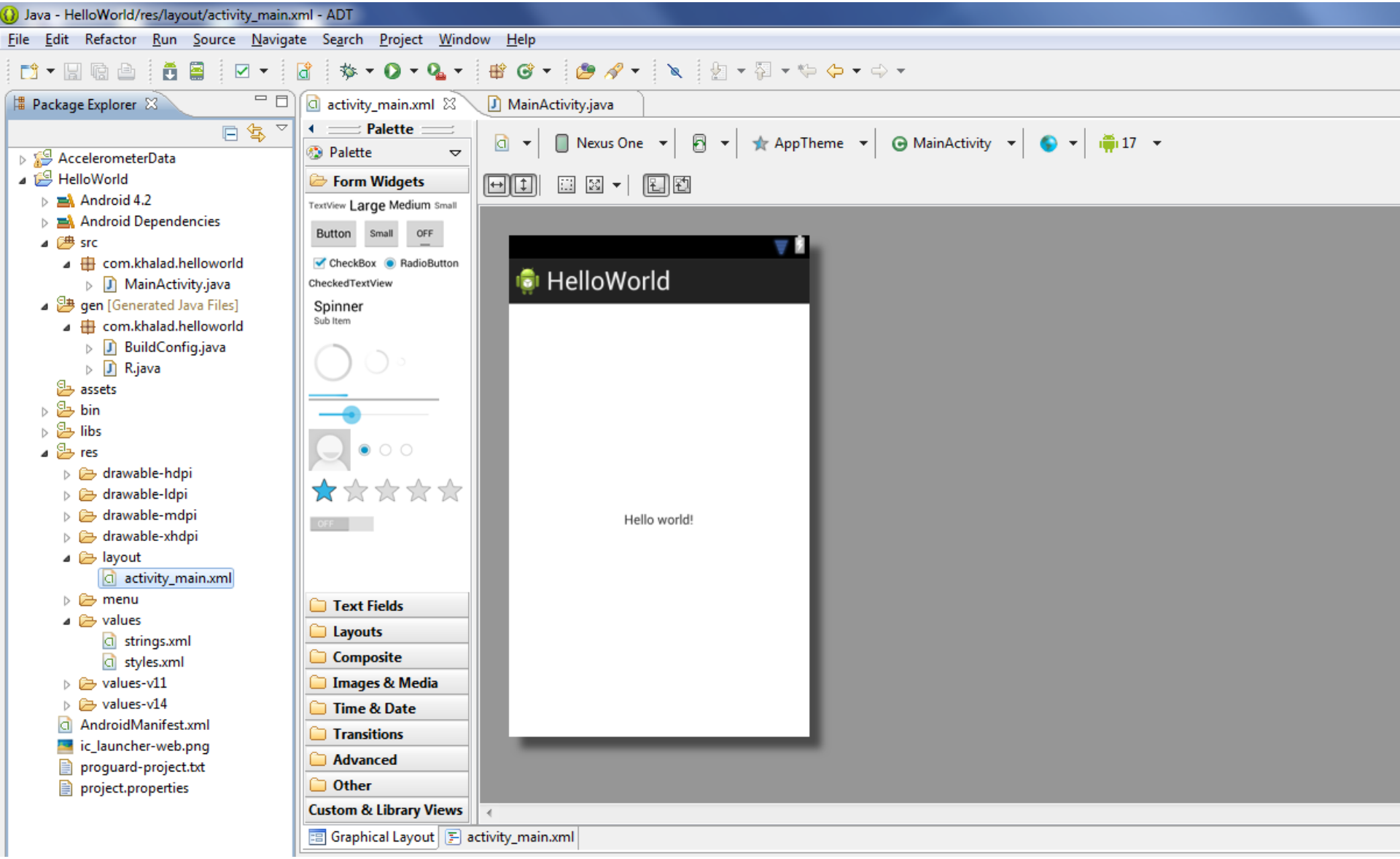

New Android Application
Configure Launcher Icon
Configure the attributes of the icon set
Foreground: Image Clipart Text
Image File: launcher_icon
Browse...
Trim Surrounding Blank Space
Additional Padding:
0%
Foreground Scaling: Crop Center
Shape None Square Circle
Background Color:
Preview:
ldpi:
mdpi:
hdpi:
xhdpi:
< Back
Next >
Finish
Cancel

3

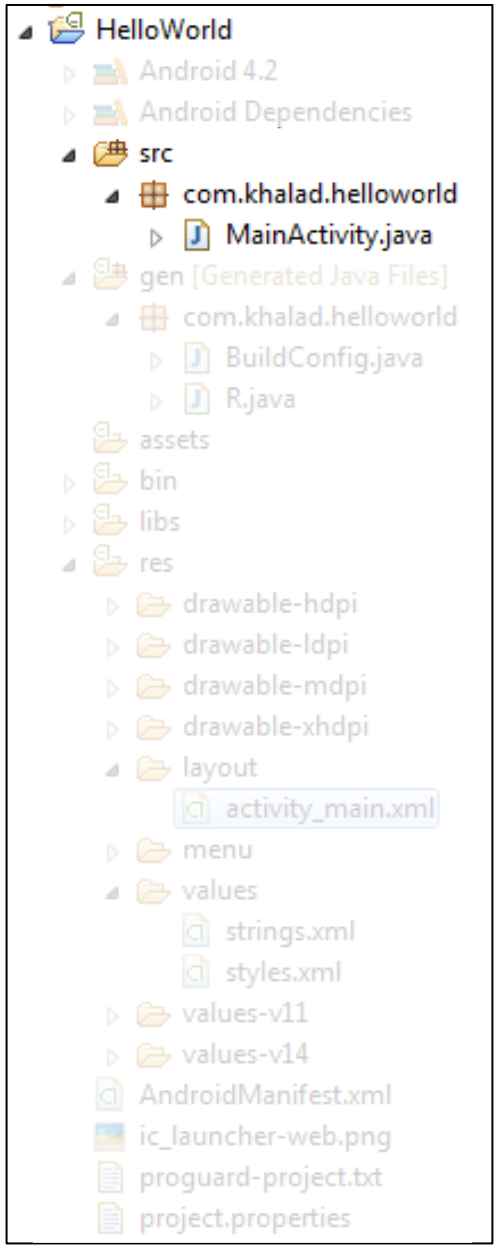

New Android Application
Create Activity
Select whether to create an activity, and if so, what kind of activity.
Create Activity
BlankActivity
FullscreenActivity
LoginActivity
MasterDetailFlow
SettingsActivity
New Blank Activity
Creates a new blank activity, with optional inner navigation.
< Back
Next >
Finish
Cancel

4


New Android Application
New Blank Activity
Creates a new blank activity, with optional inner navigation.
Activity Name: MainActivity
Layout Name: activity_main
Navigation Type: None
The name of the activity class to create
< Back
Next >
Finish
Cancel


Java - HelloWorld/res/layout/activity_main.xml - ADT
File Edit Refactor Run Source Navigate Search Project Window Help
Package Explorer
AccelerometerData
HelloWorld
Android 4.2
Android Dependencies
src
com.khalad.helloworld
MainActivity.java
gen [Generated Java Files]
com.khalad.helloworld
BuildConfig.java
R.java
assets
bin
libs
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
activity_main.xml
menu
values
strings.xml
styles.xml
values-v11
values-v14
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties
activity_main.xml
MainActivity.java
Palette
Form Widgets
Text Fields
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Other
Custom & Library Views
Nexus One
AppTheme
MainActivity
17
HelloWorld
Hello world!
Graphical Layout
activity_main.xml

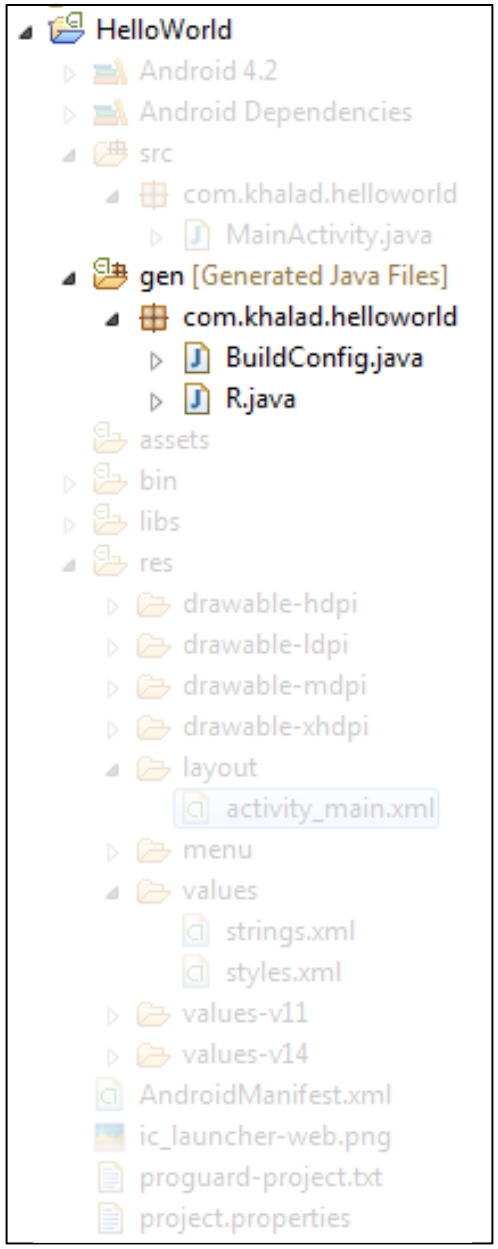
# Folder Structure For Android Project




**Src Folder** (Source folder) – contains the java code of the application.

```
package com.khalad.helloworld;

import android.os.Bundle;

public class MainActivity extends Activity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @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;
    }

}
```

# Folder Structure For Android Project



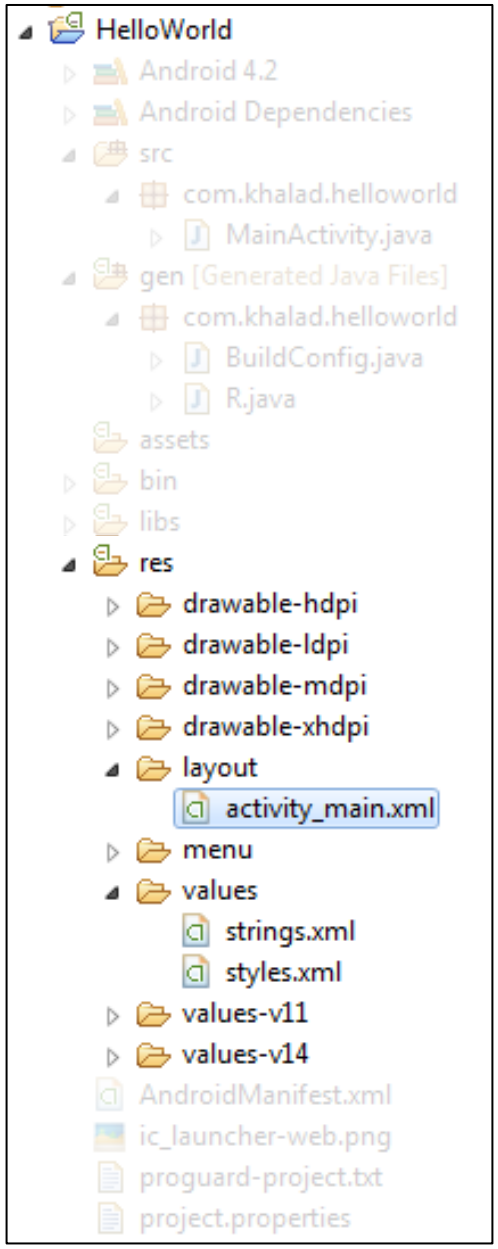

**gen Folder**: contains java files generated by ADT

These files have references to various resources placed in the application.

It contains a special class 'R' which contains all these references.

```
/* AUTO-GENERATED FILE.  DO NOT MODIFY.

package com.khalad.helloworld;

public final class R {
    public static final class attr {
    }
    public static final class drawable {
        public static final int ic_launcher=0x7f020000;
    }
    public static final class id {
        public static final int menu_settings=0x7f070000;
    }
    public static final class layout {
        public static final int activity_main=0x7f030000;
    }
    public static final class menu {
        public static final int activity_main=0x7f060000;
    }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int hello_world=0x7f040001;
        public static final int menu_settings=0x7f040002;
    }
```

# Folder Structure For Android Project



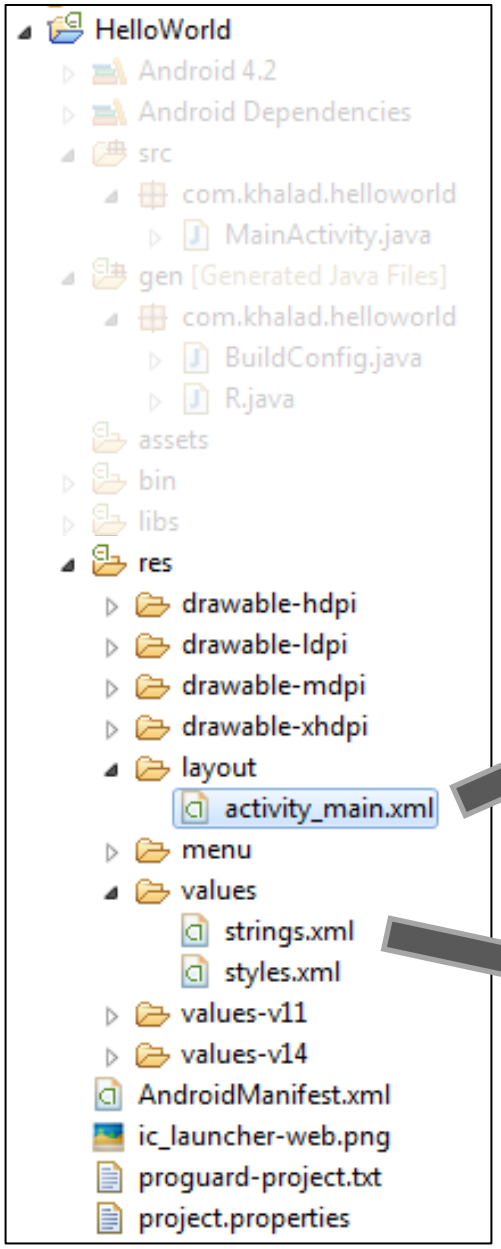

**res Folder** (Resource folder): Contains application resources, such as drawable files, layout files, and string values

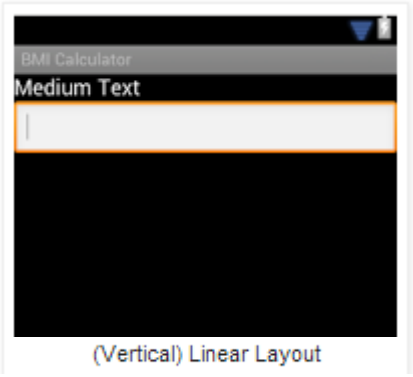
*drawable* - For bitmap files (PNG, JPEG, or GIF), image files, and XML files that describe Drawable shapes

*layout* - XML files that are compiled into screen layout

*values* - XML files that contain simple values, such as strings, integers, and colors

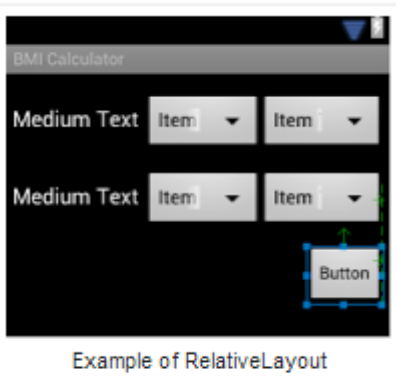
# Common Layouts




(Vertical) Linear Layout

Example of TableLayout (Portrait View)




Example of RelativeLayout

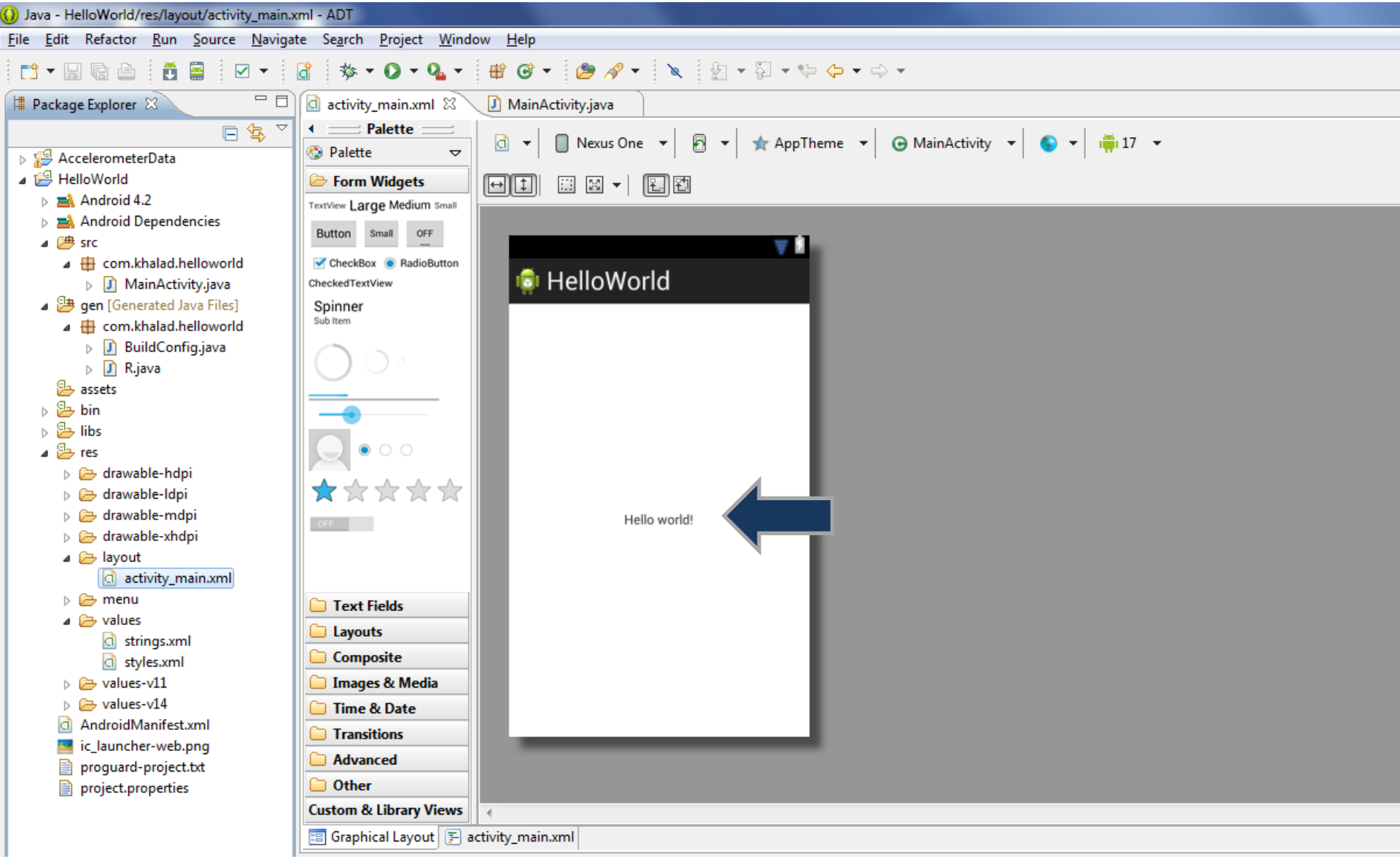
activity_main.xml

```
<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:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>
```

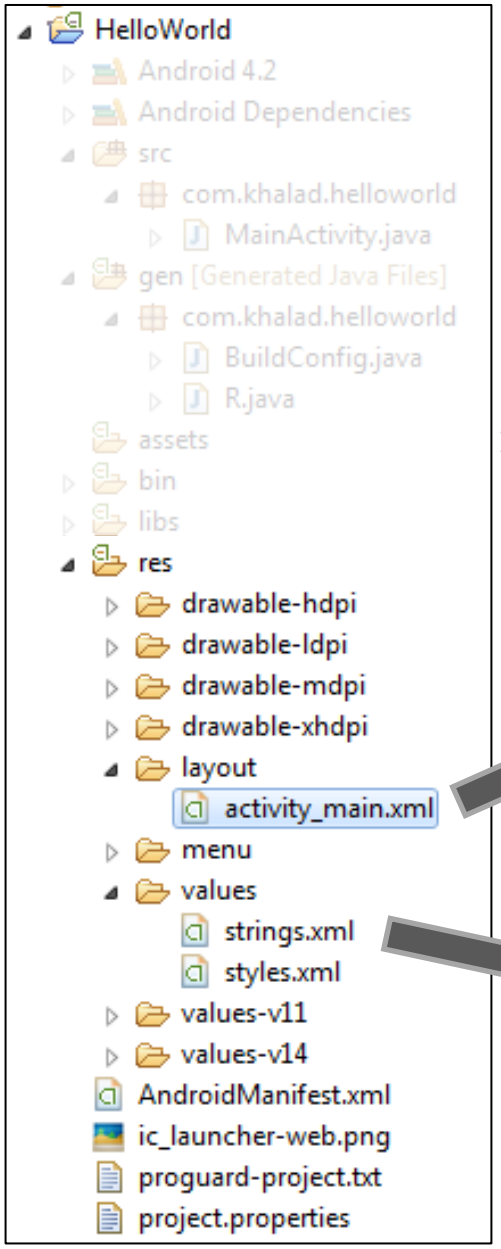
strings.xml

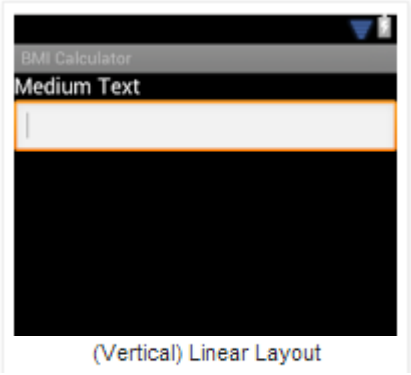
```
<?xml version="1.0" encoding="utf-8"?>
<resources>

    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

Image Source: http://android-meda.blogspot.ca/2012/02/tutorial-using-layouts-in-your-android.html

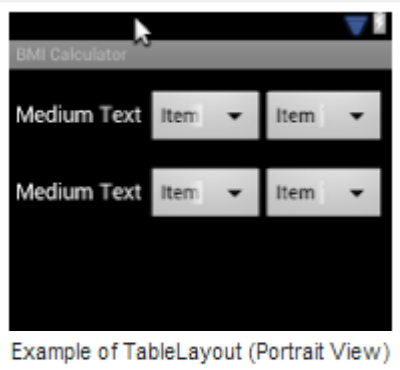

Java - HelloWorld/res/layout/activity_main.xml - ADT
File Edit Refactor Run Source Navigate Search Project Window Help
Package Explorer
AccelerometerData
HelloWorld
Android 4.2
Android Dependencies
src
com.khalad.helloworld
MainActivity.java
gen [Generated Java Files]
com.khalad.helloworld
BuildConfig.java
R.java
assets
bin
libs
res
drawable-hdpi
drawable-ldpi
drawable-mdpi
drawable-xhdpi
layout
activity_main.xml
menu
values
strings.xml
styles.xml
values-v11
values-v14
AndroidManifest.xml
ic_launcher-web.png
proguard-project.txt
project.properties
activity_main.xml
MainActivity.java
Palette
Form Widgets
Text Fields
Layouts
Composite
Images & Media
Time & Date
Transitions
Advanced
Other
Custom & Library Views
Nexus One
AppTheme
MainActivity
17
HelloWorld
Hello world!
Graphical Layout
activity_main.xml

# Common Layouts

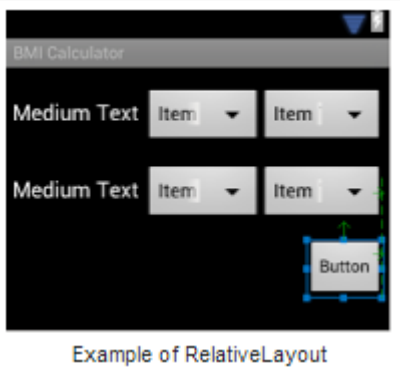



(Vertical) Linear Layout




Example of TableLayout (Portrait View)




Example of RelativeLayout

Layouts: http://developer.android.com/guide/topics/ui/declaring-layout.html#CommonLayouts

activity_main.xml

```
<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:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />
</RelativeLayout>
```

strings.xml

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

    <string name="app_name">HelloWorld</string>
    <string name="action_settings">Settings</string>
    <string name="hello_world">Hello world!</string>

</resources>
```

Image Source: http://android-meda.blogspot.ca/2012/02/tutorial-using-layouts-in-your-android.html

# activity_main.xml file

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="vertical" >

    <EditText
        android:id="@+id/editText1" android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Username" >
    </EditText>

    <EditText
        android:id="@+id/editText2"
        android:layout_width="match_parent" android:layout_height="wrap_content"
        android:ems="10"
        android:hint="Password"
    android:inputType="textPassword" />

    <Button
        android:id="@+id/button1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Button" />

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        >
        <TextView
            android:id="@+id/textView1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Gender"
            android:textAppearance="?android:attr/textAppearanceMedium" />

        <RadioButton
            android:id="@+id/radioButton1"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Male" />

        <RadioButton
            android:id="@+id/radioButton2"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="Female" />
    </LinearLayout>
</LinearLayout>
```

**Text field**

**Text field**

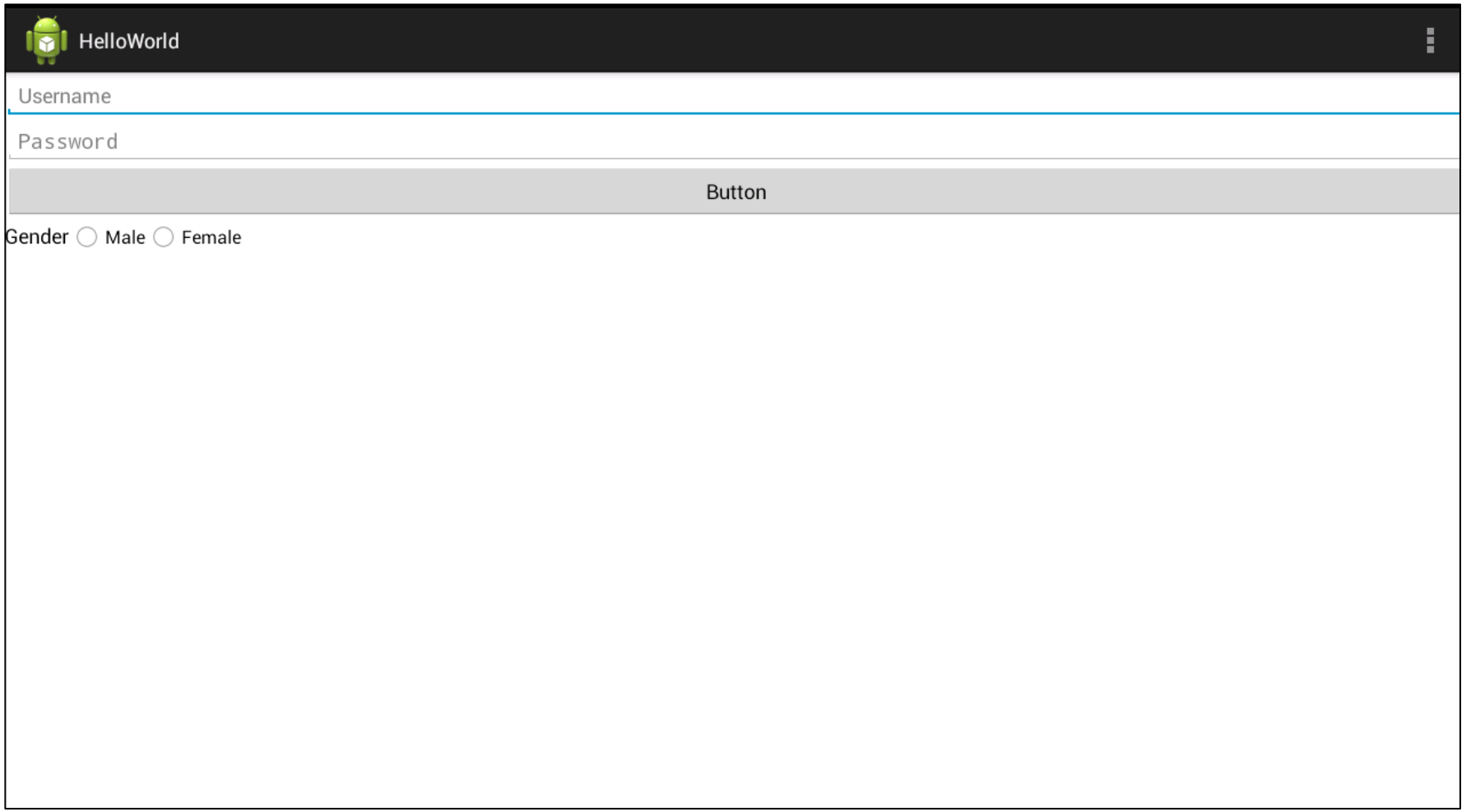
**Button**

**Text view**

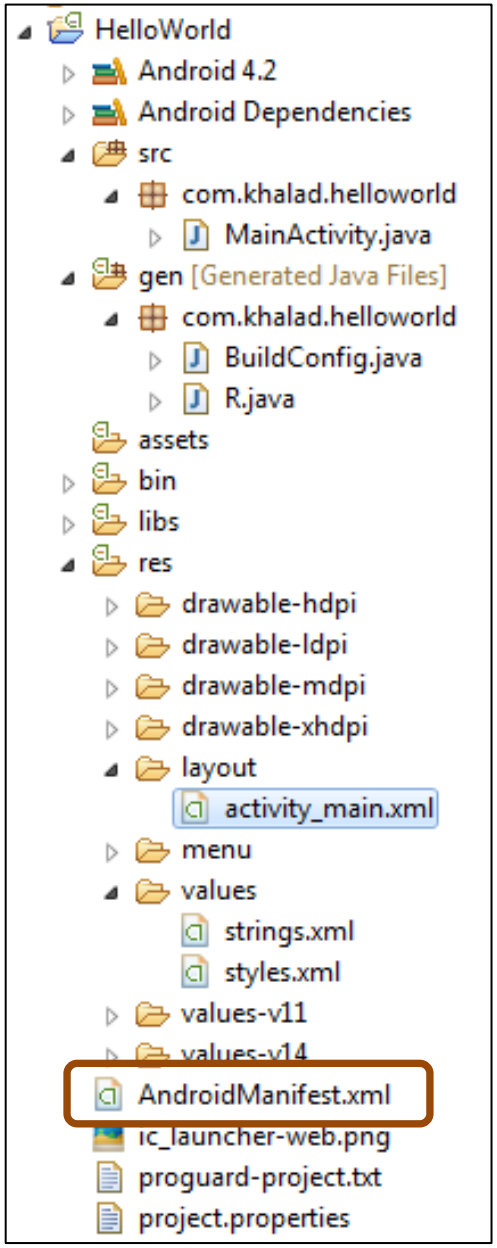
**Radio Button**

**Radio Button**

# Output



# Folder Structure For Android Project



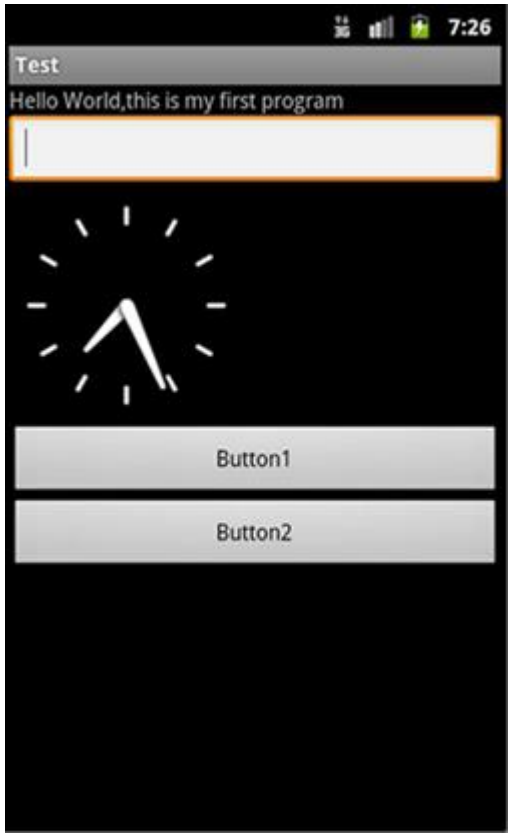

**Manifest file**:describe the application

- Declare app's name, version, icon, permission, etc…
- Declare the application's components: activity, service ,receiver or provider

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.khalad.helloworld"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.khalad.helloworld.MainActivity"
            android:label="@string/app_name" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>

</manifest>
```

# Core Components

Activity




- An *activity* is a user interface screen where visual elements (Views or widgets) can be placed
- In this example, there are five widgets(TextView, EditText, AnalogClock and two Buttons)
- An application might consist of just one activity or several

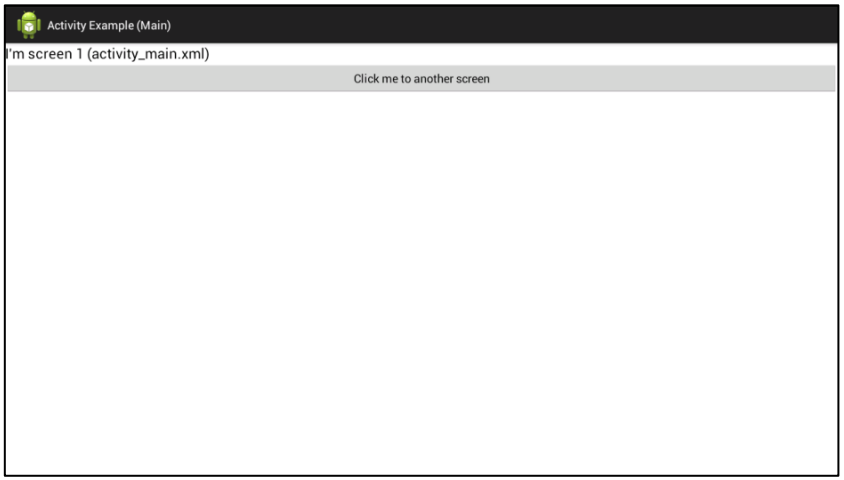
# Core Components

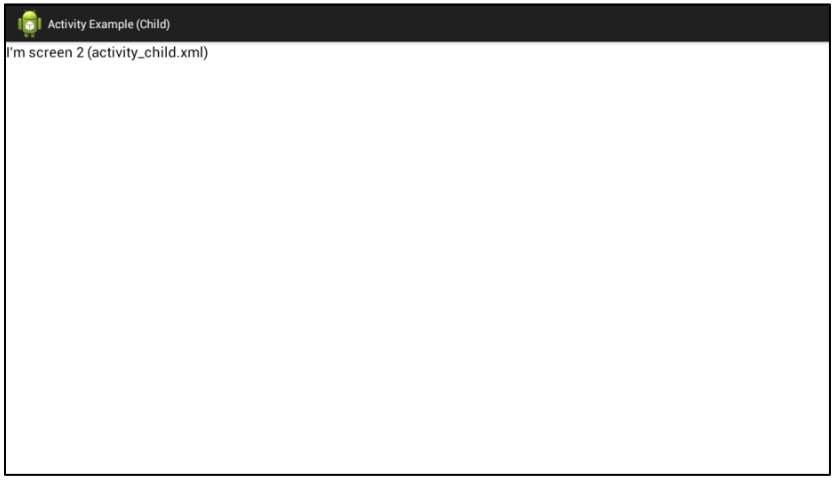
Intent, Service

- *Intent* is a mechanism to describe specific action, such a "Send an email"

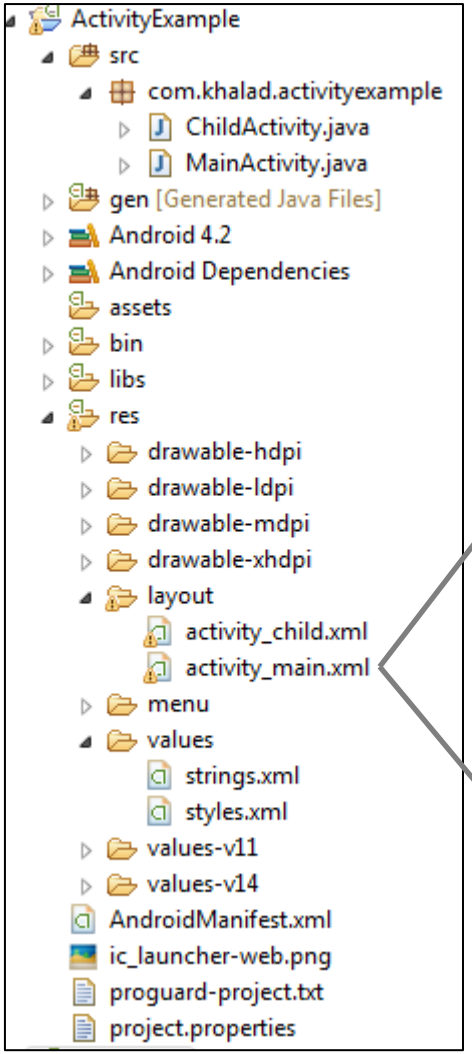
- A *service* is a task that runs in the background without the user's direct interaction

# Activity Example



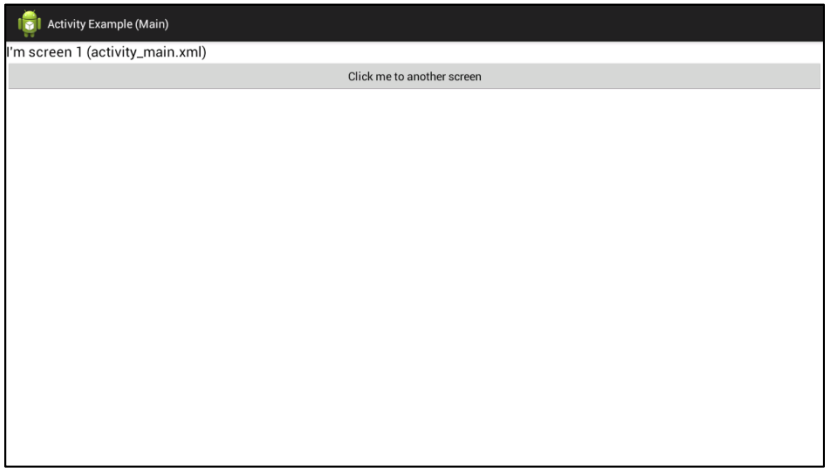
# Activity Example



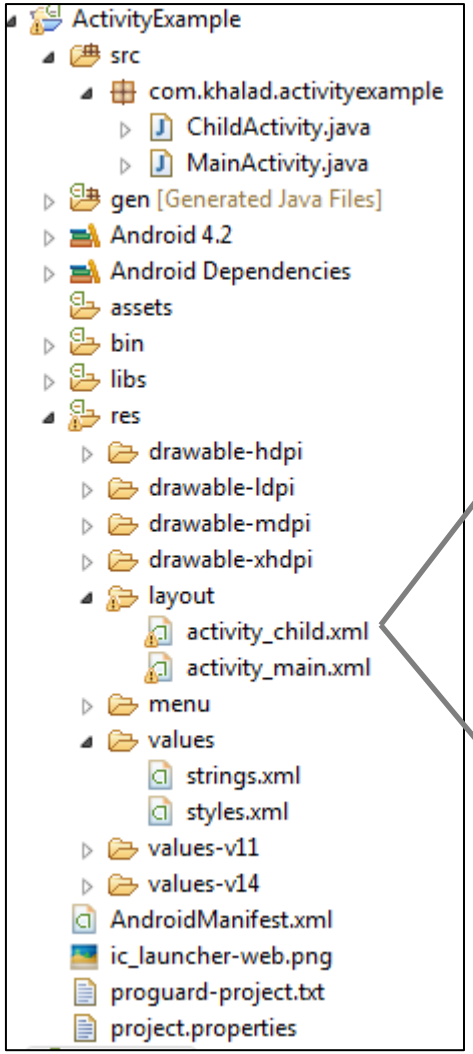

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I&apos;m screen 1 (activity_main.xml)"
        android:textAppearance="?android:attr/textAppearanceLarge" />

    <Button
        android:id="@+id/button1"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="Click me to another screen" />

</LinearLayout>
```

Text View

Button

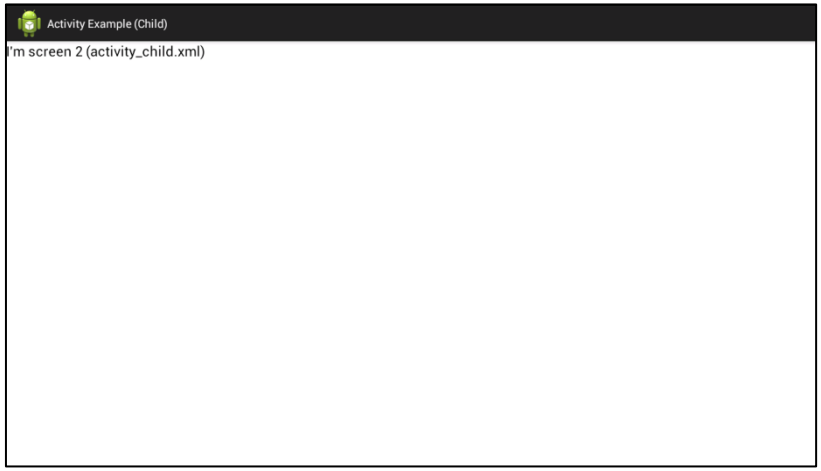


# Activity Example

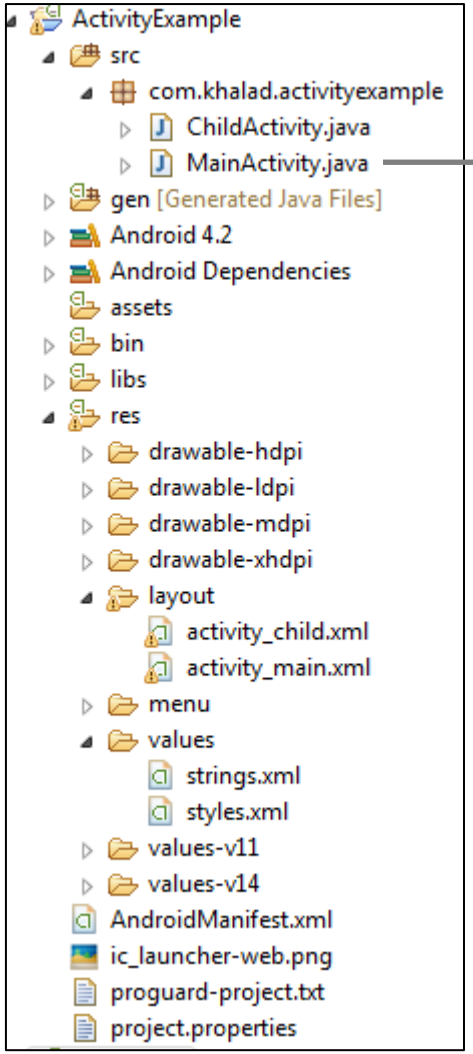



```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/linearLayout1"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="I&apos;m screen 2 (activity_child.xml)"
        android:textAppearance="?android:attr/textAppearanceLarge" />

</LinearLayout>
```

Text View



# Activity Example



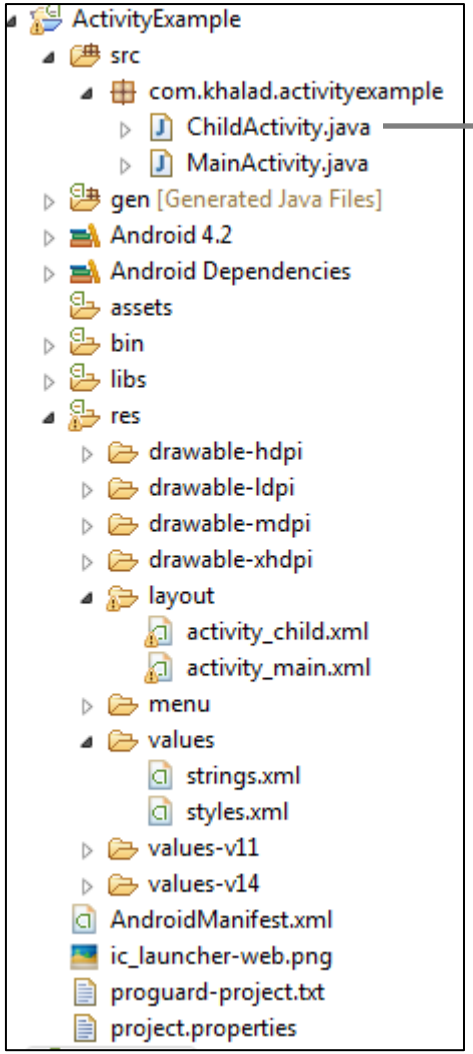

```
package com.khalad.activityexample;

import android.app.Activity;


public class MainActivity extends Activity {

    Button button;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        addListenerOnButton();
    }

    public void addListenerOnButton() {

        final Context context = this;
        button = (Button) findViewById(R.id.button1);
        button.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View arg0) {

                Intent intent = new Intent(context, ChildActivity.class);
                startActivity(intent);
            }
        });
    }
}
```

**Inherit from the activity class**

**Set layout as describe in *activity_main.xml***

**Find "button1" and set actions**

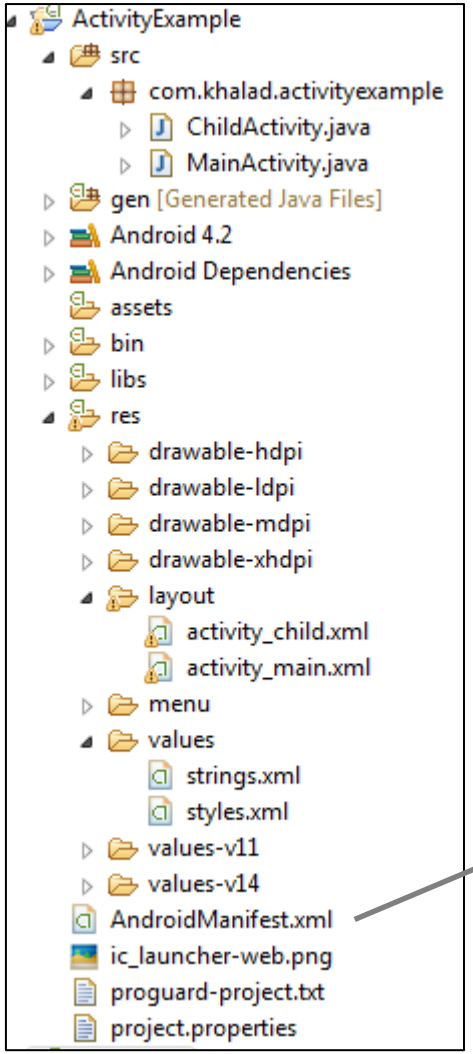
# Activity Example




```java
package com.khalad.activityexample;

import android.app.Activity;

public class ChildActivity extends Activity {

    Button button;

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_child);
    }

}
```

**Set layout as describe in *activity_child.xml***

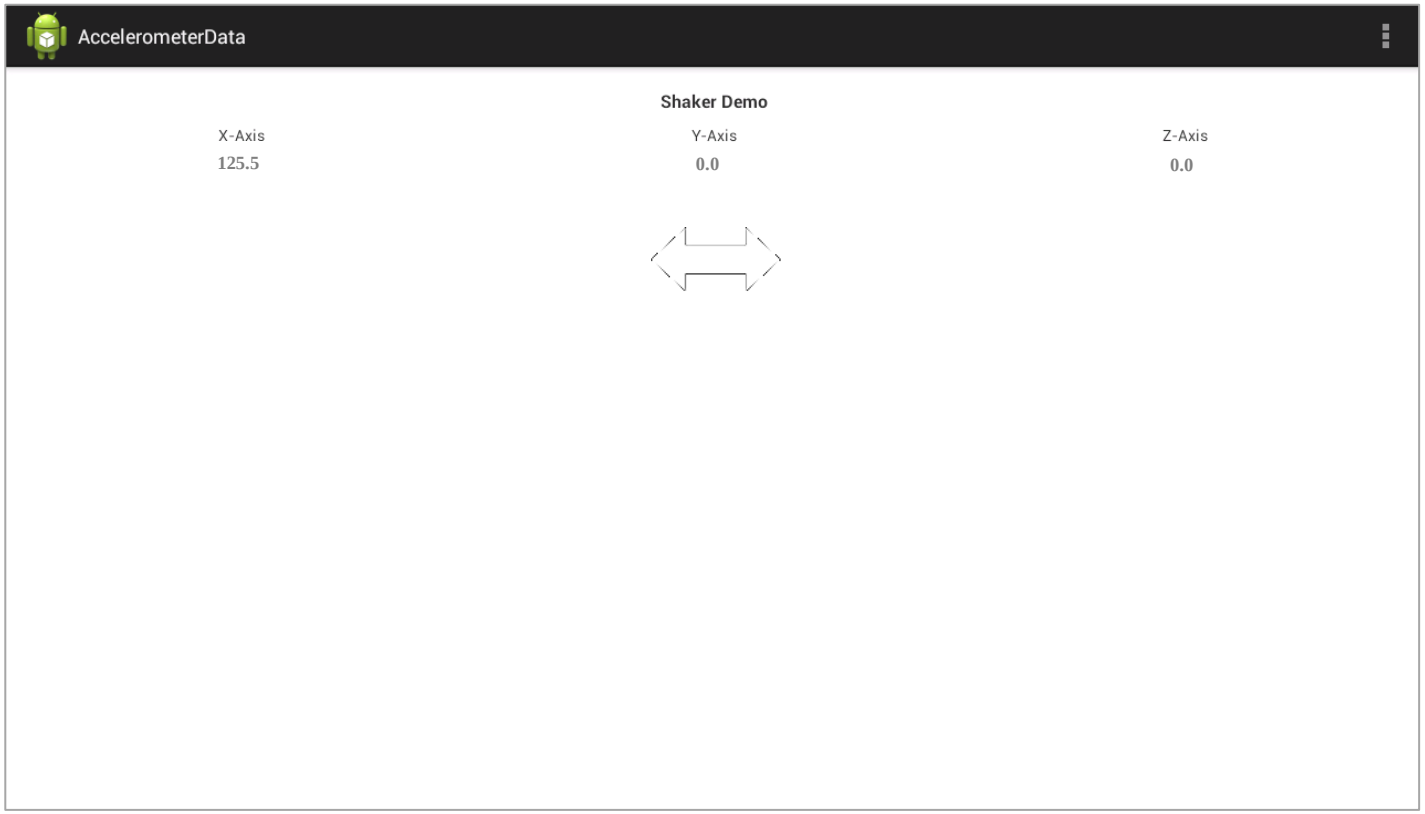
# Activity Example




```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.khalad.activityexample"
    android:versionCode="1"
    android:versionName="1.0" >

    <uses-sdk
        android:minSdkVersion="8"
        android:targetSdkVersion="17" />

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name1"
        android:theme="@style/AppTheme" >
        <activity
            android:name="com.khalad.activityexample.MainActivity"
            android:label="@string/app_name1" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>

        <activity
            android:label="@string/app_name2"
            android:name="com.khalad.activityexample.ChildActivity" >
        </activity>

    </application>

</manifest>
```

# Accelerometer Example

An accelerometer is defined as an instrument for measuring the time rate of change of velocity with respect to magnitude or direction.




**7 Text View, 1 Image View**

# Step 1-> Layout

```
<LinearLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:gravity="center"
        android:paddingTop="20dip"
        android:text="Shaker Demo"
        android:textSize="16sp"
        android:textStyle="bold" />
    <TableLayout
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:paddingTop="10dip"
        android:stretchColumns="*" >
        <TableRow>
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="X-Axis"
                android:textSize="14sp" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Y-Axis"
                android:textSize="14sp" />
            <TextView
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center"
                android:text="Z-Axis"
                android:textSize="14sp" />
        </TableRow>
        <TableRow>
            <TextView
                android:id="@+id/x_axis"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center" />
            <TextView
                android:id="@+id/y_axis"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center" />
            <TextView
                android:id="@+id/z_axis"
                android:layout_width="wrap_content"
                android:layout_height="wrap_content"
                android:gravity="center" />
        </TableRow>
    </TableLayout>
    <ImageView
        android:id="@+id/image"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:layout_gravity="center"
        android:paddingTop="15dip"
        android:visibility="invisible" />
</LinearLayout>
```

# Step 2 -> Java Main File

```
public class MainActivity extends Activity implements SensorEventListener {

    private float mLastX, mLastY, mLastZ;
    private boolean mInitialized;
    private SensorManager mSensorManager;
    private Sensor mAccelerometer;
    private final float NOISE = (float) 2.0;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        mInitialized = false;
        mSensorManager = (SensorManager) getSystemService(Context.SENSOR_SERVICE);
        mAccelerometer = mSensorManager.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
        mSensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
    }
```

SensorManager → access the device's sensor

Get an instance of this class by calling getSystemService()

# Step 2 -> Java Main File

```
protected void onResume() {
    super.onResume();
    mSensorManager.registerListener(this, mAccelerometer, SensorManager.SENSOR_DELAY_NORMAL);
}

protected void onPause() {
    super.onPause();
    mSensorManager.unregisterListener(this);
}
```

**onResume()**

**onPaues()**

```
    public void onSensorChanged(SensorEvent event)
    {
        TextView tvX= (TextView)findViewById(R.id.x_axis);
        TextView tvY= (TextView)findViewById(R.id.y_axis);
        TextView tvZ= (TextView)findViewById(R.id.z_axis);
        ImageView iv = (ImageView)findViewById(R.id.image);
        float x = event.values[0];
        float y = event.values[1];
        float z = event.values[2];

        if (!mInitialized) {
            mLastX = x;
            mLastY = y;
            mLastZ = z;
            tvX.setText("0.0");
            tvY.setText("0.0");
            tvZ.setText("0.0");
            mInitialized = true;
        } else {
            float deltaX = Math.abs(mLastX - x);
            float deltaY = Math.abs(mLastY - y);
            float deltaZ = Math.abs(mLastZ - z);
            if (deltaX < NOISE) deltaX = (float)0.0;
            if (deltaY < NOISE) deltaY = (float)0.0;
            if (deltaZ < NOISE) deltaZ = (float)0.0;
            mLastX = x;
            mLastY = y;
            mLastZ = z;
            tvX.setText(Float.toString(deltaX));
            tvY.setText(Float.toString(deltaY));
            tvZ.setText(Float.toString(deltaZ));
            iv.setVisibility(View.VISIBLE);
            if (deltaX > deltaY) {
                iv.setImageResource(R.drawable.horizontal);
            } else if (deltaY > deltaX) {
                iv.setImageResource(R.drawable.vertical);
            } else {
                iv.setVisibility(View.INVISIBLE);
            }
        }
    }
}
```

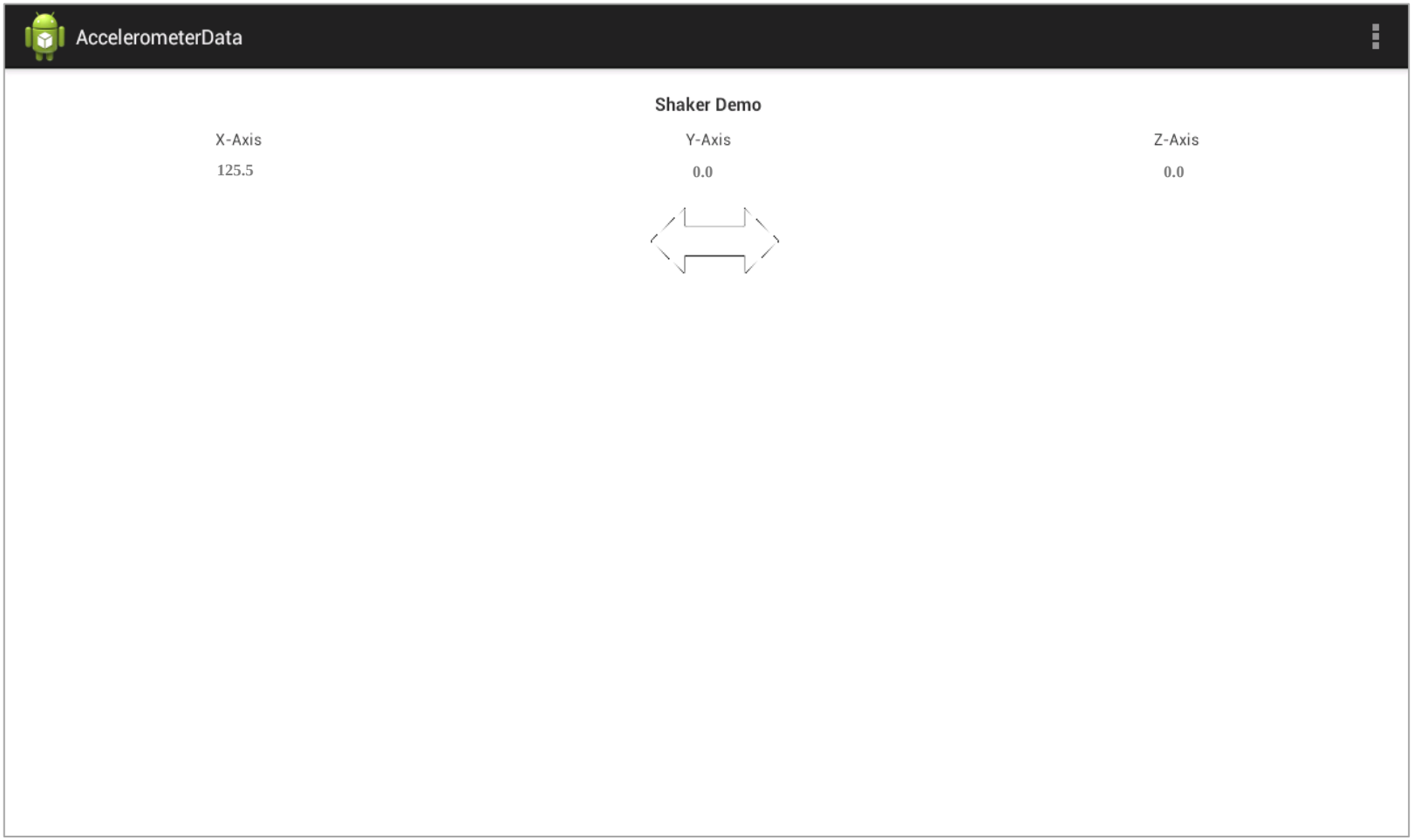
Sensor values

Difference between current and last values

Display in Text View

Display Image

# Accelerometer Example



# Permission

- Manifest update:
  - Add permission

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    //…
    android:versionName="1.0">

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

   <uses-sdk android:minSdkVersion="8" />
        //…
```

# Android Development Tutorial

Human-Computer Interaction (COMP 4020)
Winter 2014