Android 5 Lollipop Development Tools

Norman McEntire
norman.mcentire@servin.com

Opening Remarks

- **Welcome**!
- **Thank you** for attending!
- **My promise** to you
  - Provide a solid introduction to **Android 5 Lollipop Development Tools**
    - **Android Studio**

About Myself

- Norman McEntire
  - norman.mcentire@servin.com
- BS/MS Computer Engineering
  - USC - University of South Carolina
- 30+ Years Computer Engineering Experience
  - Hardware Engineering (chips, boards, systems)
  - Software Engineering (drivers, systems software, mobile apps)
  - Current Software Focus: **Android/Java,** iOS 8/ObjC/Swift, IoT, Linux

How To Take This Course

- Option 1 - **Corporate Training**
  - Contact me at norman.mcentire@servin.com to schedule this Android course at your corporation
- Option 2 - **UCSD Extension Course**
  - http://extension.ucsd.edu/studyarea/index.cfm?vAction=singleCourse&vCourse=CSE-41145&vsacategoryid=166&vStudyAreaID=14

My Assumptions About You

- You are in one of two broadly defined groups
  - Group 1. **Experienced** Android Software Developer
    - You have been using **Eclipse ADT**
  - Group 2. **New** to Android Software Development
    - No experience with Eclipse ADT
- **Common** to both groups: **Learn Android Studio**!

Agenda

- Android 5 Development **Tool Options**
- **Migrating from Eclipse ADT** to Android Studio
- Using **Android Studio**
- Using Android Studio **Tools**

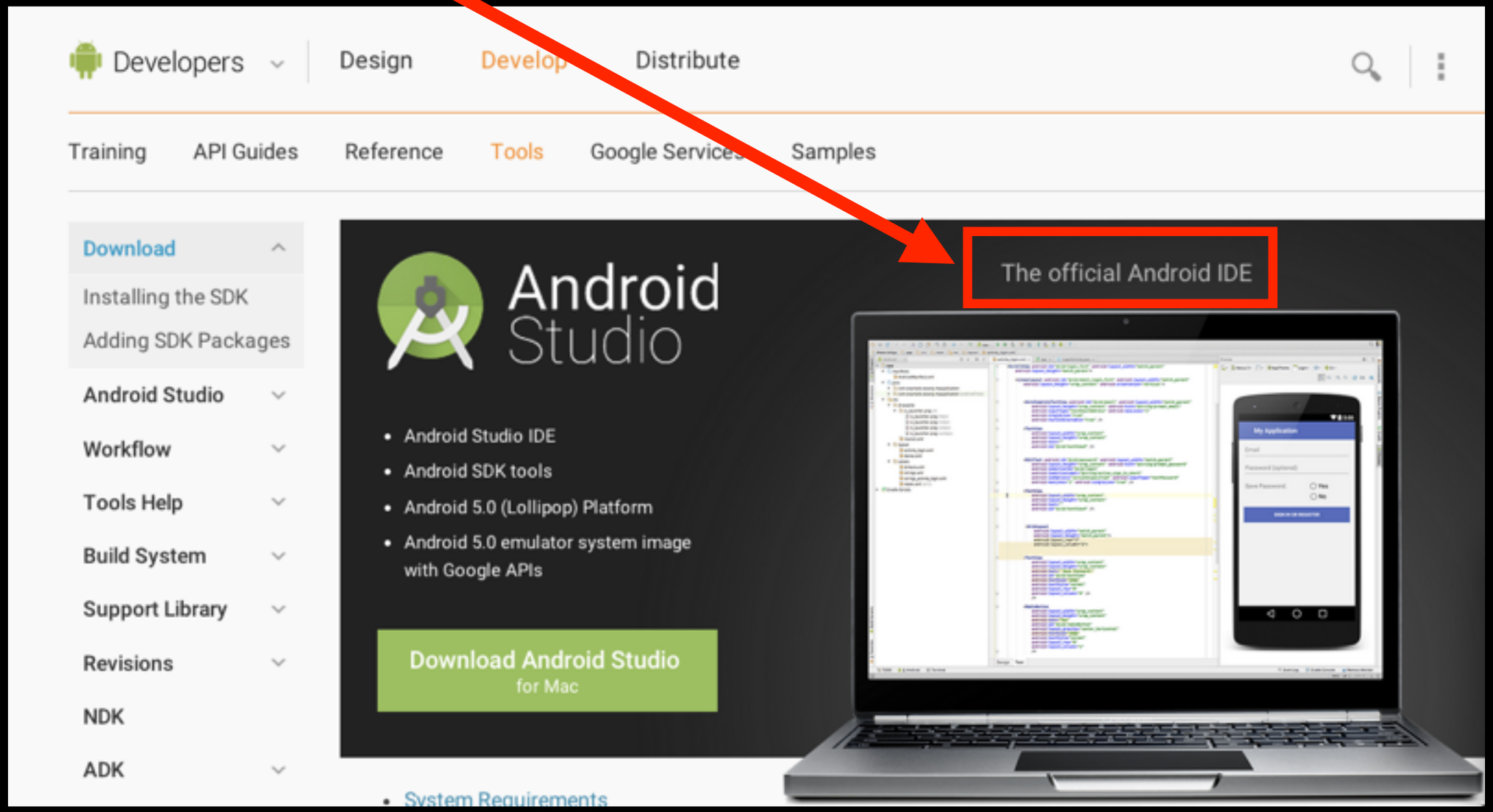
Android 5 Development

Tool Options

Android 5 Development Tools Options

- You have three major options
  - Option 1. Start Using **Android Studio**
    - The official Android 5 IDE
  - Option 2. Continue to use **Eclipse ADT** for now
  - Migrate to Android Studio when you can
  - Option 3. Use the **command-line** (no IDE)

Why Android Studio



System Requirements

- **Linux** (e.g. Ubuntu 12.04)
  - GNOME or KDE, glibc 2.11 or higher
  - **JDK 7**
- **Mac** OS X
  - 10.8.5 or higher
  - **JDK 7**
  - NOTE: Run Android Studio with **JDK 1.6** for optimized rendering, but use JDK 7 for building Android code
- **Windows**
  - Windows 7, 8
  - JDK 7

Android Studio Downloads

All Android Studio Packages

Select a specific Android Studio package for your platform. Also see the Android Studio release notes.

Platform	Package	Size	SHA-1 Checksum
Windows	android-studio-bundle-135.1641136.exe (Recommended)	868344232 bytes	9c1c8ea6aa17fb74e0593c62fd48ee62a8950be7
	android-studio-ide-135.1641136.exe (No SDK tools included)	260272840 bytes	464d1c5497ab3d1bdef441365791ab36c89cd5ae
	android-studio-ide-135.1641136-windows.zip	246249059 bytes	6d6856aca83f6ff747ca40b10f70edfbbcccd91c
Mac OS X	android-studio-ide-1641136.dmg	245729073 bytes	49506ba2cf6b56be4f7d07e6a00c4ec3ba2249d5
Linux	android-studio-ide-135.1641136-linux.zip	243917559 bytes	7c8f2d0cec21b98984cdba45ab5a25f26d67f23a

Eclipse IDE and Android Studio IDE

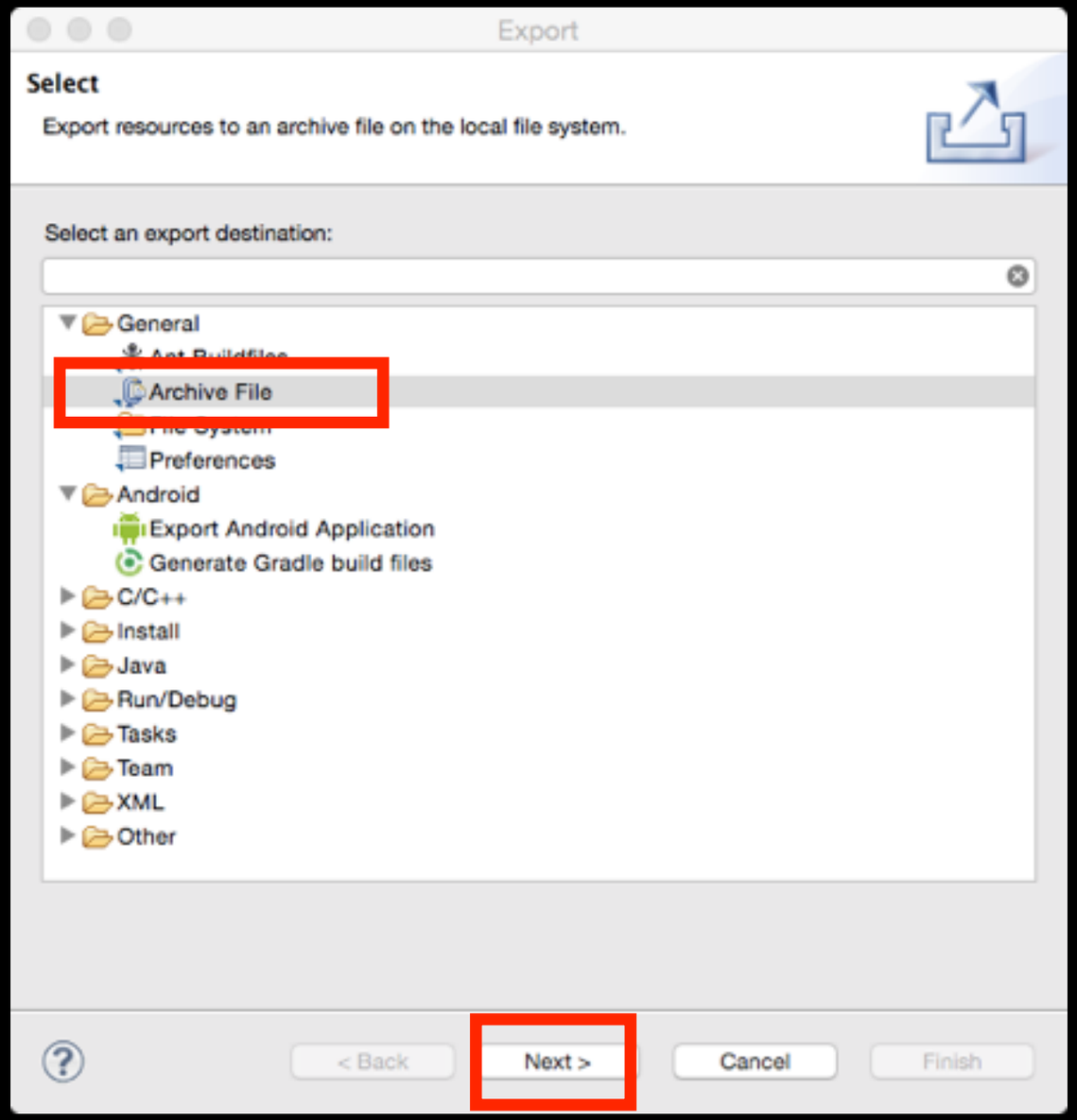
- The **Good News** About **Android Studio**
    - It is **highly integrated** with the **Android Java** build environment
        - Example 1: **GUI layout much easier** (see multiple layouts at same time)
        - Example 2: **Build Variants** (build “free" and “paid” version of app at same time)
- The **Bad News for Eclipse Users**: You may be using Eclipse for other software development projects (Android, C/C++, PHP, etc) and now you have “yet another IDE” to learn

Exporting Your Eclipse Project for Import to Android Studio

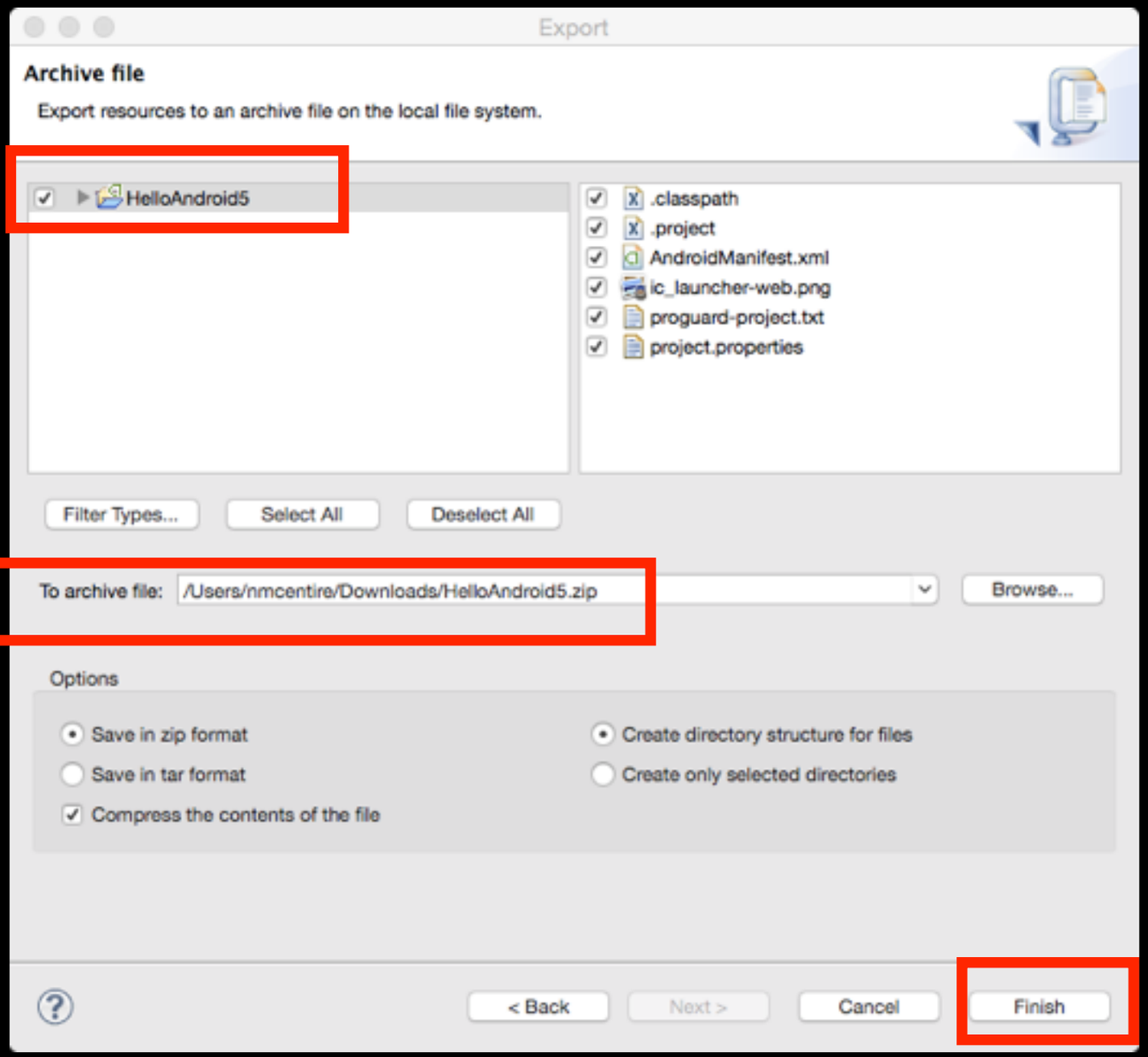
Steps To Migrate Your Eclipse Project To Android Studio

- Step 1 [In **Eclipse**]. **Export** your project
- Step 2 [In **Studio**]. Close all open projects
- Step 3 [In **Studio**]. Select **Import Non-Android Studio Project**

To Create an Archive File In Eclipse



Archive File In Eclipse



Migrating From Eclipse ADT to Android Studio

Migrating To Android Studio

Migrating to Android Studio

If you have been using Eclipse with ADT, be aware that Android Studio is now the official IDE for Android, so you should migrate to Android Studio to receive all the latest IDE updates.

Steps To Migrate Your Eclipse Project To Android Studio

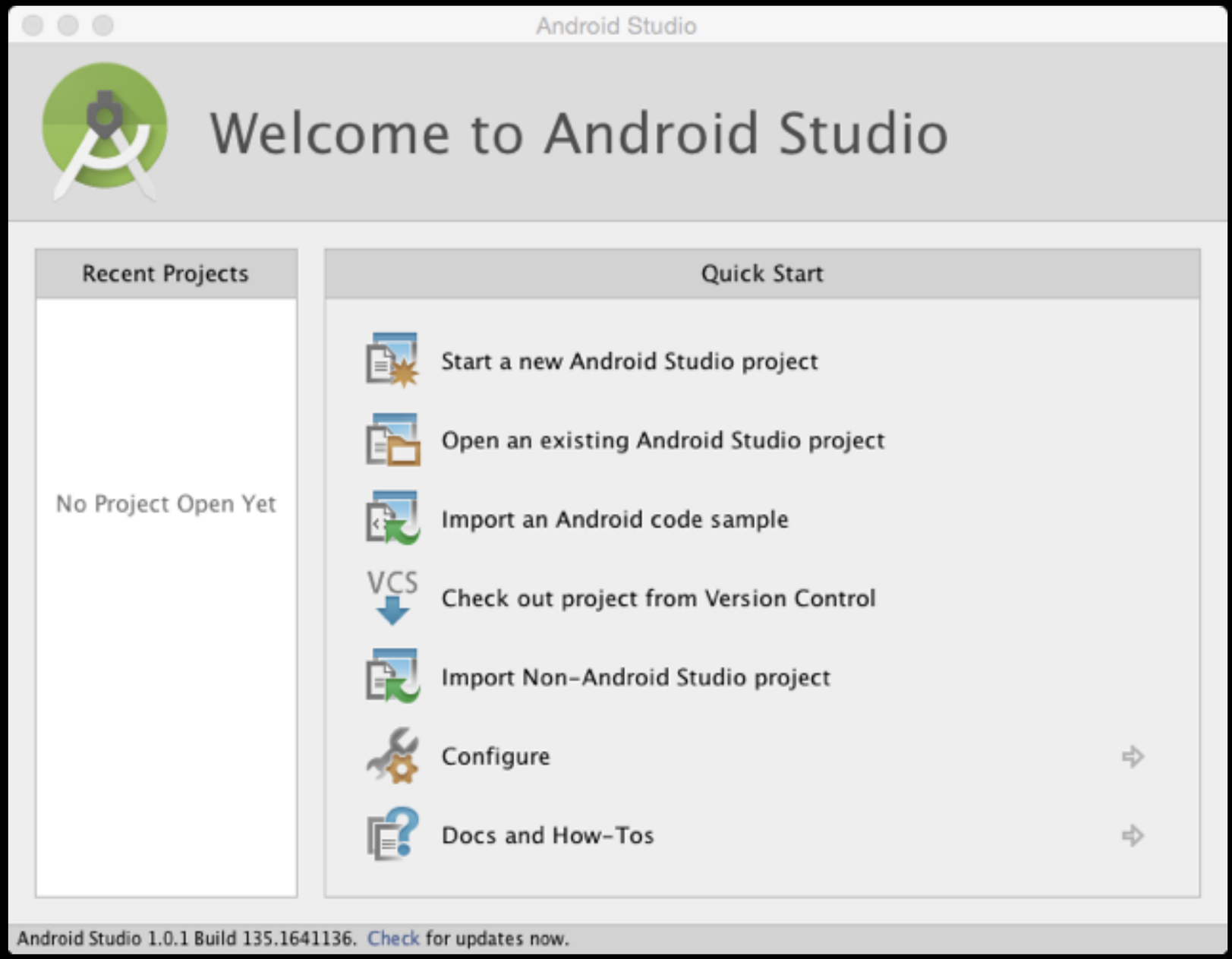
- Step 1 [In **Eclipse**]. **Export** your project
- Step 2 [In **Studio**]. Close all open projects
- Step 3 [In **Studio**]. Select **Import Non-Android Studio Project**

Demo

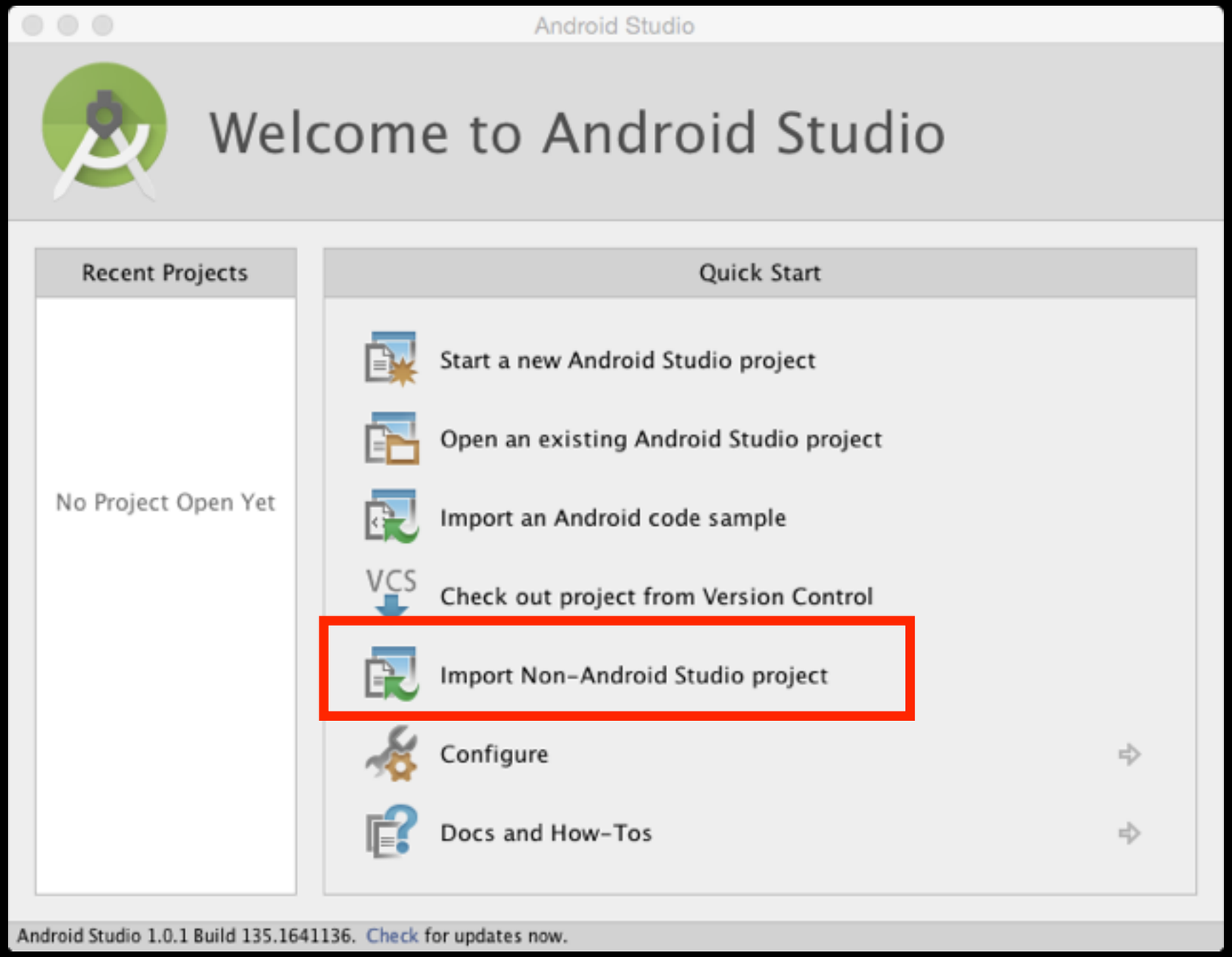
Migrating from Eclipse ADT Project to Android Studio

Android Studio 1.1 Startup

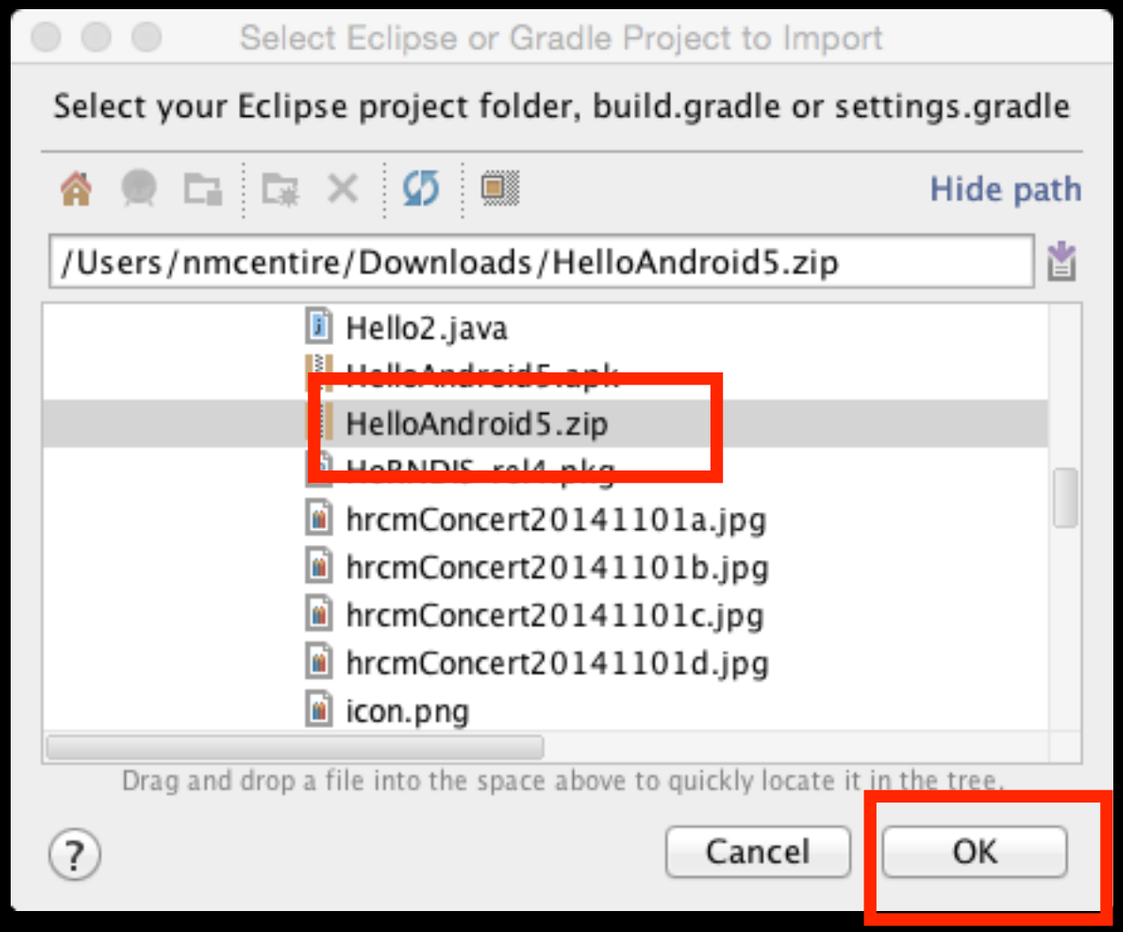
Android Studio Startup



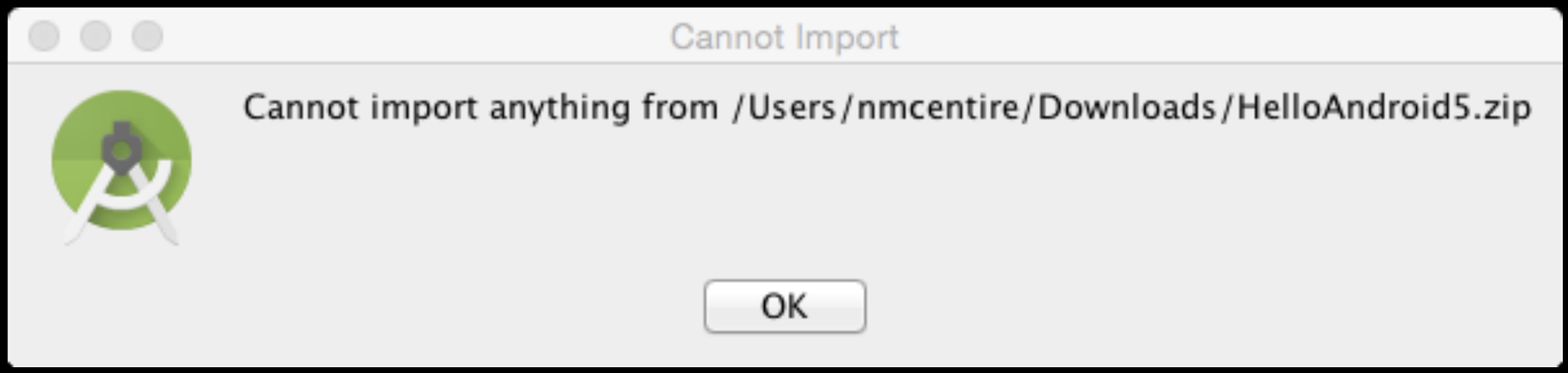
Import Non-Android Studio Project



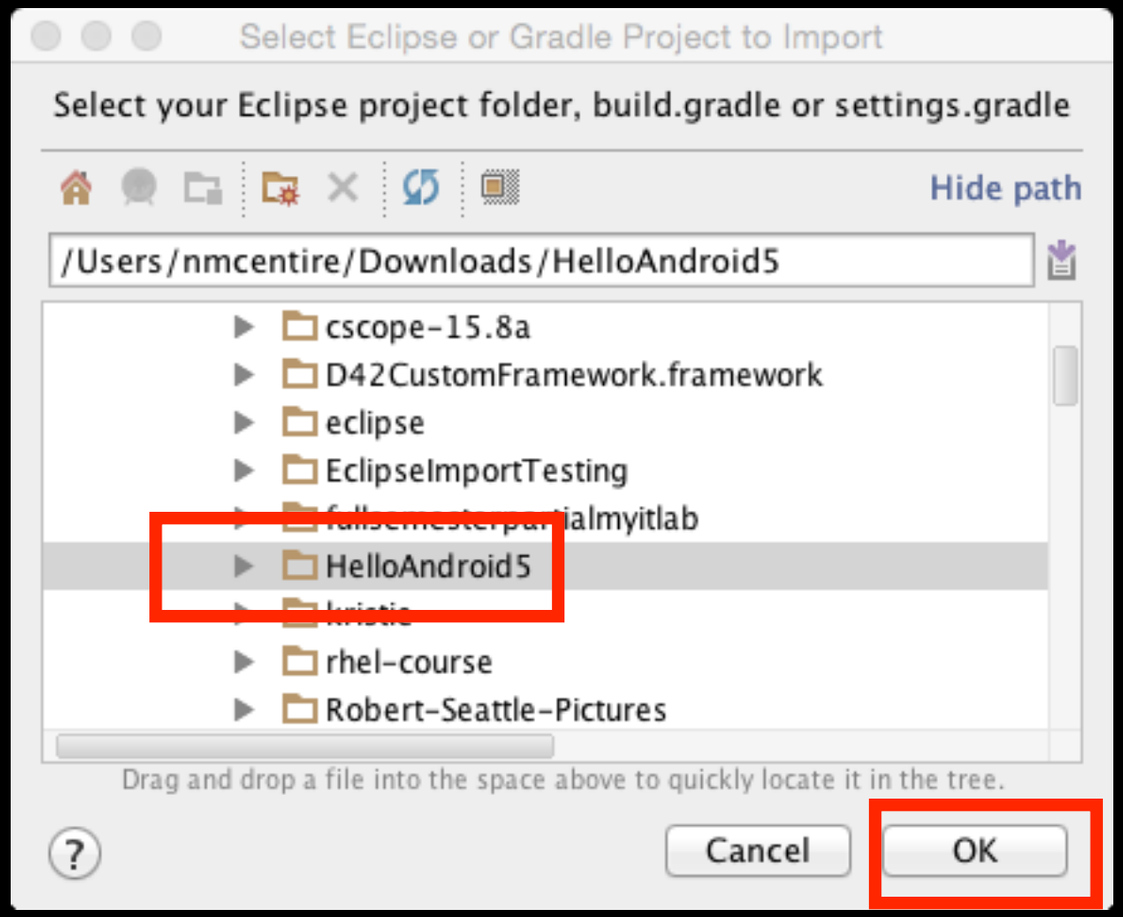
Select Project To Import



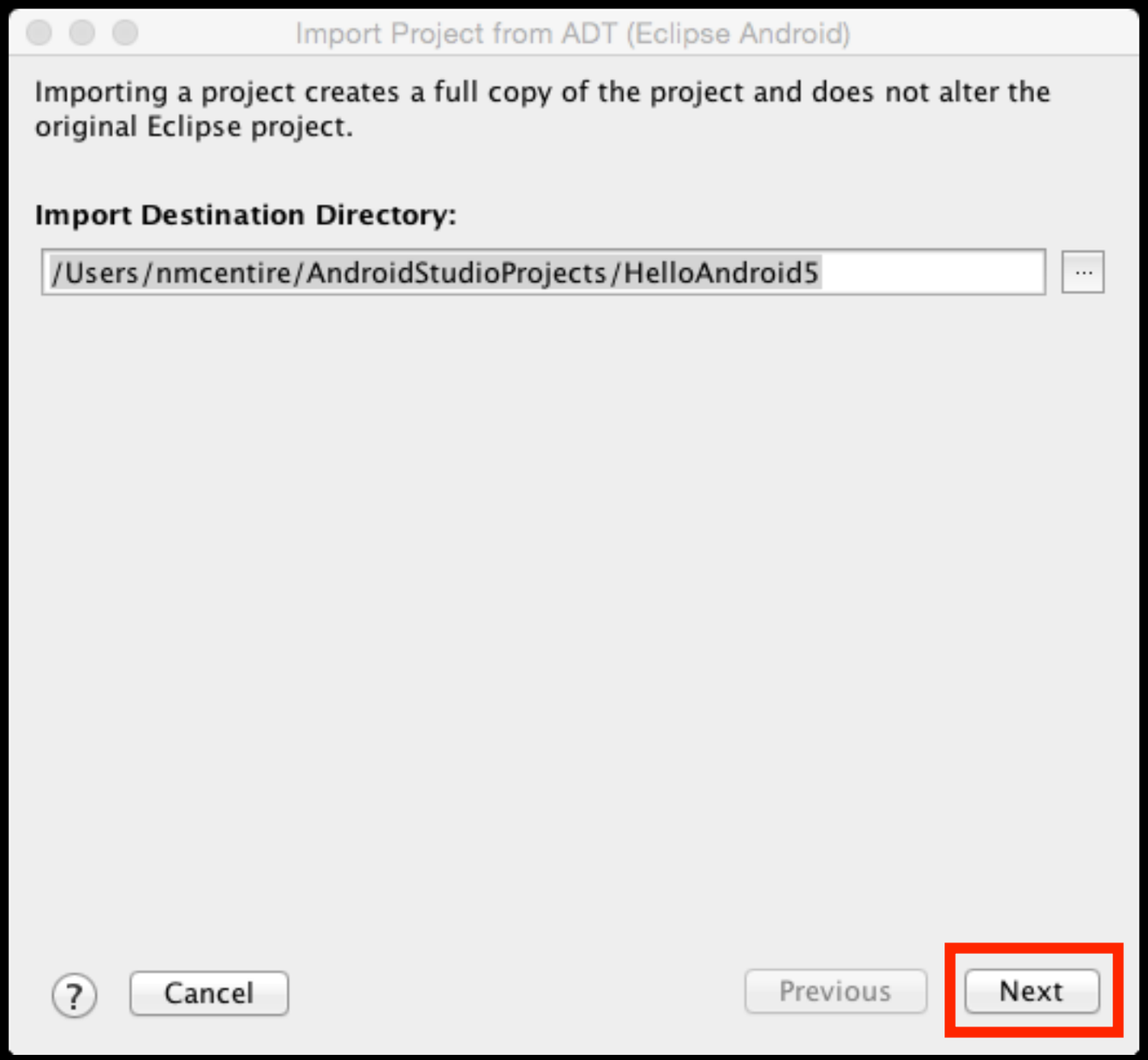
Cannot Directly Import From Zip File



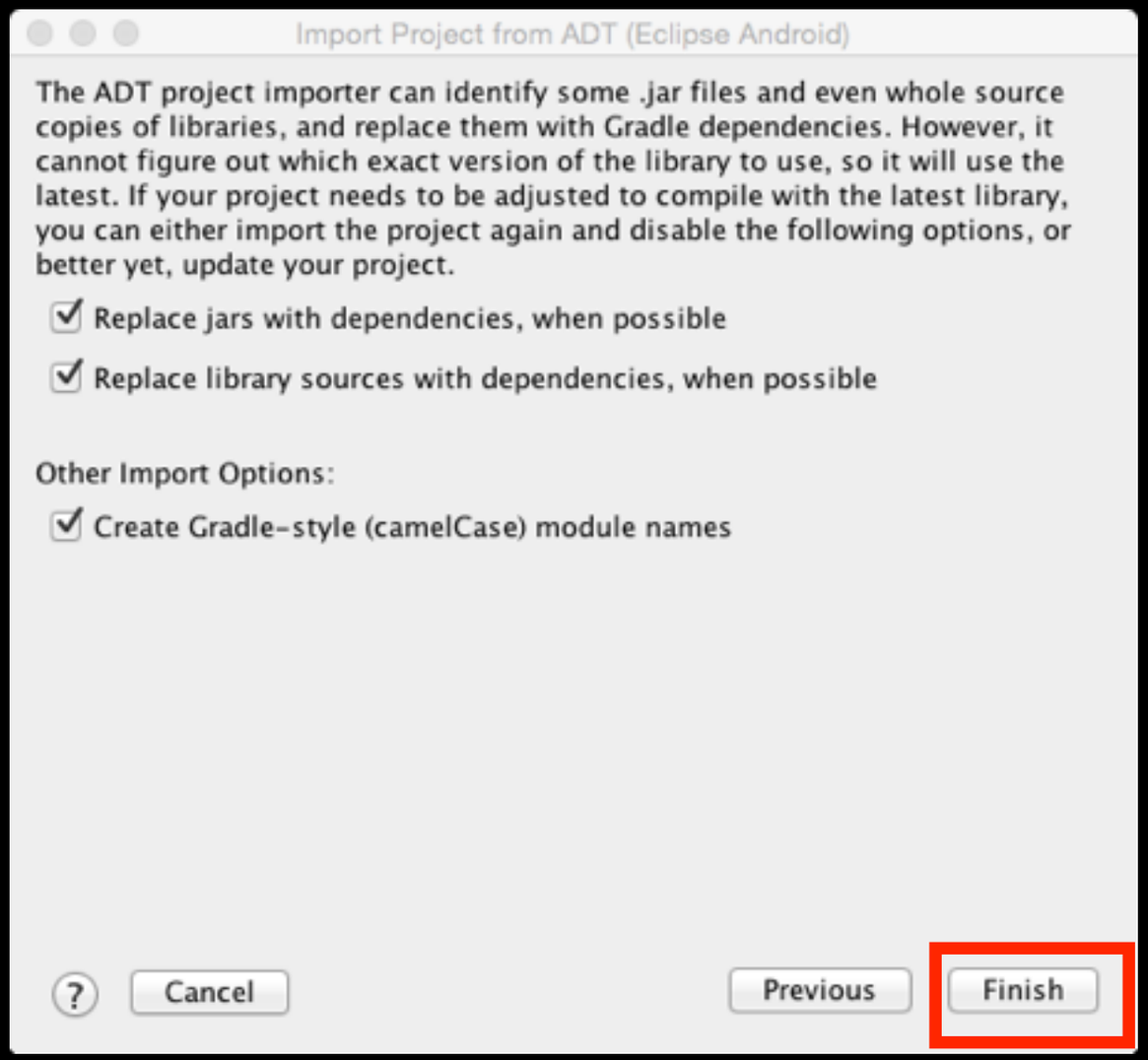
Unzip File, Then Import



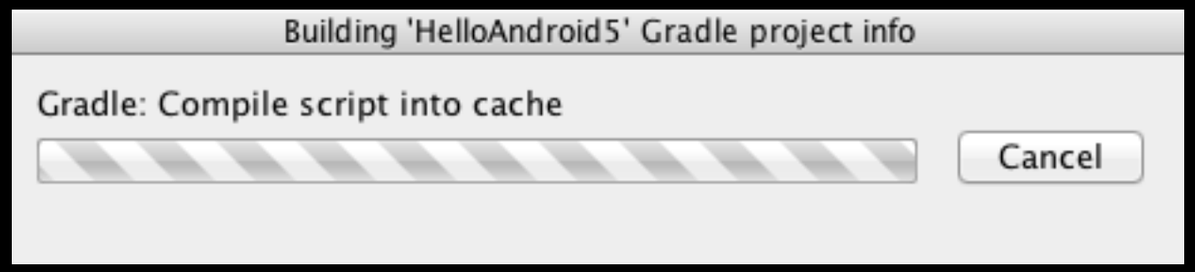
Import Destination Directory



Import Project from Eclipse ADT

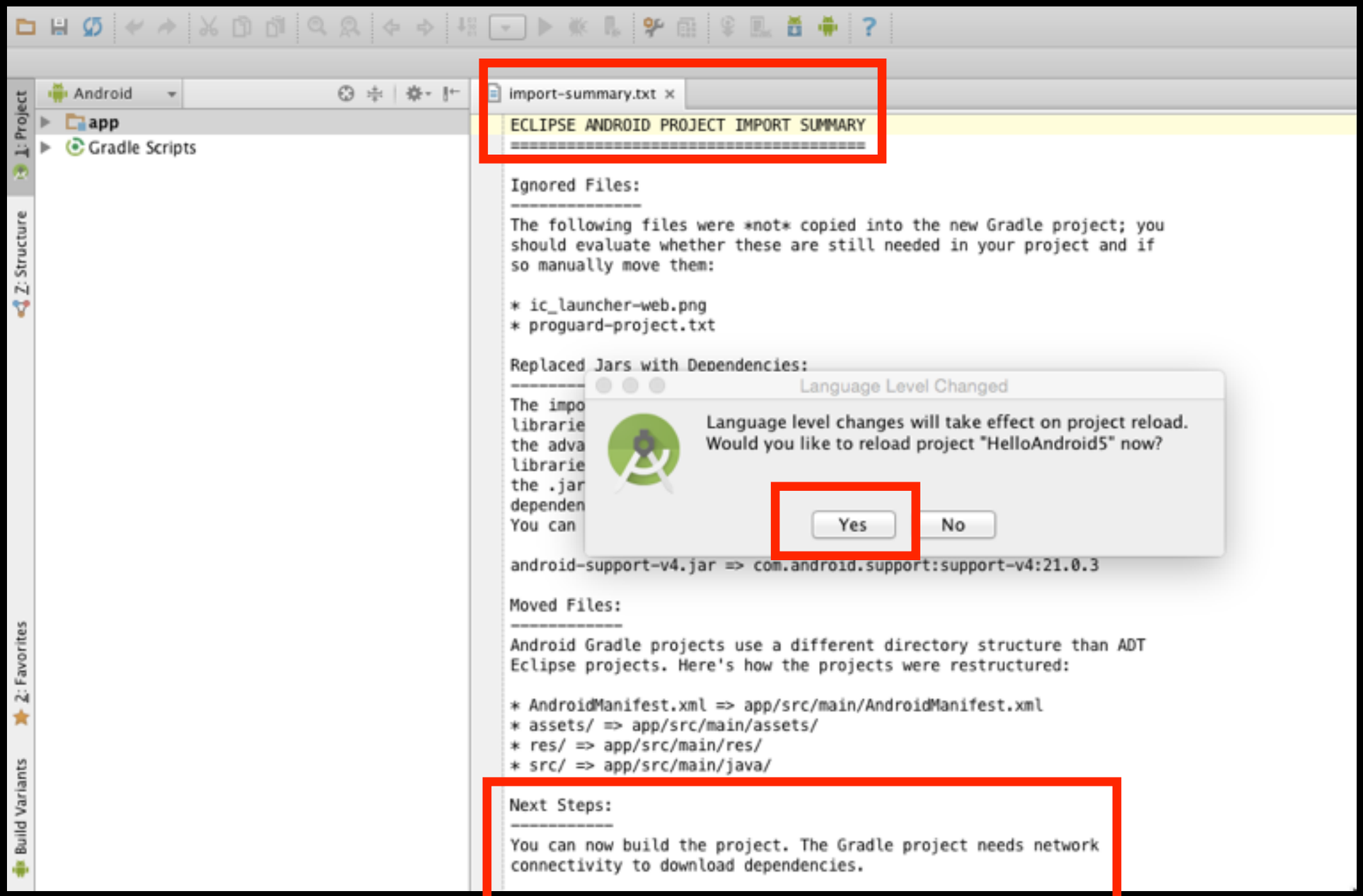


Import In Progress

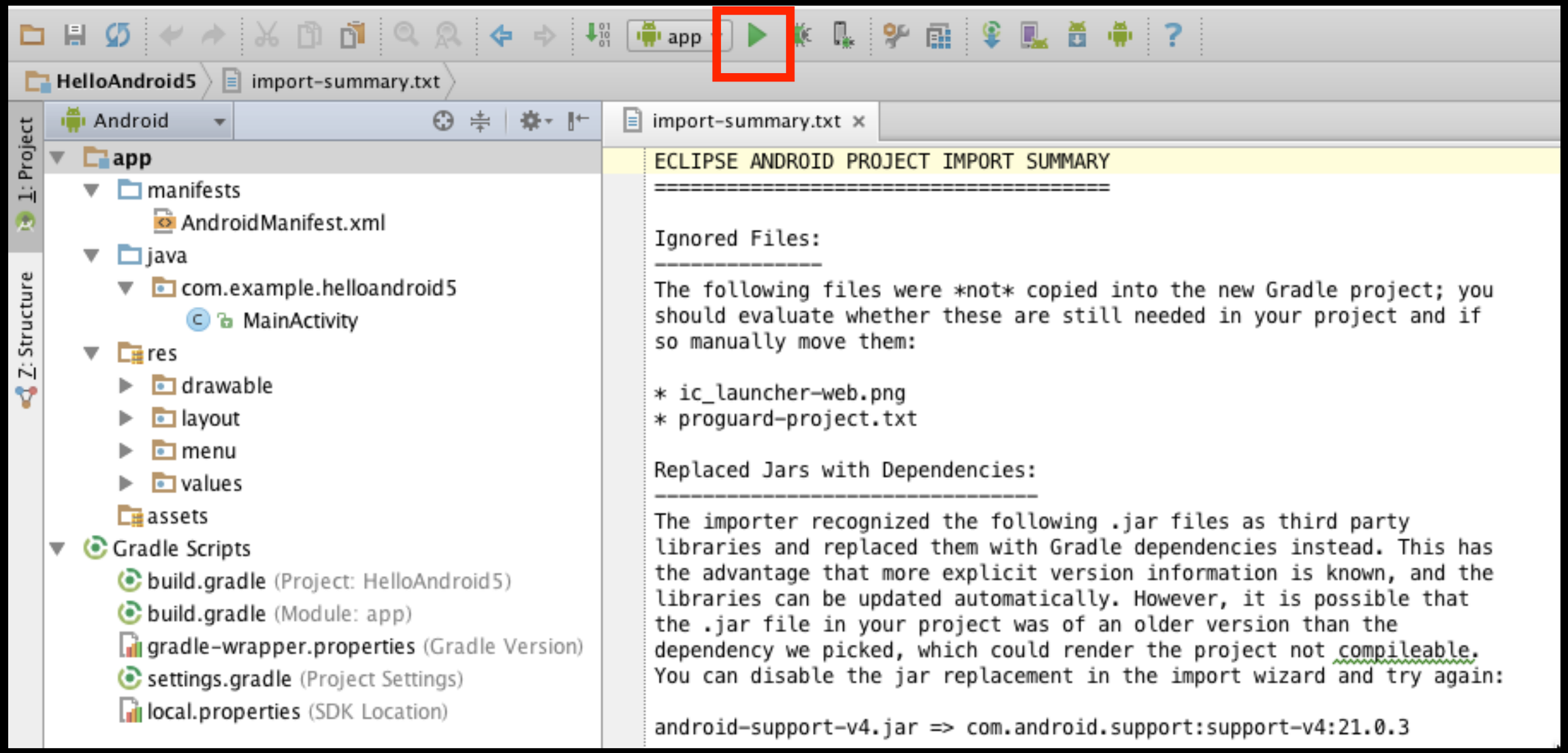


Note: Notice use of Gradle to build project.

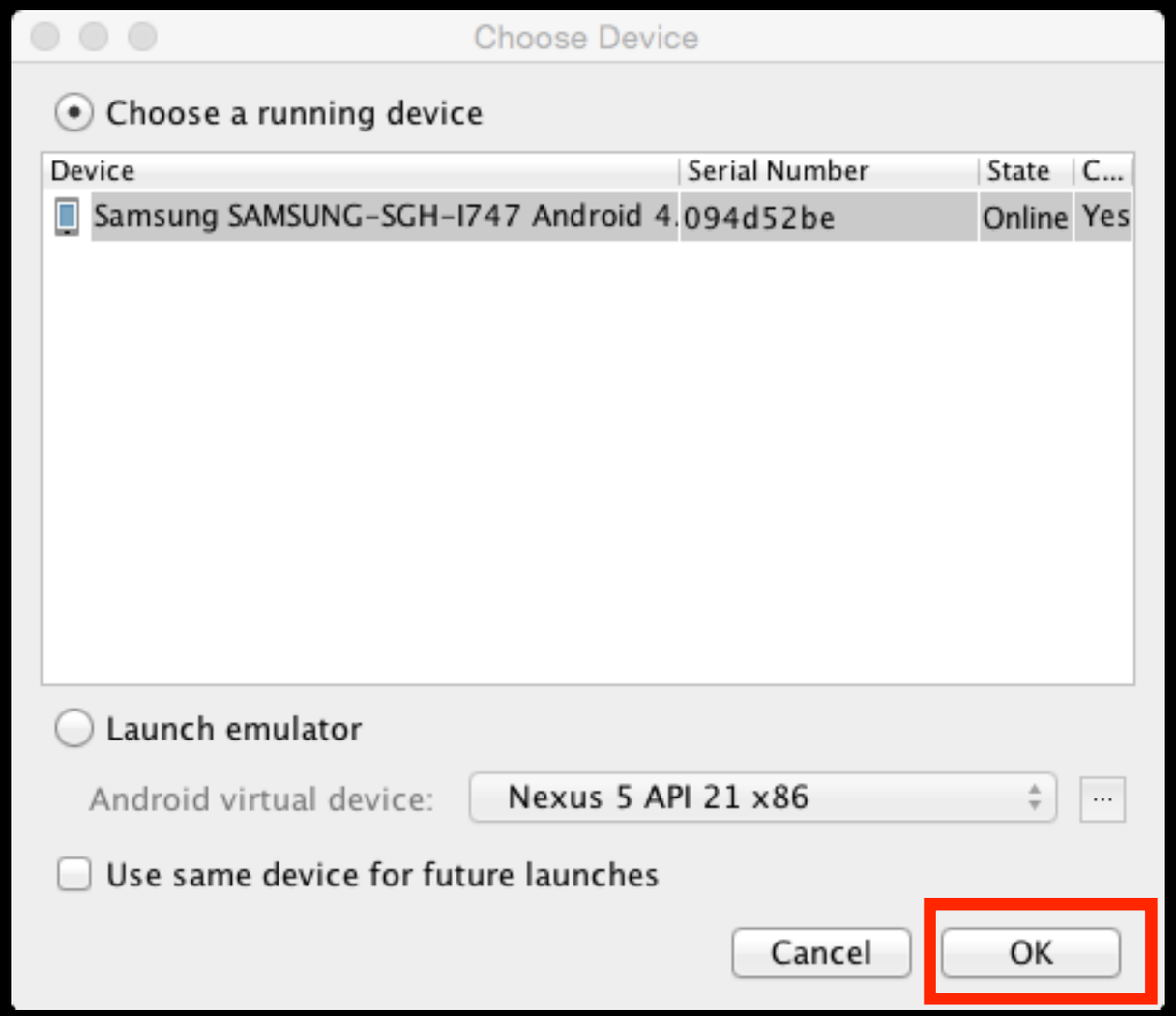
Results of Import



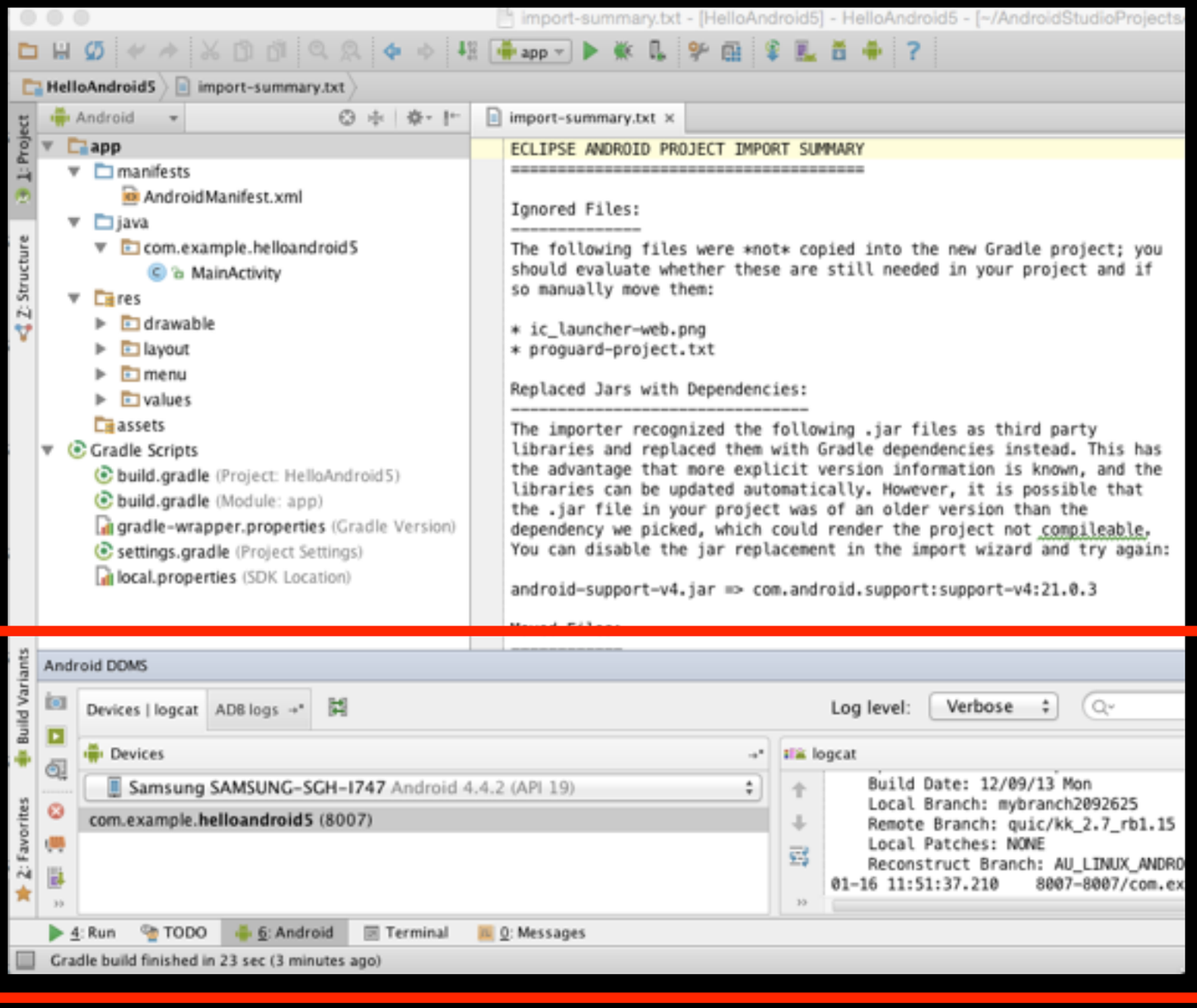
Build and Run In Android Studio



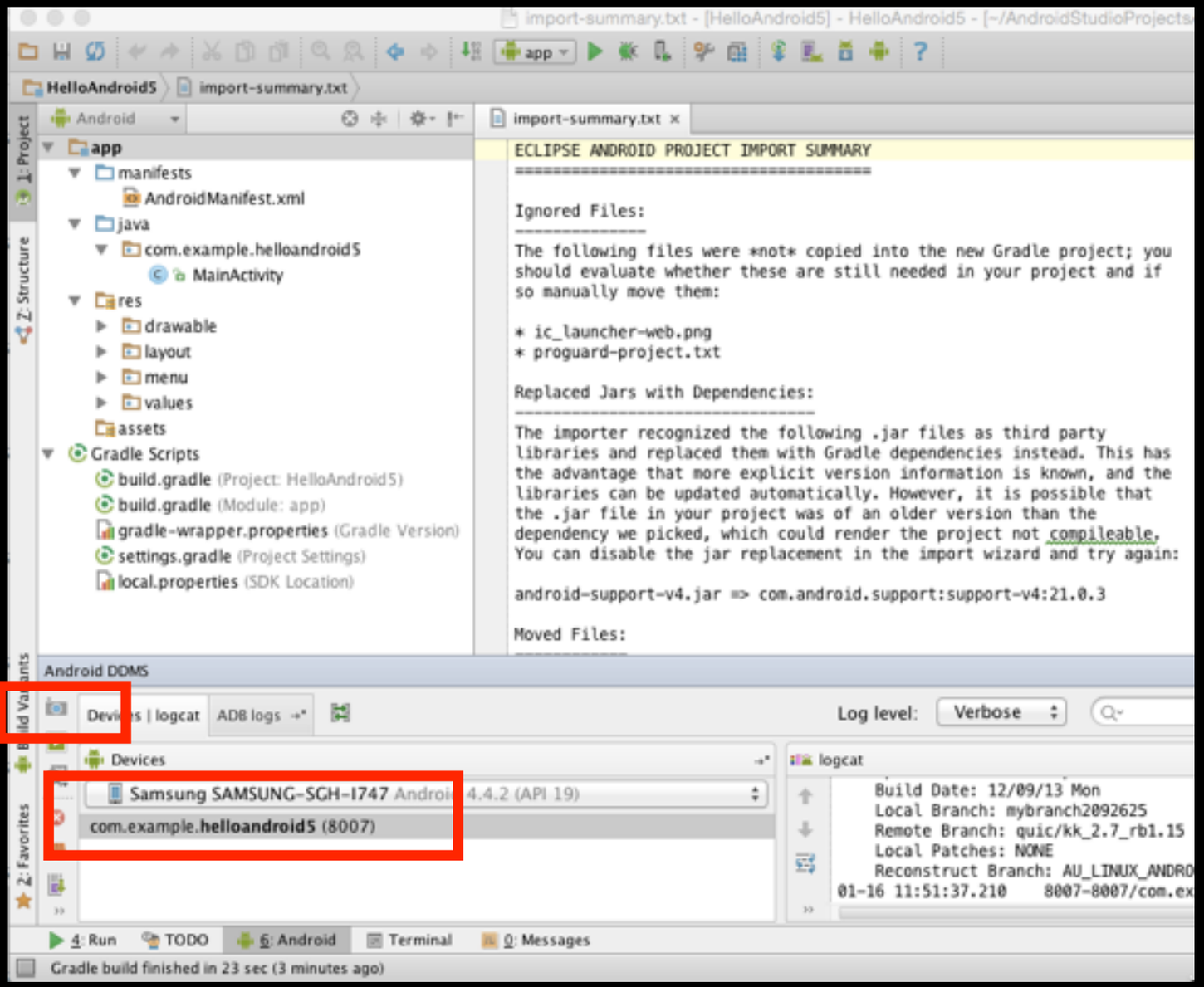
Device Chooser



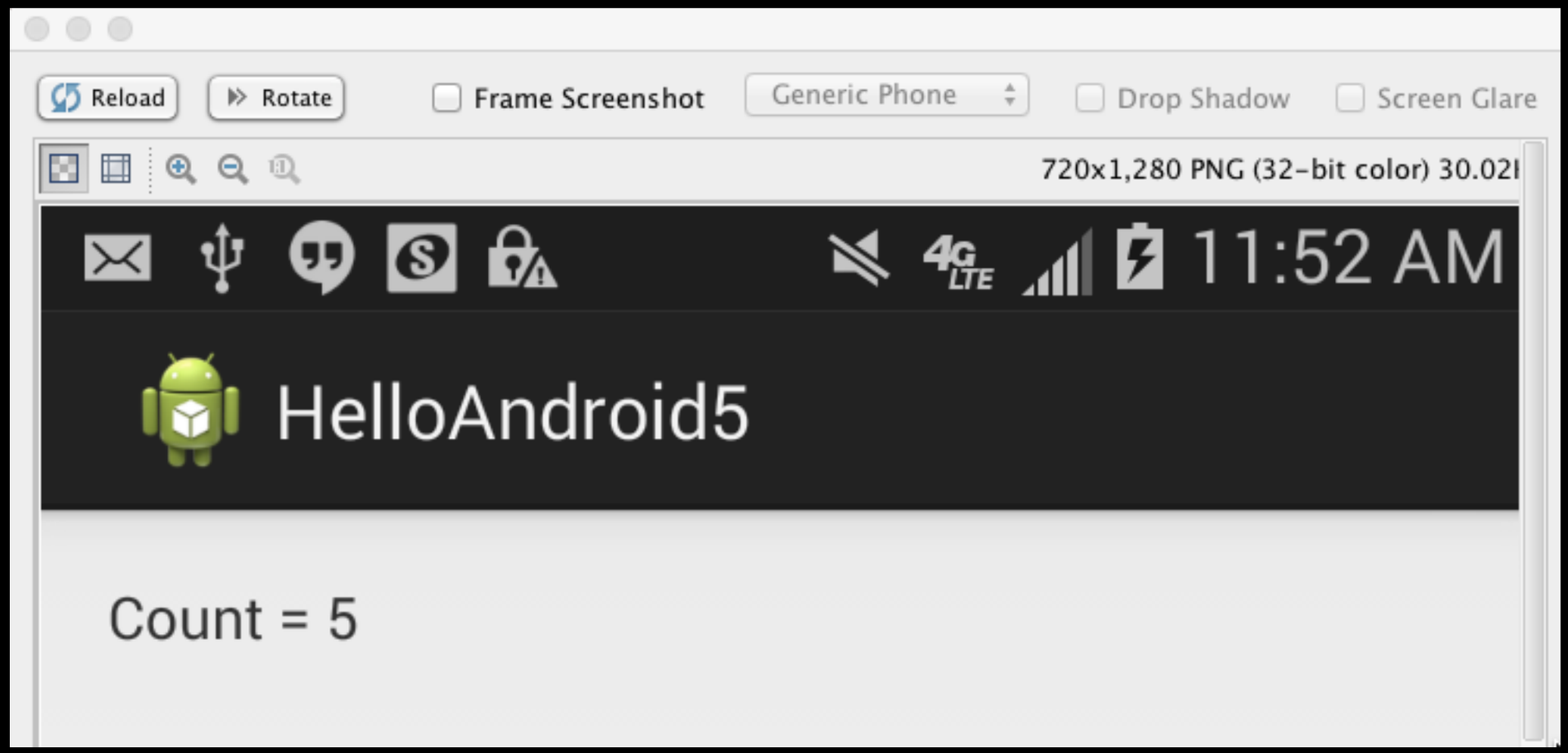
Android DDMS



Screen Capture



Captured Screen



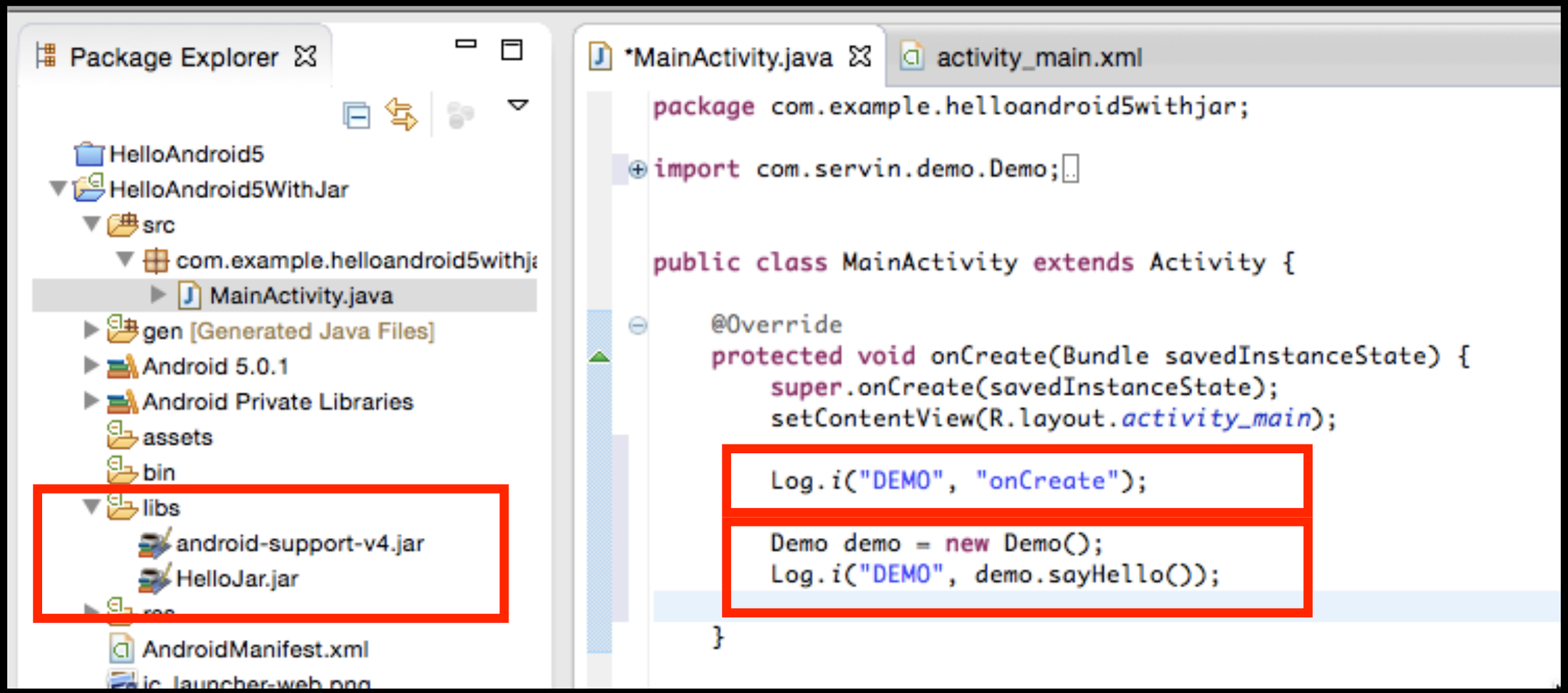
Summary

- Built and Ran Android 5 App using **Eclipse**
- **Exported** from **Eclipse** to **Android Studio**
- Built and Ran Android 5 App using **Android Studio**

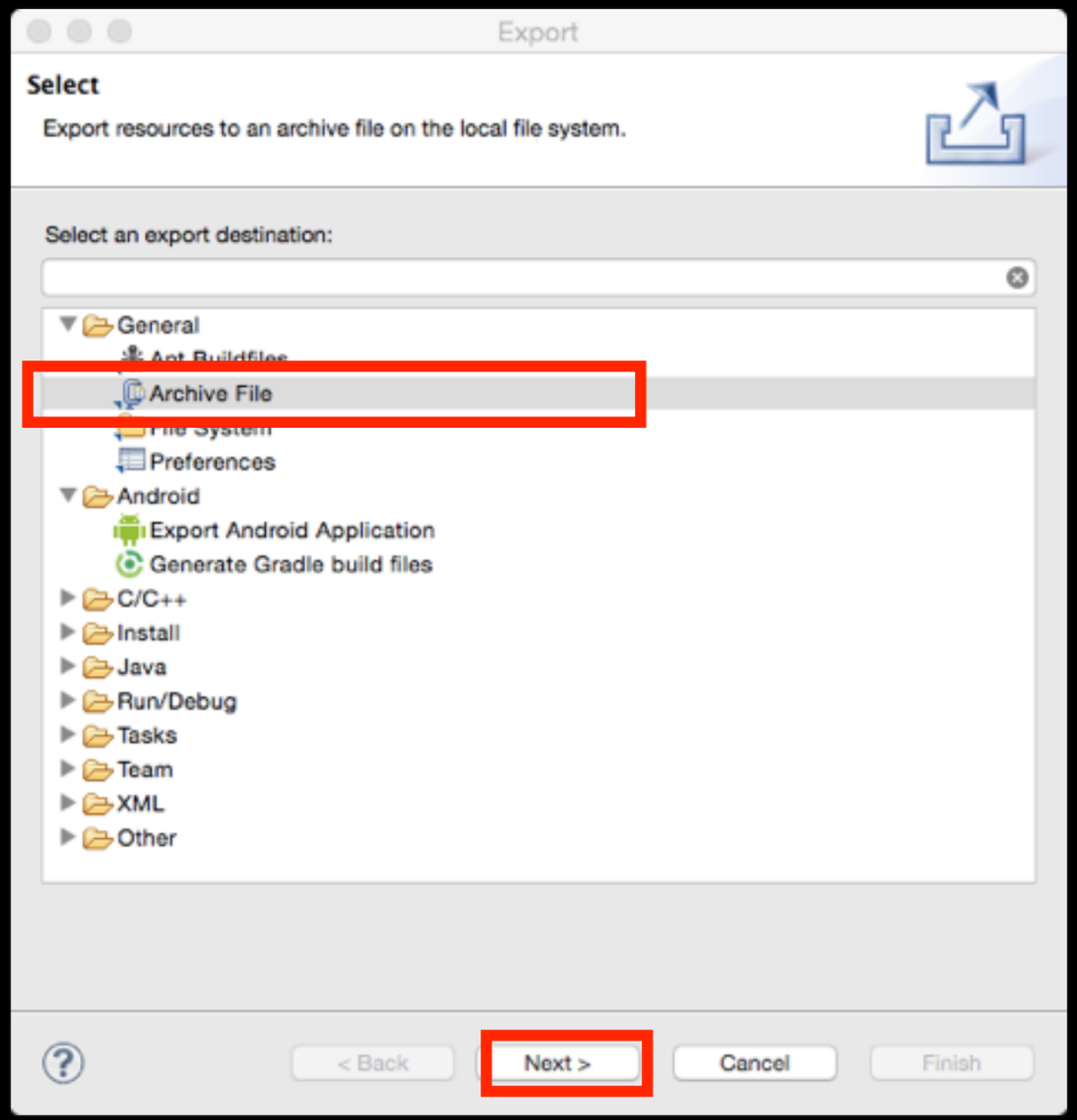
Demo

Import Eclipse ADT Project that has .jar files in the project

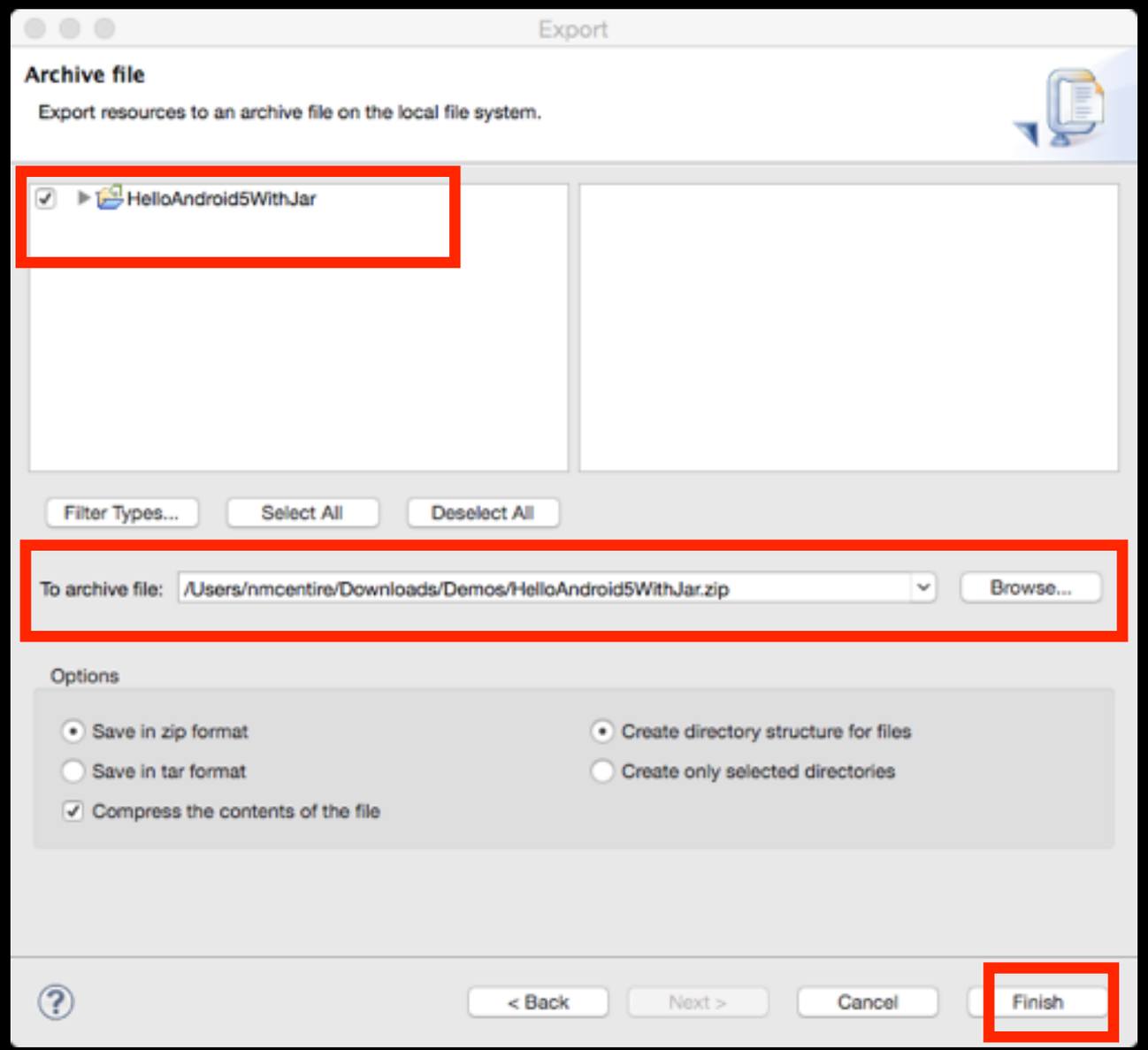
Eclipse ADT Project with .jar file (HelloJar.jar)



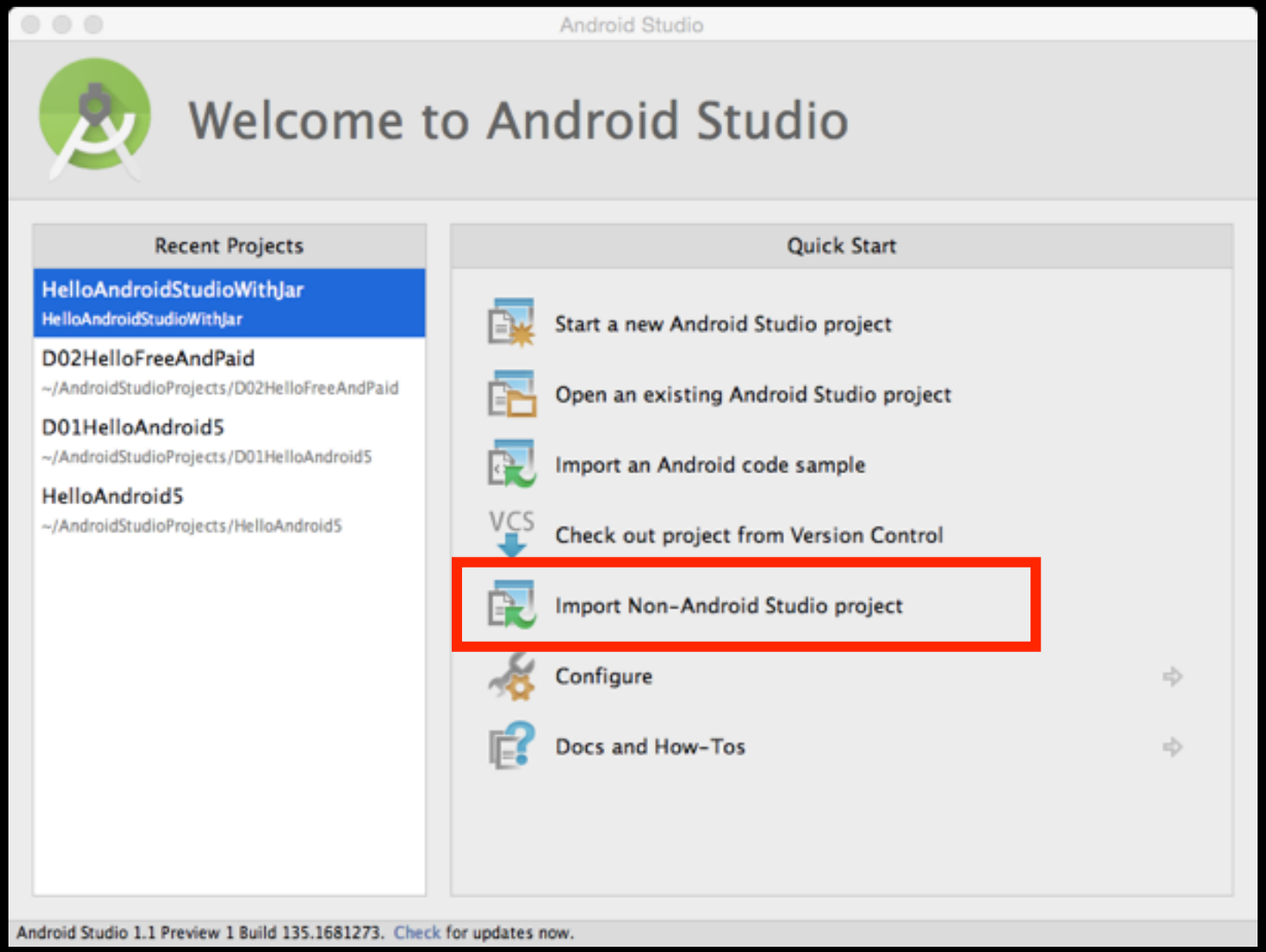
Export Project from Eclipse (Select Archive File)



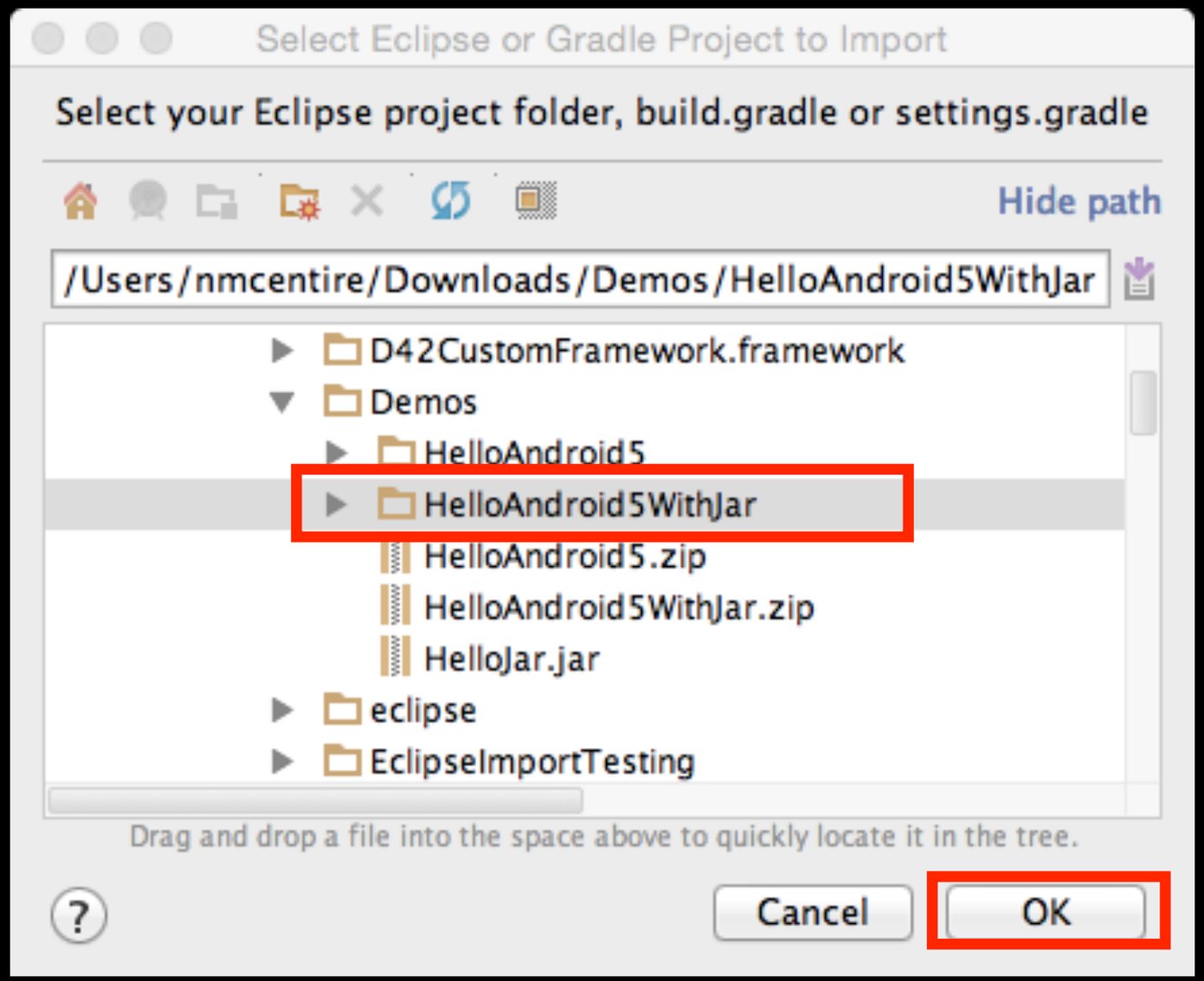
Archive File (HelloAndroid5WithJar)



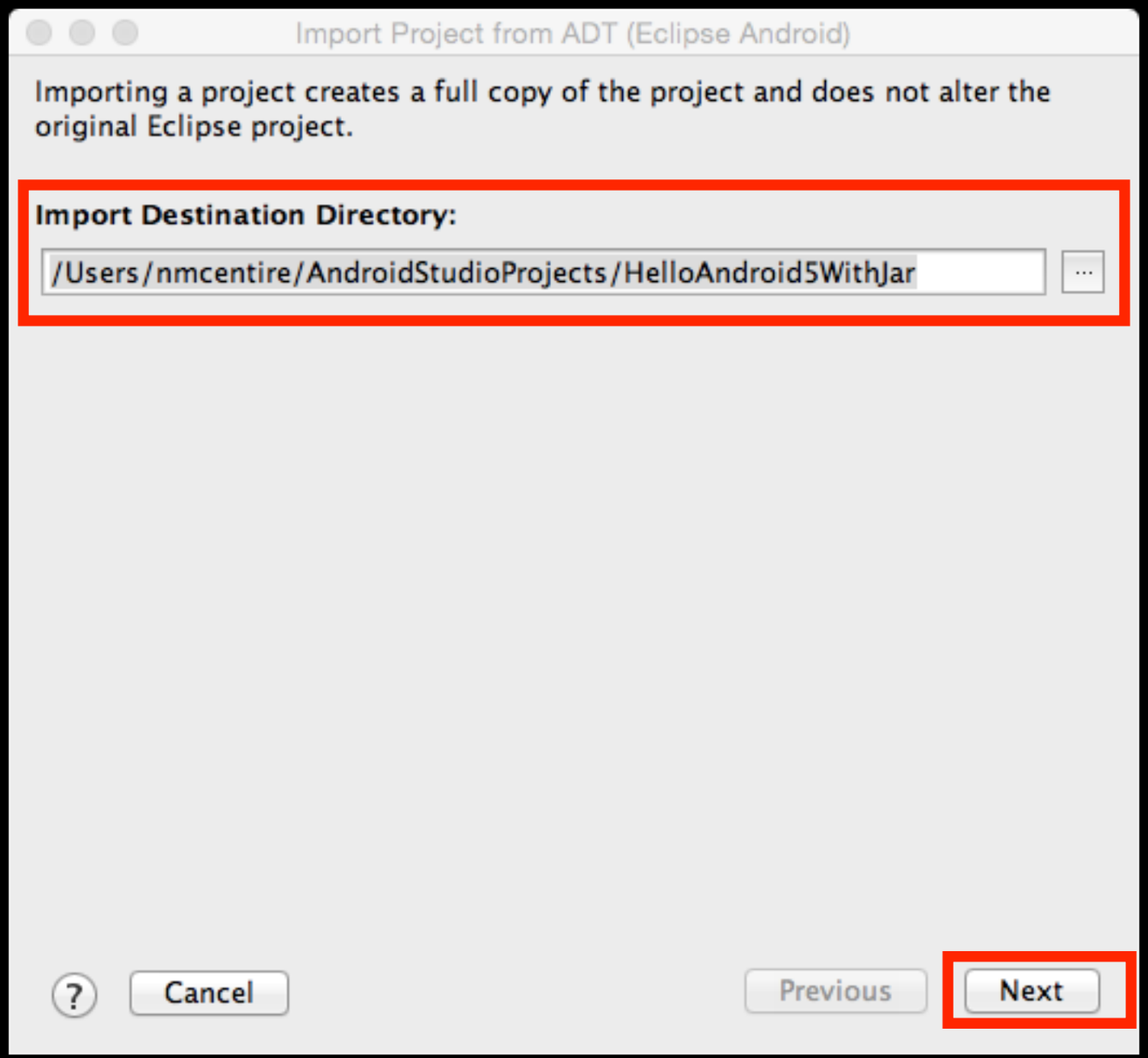
Import Into Android Studio



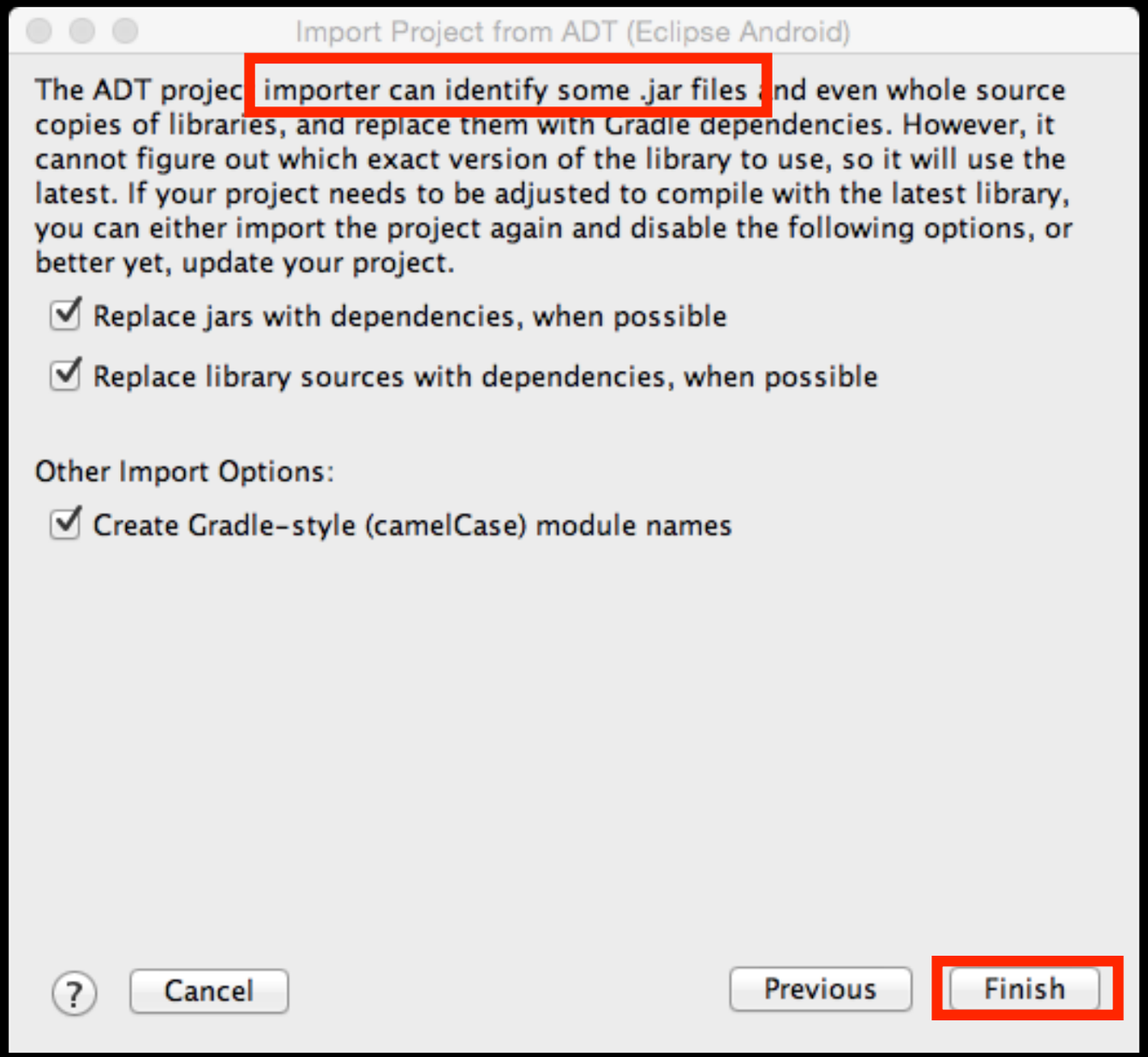
Select Directory



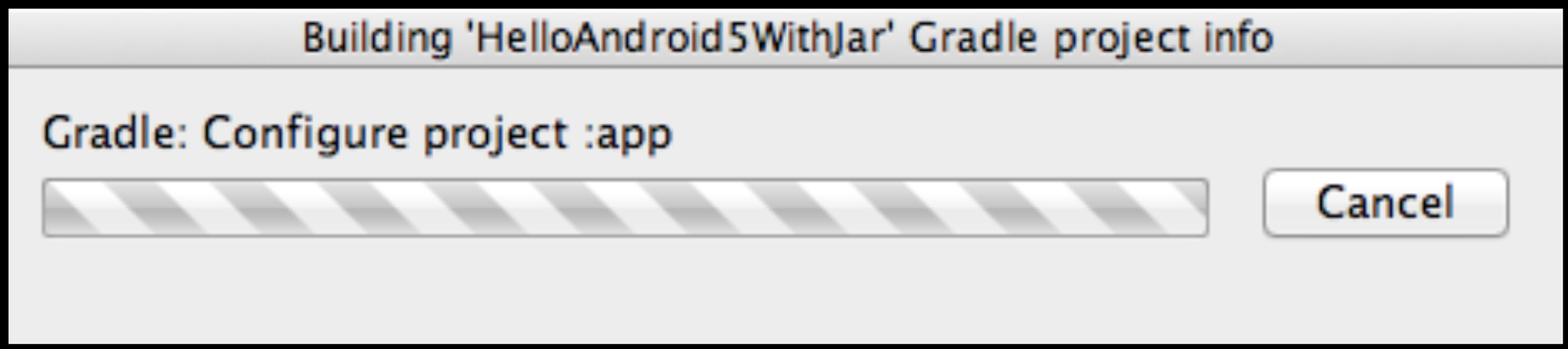
Destination Directory



“ADT project importer can identify some .jar files”



Gradle Configuration



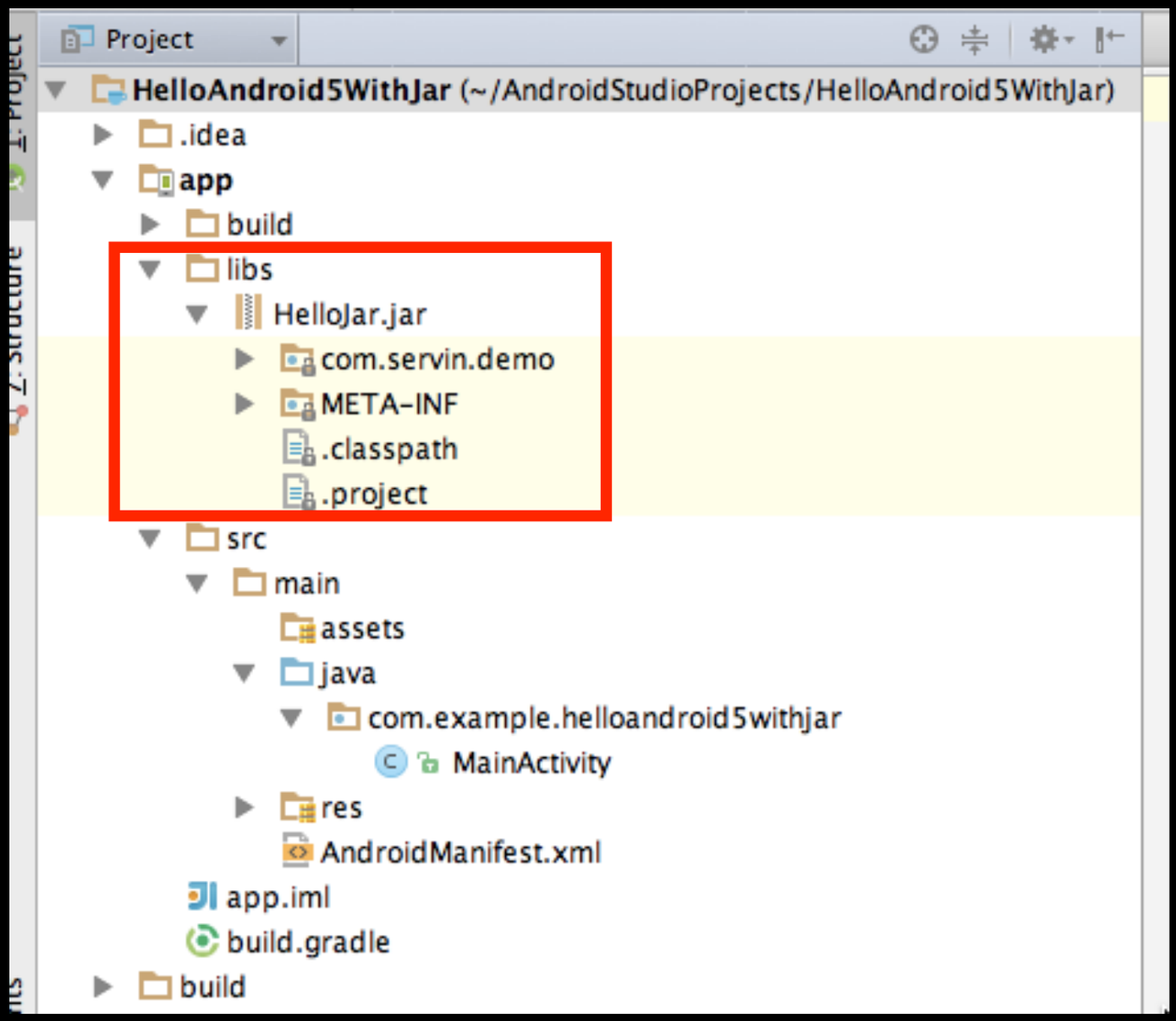
build.gradle(Module: app)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.example.helloandroid5withjar"
        minSdkVersion 19
        targetSdkVersion 21
    }

    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), '
        }
    }
}

dependencies {
    compile 'com.android.support:support-v4:21.0.3'
    compile files('libs/HelloJar.jar')
}
```

Moved Files During Import

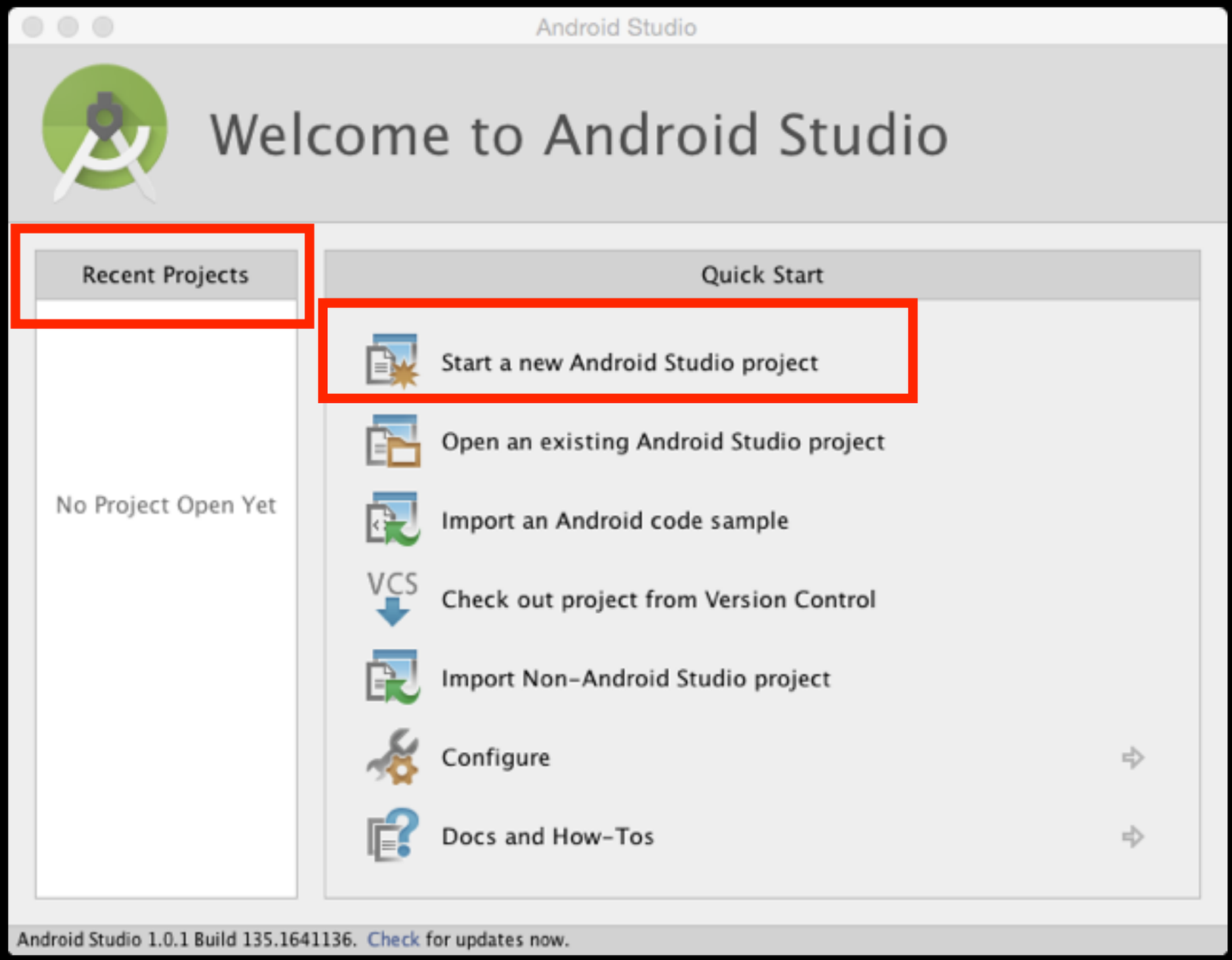
```
Moved Files:
------------
Android Gradle projects use a different directory structure than ADT
Eclipse projects. Here's how the projects were restructured:

* AndroidManifest.xml => app/src/main/AndroidManifest.xml
* assets/ => app/src/main/assets/
* libs/HelloJar.jar => app/libs/HelloJar.jar
* res/ => app/src/main/res/
* src/ => app/src/main/java/
```

Results Shown In Android Studio

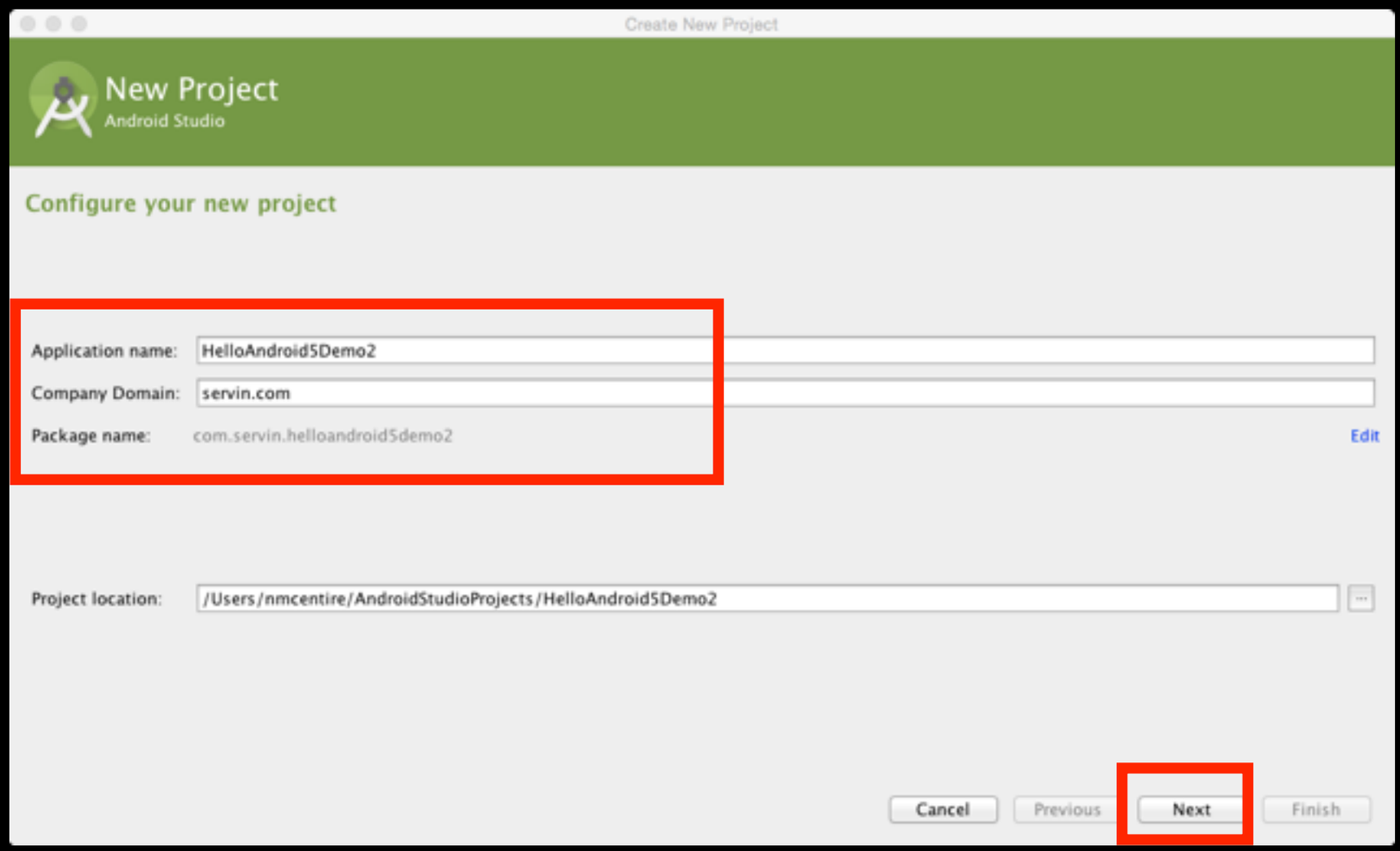


Using Android Studio

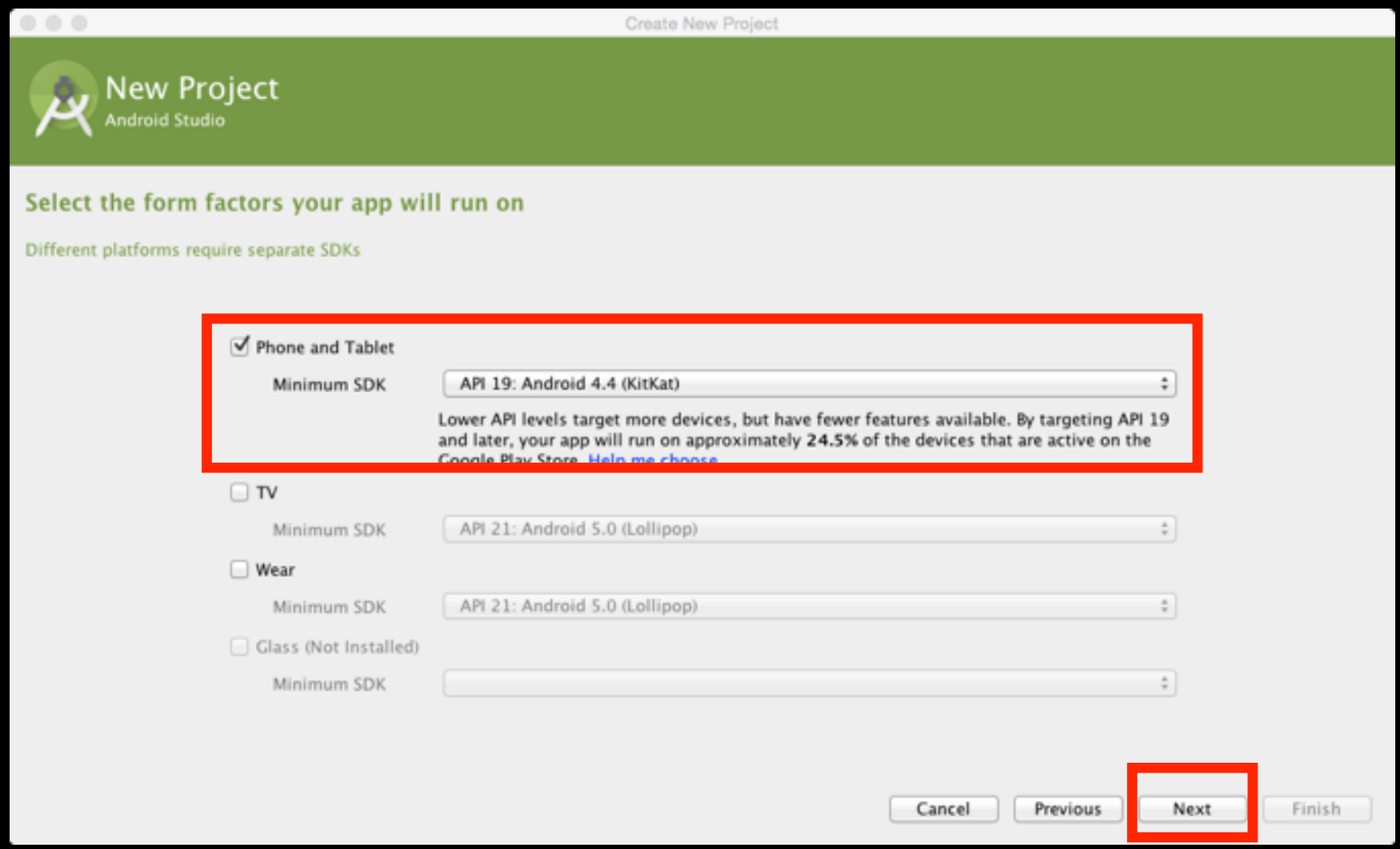
Android Studio Startup



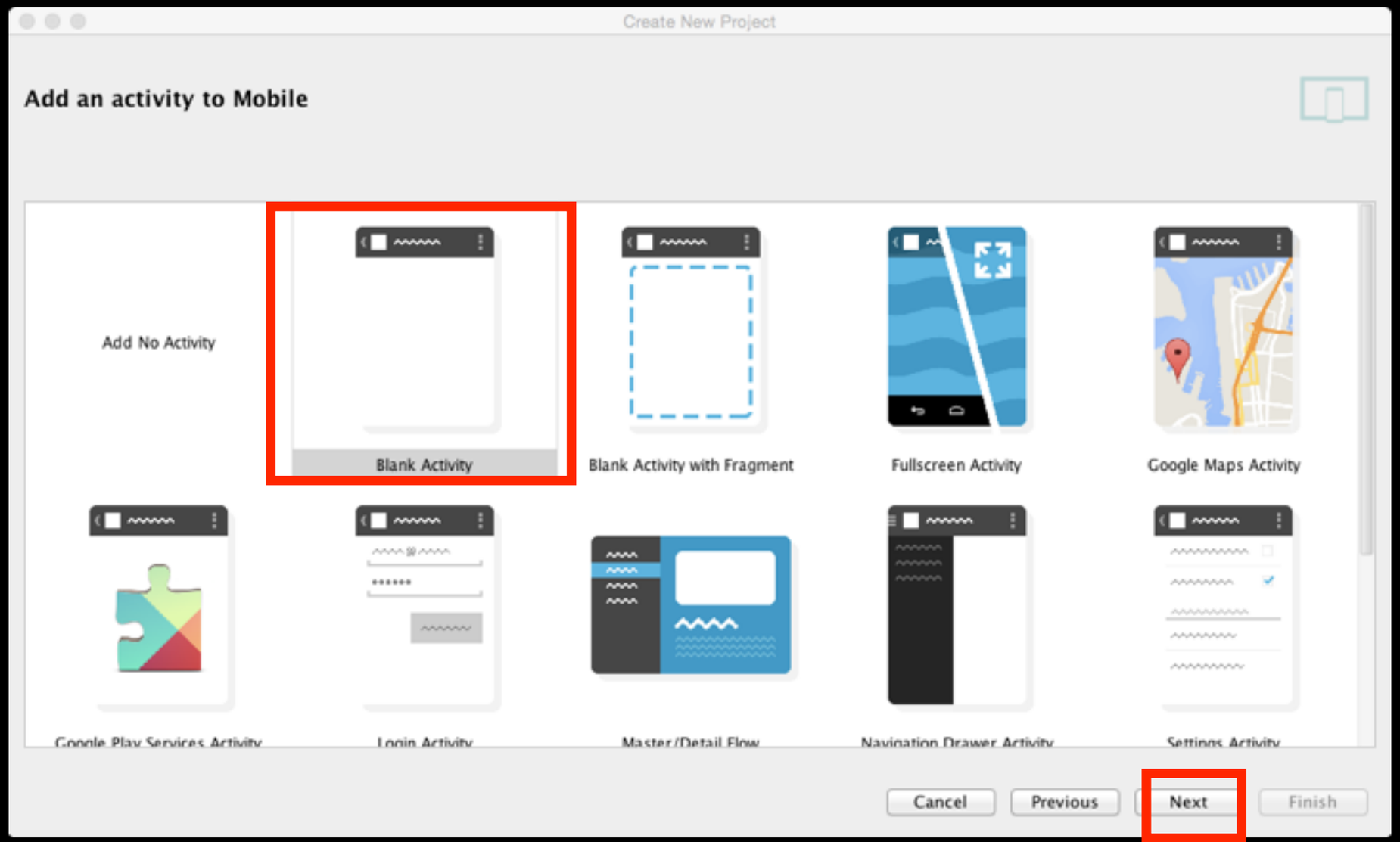
New Project



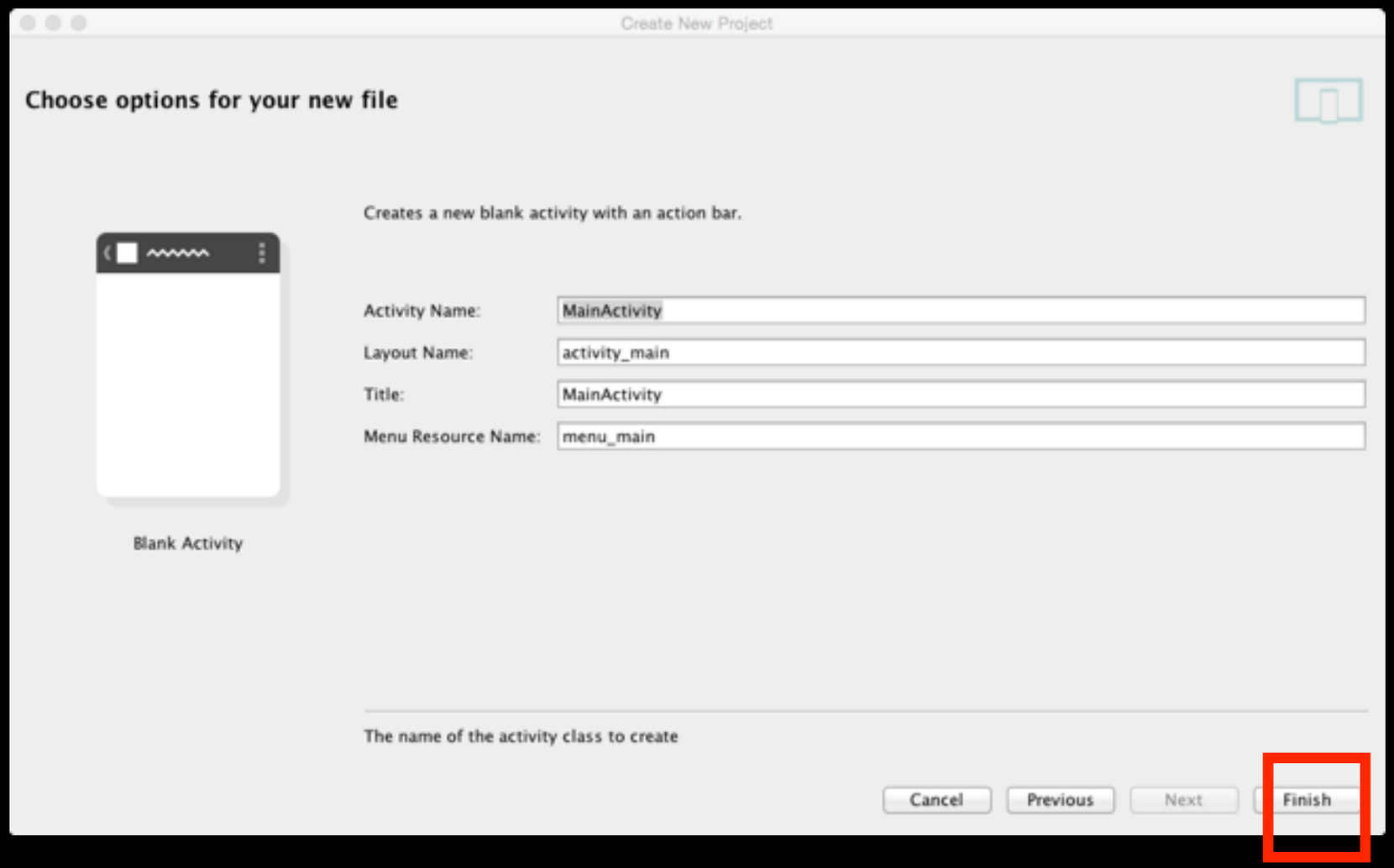
Select Form Factors



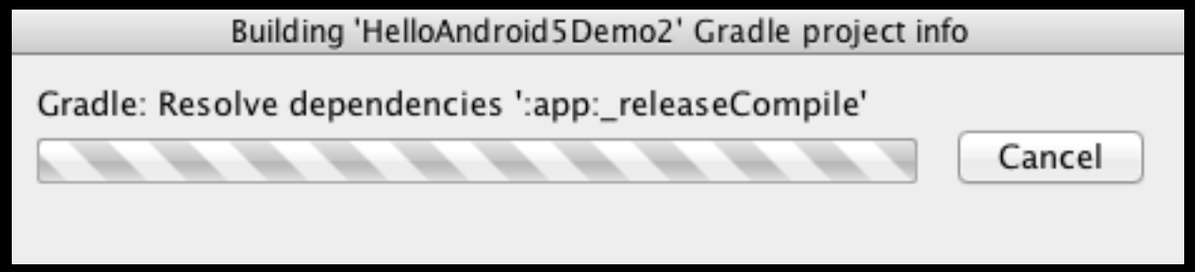
Add Activity



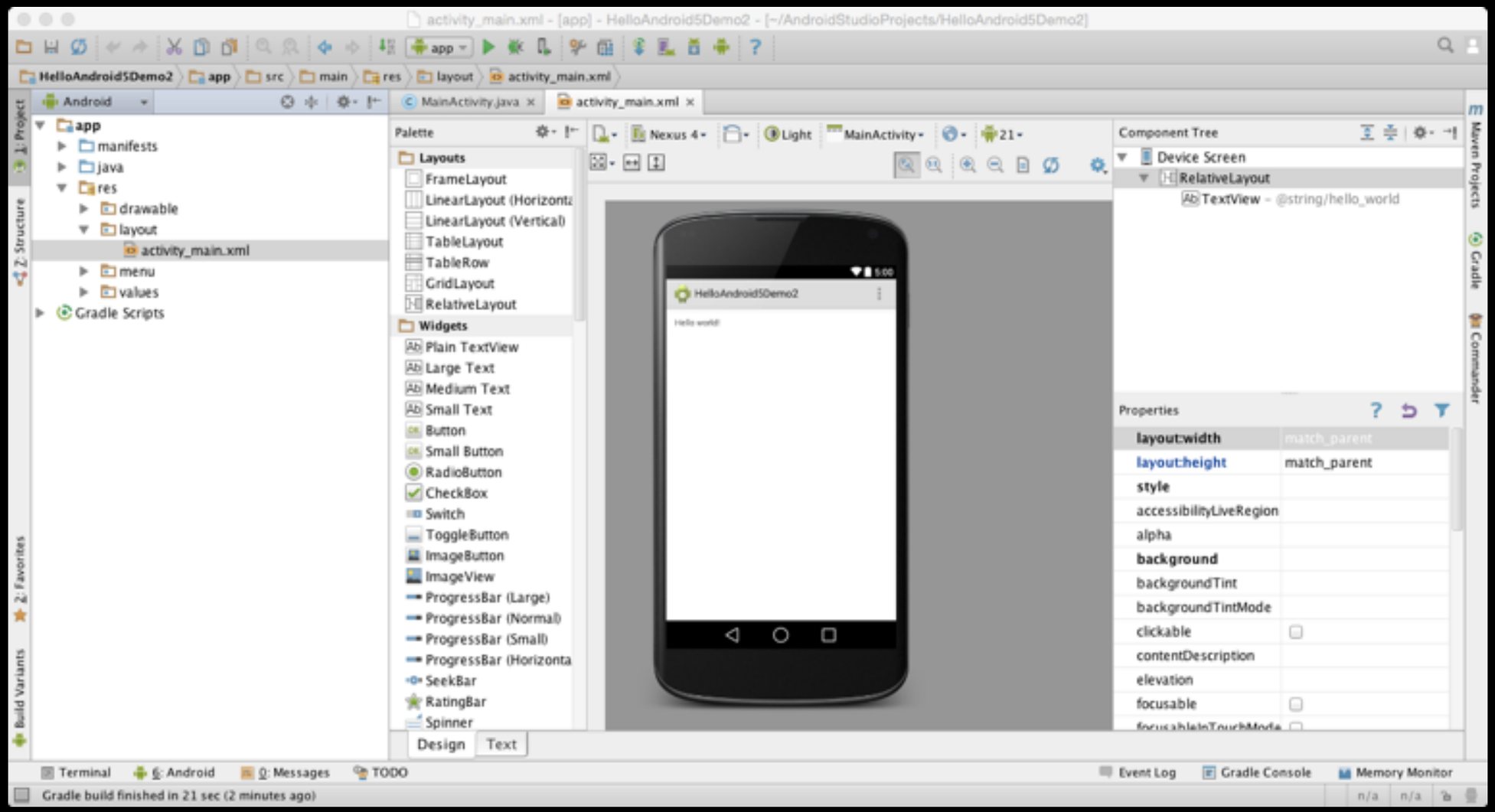
Options for New File (Use Defaults)



Getting Project Ready (Takes a few moments)



Project Ready for Editing



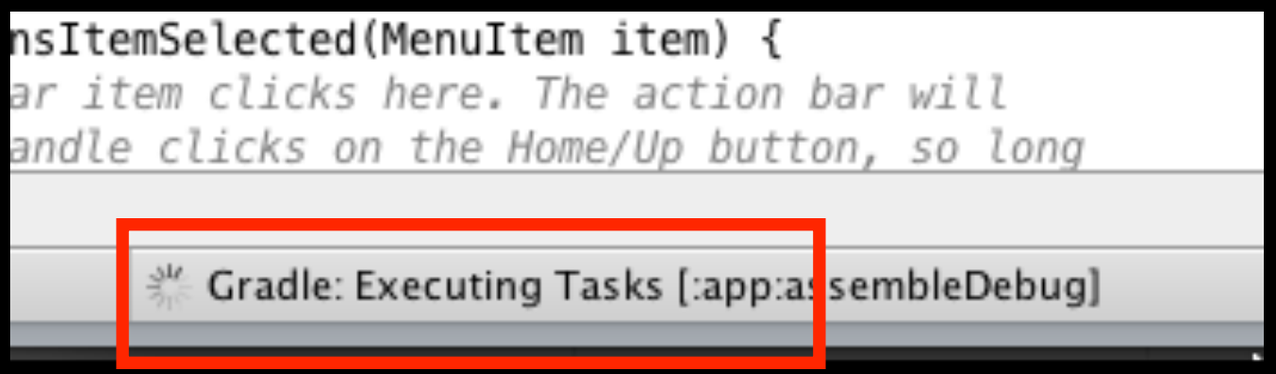
Editing XML Layout

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_pare
    android:layout_height="match_parent" android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    android:paddingBottom="16dp" tools:context=".MainActivity">

    <TextView
        android:id="@+id/textView1"
        android:text="Hello world!" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

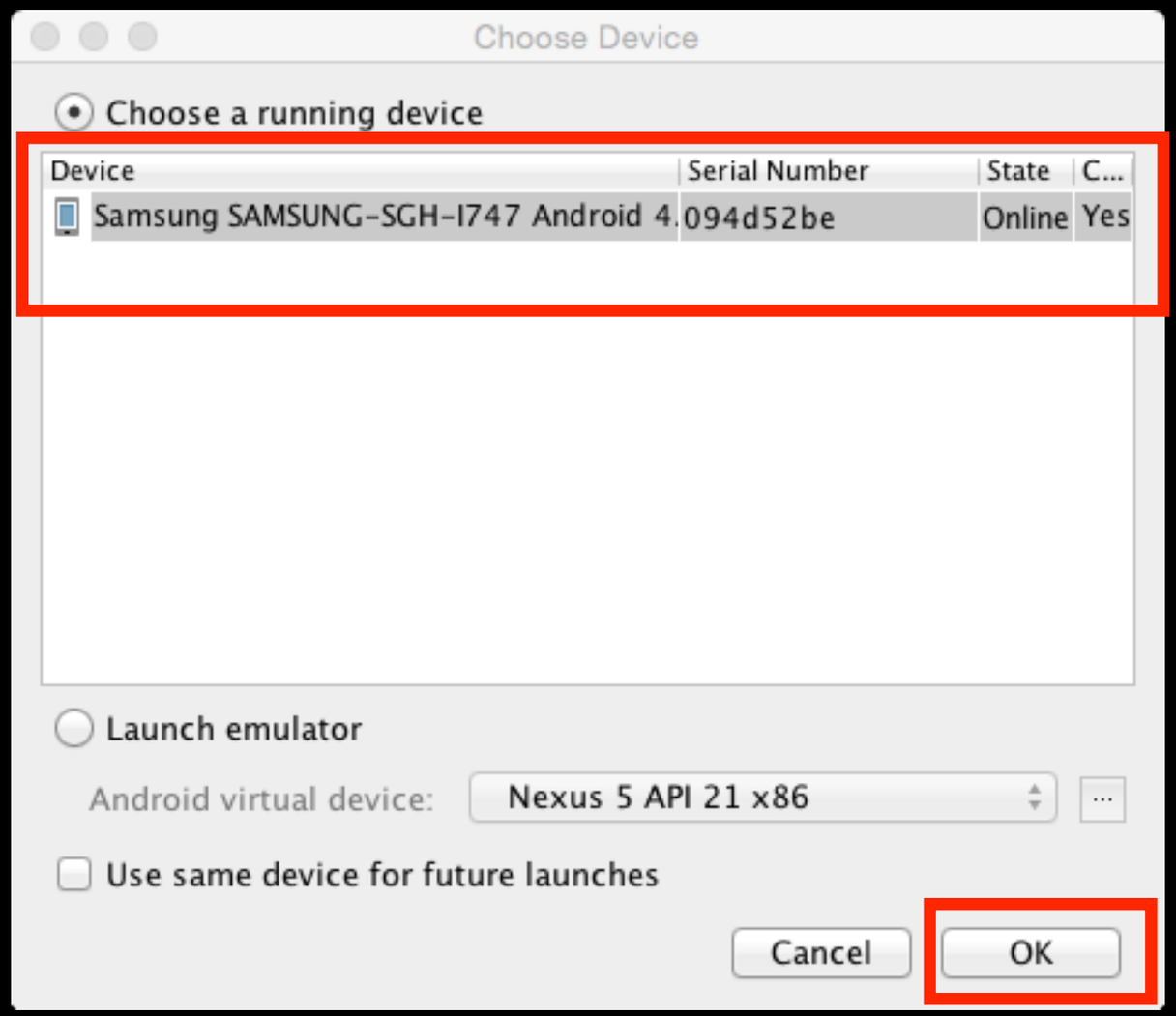
Editing MainActivity.java

```
public class MainActivity extends ActionBarActivity {

    TextView textView;
    int count;

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

        textView = (TextView) findViewById(R.id.textView1);
        textView.setText("Count = 0");
    }

    @Override
    public boolean onTouchEvent(MotionEvent event) {
        if (event.getAction() == MotionEvent.ACTION_DOWN) {
            count++;
            textView.setText("Count = " + count);
        }
        return super.onTouchEvent(event);
    }
```

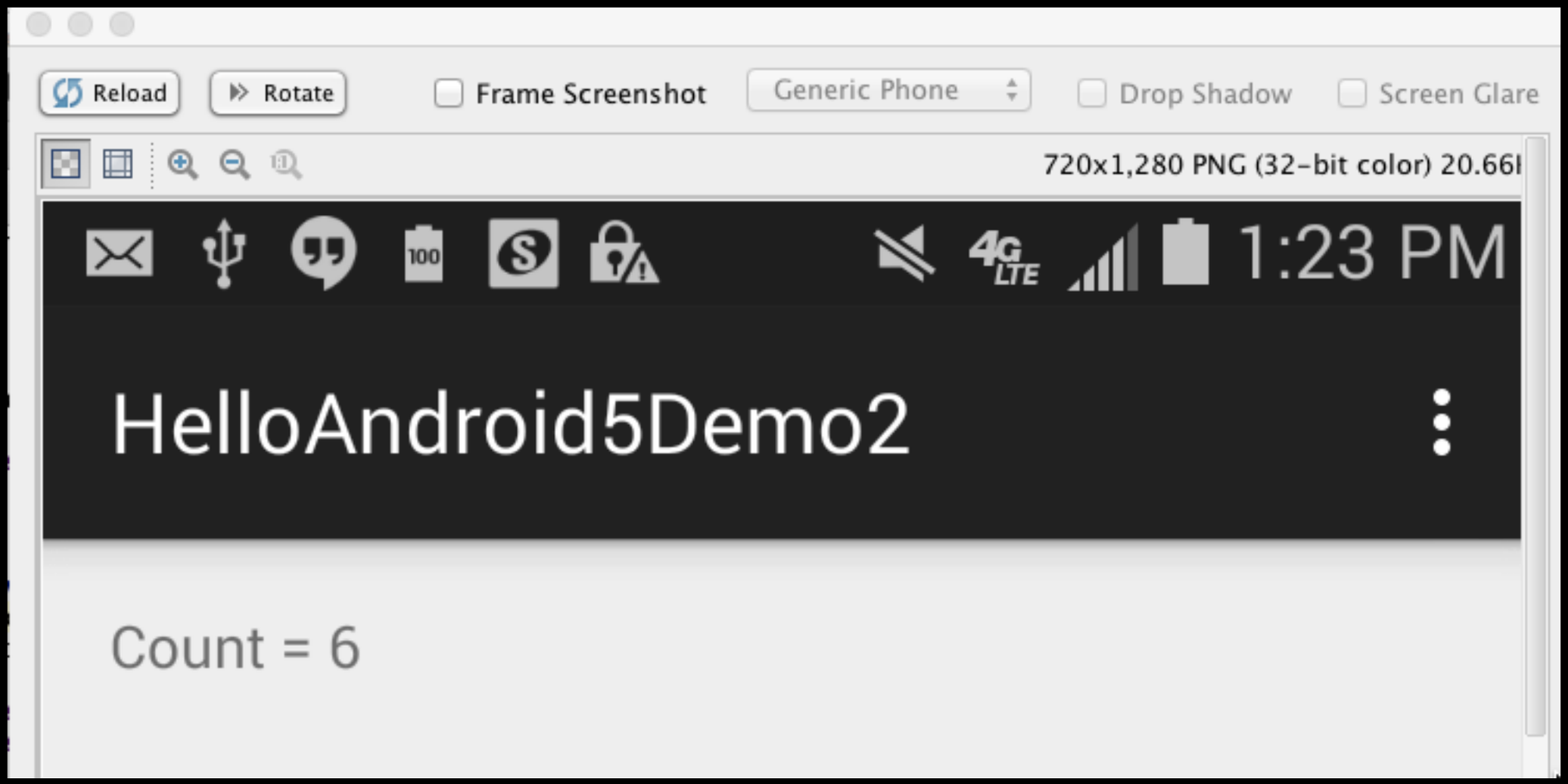
Tracking status of the build



Device Chooser



Screen Capture



Android Studio Skills

Android Studio Skill
Code Completion

- Just start typing and Android Studio will show you choices

- Just press **ENTER** to complete a given choice

- Also press **TAB** go to next parameter

- Example

    - Toast.makeText(this,”hi”,Toast.LENGTH_LONG).show()

Android Studio Skill
SmartType Code Completion

- When you get to certain locations in your code, press **Shift+Control+Spacebar** to see choices based on context

- Example
  - StringBuffer sb = new **Shift+Control+SpaceBar**
  - Calendar now = new GregorianCalendar(**Shift +Control+SpaceBar**)

Android Studio Skill
Navigate in Current File

- To navigate in the current file
    - Option 1 [Menu]. **Navigate**, **File Structure**
        - Then select member you want to navigate to
    - Option 2 [Kbd]. **Command+F12**
        - Then select member you want to navigate to

Android Studio Skill
Quickly View Definition

- To view a quick **definition** of a class
- Step 1. Highlight a given class name
- Step 2 [Menu]. **View**, **Quick Definition**

Android Studio Skill
Quickly View Documentation

- To quickly view documentation of a class
  - Step 1. Highlight a given class name
  - Step 2.
    - Option 1 [Menu]. **View**, **Quick Documentation**
    - Option 2 [Kbd]. **F1**

Android Studio Skill
Navigate to Declaration

- To navigate to the declaration of a class, variable, or method
- Step 1. Position cursor on name
- Step 2.
  - Option 1 [Menu]. **Navigate**, **Declaration**
  - Option 2 [Kbd]. **Command+b**

Android Studio Skill
Refactor a Name

- To refactor a name of a class, method, or variable

- Step 1. Position cursor on name

- Step 2.

  - Option 1 [Menu]. **Refactor**, **Rename**

  - Option 2 [Kbd]. **Shift+F6**

Android Studio Skill
Display Override Methods

- To display list of methods you can override in the base class

- Option 1 [Menu]. **Code, Override Methods**

- Option 2 [Kbd]. **Control+o**

- **NOTE: You can also “just start typing the name”**

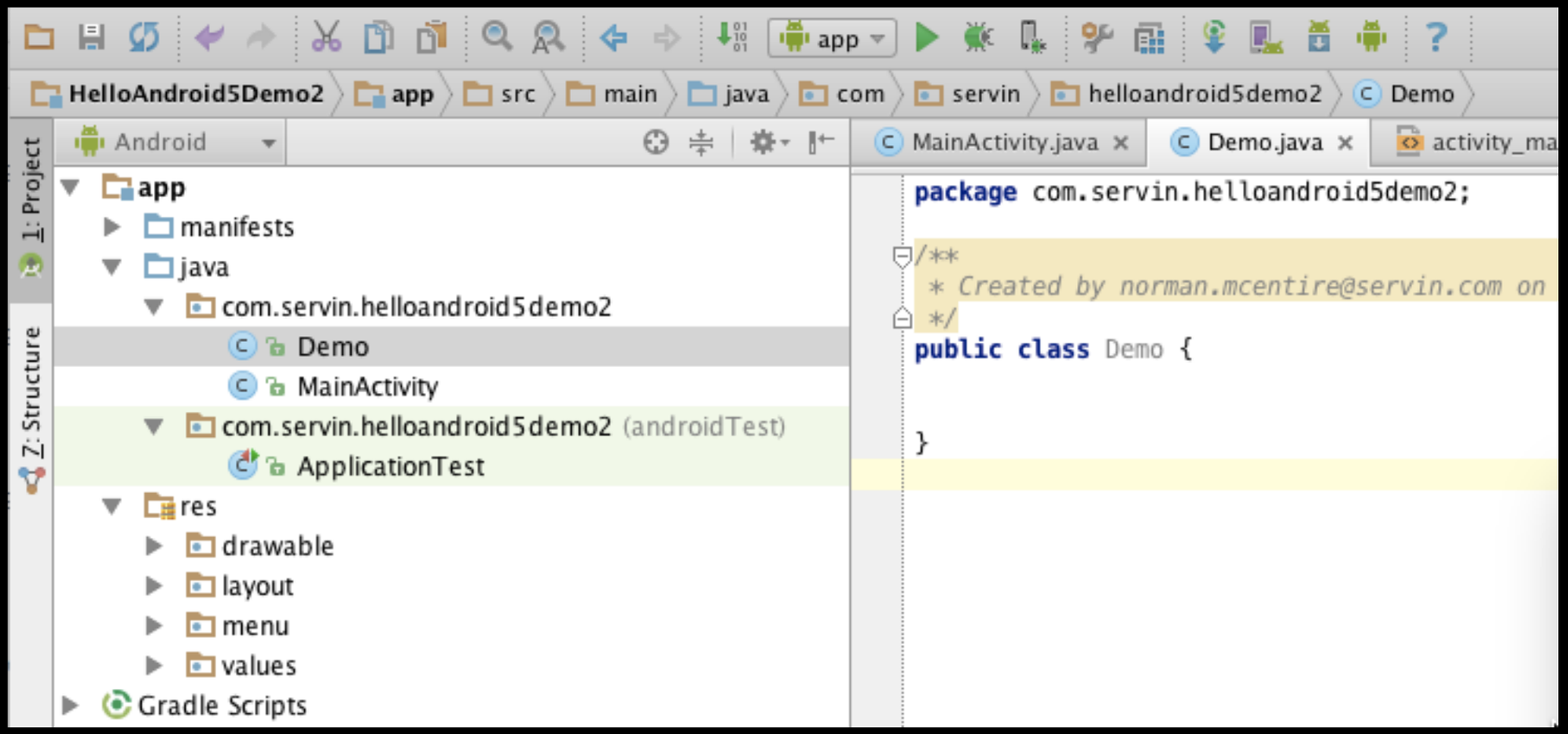
Android Studio Skill
Display Methods of Interface

- To display list of methods you can implement in an interface
- Option 1 [Menu]. **Code, Implement Methods**
- Option 2 [Kbd]. **Control+i**

Android Studio Skill
Add New File To Project

- To add a new file to your project
  - Step 1. Click on “folder” where you want to create the file (e.g. java, res, etc)
- Option 1 [Menu]. **File, New**
- Option 2 [MacKbd]. **Command+n**
- Option 2 [WinKbd]. **Alt+Insert**

Results of Adding Demo Class



Android Studio Skill
Generate Code

- To generate code for a class (e.g. constructor, getter/setter, etc.)
    - Step 1. Click on location where you want the code generated
- Option 1 [Menu]. **Code, Generate**
- Option 2 [MacKbd]. **Command+n**
- Option 2 [WinKbd]. **Alt+Insert**

Results of Code Generation

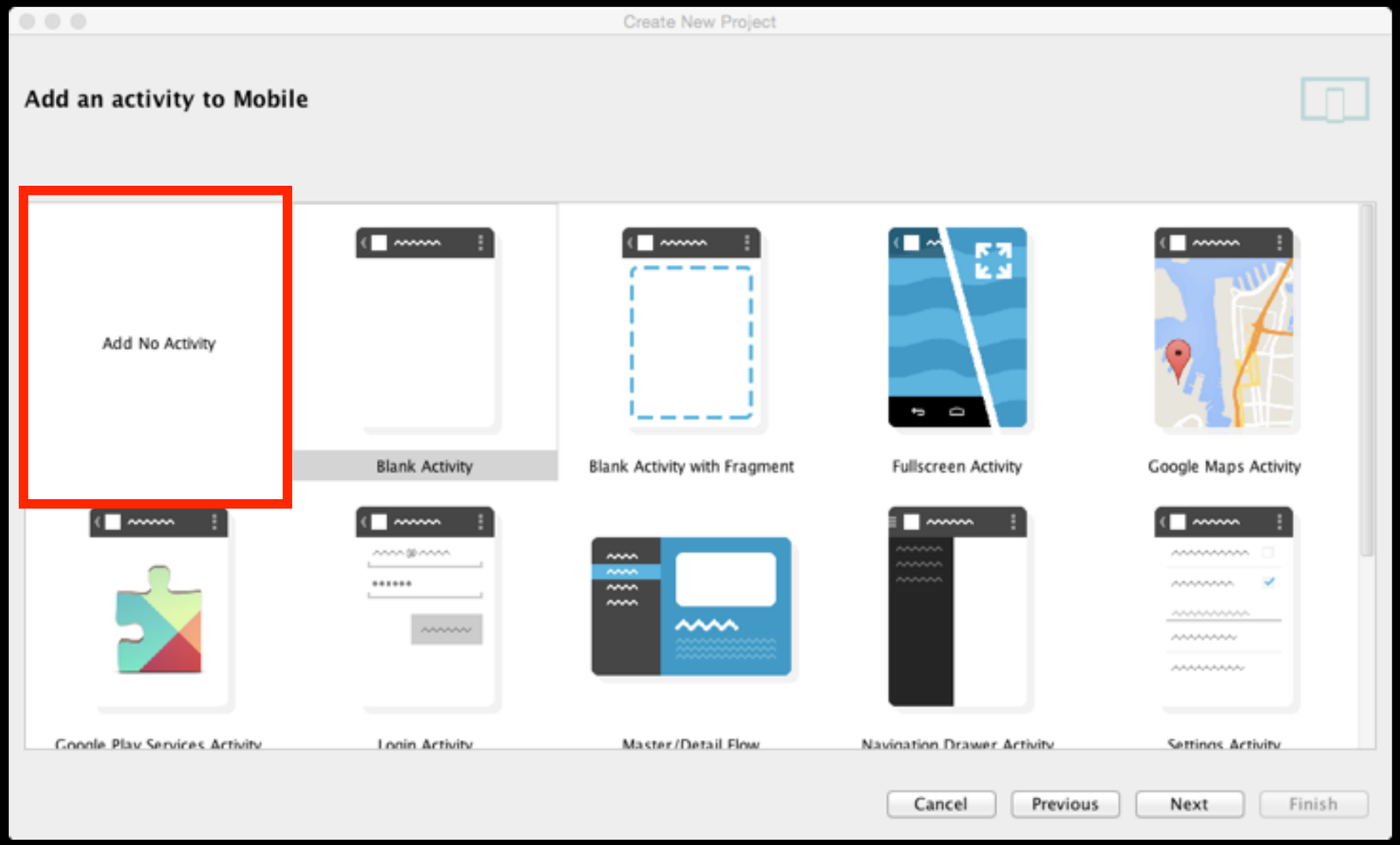
```java
/**
 * Created by norman.mcentire@servin.com on 1/16/15.
 */
public class Demo {

    private String name;
    private int number;

    public Demo() {
    }

    public Demo(int number, String name) {
        this.number = number;
        this.name = name;
    }

    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }
}
```

Using Android Studio

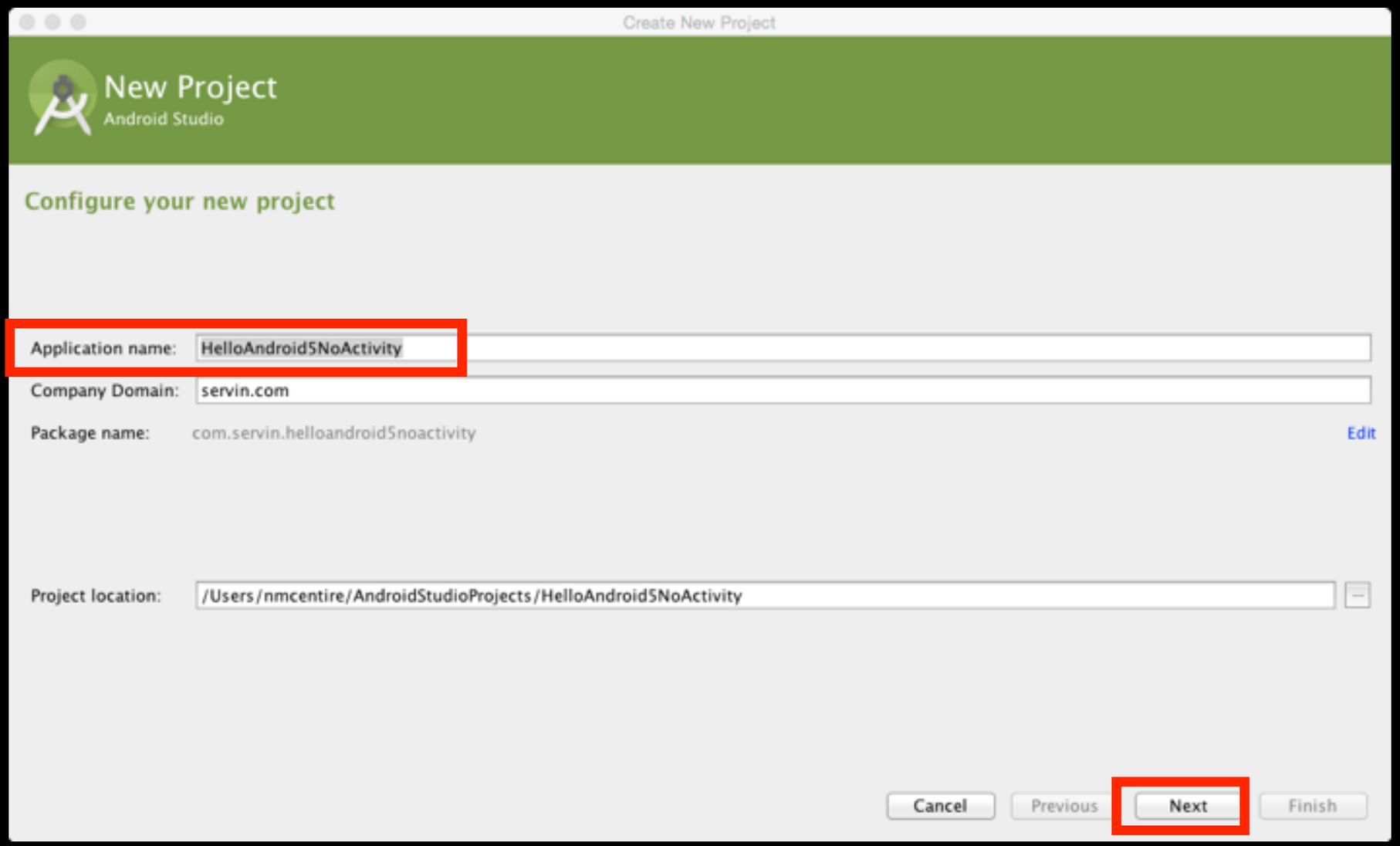
Templates

No Activity

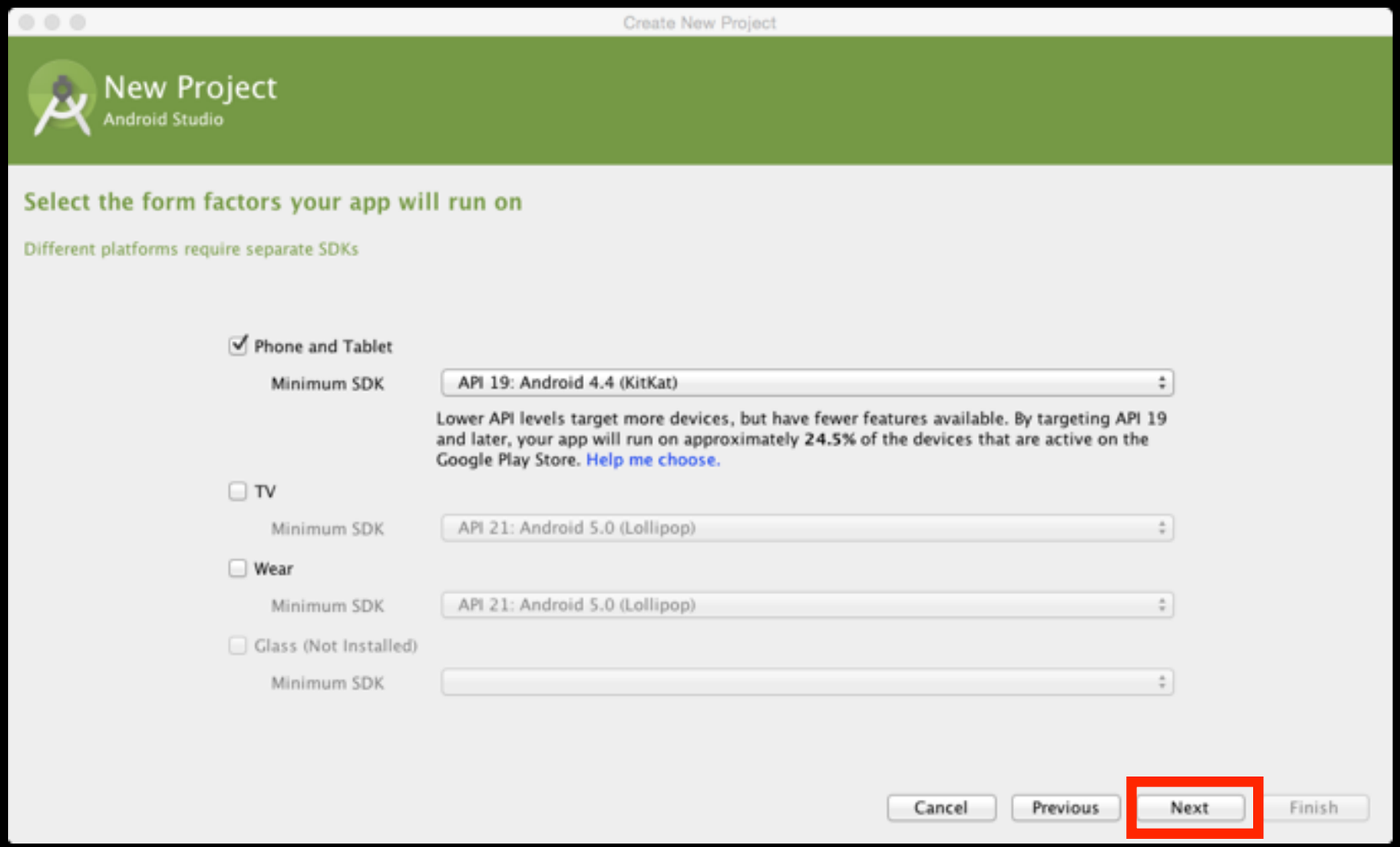
No Activity



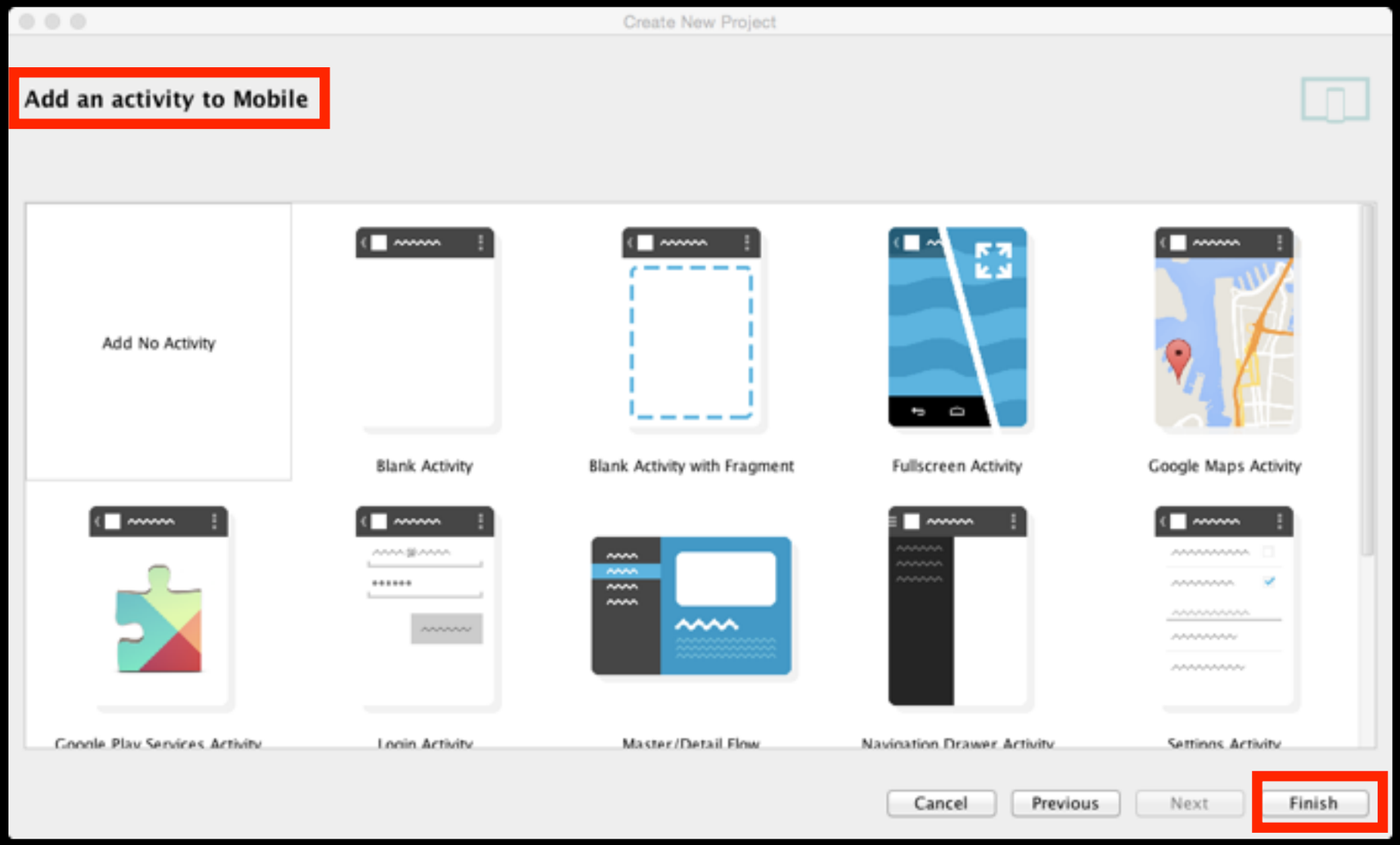
New Project



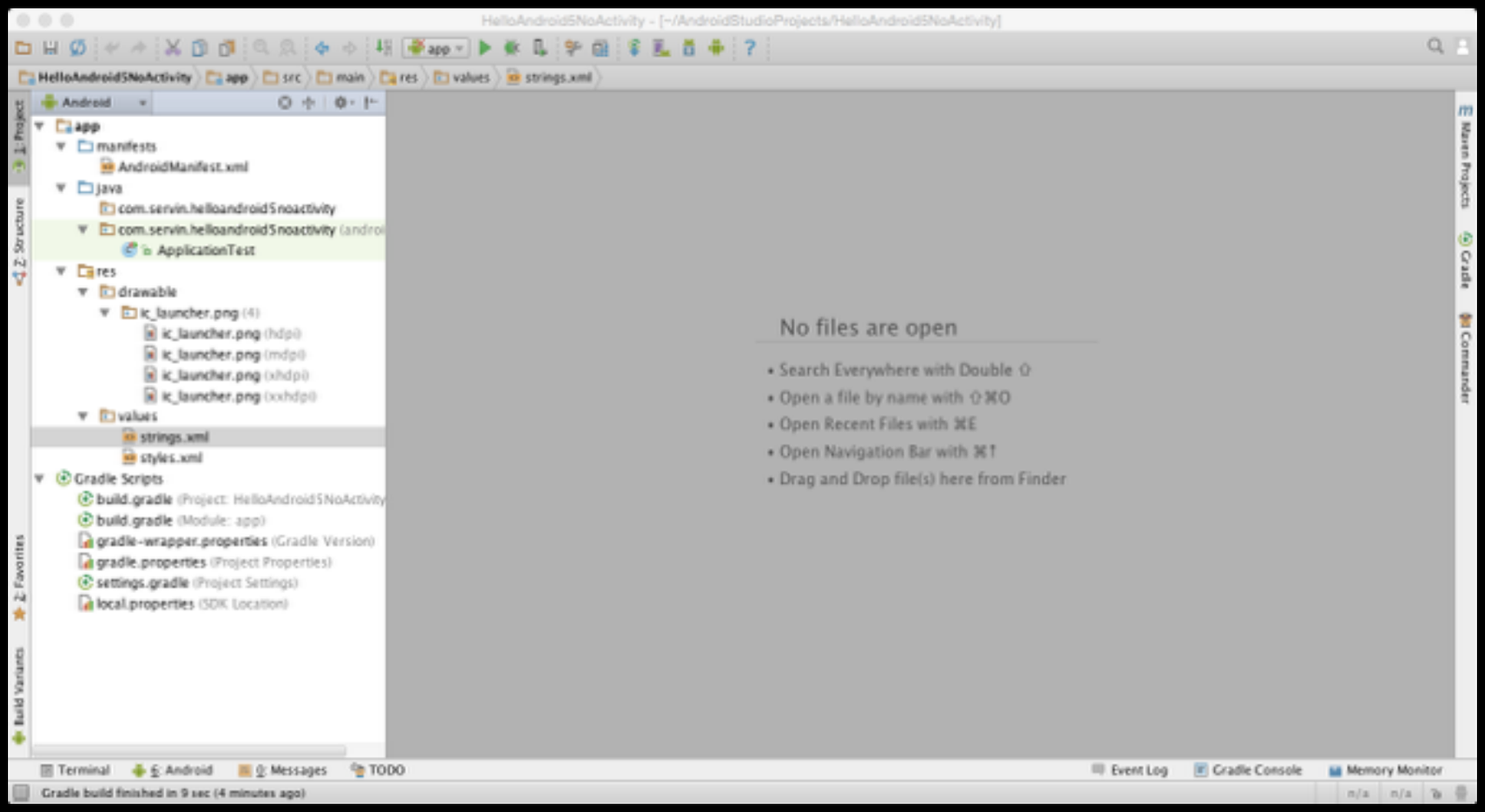
Select Form Factor



(Optional)
Add Activity



Result



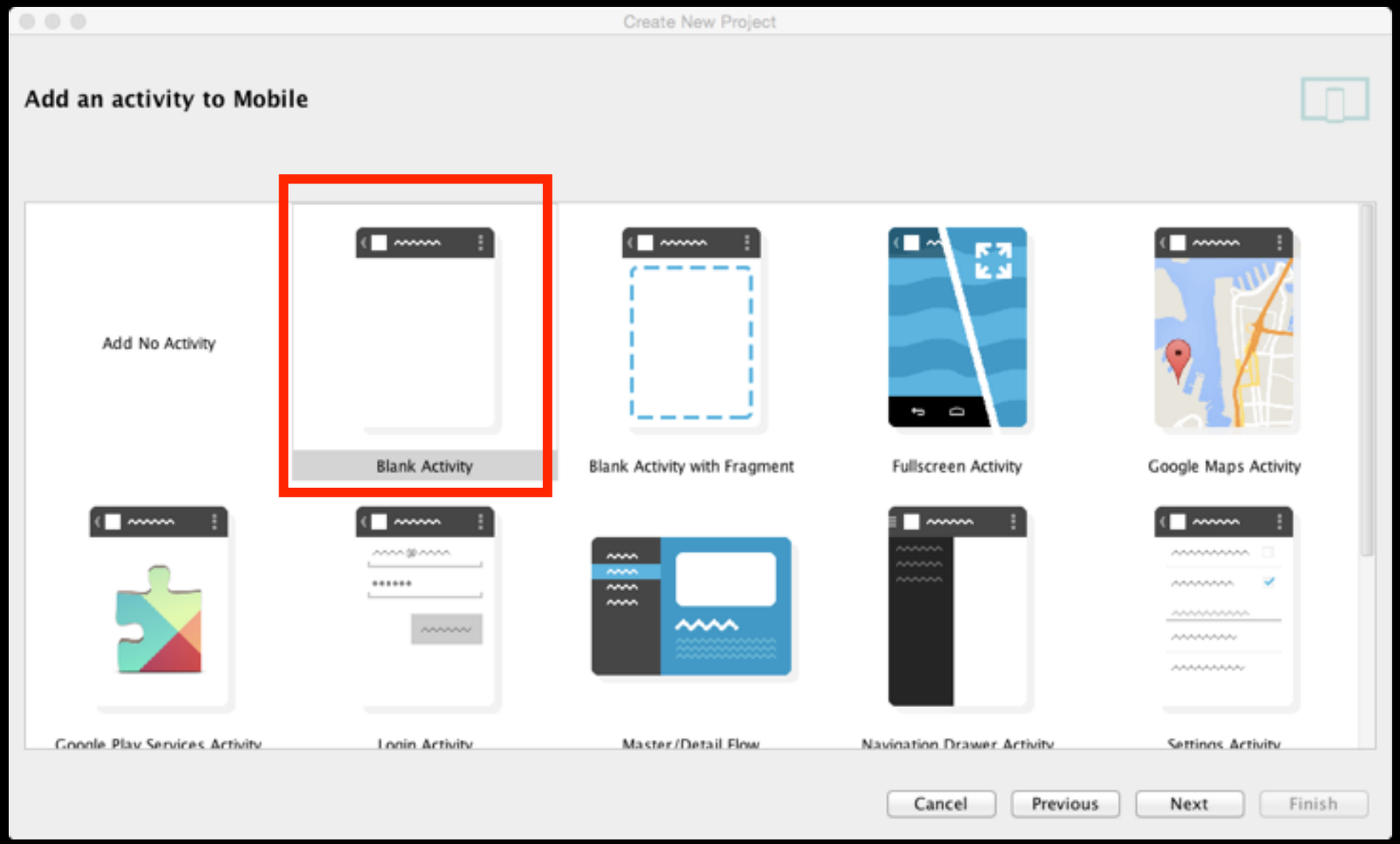
AndroidManifest.xml

HelloAndroid5NoActivity × | app × | AndroidManifest.xml × | gradle-wrapper.properties × | gradle.properti

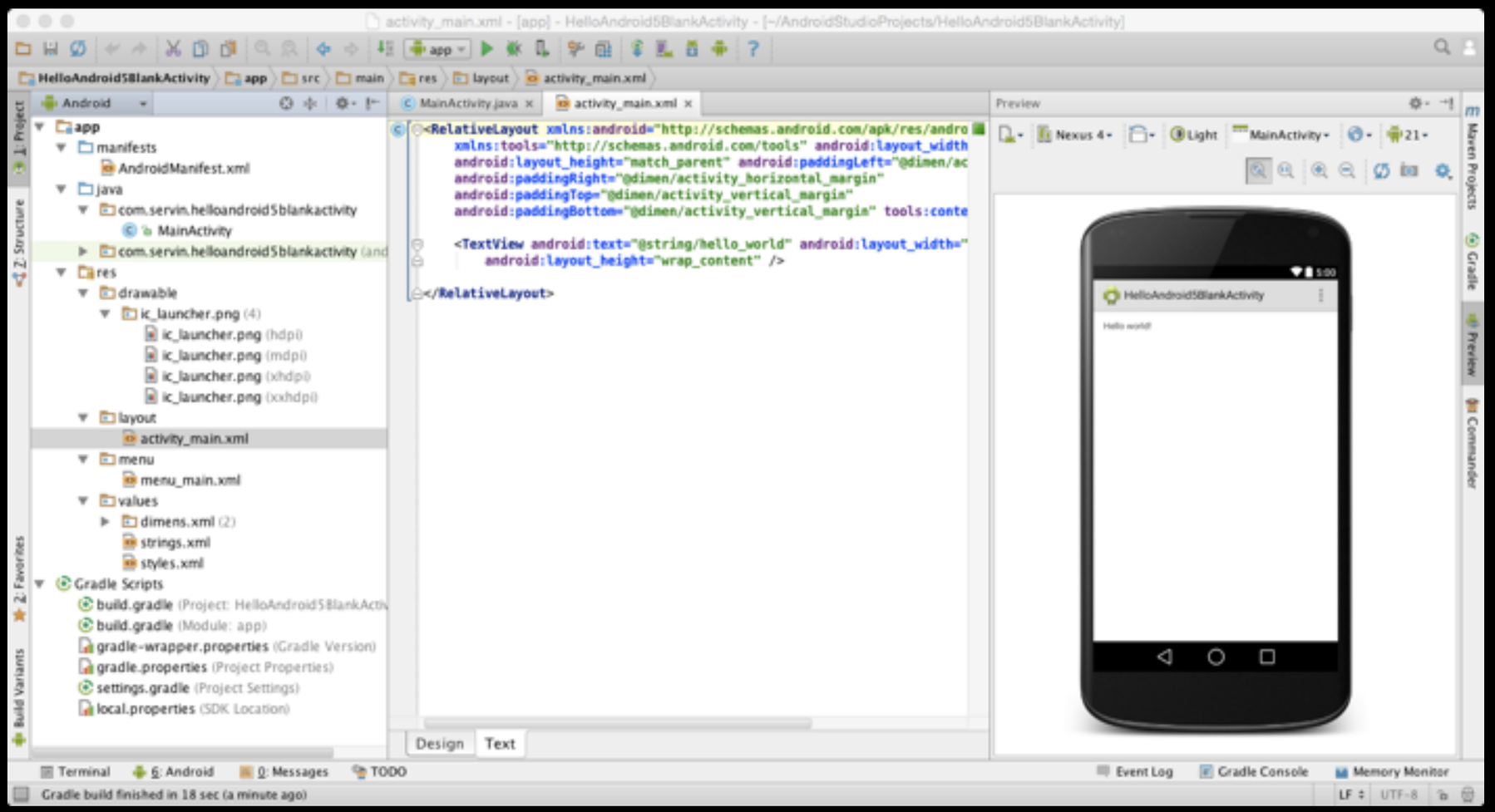
```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.servin.helloandroid5noactivity">

    <application android:allowBackup="true" android:label="HelloAndroid5NoActivity"
        android:icon="@drawable/ic_launcher" android:theme="@style/AppTheme">

    </application>

</manifest>
```

Blank Activity

Blank Activity



Blank Activity



AndroidManifest.xml

MainActivity.java | activity_main.xml | AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.servin.helloandroid5blankactivity" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="HelloAndroid5BlankActivity"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="HelloAndroid5BlankActivity" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java × | activity_main.xml × | AndroidManifest.xml ×

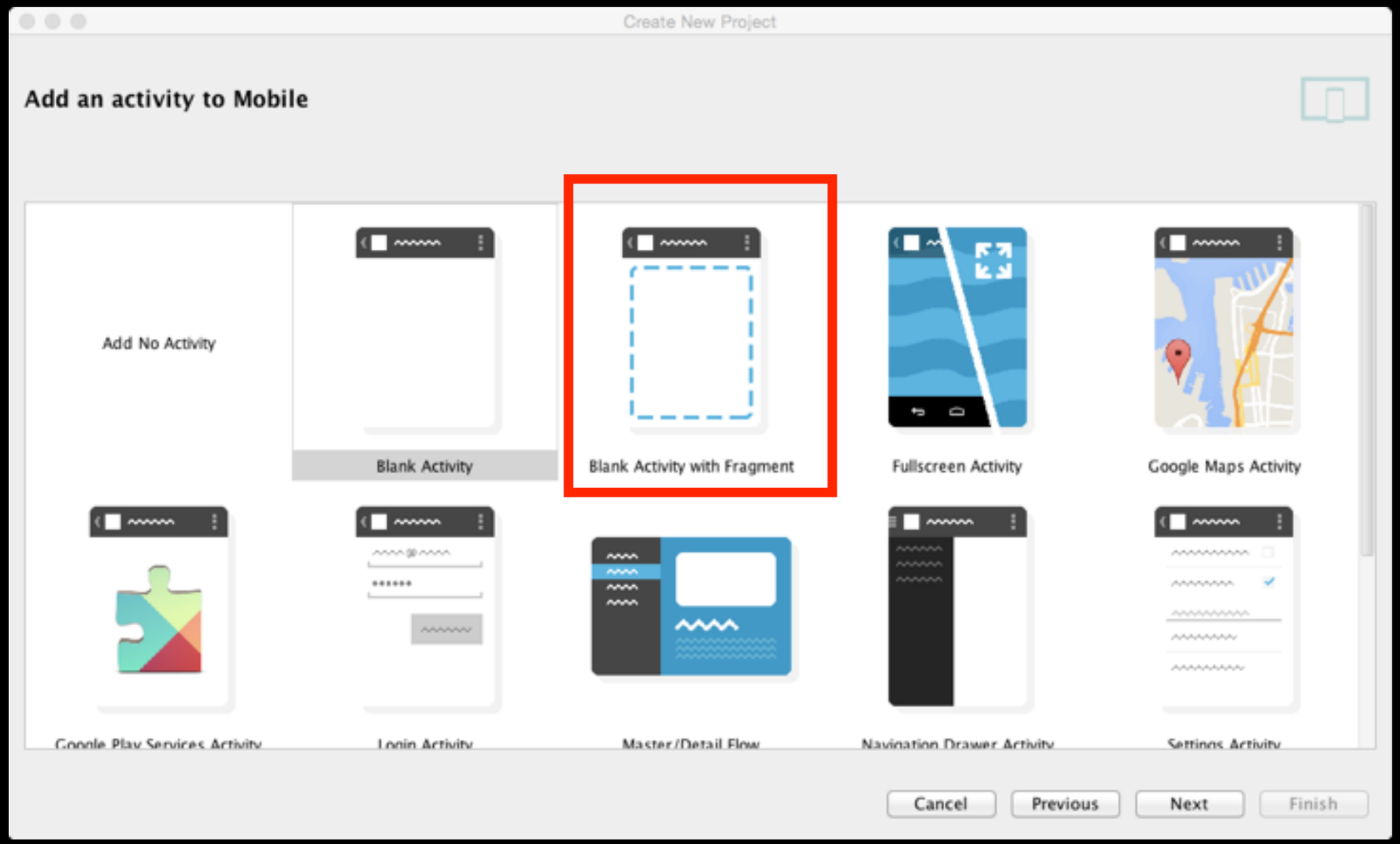
```java
package com.servin.helloandroid5blankactivity;

import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;


public class MainActivity extends ActionBarActivity {

    @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.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
}
```

MainActivity.java | activity_main.xml | AndroidManifest.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" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin" tools:context=".MainActivity">

    <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Blank Activity with Fragment

Blank Activity



AndroidManifest.xml

MainActivity.java | fragment_main.xml | activity_main.xml | AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.servin.helloandroid5blankactivitywithfragment" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="HelloAndroid5BlankActivityWithFragment"
        android:theme="@style/AppTheme" >
        <activity
            android:name=".MainActivity"
            android:label="HelloAndroid5BlankActivityWithFragment" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

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

</manifest>
```

MainActivity.java - 1

MainActivity.java × | fragment_main.xml × | activity_main.xml × | AndroidManifest.xml ×

```java
public class MainActivity extends ActionBarActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        if (savedInstanceState == null) {
            getSupportFragmentManager().beginTransaction()
                    .add(R.id.container, new PlaceholderFragment())
                    .commit();
        }
    }


    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is present.
        getMenuInflater().inflate(R.menu.menu_main, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
        // Handle action bar item clicks here. The action bar will
        // automatically handle clicks on the Home/Up button, so long
        // as you specify a parent activity in AndroidManifest.xml.
        int id = item.getItemId();

        //noinspection SimplifiableIfStatement
        if (id == R.id.action_settings) {
            return true;
        }

        return super.onOptionsItemSelected(item);
    }
```

MainActivity.java - 2

```
    /**
     * A placeholder fragment containing a simple view.
     */
    public static class PlaceholderFragment extends Fragment {

        public PlaceholderFragment() {
        }

        @Override
        public View onCreateView(LayoutInflater inflater, ViewGroup container,
                                 Bundle savedInstanceState) {
            View rootView = inflater.inflate(R.layout.fragment_main, container, false);
            return rootView;
        }
    }
}
```

activity_main.xml

MainActivity.java | fragment_main.xml | activity_main.xml | AndroidManifest.xml

```
<FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:id="@+id/container"
    android:layout_width="match_parent" android:layout_height="match_parent"
    tools:context=".MainActivity" tools:ignore="MergeRootFrame" />
```

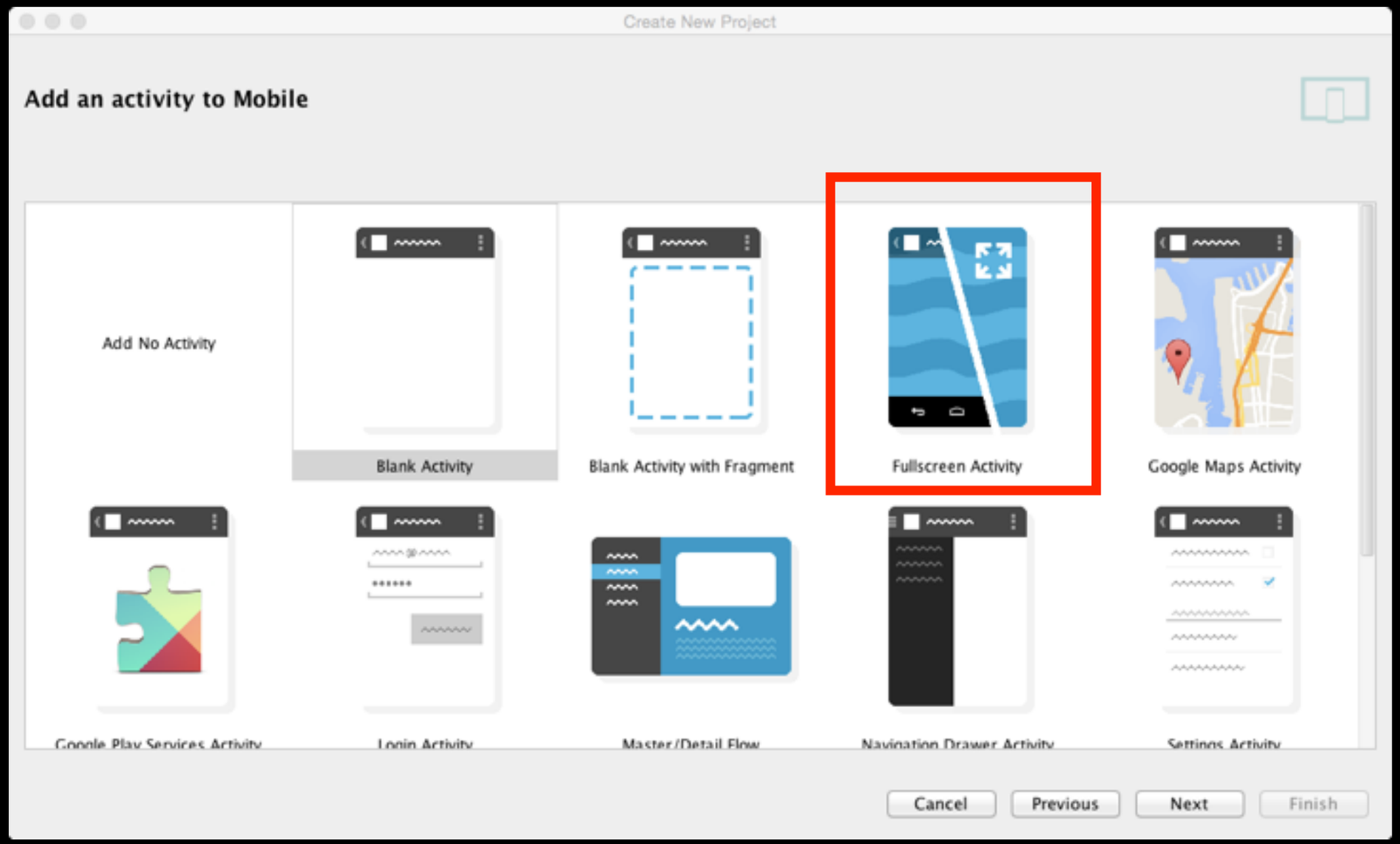
fragment_activity_main.xml

MainActivity.java | fragment_main.xml | activity_main.xml | AndroidManifest.xml

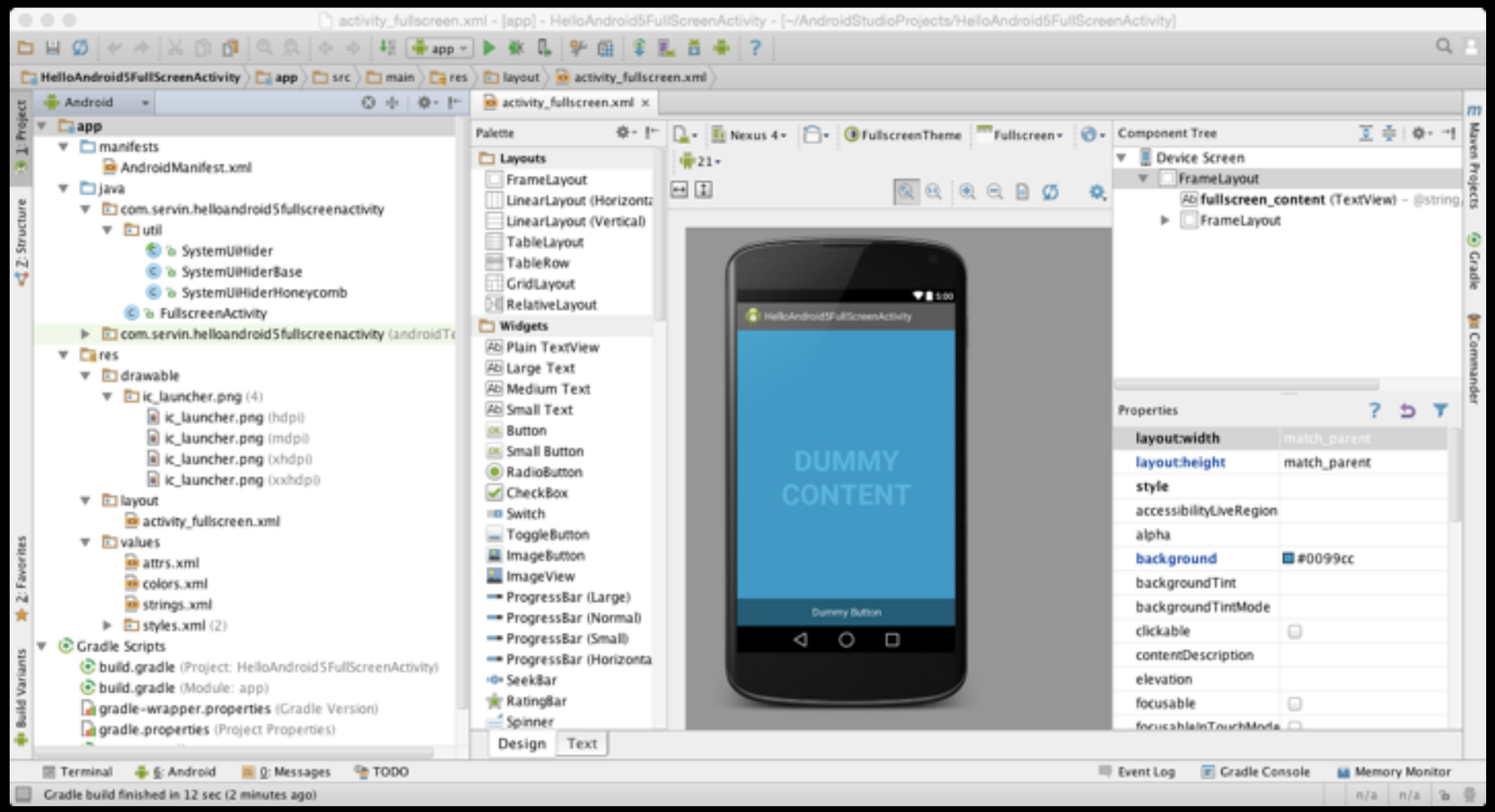
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent"
    android:layout_height="match_parent" android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    android:paddingBottom="@dimen/activity_vertical_margin"
    tools:context=".MainActivity$PlaceholderFragment">

    <TextView android:text="@string/hello_world" android:layout_width="wrap_content"
        android:layout_height="wrap_content" />

</RelativeLayout>
```

Fullscreen Activity

Fullscreen Activity

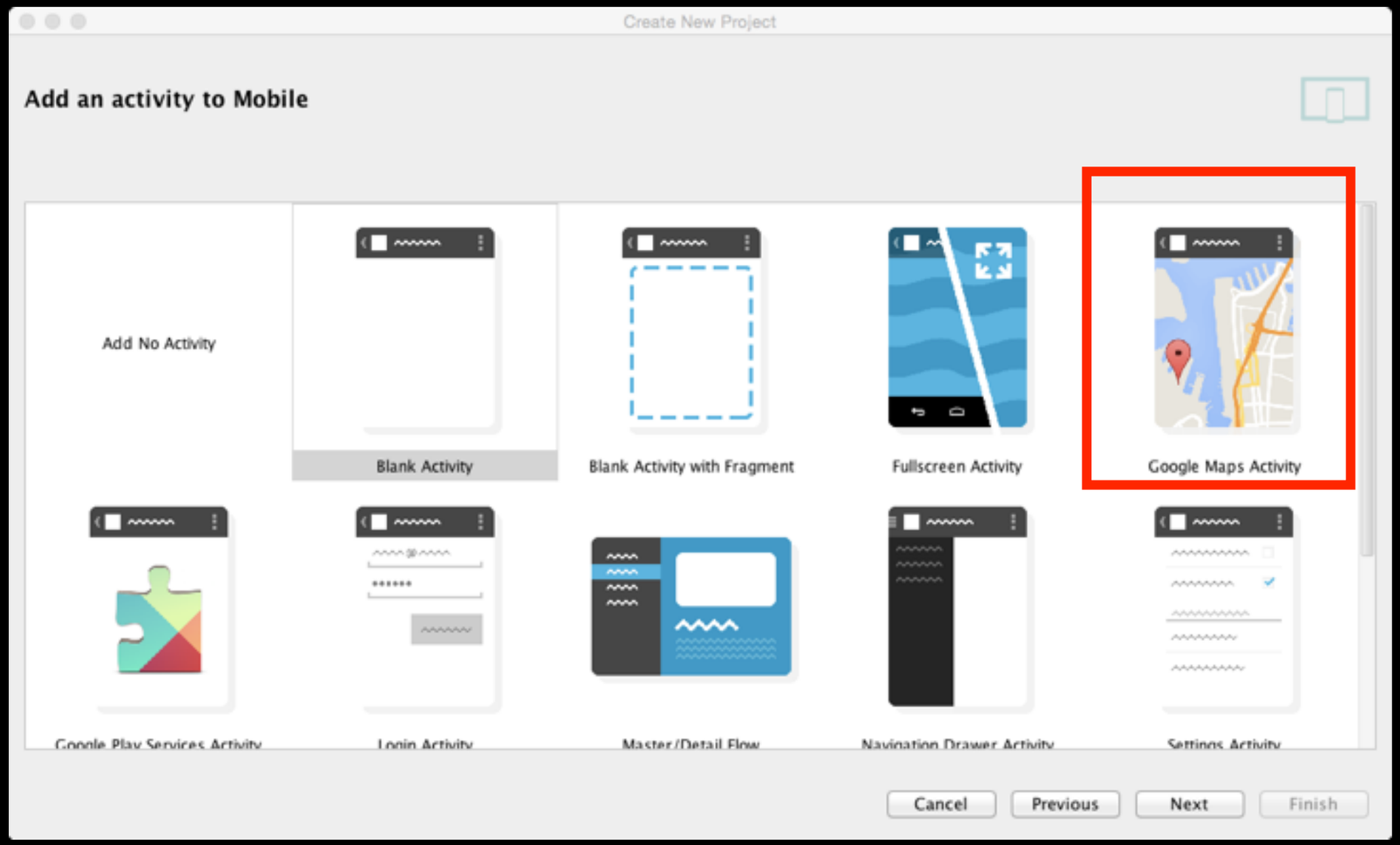


Result



Google Maps Activity

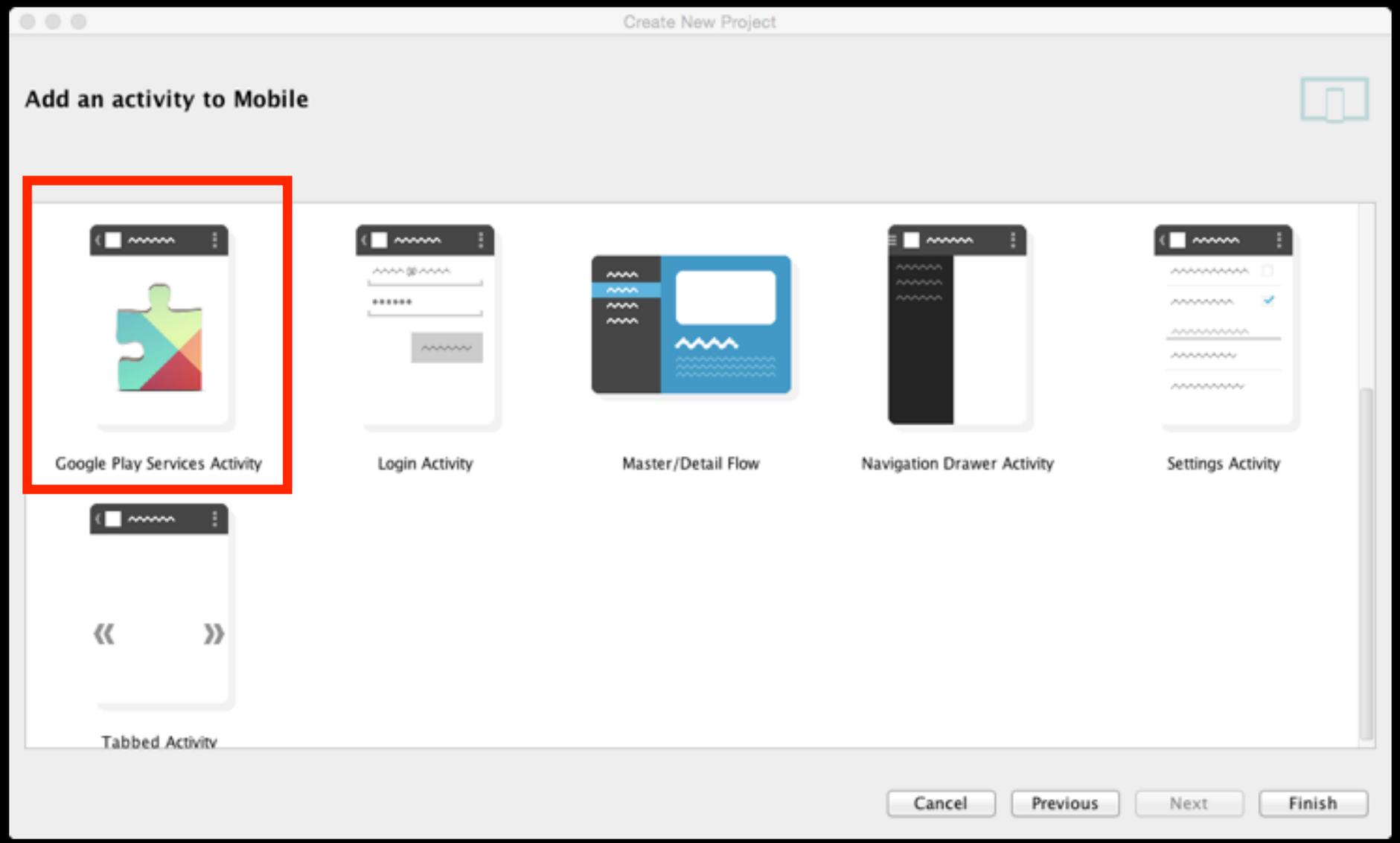
Google Maps Activity



Google Play Services

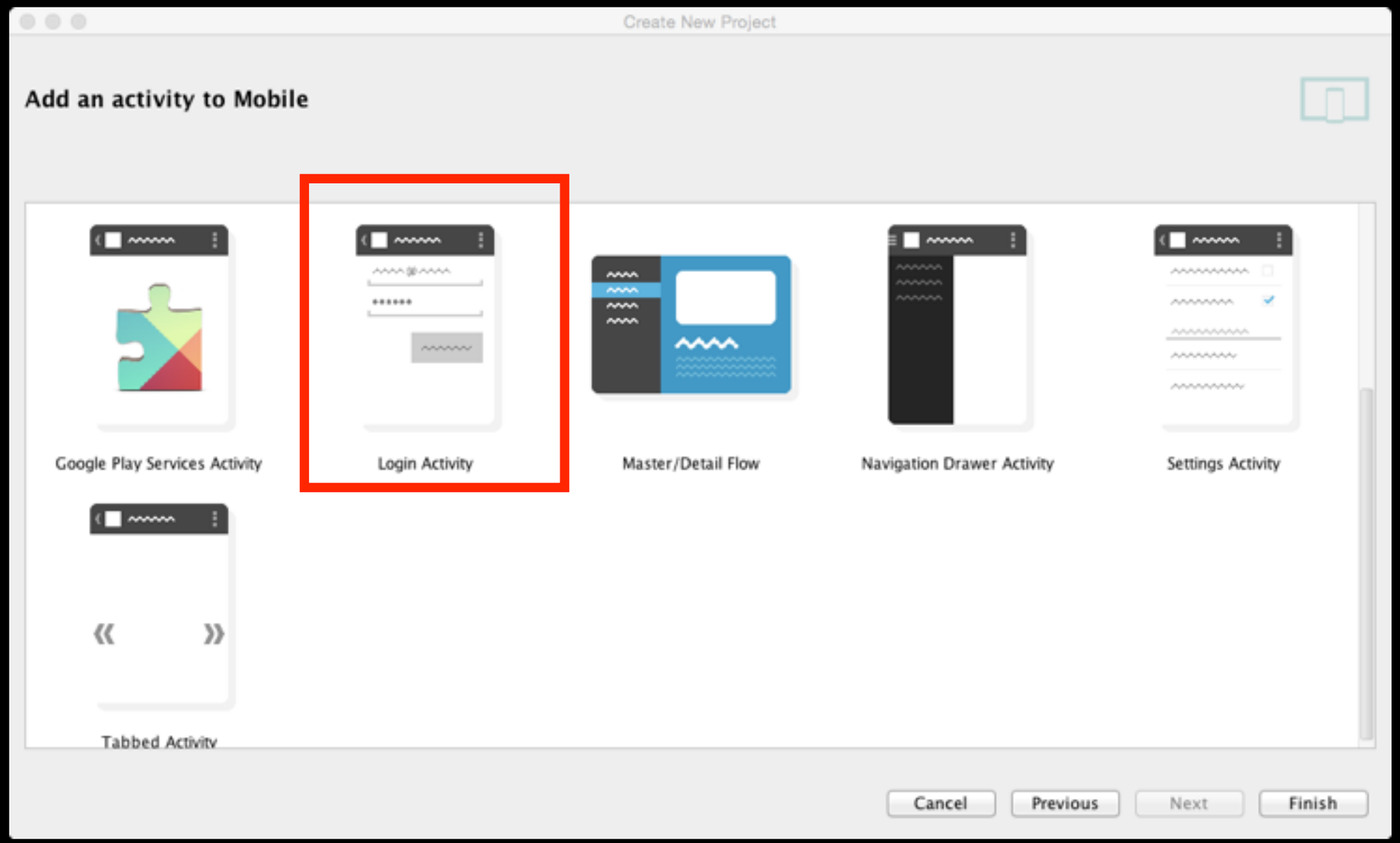
Activity

Google Play Services Activity



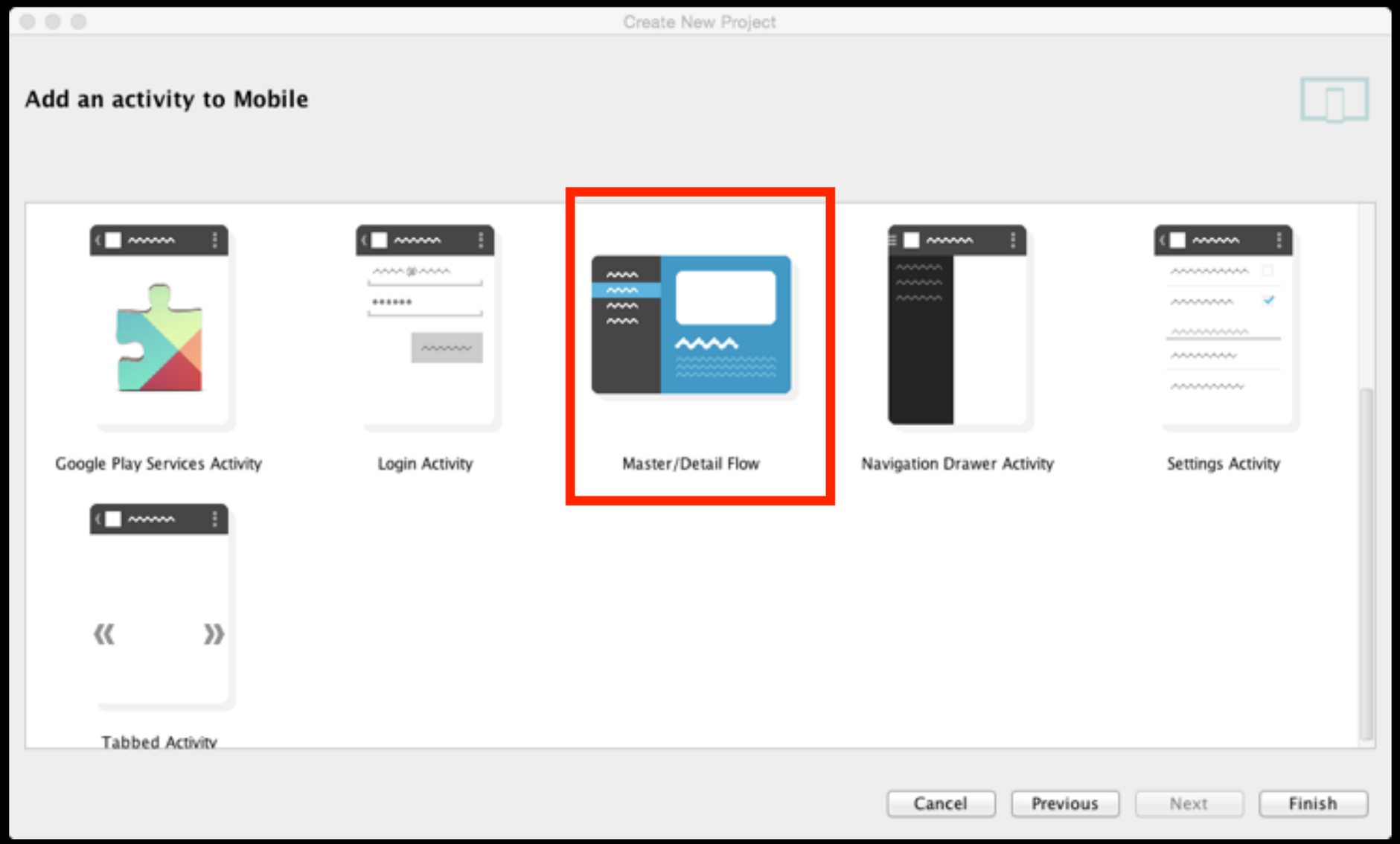
Login Activity

Login Activity



Master/Detail Flow

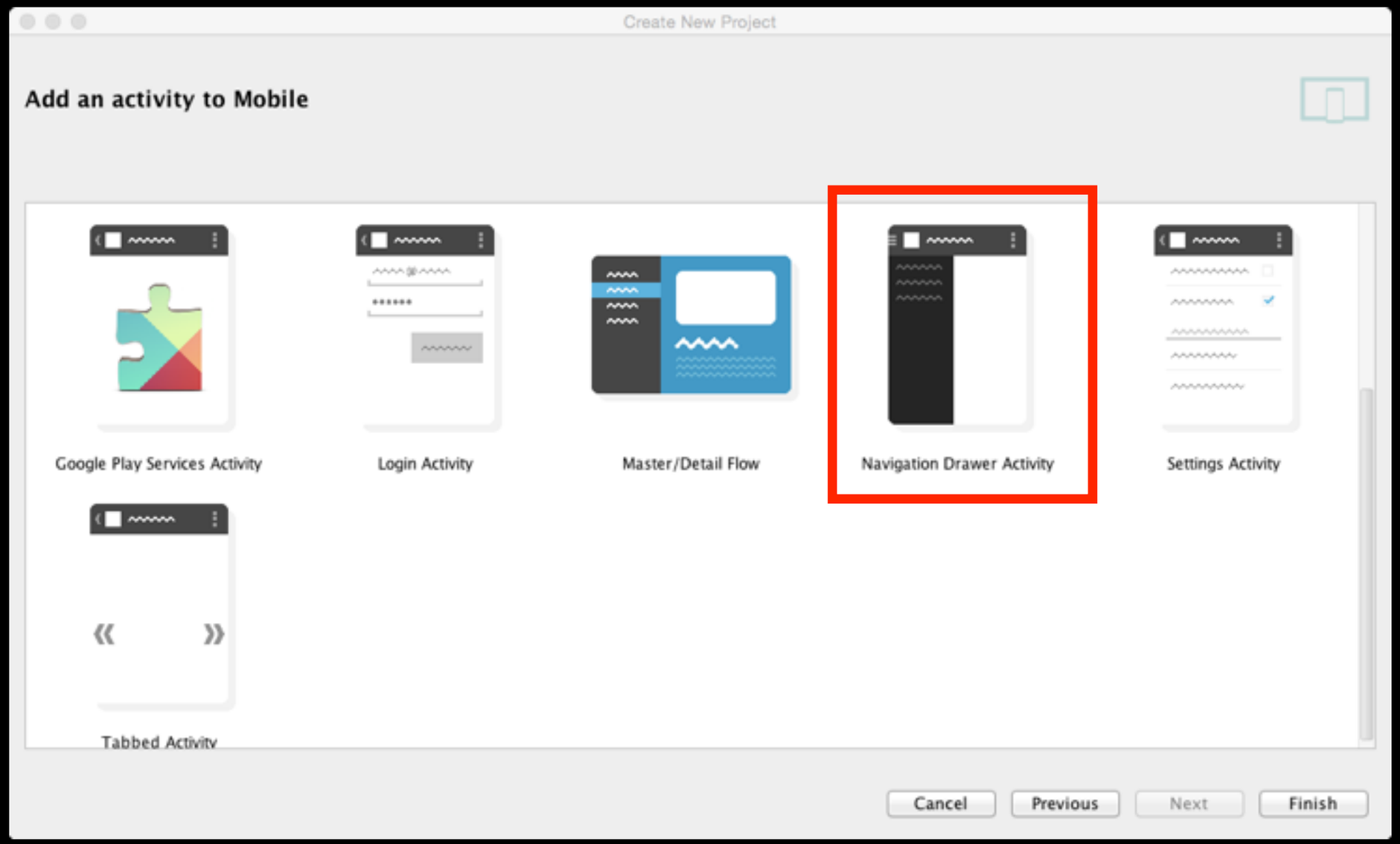
Master/Detail Flow



Navigation Draw

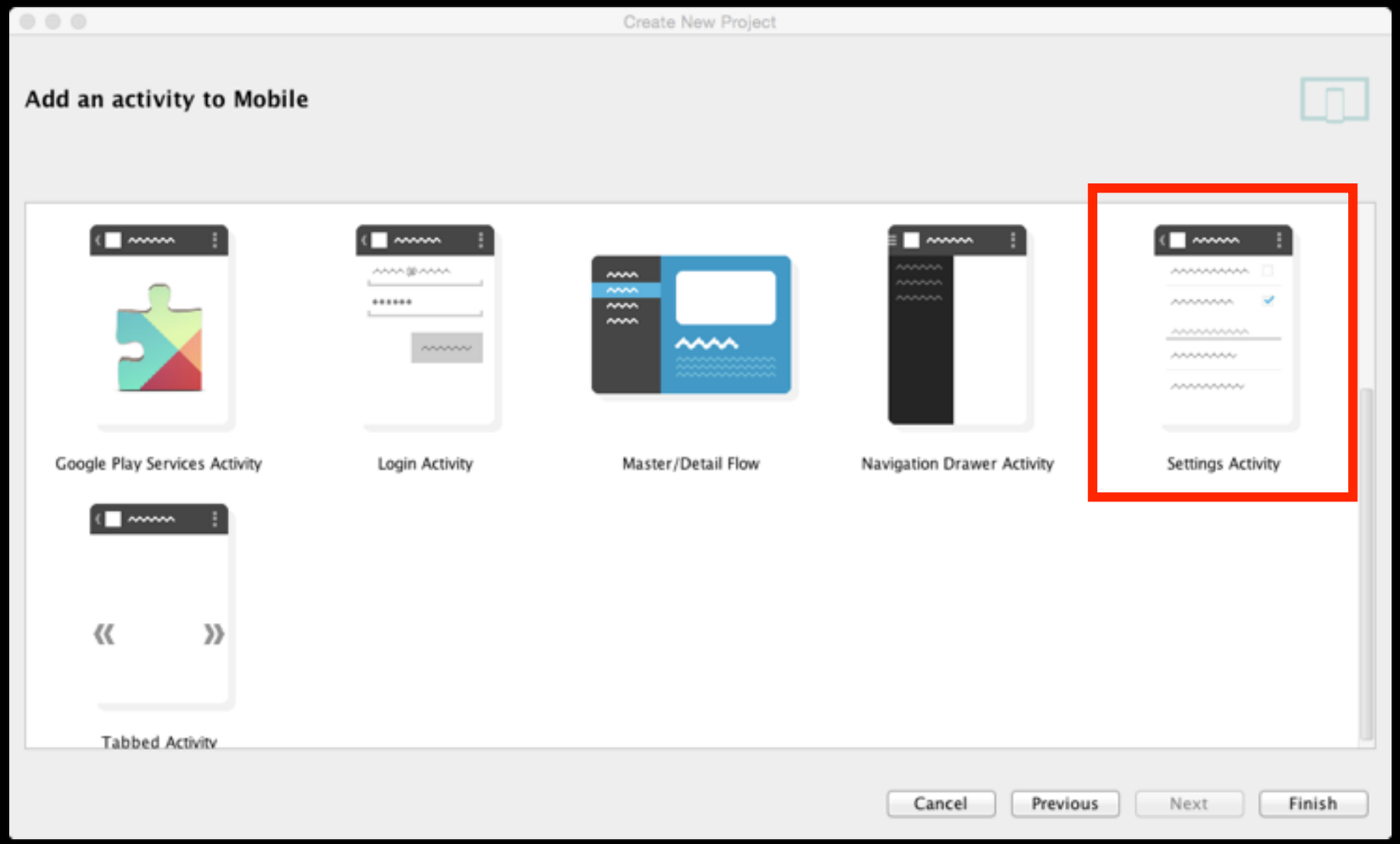
Activity

Navigation Draw Activity



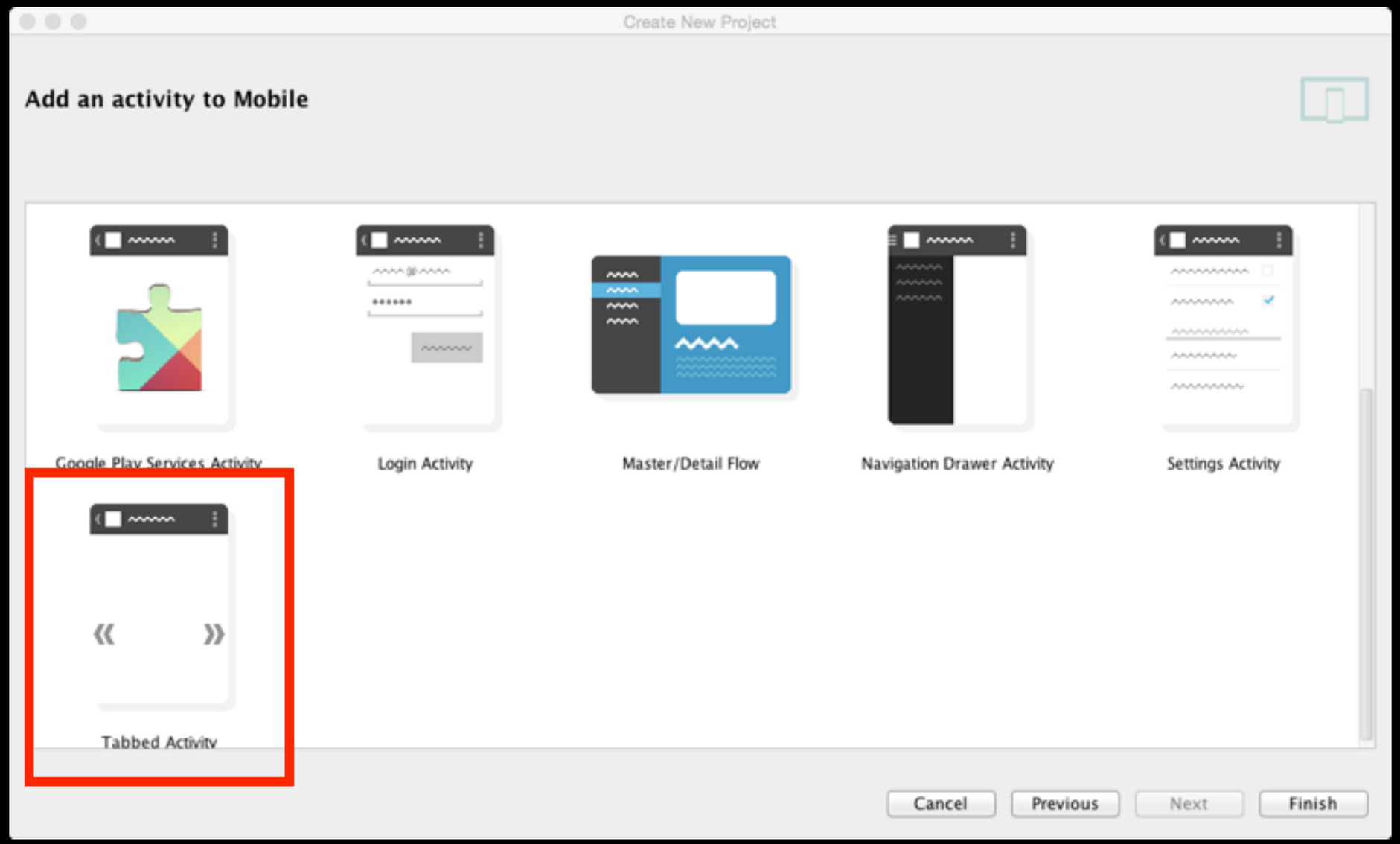
Settings Activity

Settings Activity



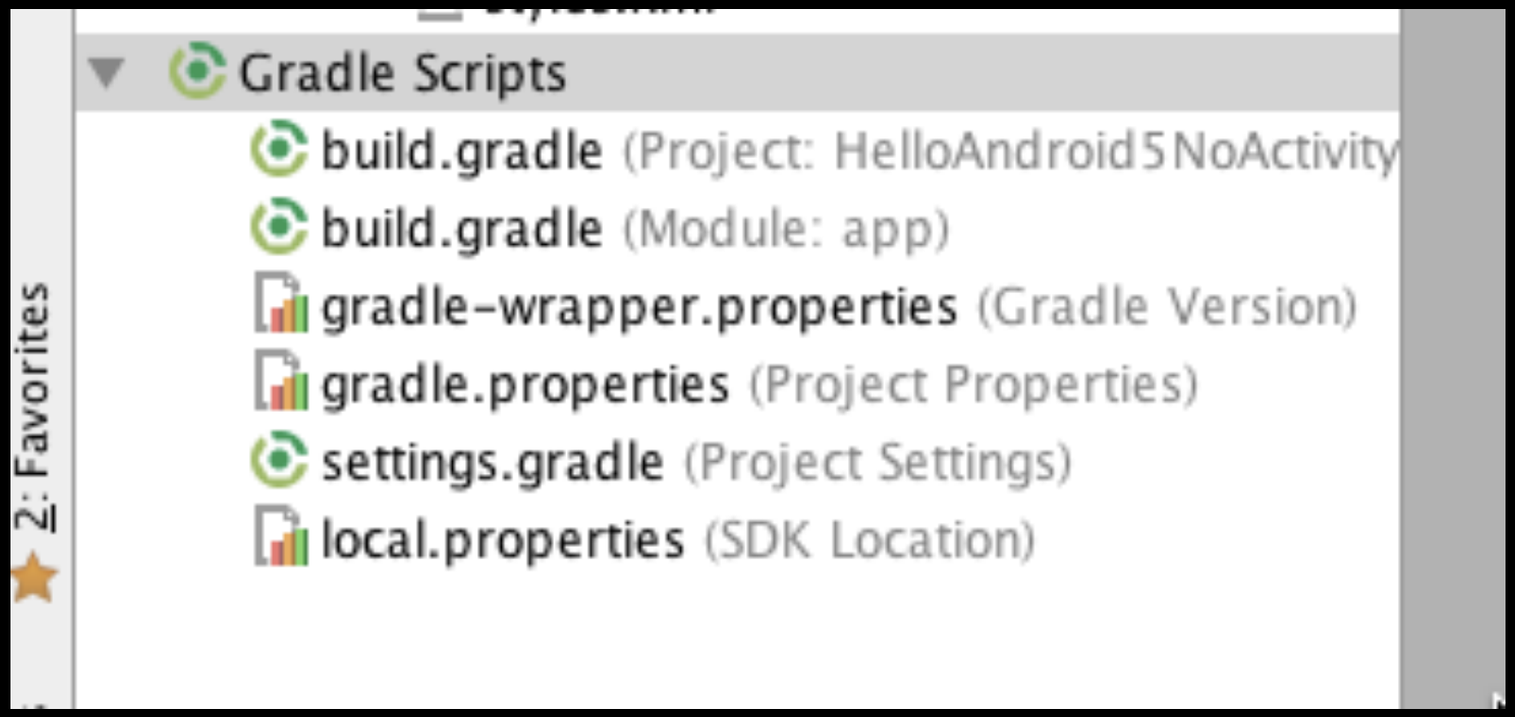
Tabbed Activity

Tabbed Activity



Gradle Build Scripts

Gradle Build Scripts



build.grade (Project)

HelloAndroid5NoActivity

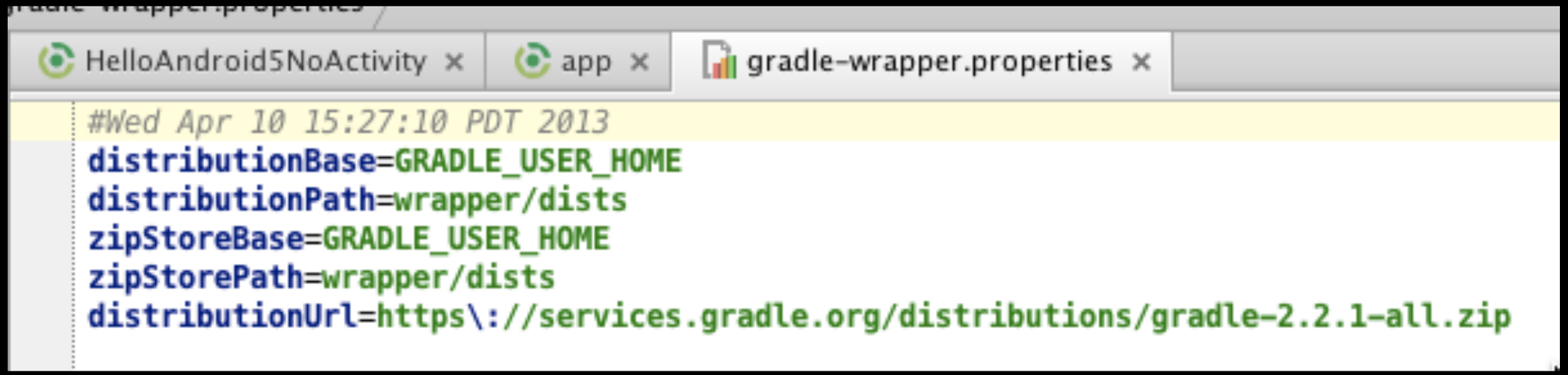
```
// Top-level build file where you can add configuration options common to all sub-projects/modules.

buildscript {
    repositories {
        jcenter()
    }
    dependencies {
        classpath 'com.android.tools.build:gradle:1.0.0'

        // NOTE: Do not place your application dependencies here; they belong
        // in the individual module build.gradle files
    }
}

allprojects {
    repositories {
        jcenter()
    }
}
```

build.grade
(app)

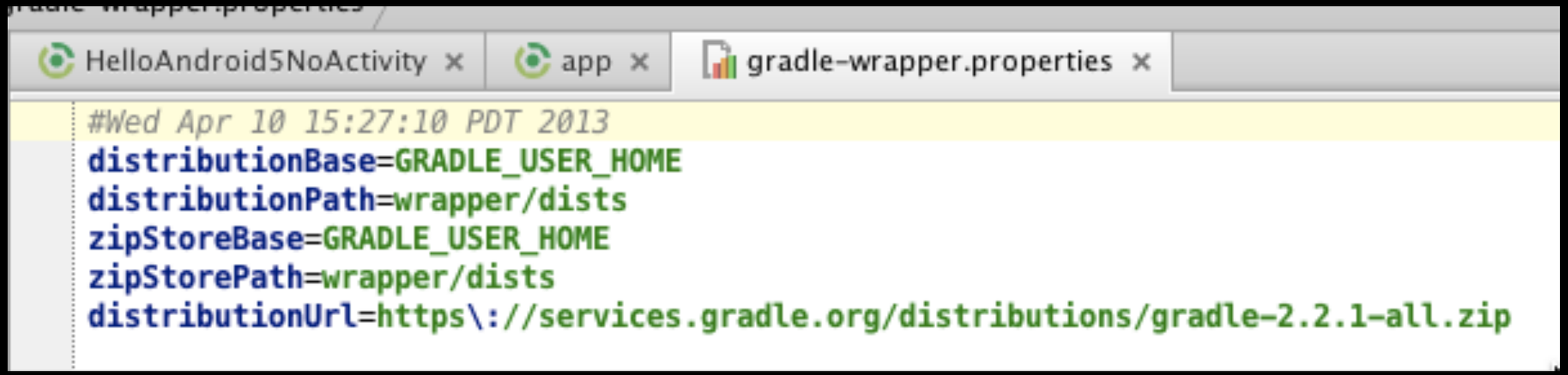
HelloAndroid5NoActivity × app ×

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5noactivity"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

gradle-wrapper.properties



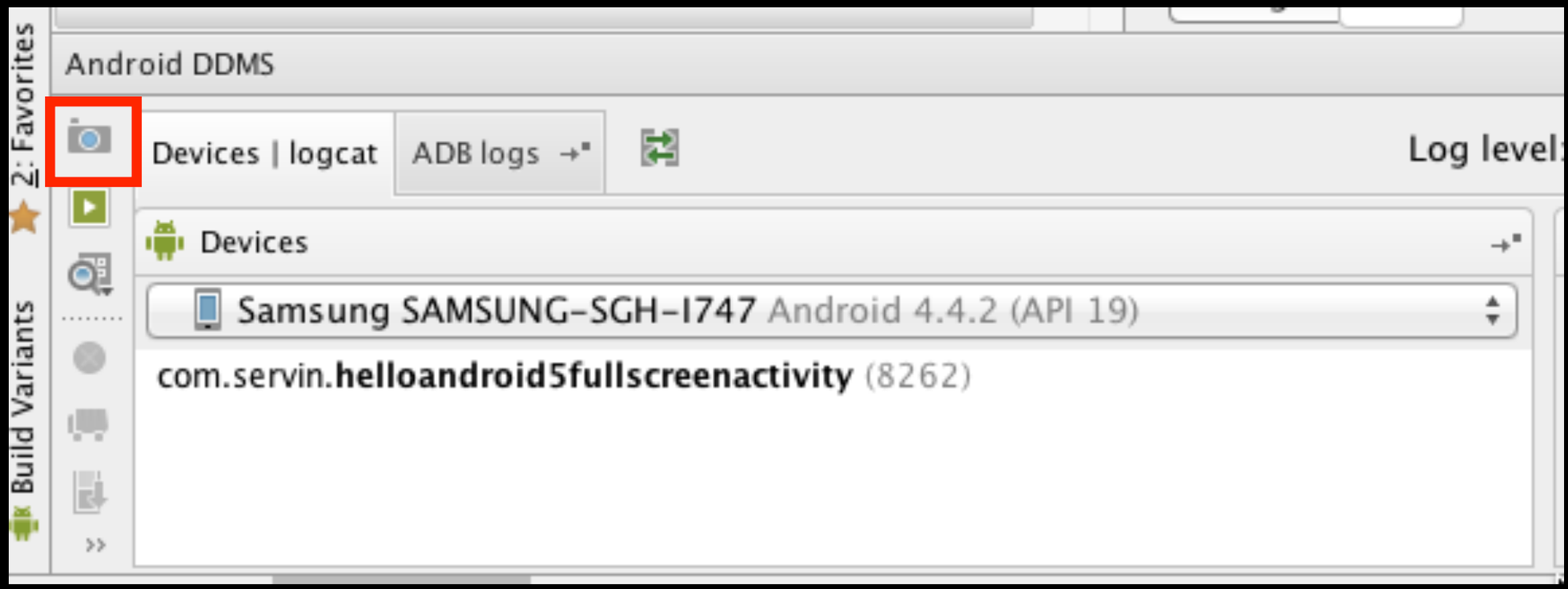
gradle-wrapper.properties



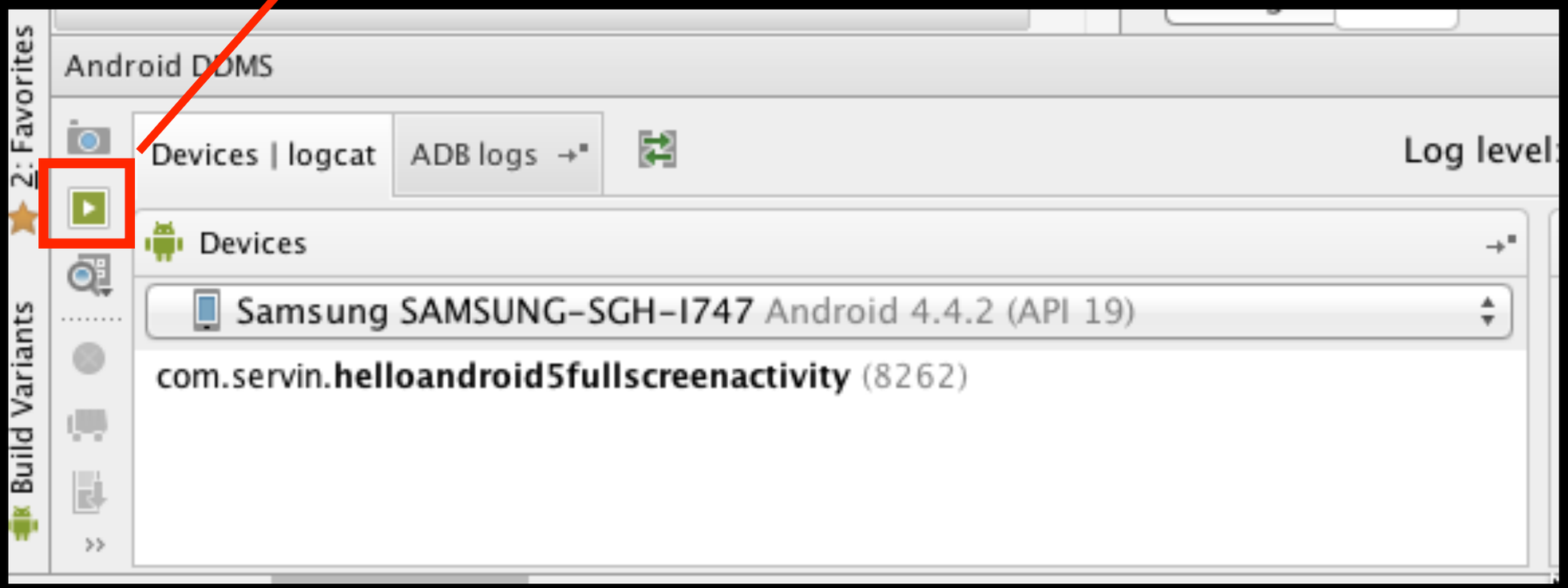
Android Studio

DDMS Options

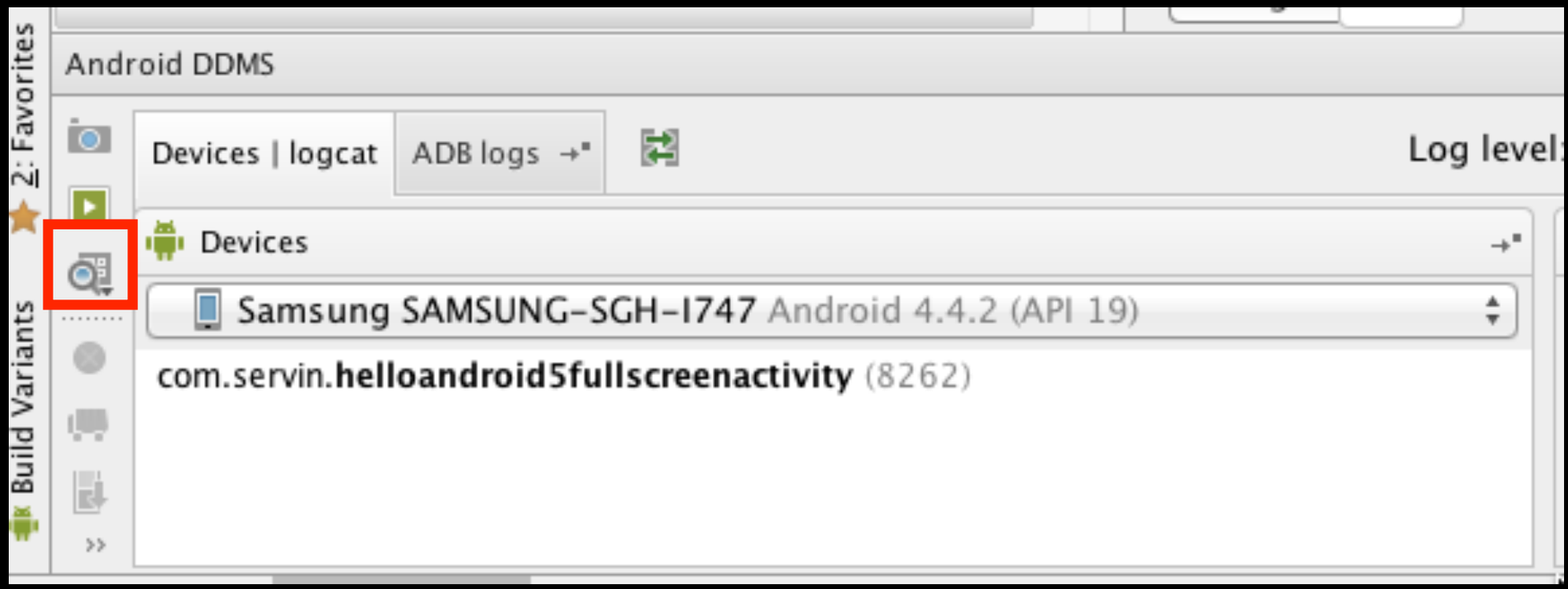
Screen Capture



Screen Recorder



System Information



System Information
Activity Manager State

activity_fullscreen.xml × | dumpsys-8557422377698455889.txt × | AndroidManifest.xml × | Sy

```
ACTIVITY MANAGER PENDING INTENTS (dumpsys activity intents)
  * PendingIntentRecord{42cc6ee0 com.sec.spp.push broadcastIntent}
  * PendingIntentRecord{42ea2f08 com.yellowpages.android.ypmobile broadcastIntent}
  * PendingIntentRecord{4225f830 android broadcastIntent}
  * PendingIntentRecord{42c2a6e8 com.google.android.gms startService}
  * PendingIntentRecord{42d60ec0 com.sec.knox.seandroid broadcastIntent}
  * PendingIntentRecord{42c7dfd0 com.sec.spp.push broadcastIntent}
  * PendingIntentRecord{42be1050 com.sec.knox.seandroid broadcastIntent}
  * PendingIntentRecord{42d45bb0 com.sec.android.app.music startService}
  * PendingIntentRecord{42dbdb28 com.google.android.gms broadcastIntent}
  * PendingIntentRecord{42d84268 com.google.android.gms broadcastIntent}
  * PendingIntentRecord{42aa8498 android broadcastIntent}
  * PendingIntentRecord{42de5688 com.google.android.gms broadcastIntent}
  * PendingIntentRecord{42407850 com.google.android.gms startService}
  * PendingIntentRecord{429f3928 com.wssyncmldm broadcastIntent}
  * PendingIntentRecord{42e97a10 com.yellowpages.android.ypmobile broadcastIntent}
  * PendingIntentRecord{42ce73c0 com.sec.android.app.music startService}
  * PendingIntentRecord{42e85e28 android broadcastIntent}
  * PendingIntentRecord{42dfd2d0 com.sec.spp.push broadcastIntent}
  * PendingIntentRecord{424456c8 com.yellowpages.android.ypmobile broadcastIntent}
  * PendingIntentRecord{42f1c380 com.sec.spp.push broadcastIntent}
  * PendingIntentRecord{42dae7d8 com.osp.app.signin startActivity}
  * PendingIntentRecord{42d80678 com.google.android.gms broadcastIntent}
  * PendingIntentRecord{42e885f0 com.android.settings startActivity}
  * PendingIntentRecord{42e86fa0 com.android.settings startActivity}
```

System Information
Package Information

activity_fullscreen.xml × | dumpsys-8557422377698455889.txt × | dumpsys-1110280893919688926.txt × | Andro

```
Verifiers:
  Required: com.android.vending (uid=10067)

Libraries:
  sws -> (jar) /system/framework/sws.jar
  sechardware -> (jar) /system/framework/sechardware.jar
  com.android.future.usb.accessory -> (jar) /system/framework/com.android.future.usb.accessory.jar
  allshare -> (jar) /system/framework/allshare.jar
  com.sec.android.mdm -> (jar) /system/framework/sec_edm.jar
  android.test.runner -> (jar) /system/framework/android.test.runner.jar
  com.sec.android.visualeffect -> (jar) /system/framework/com.sec.android.visualeffect.jar
  seccamera -> (jar) /system/framework/seccamera.jar
  videowall -> (jar) /system/framework/videowall.jar
  secvision -> (jar) /system/framework/secvision.jar
  libvtmanagerjar -> (jar) /system/framework/libvtmanagerjar.jar
  sec_platform_library -> (jar) /system/framework/sec_platform_library.jar
  com.google.android.media.effects -> (jar) /system/framework/com.google.android.media.effects.jar
  secmediarecorder -> (jar) /system/framework/secmediarecorder.jar
  com.broadcom.bt -> (jar) /system/framework/com.broadcom.bt.jar
  com.policydm.features -> (jar) /system/framework/com.policydm.features.jar
  com.android.location.provider -> (jar) /system/framework/com.android.location.provider.jar
  com.sec.smartcard.auth -> (jar) /system/framework/secsmartcard.jar
  org.simalliance.openmobileapi -> (jar) /system/framework/org.simalliance.openmobileapi.jar
  qcom.digitalpen.util -> (jar) /system/framework/digitalpenutil.jar
  javax.obex -> (jar) /system/framework/javax.obex.jar
```

System Information
Memory Usage

activity_fullscreen.xml × | dumpsys-4327684967024033230.txt × | AndroidManifest.xml ×

```
Applications Memory Usage (kB):
Uptime: 3239834 Realtime: 4021446

Total PSS by process:
    76793 kB: com.sec.android.app.launcher (pid 1116 / activities)
    48778 kB: system (pid 692)
    47899 kB: com.android.systemui (pid 971)
    33022 kB: com.sec.android.inputmethod (pid 1919)
    27438 kB: com.google.android.apps.plus (pid 3643)
    20363 kB: com.google.android.gms (pid 1619)
    20200 kB: com.google.android.googlequicksearchbox:search (pid 1805)
    19322 kB: com.android.vending (pid 3423)
    15996 kB: com.android.phone (pid 1091)
    15779 kB: com.google.process.location (pid 1406)
    15144 kB: com.google.process.gapps (pid 1593)
    14948 kB: com.sec.android.gallery3d (pid 3916)
    14635 kB: zygote (pid 174)
    13075 kB: com.google.android.apps.magazines (pid 8301)
    12427 kB: com.sec.spp.push (pid 3539)
    11589 kB: com.android.contacts (pid 2096)
     9640 kB: com.sec.android.inputmethod:ACService (pid 1977)
     9441 kB: android.process.media (pid 3526)
     9225 kB: com.samsung.android.MtpApplication (pid 7590)
     8865 kB: mediaserver (pid 176)
     8862 kB: com.google.android.talk (pid 8752)
     8530 kB: com.android.email (pid 7875)
```

System Information
Memory Usage Over Time

activity_fullscreen.xml × | dumpsys-2657125578409594274.txt × | AndroidManifest.xml ×

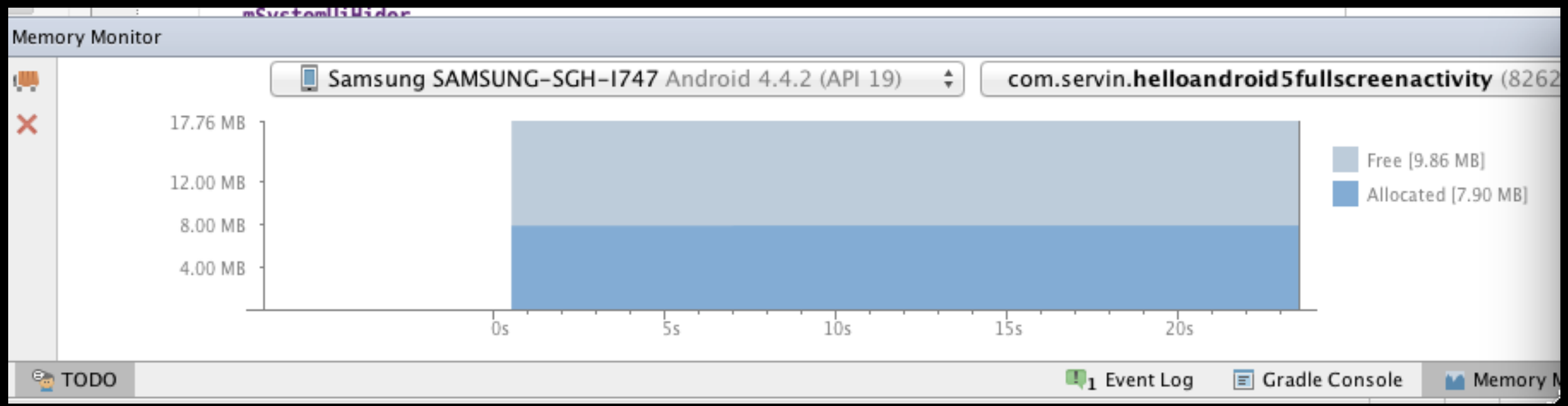
```
AGGREGATED OVER LAST 24 HOURS:
  * system / 1000:
            TOTAL: 100% (44MB-47MB-53MB/40MB-43MB-48MB over 28)
      Persistent: 100% (44MB-47MB-53MB/40MB-43MB-48MB over 28)
  * com.android.systemui / u0a179:
            TOTAL: 100% (41MB-45MB-59MB/35MB-39MB-52MB over 28)
      Persistent: 100% (41MB-45MB-59MB/35MB-39MB-52MB over 28)
         Imp Fg: 0.03%
  * com.sec.knox.eventsmanager / 1000:
            TOTAL: 100% (2.3MB-2.6MB-2.8MB/1.6MB-1.7MB-1.7MB over 28)
      Persistent: 100% (2.3MB-2.6MB-2.8MB/1.6MB-1.7MB-1.7MB over 28)
         Service: 0.16%
  * com.android.phone / 1001:
            TOTAL: 100% (16MB-16MB-16MB/14MB-14MB-15MB over 28)
      Persistent: 100% (16MB-16MB-16MB/14MB-14MB-15MB over 28)
  * com.android.nfc / 1027:
            TOTAL: 100% (4.8MB-6.2MB-7.5MB/3.9MB-5.1MB-6.2MB over 28)
      Persistent: 100% (4.8MB-6.2MB-7.5MB/3.9MB-5.1MB-6.2MB over 28)
  * org.simalliance.openmobileapi.service:remote / 1101:
            TOTAL: 100% (2.4MB-2.6MB-2.7MB/1.8MB-1.9MB-1.9MB over 31)
         Imp Fg: 100% (2.4MB-2.6MB-2.7MB/1.8MB-1.9MB-1.9MB over 31)
  * android.process.acore / u0a9:
            TOTAL: 100% (8.1MB-8.7MB-9.4MB/7.5MB-8.0MB-8.6MB over 28)
      Persistent: 100% (8.1MB-8.7MB-9.4MB/7.5MB-8.0MB-8.6MB over 28)
         Imp Fg: 0.07%
        (Cached): 0.03%
```

System Information Graphics State

activity_fullscreen.xml × | dumpsys-2657125578409594274.txt × | dumpsys-6693930777525773694.txt ×

```
Applications Graphics Acceleration Info:
Uptime: 3443082 Realtime: 4224693

** Graphics info for pid 8262 [com.servin.helloandroid5fullscreenactivity] **

Recent DisplayList operations
            DrawText
            RestoreToCount
          RestoreToCount
DrawRect
DrawRect
multiDraw
  DrawText
DrawText
DrawBitmap
DrawDisplayList
  DrawDisplayList
    DrawDisplayList
      DrawRect
      DrawDisplayList
        Save
        ClipRect
        Translate
        DrawText
        DrawText
```

Memory Monitor



Android Studio

Dynamic Layout

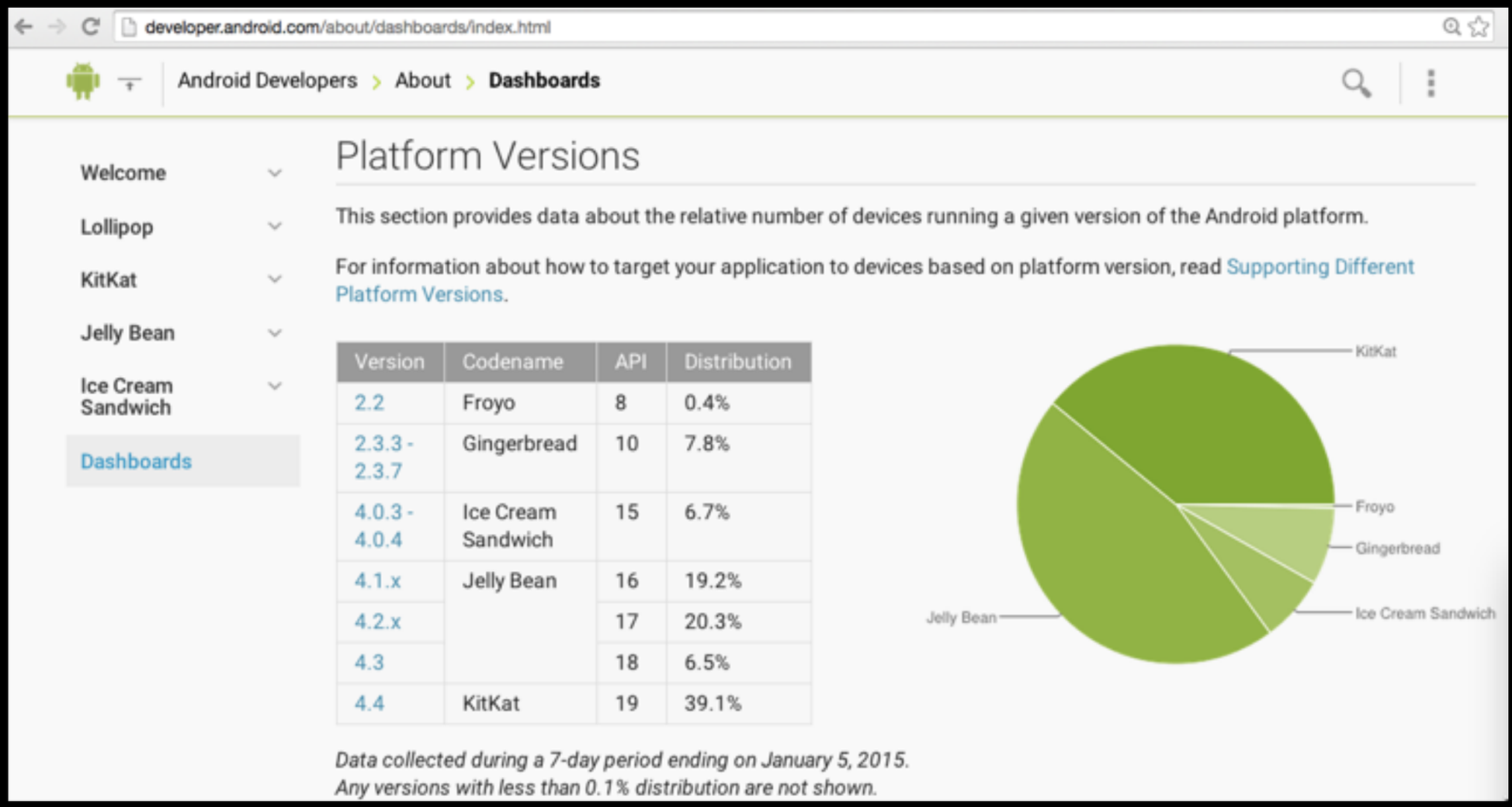
Dynamic Layout

- Three Options
  - Design View
    - Drag from Palette to Preview or Component Tree
  - Text View
  - Preview

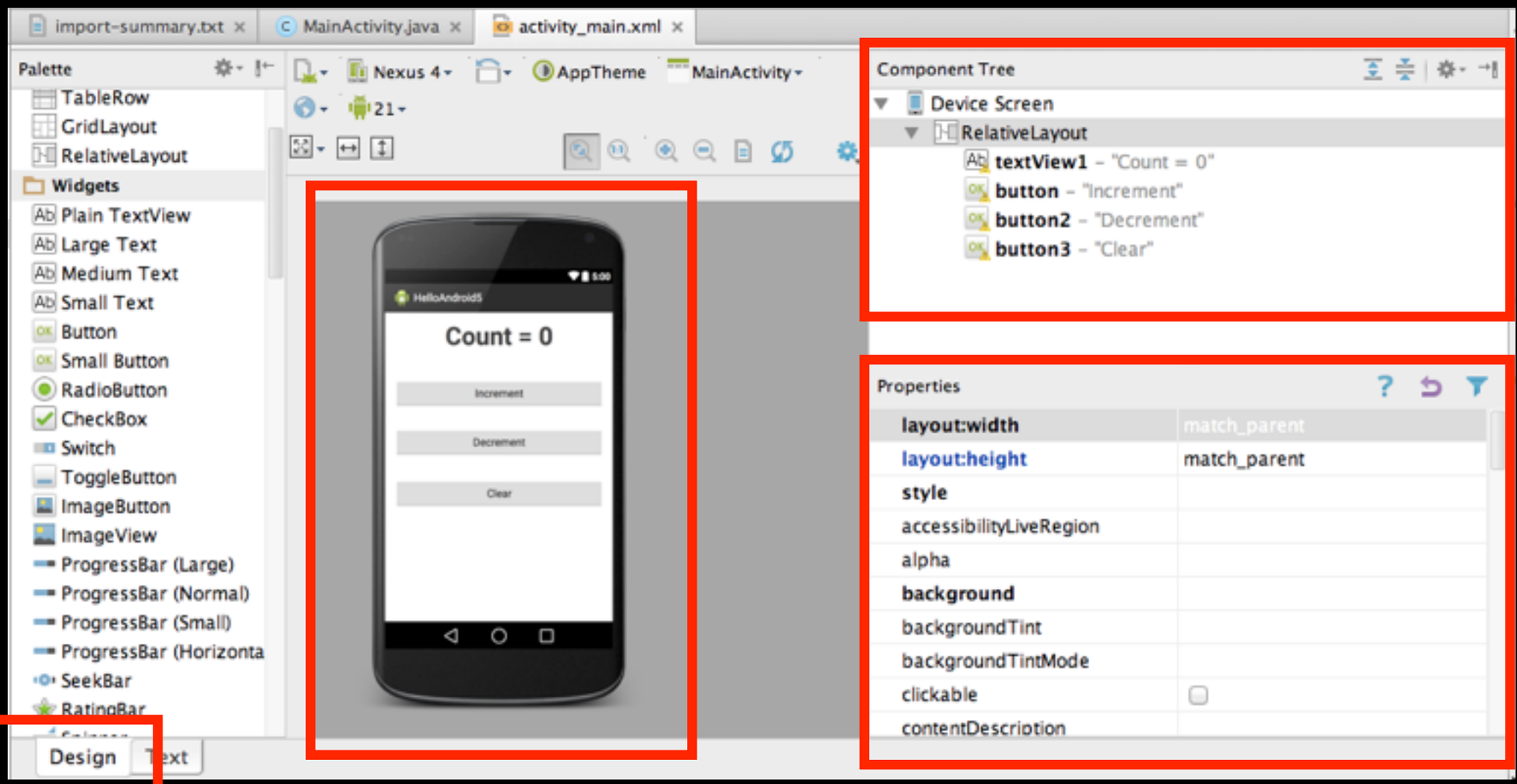
Dynamic Design Preview Choices

- You can preview all of the following
  - Different Device Images
  - Different Display Densities
  - Different UI Modes
  - Different Android Versions

Motivation For Dynamic Layout



Design View



Text View

import-summary.txt | MainActivity.java | activity_main.xml

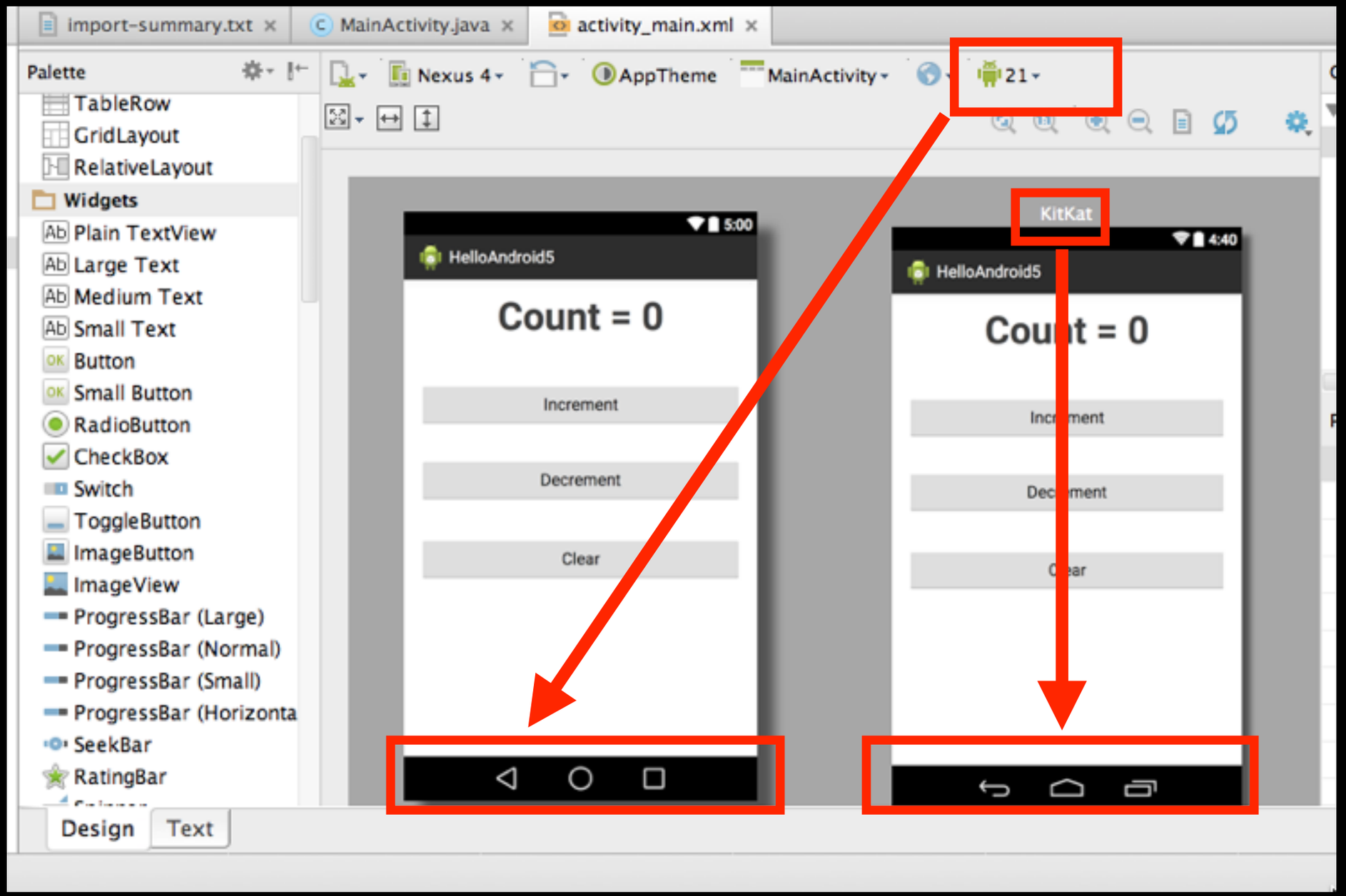
```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/andro
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="16dp"
    android:paddingLeft="16dp"
    android:paddingRight="16dp"
    android:paddingTop="16dp"
    tools:context="com.example.helloandroid5.MainActivity" >

    <TextView
        android:id="@+id/textView1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Count = 0"
        android:layout_alignEnd="@+id/button"
        android:layout_alignParentStart="false"
        android:gravity="center"
        android:textSize="42dp"
        android:textStyle="bold" />

    <Button
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:text="Increment"
        android:id="@+id/button"
        android:layout_below="@+id/textView1"
        android:layout_alignParentEnd="true"
```

Design | Text

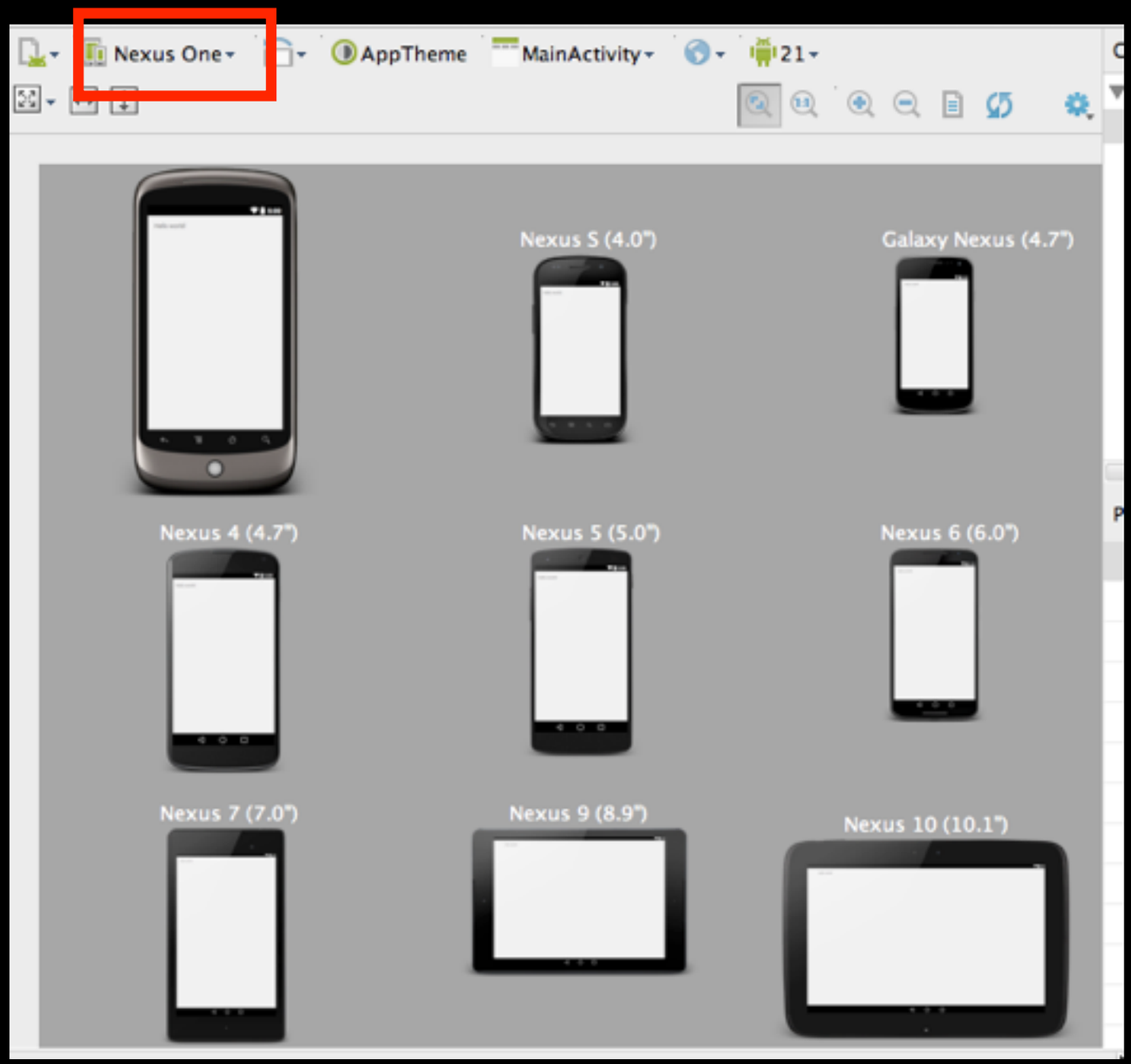
Preview

Nexus 4 | AppTheme | MainActivity

Dynamic Preview based on API Version



Dynamic Preview based on Screen Size



Keeping Android Studio Up-To-Date

Four Update Channels

Update channels

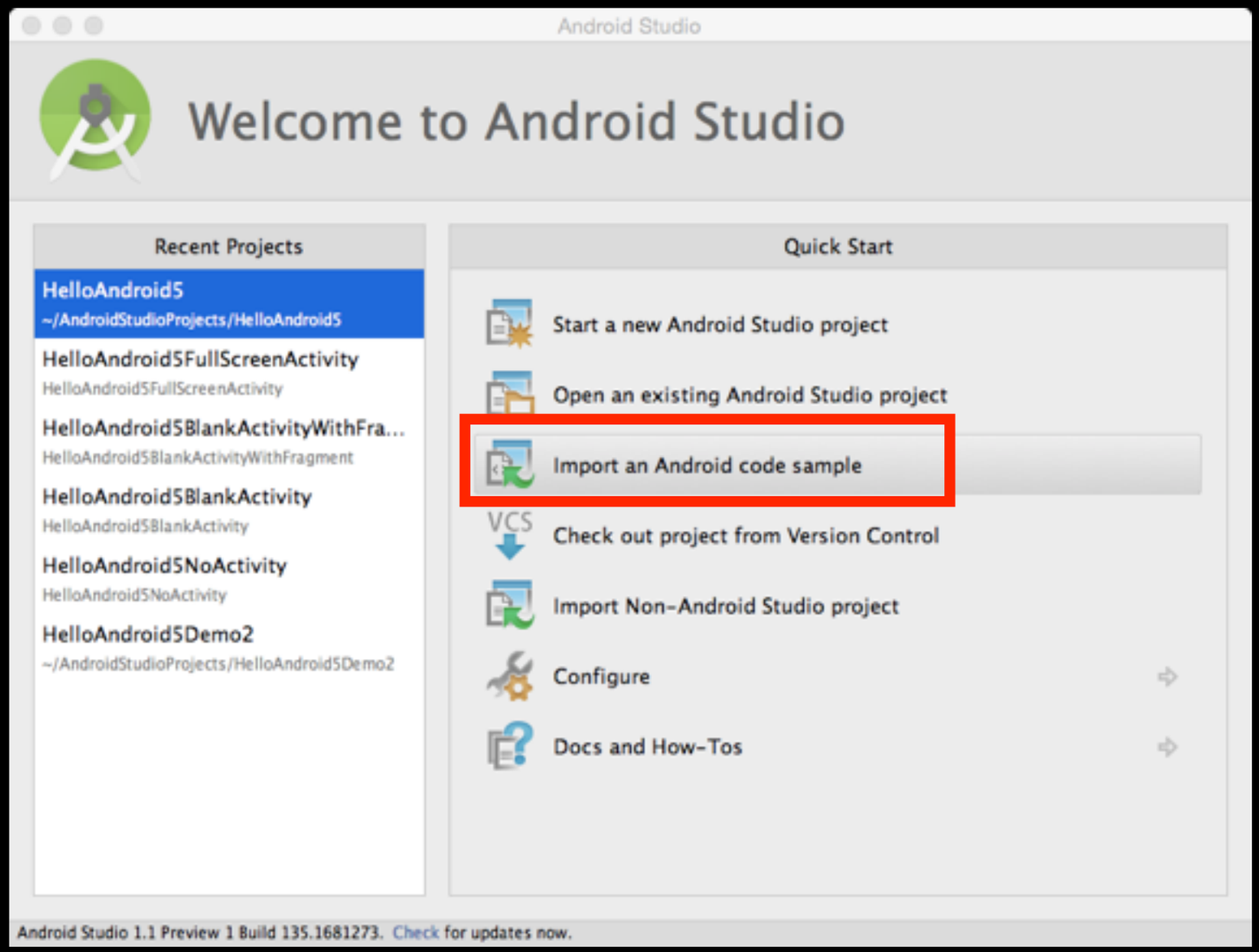
Android Studio provides four update channels to keep Android Studio up-to-date based on your code-level preference:

- **Canary channel**: Canary builds provide bleeding edge releases, updated about weekly. While these builds do get tested, they are still subject to bugs, as we want people to see what's new as soon as possible. This is not recommended for production.
- **Dev channel**: Dev builds are hand-picked older canary builds that survived the test of time. They are updated roughly bi-weekly or monthly.
- **Beta channel**: Beta builds are used for beta-quality releases before a production release.
- **Stable channel**: Used for stable, production-ready versions.

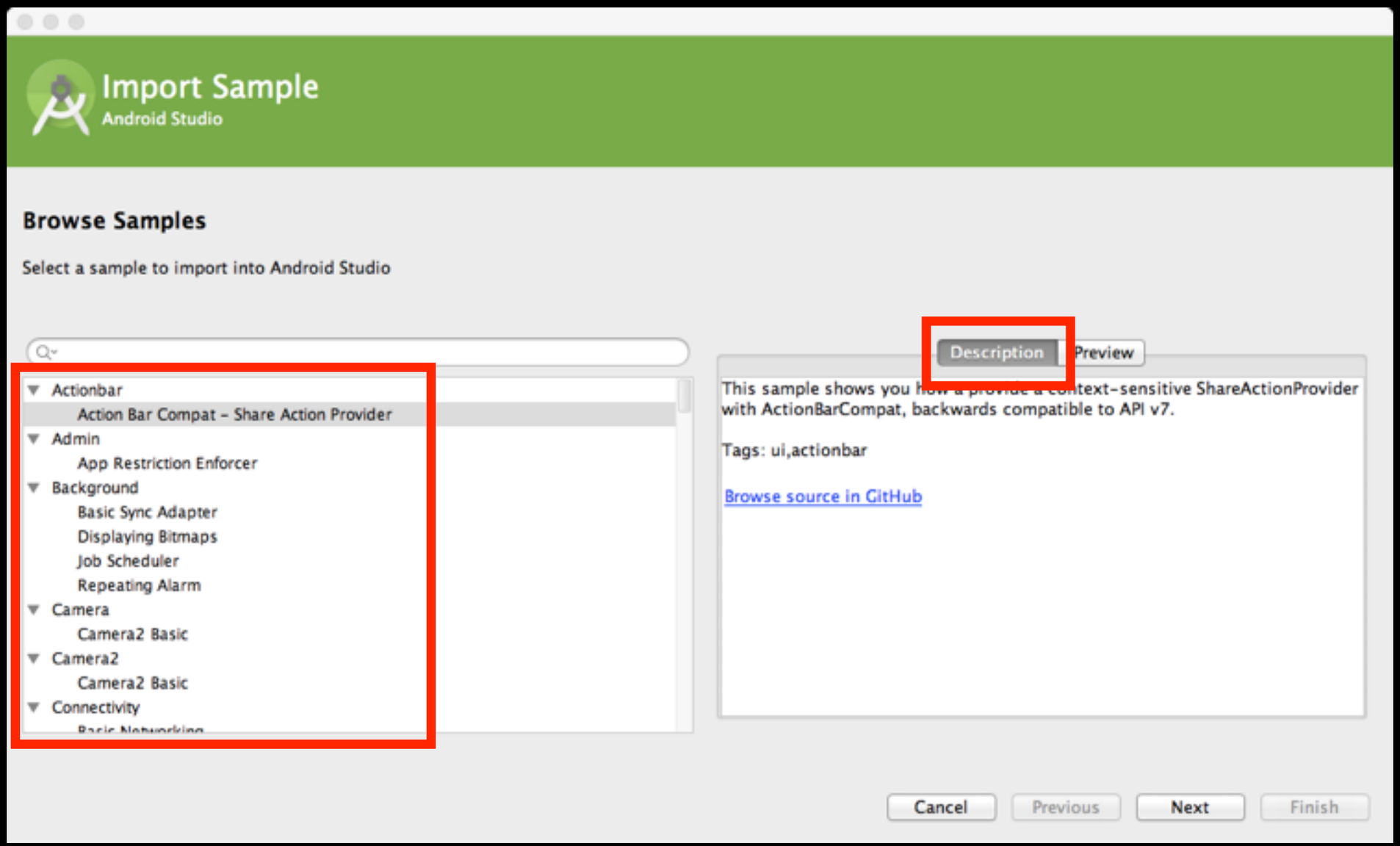
By default, Android Studio uses the *Stable* channel. Use **File > Settings > Updates** to change your channel setting.

Easy Access to Android Code Samples via GitHub

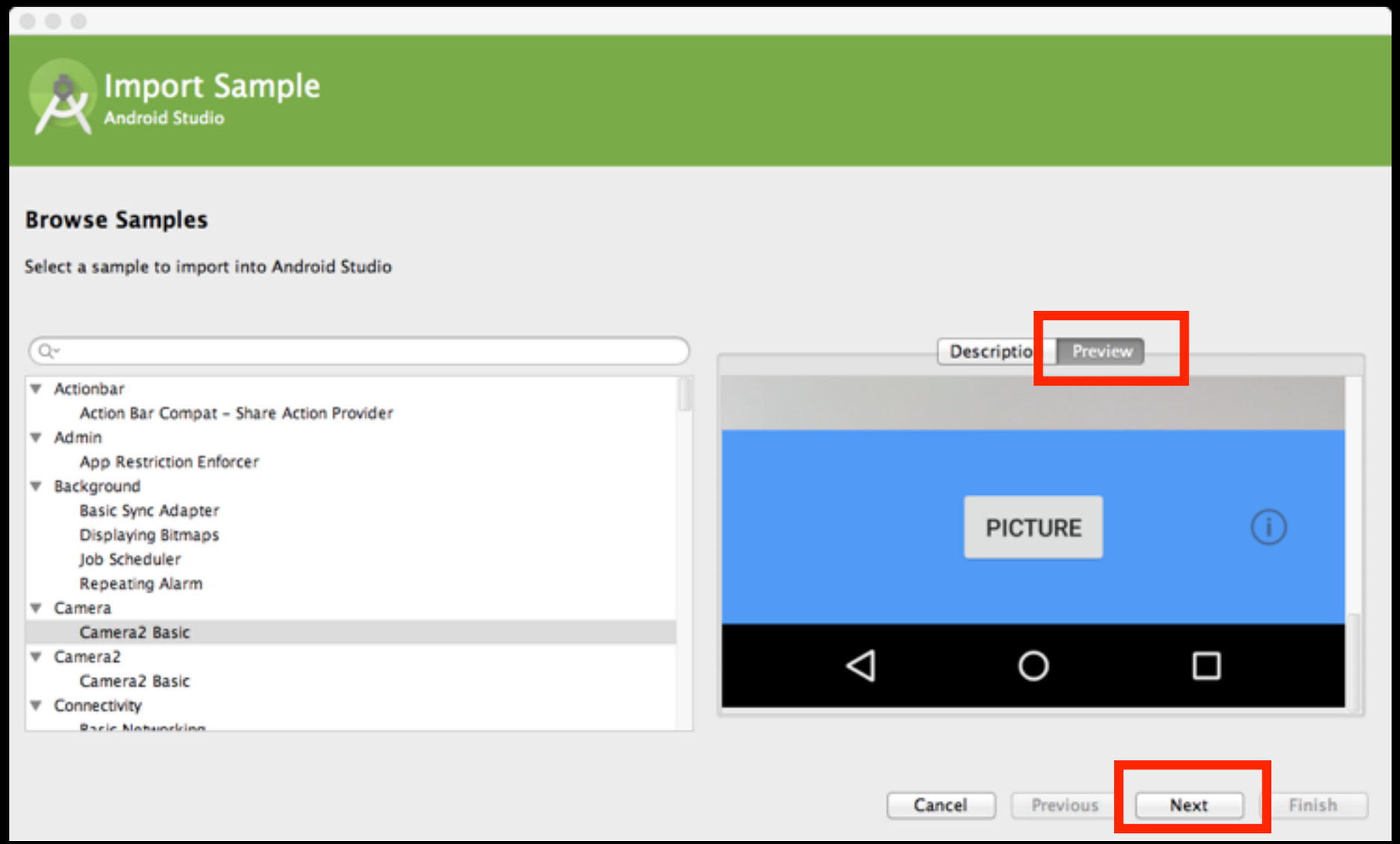
Open Project From GitHub Code Samples



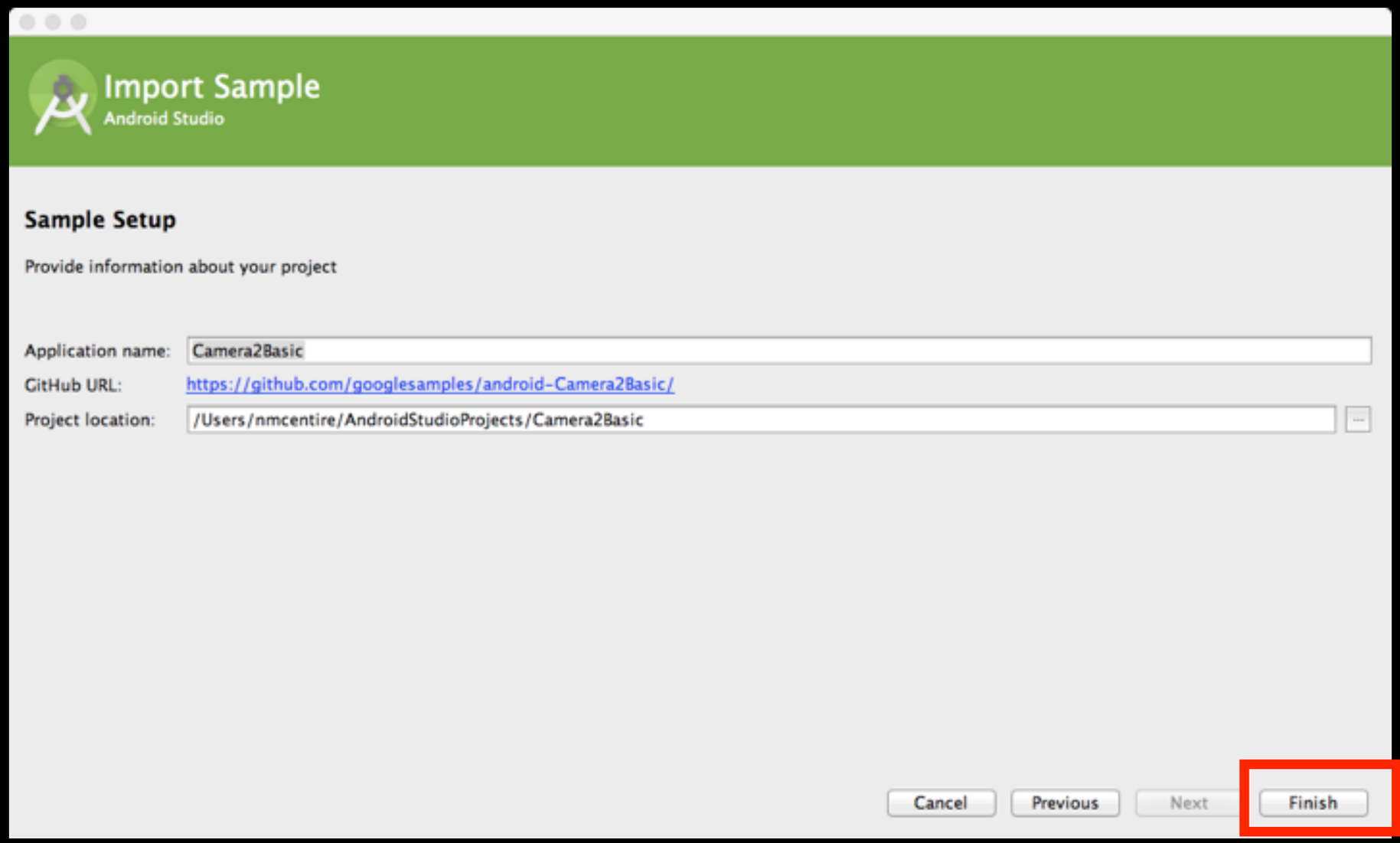
Code Samples from GitHub Description



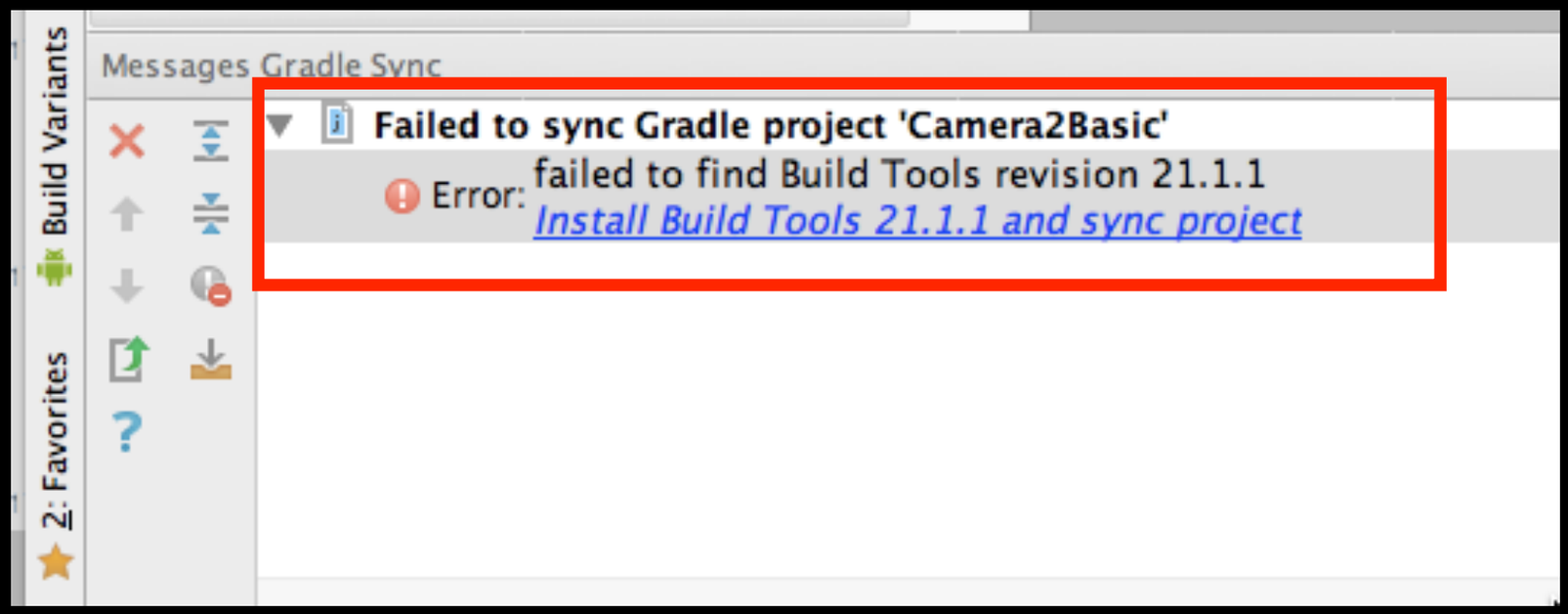
Code Samples from GitHub Preview



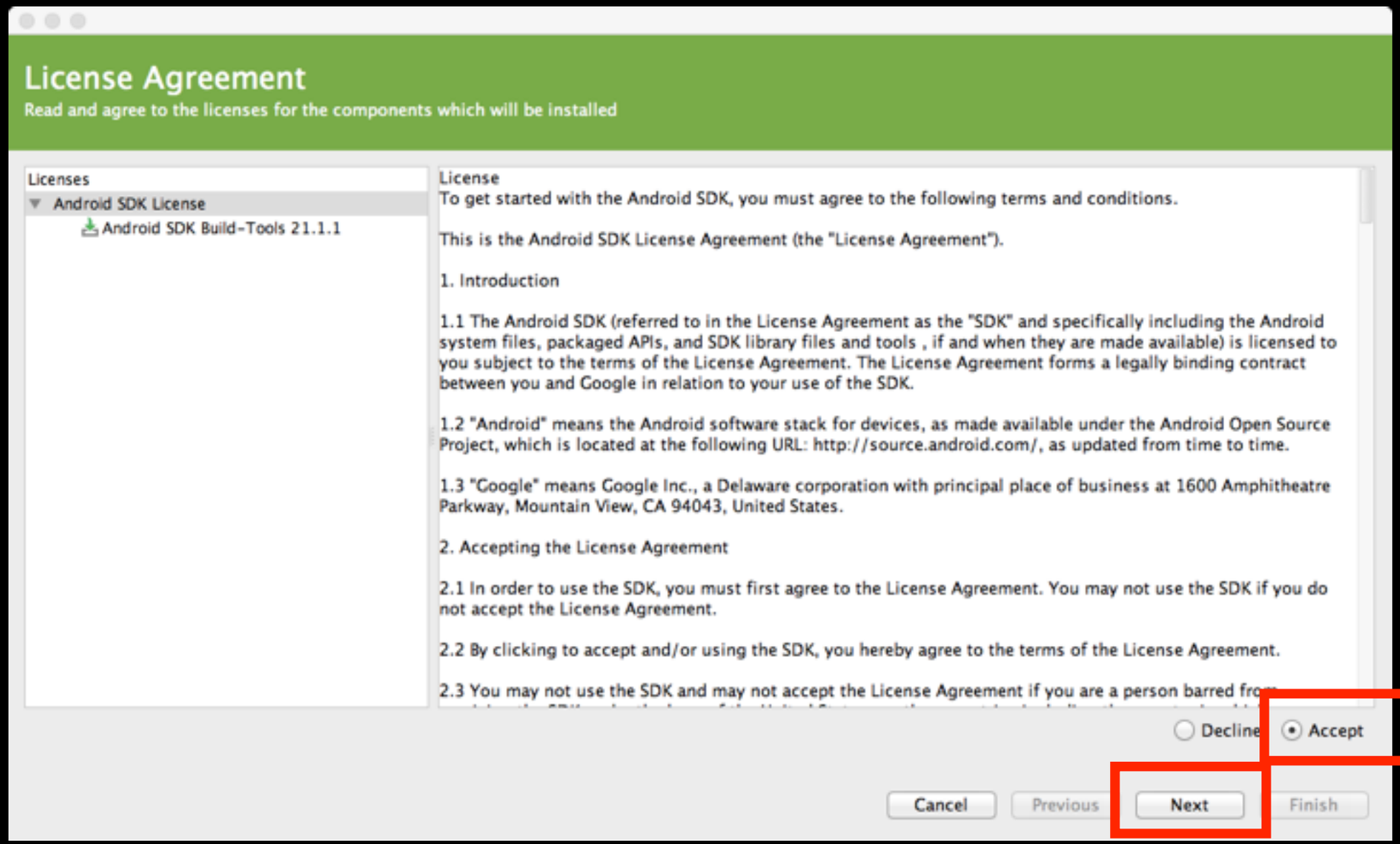
Code Samples from GitHub Import Sample



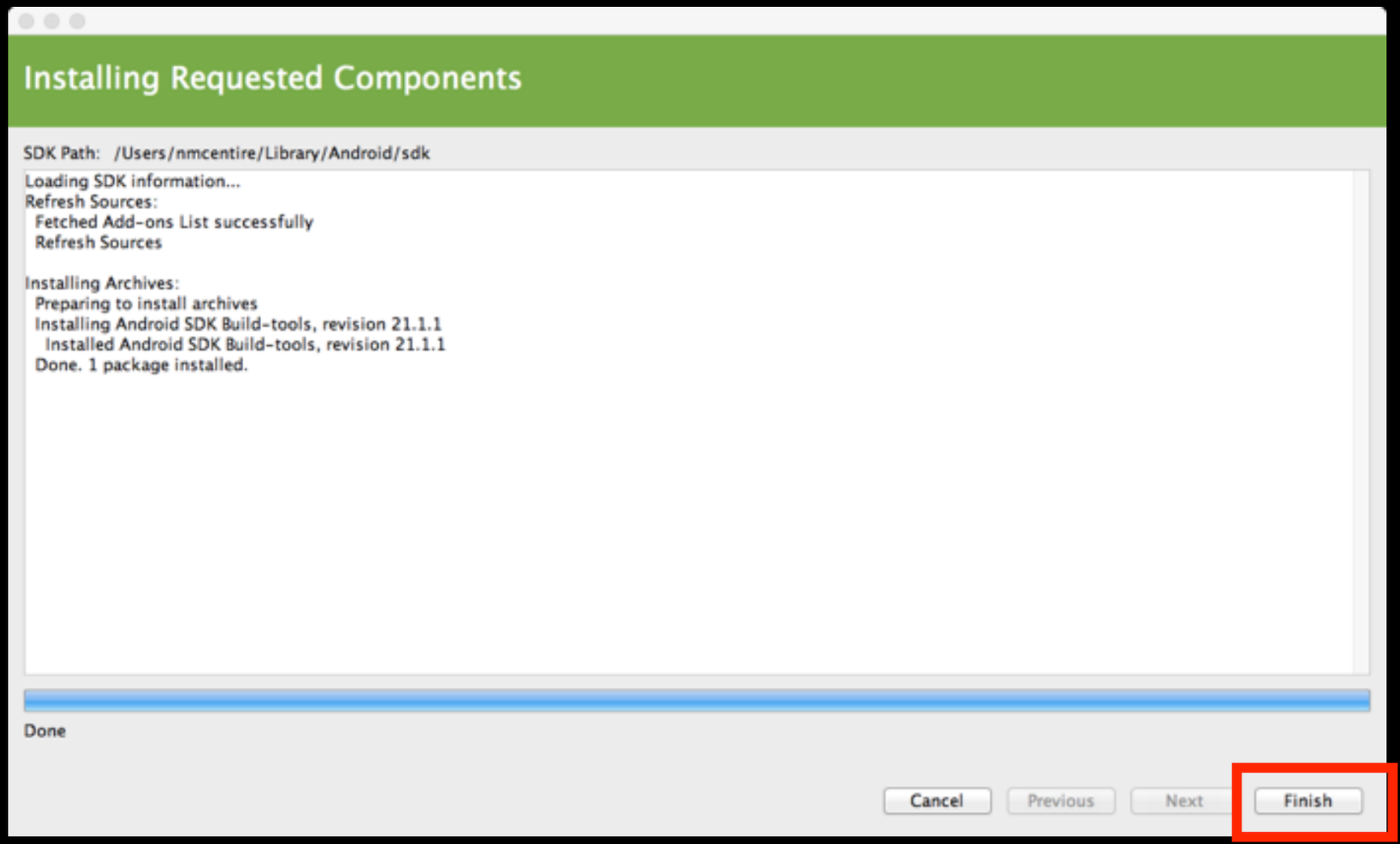
Possible Error When Importing Sample Projects



Installing Build Tools License Agreement

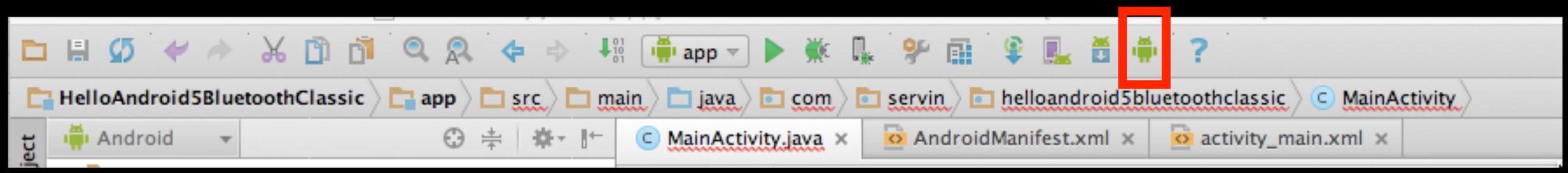


Installing Build Tools
Installing Requested Components

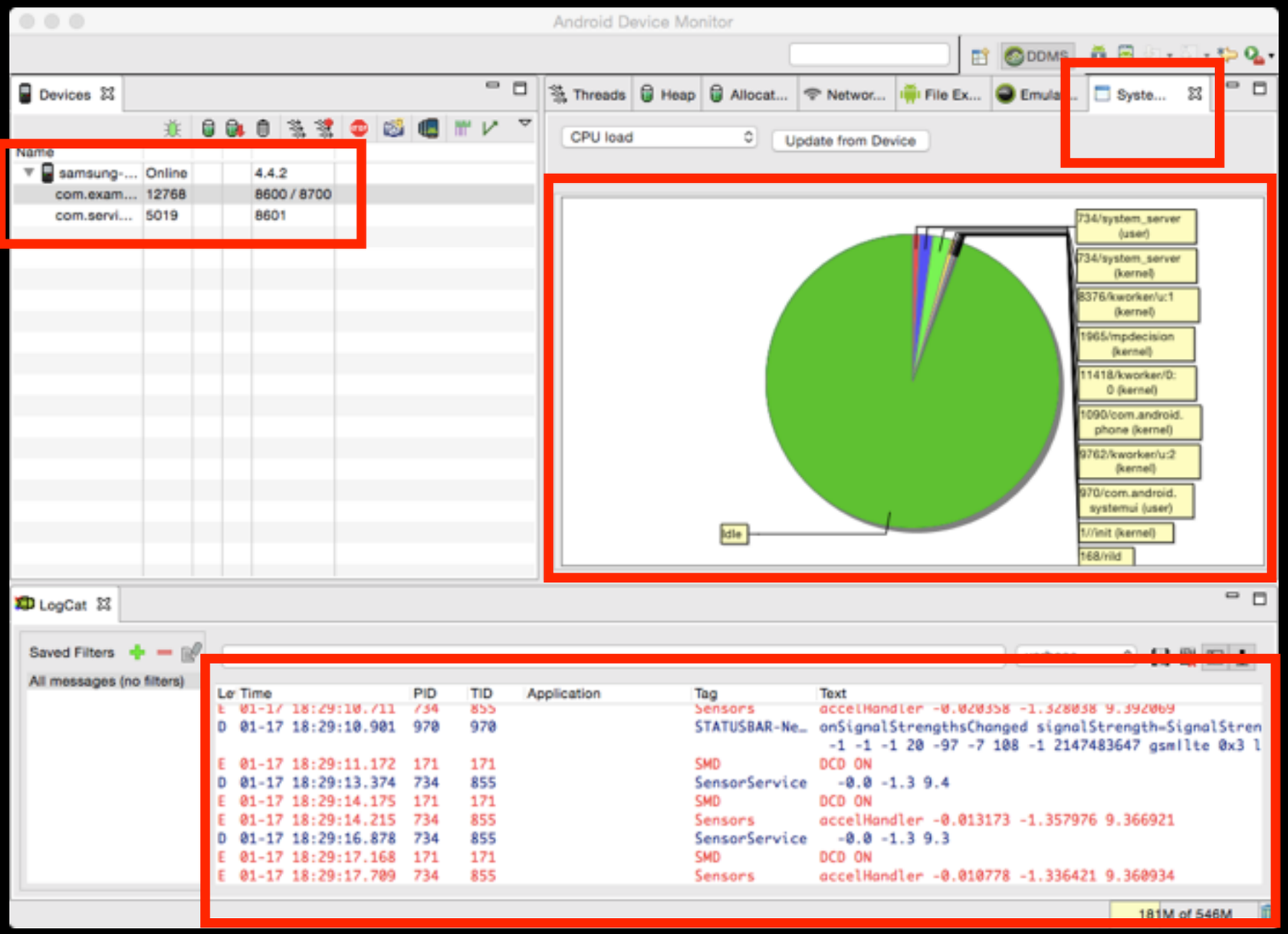


Android Device Monitor

Starting **Android Device Monitor** From Android Studio

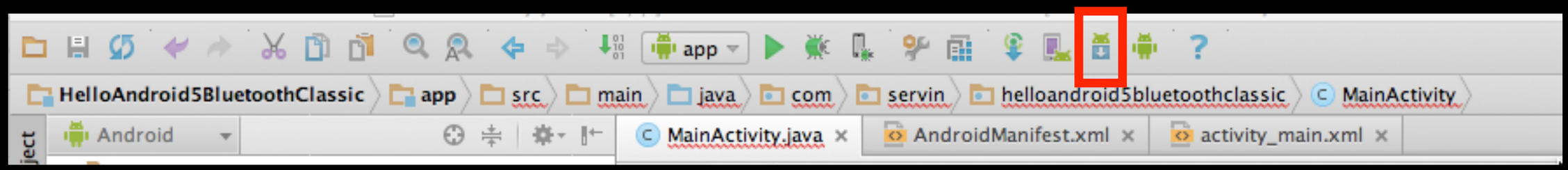


Android Device Monitor

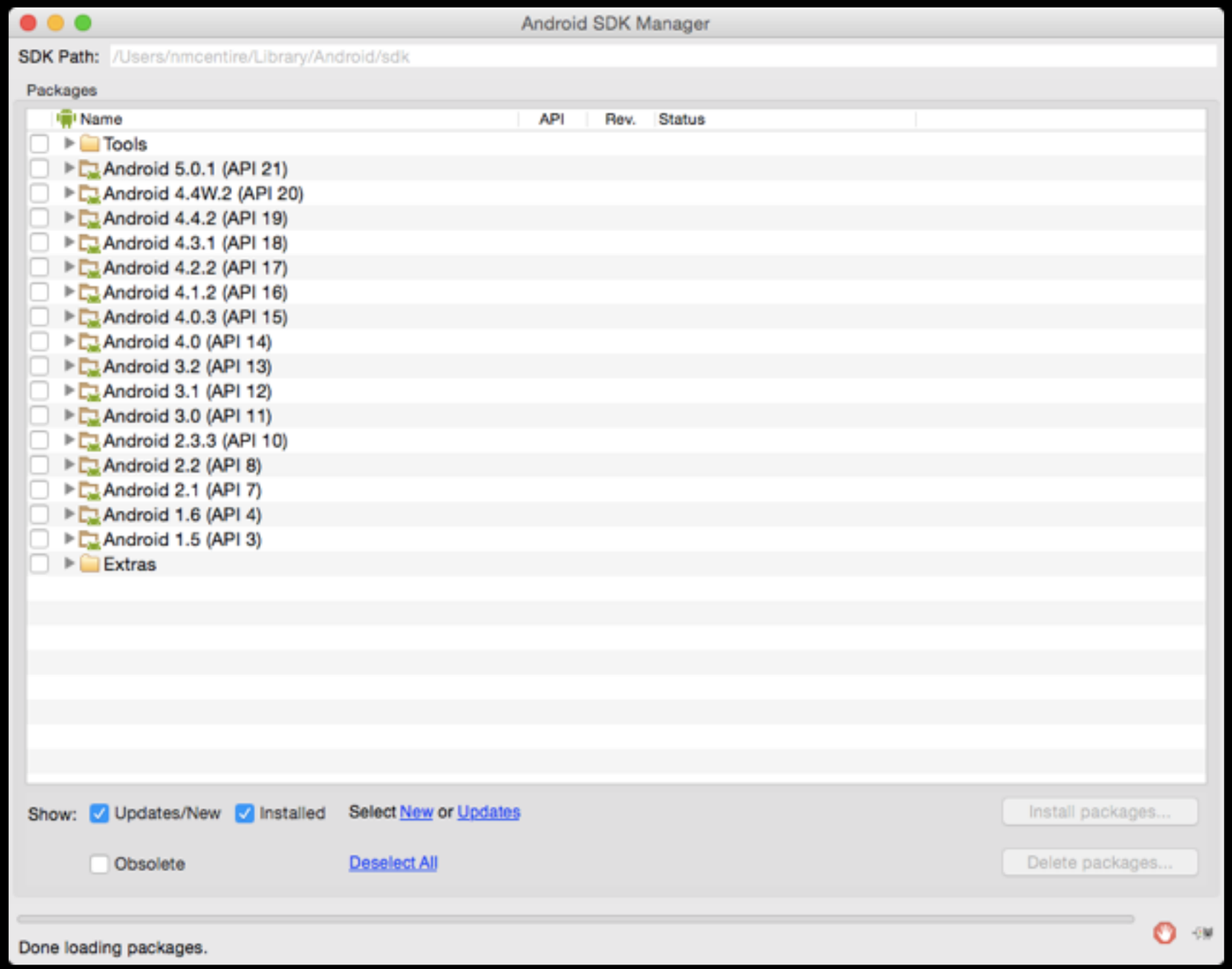


Android SDK Manager

Starting **Android SDK Manager** From Android Studio

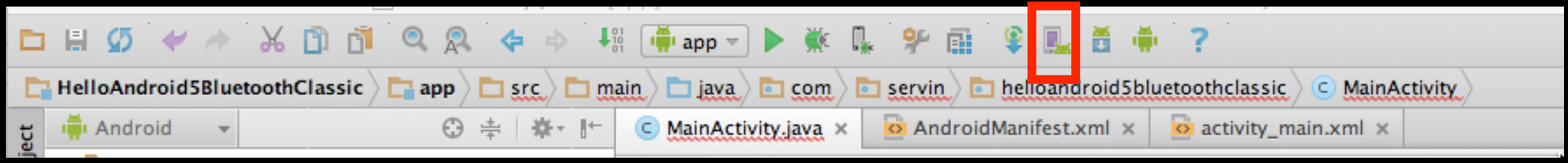


Android SDK Manager

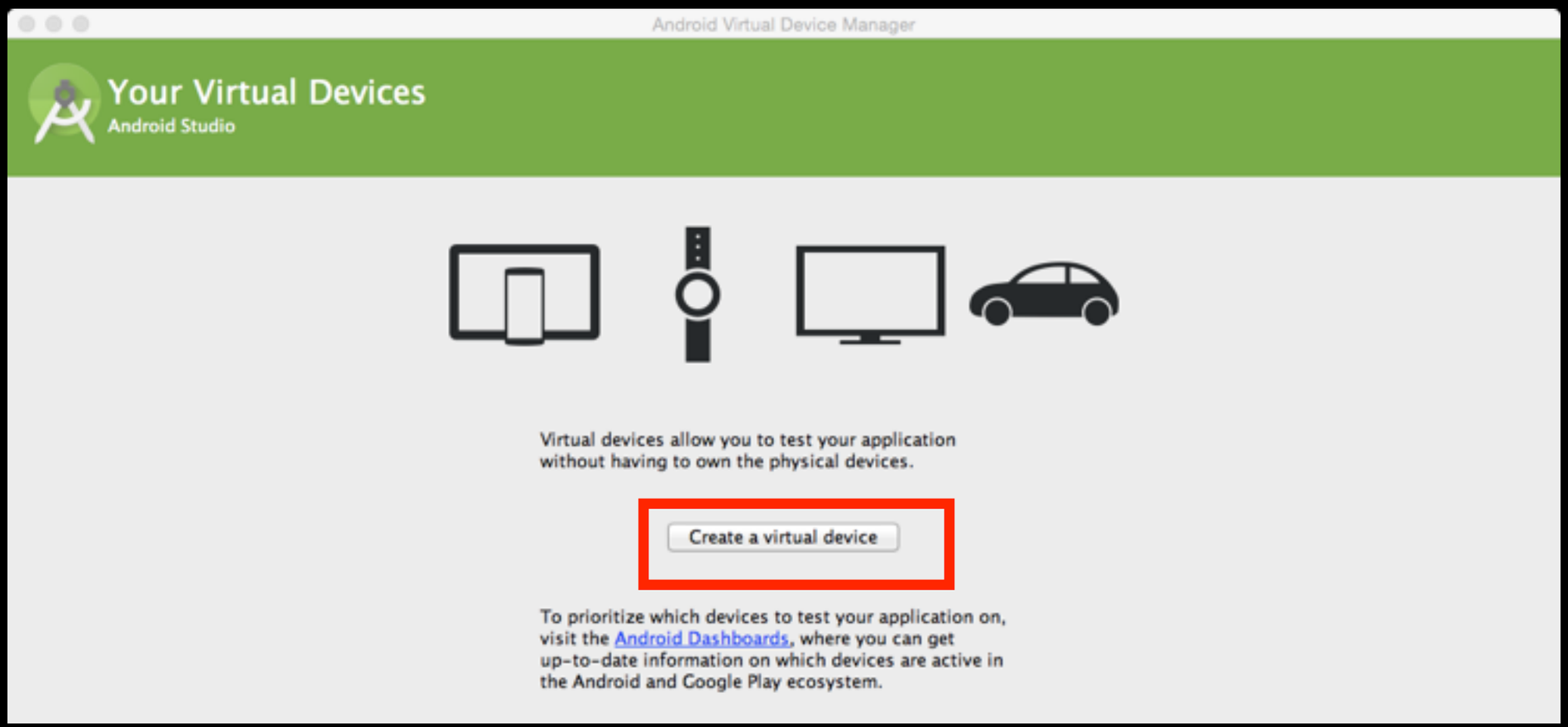


Android Virtual Device (AVD) Manager

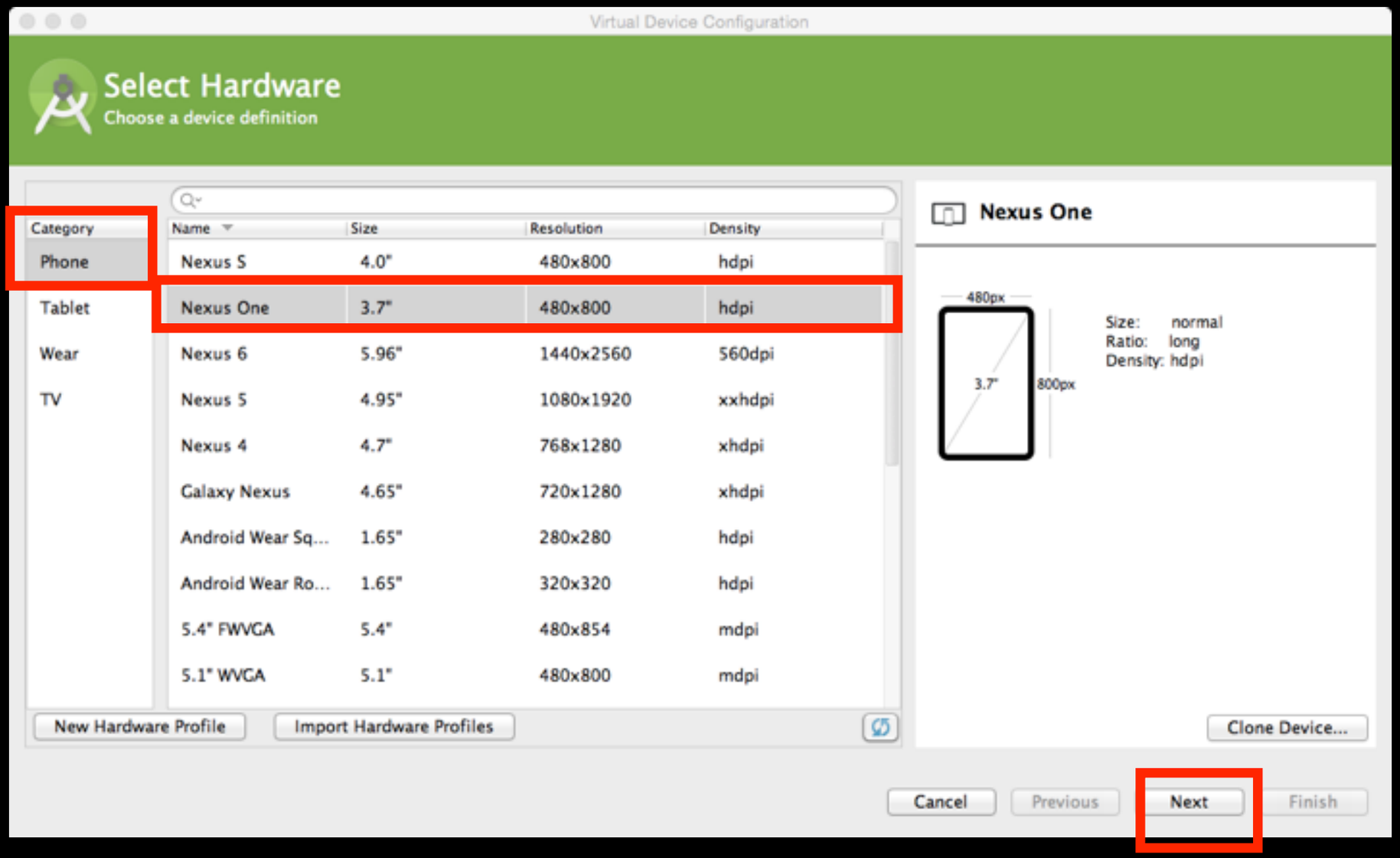
Starting **Android Virtual Device (AVD) Manager** From Android Studio



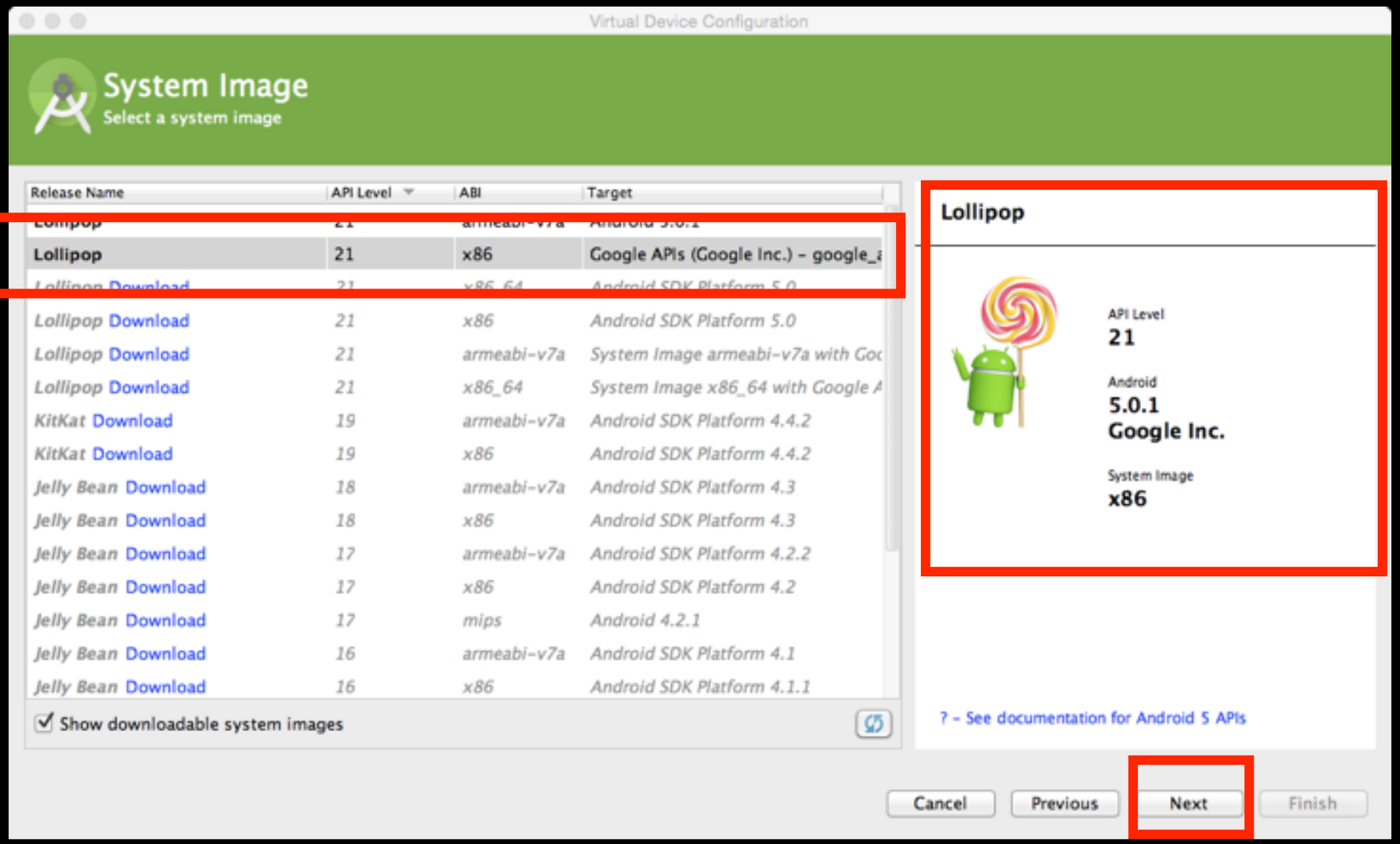
Android Virtual Device (AVD) Manager - Create Device



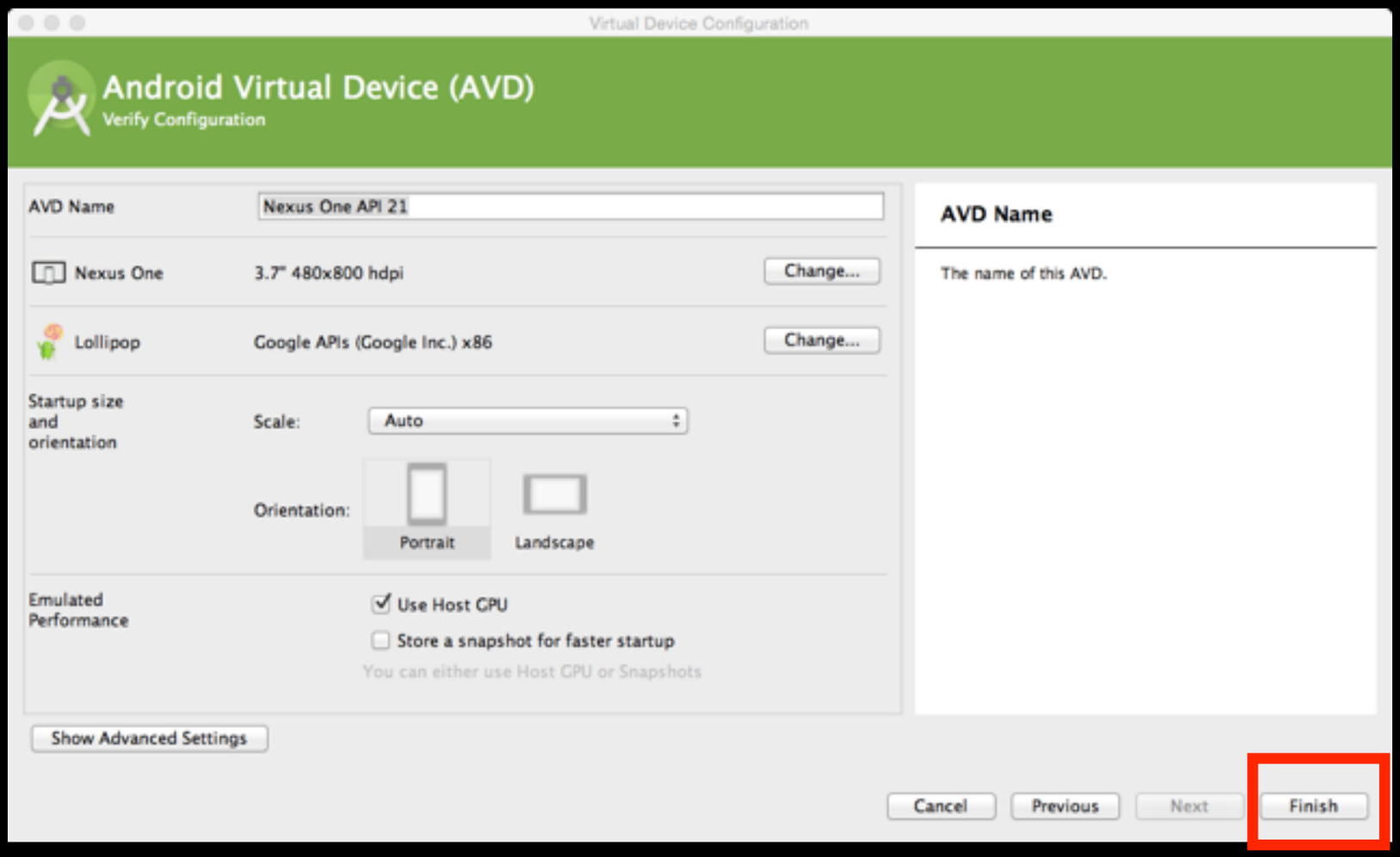
Android Virtual Device (AVD) Manager - Choose Device



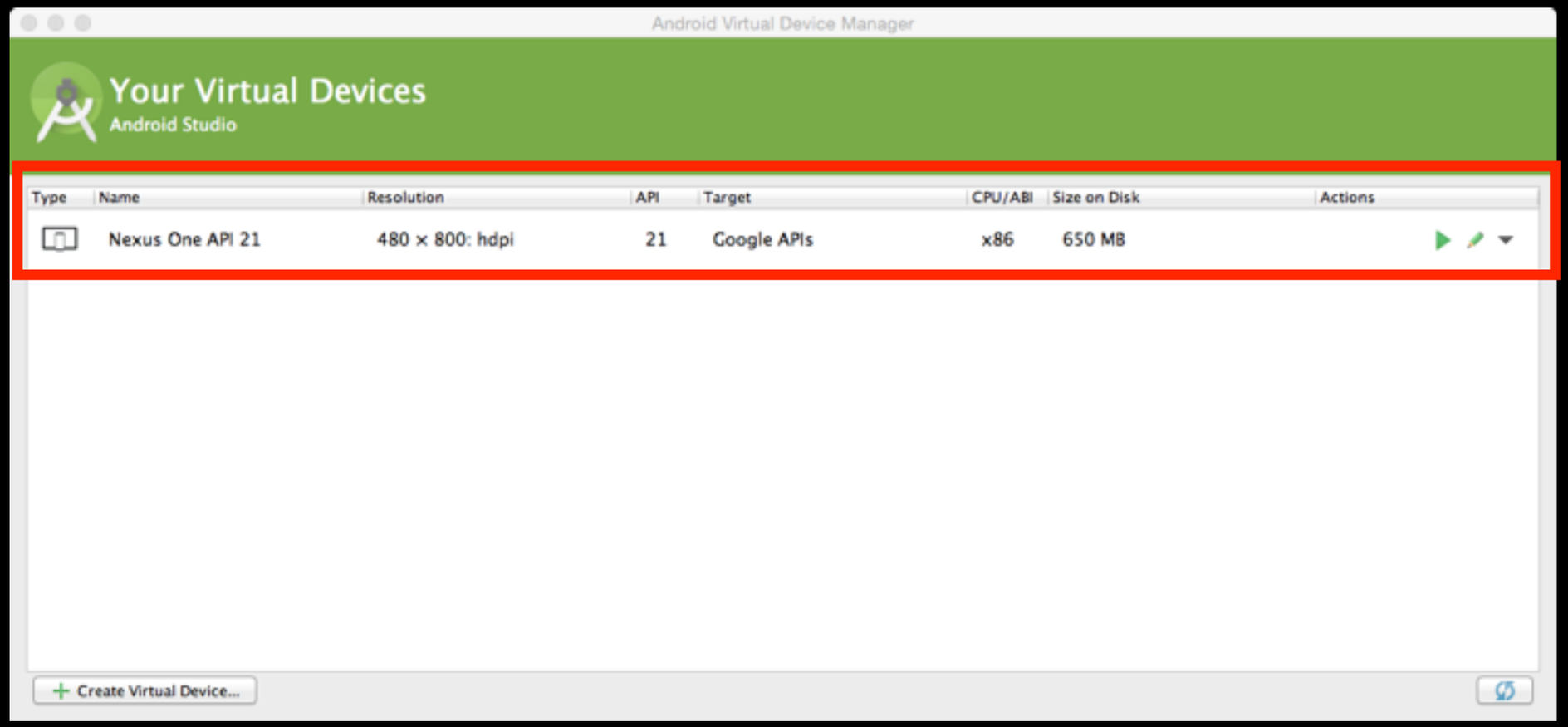
Android Virtual Device (AVD) Manager - System Image



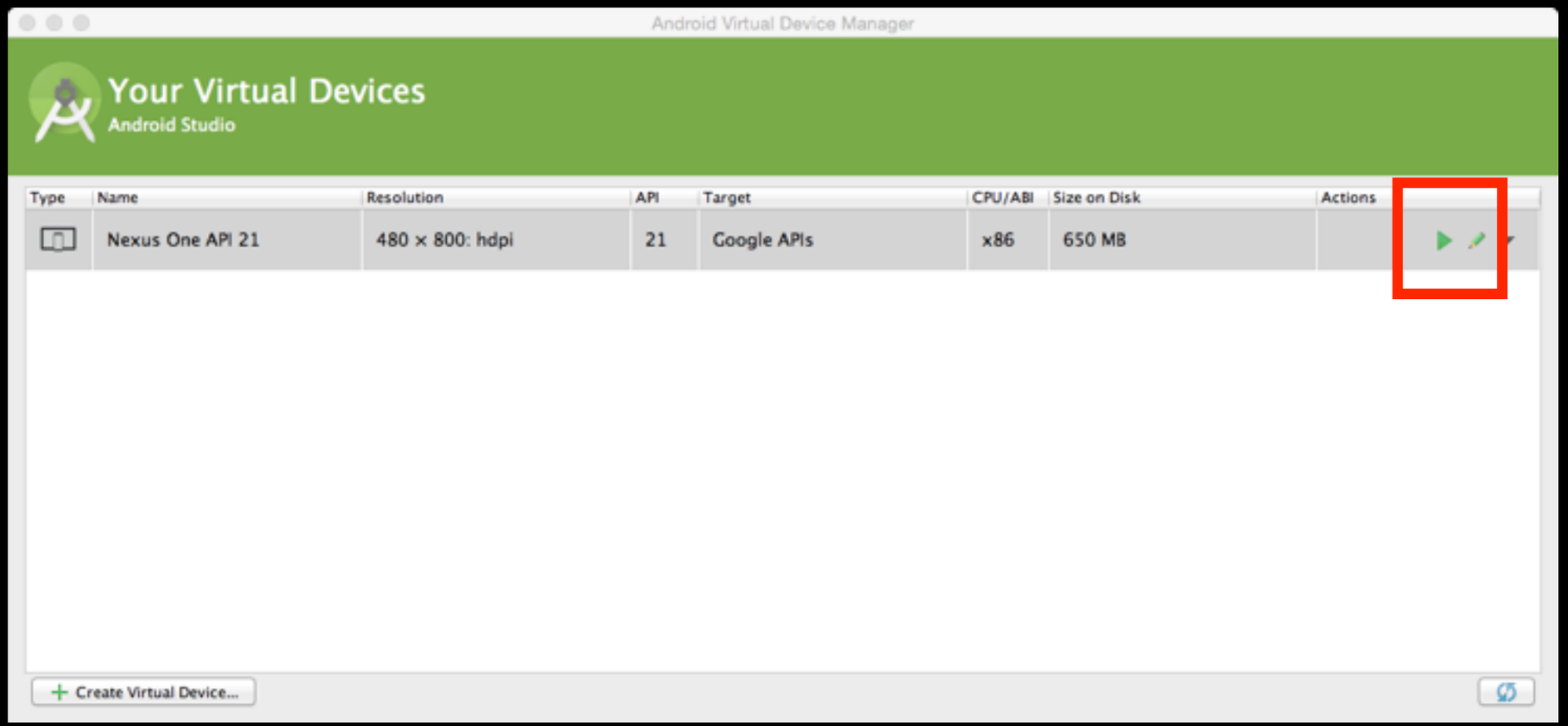
Android Virtual Device (AVD) Manager - Verify Configuration



Android Virtual Device (AVD) Manager - Your Virtual Device



Android Virtual Device (AVD) Manager - Startup



Android Virtual Device (AVD) Manager - Startup

Code Running on Emulator

Android Build System

- Previous Android Build System
  - Eclipse ADT with ANT
- New Build System
  - Android Studio with Gradle

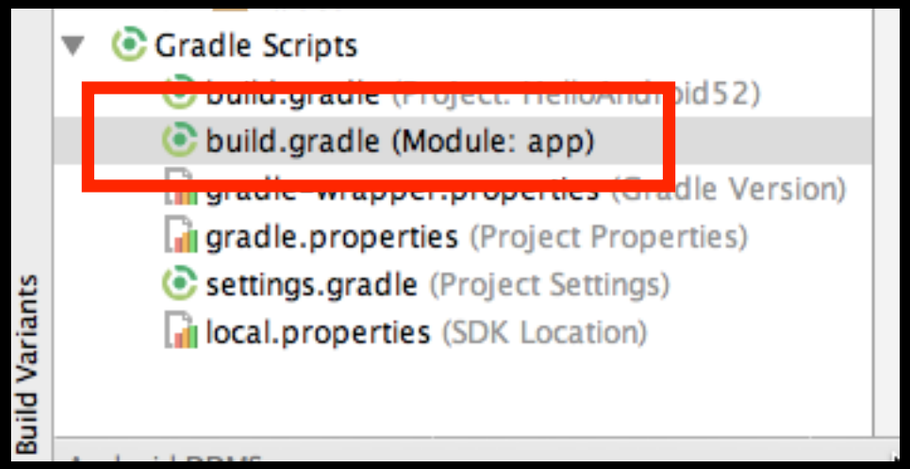
Two Options for the Build System

- Option 1.
  - Use features of Android Studio without any focus on the underlying build system
  - Create single APK
- Option 2.
  - Customize the build system
  - Create multiple APKs using same project
  - Reuse code and resources across project sets

Gradle Build Script

- The build files are called build.gradle
- They are plain text files
- They use the Groovy syntax to configure the build
- With elements provided by the Android plugin for Gradle
- In most cases, you only need to edit build files at the module level (see next slide)

Normally only edit build.gradle at Module Level



Key Point:
The Android Build System enables you to customize build WITHOUT modification to app source files!

The "apply" line includes Android-Specific Elements

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

The android Element configures all Android-specific build options

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

compileSdkVersion

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

buildToolsVersion
(always higher than compileSdkVersion)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

defaultConfig
(override settings in AndroidManifest.xml)
(apply to all build variants)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

buildTypes
(how to build and package app)

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

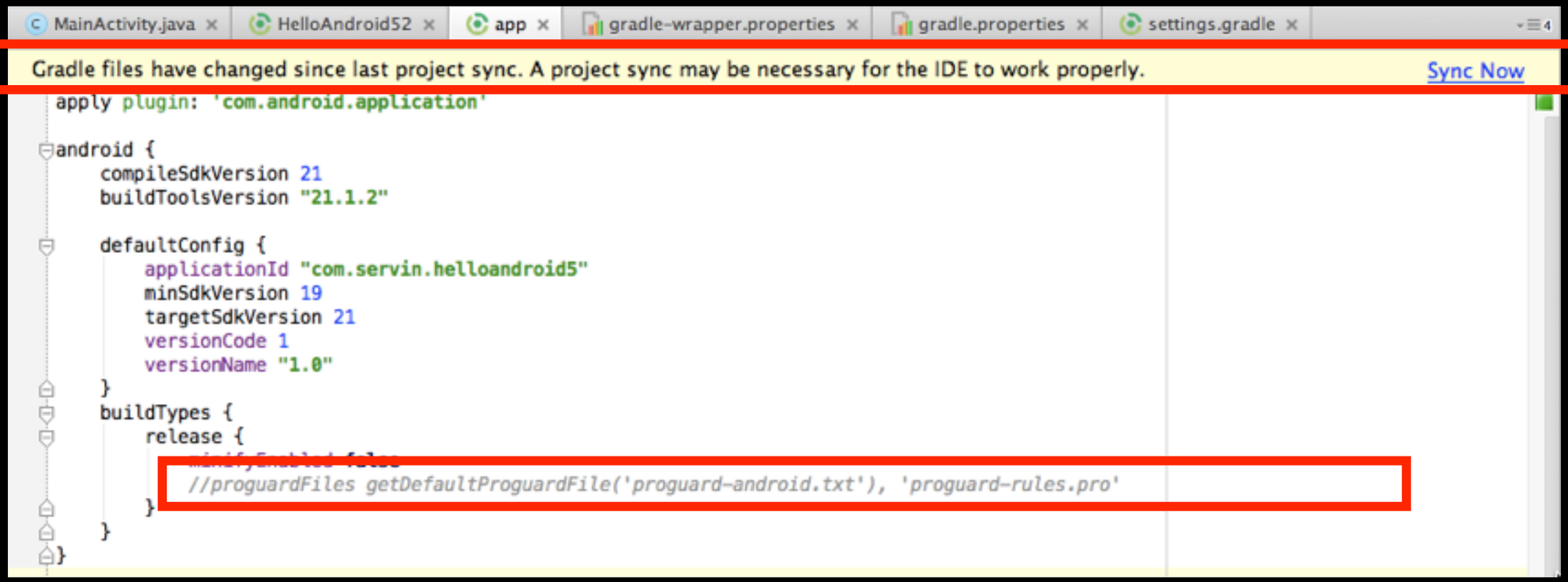
Note: By default, build system defines two types:
debug and **release**

dependencies

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

Important!
Making changes to build system
requires a sync!
(see next slide)

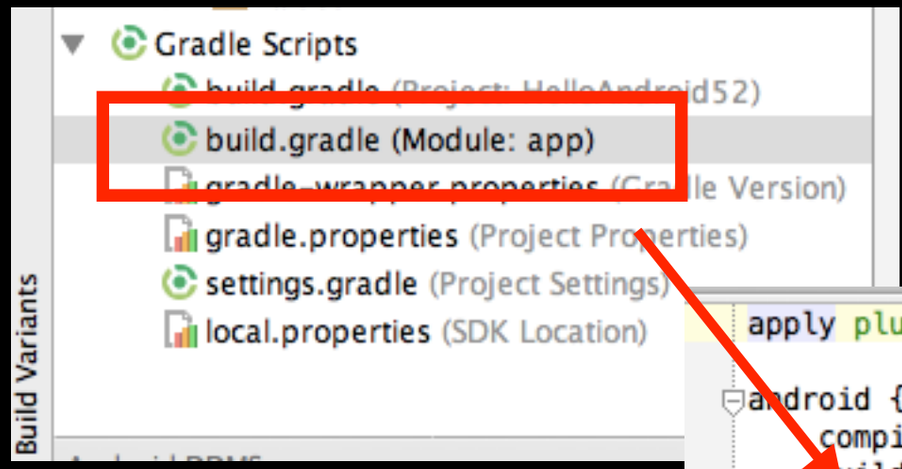
Sync Now



applicationId

- Use applicationId to uniquely identify application packages for publishing

applicationId



Key Point:
applicationId specified in build.grade, and NOT in AndroidManifest.xml

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 21
    buildToolsVersion "21.1.2"

    defaultConfig {
        applicationId "com.servin.helloandroid5"
        minSdkVersion 19
        targetSdkVersion 21
        versionCode 1
        versionName "1.0"
    }
    buildTypes {
        release {
            minifyEnabled false
            proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
        }
    }
}

dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
}
```

Build Variants Demo

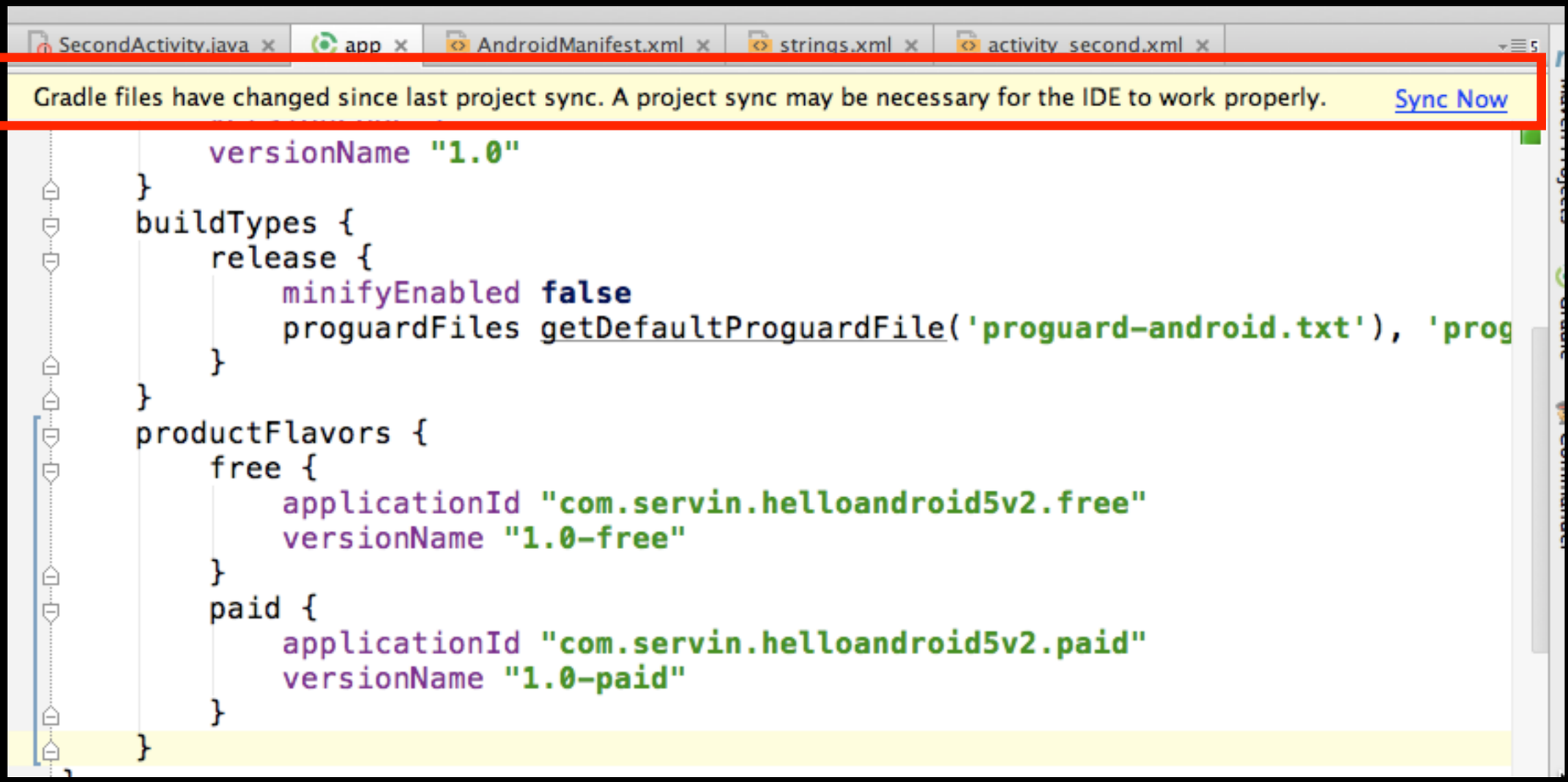
Build Variants Demo

- This demo will show how to create to build variants from a single project
  - A Free (Limited Featured) Version
  - A Paid (Fully Featured) Version
- Both versions will have the same MainActivity
- Each version will have a different SecondActivity

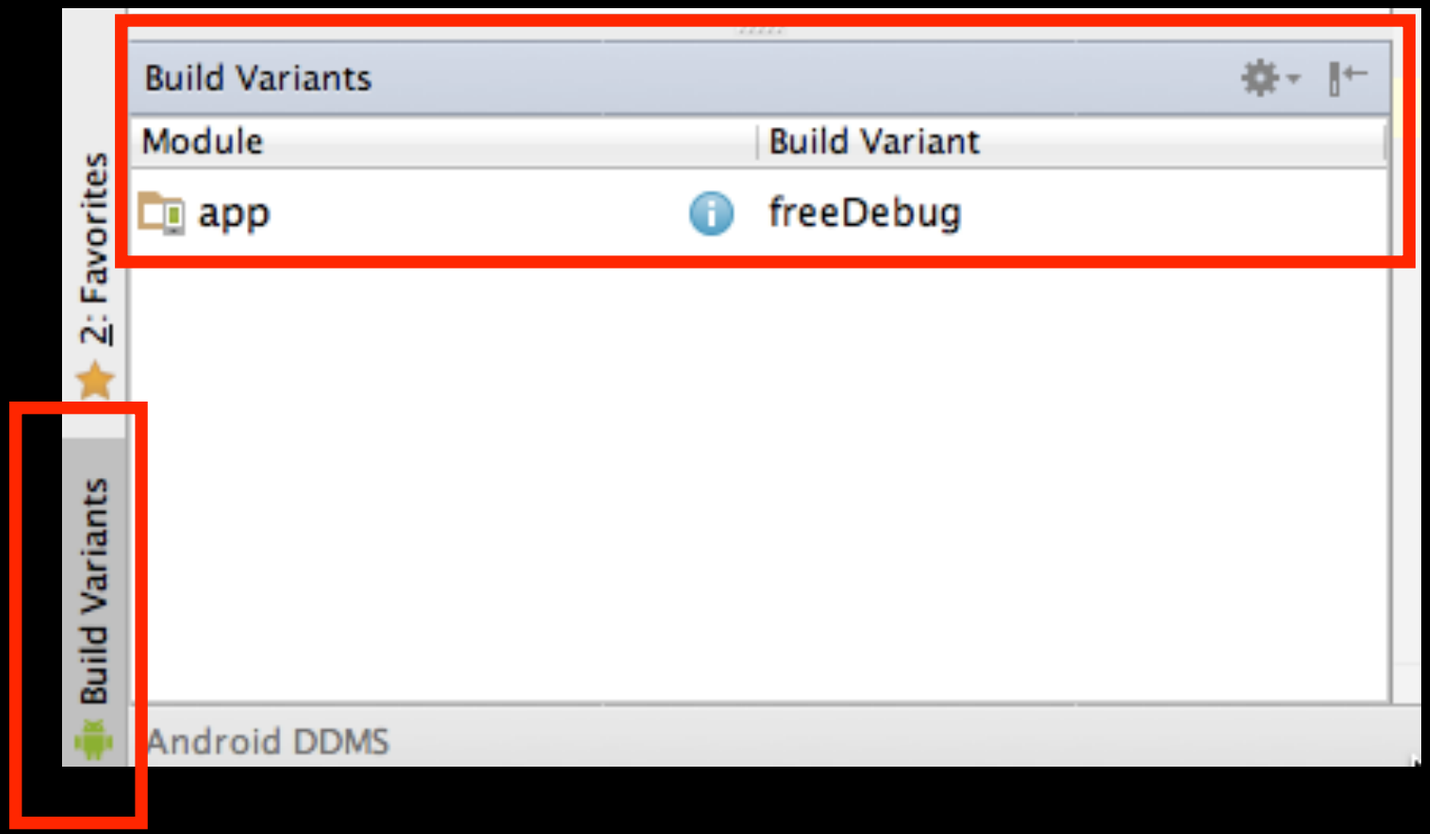
Step 1. Define Two Product Flavors

```
    }
    productFlavors {
        free {
            applicationId "com.servin.buildsystemexample.free"
            versionName "1.0-free"
        }
        paid {
            applicationId "com.servin.buildsystemexample.paid"
            versionName "1.0-paid"
        }
    }
}
```

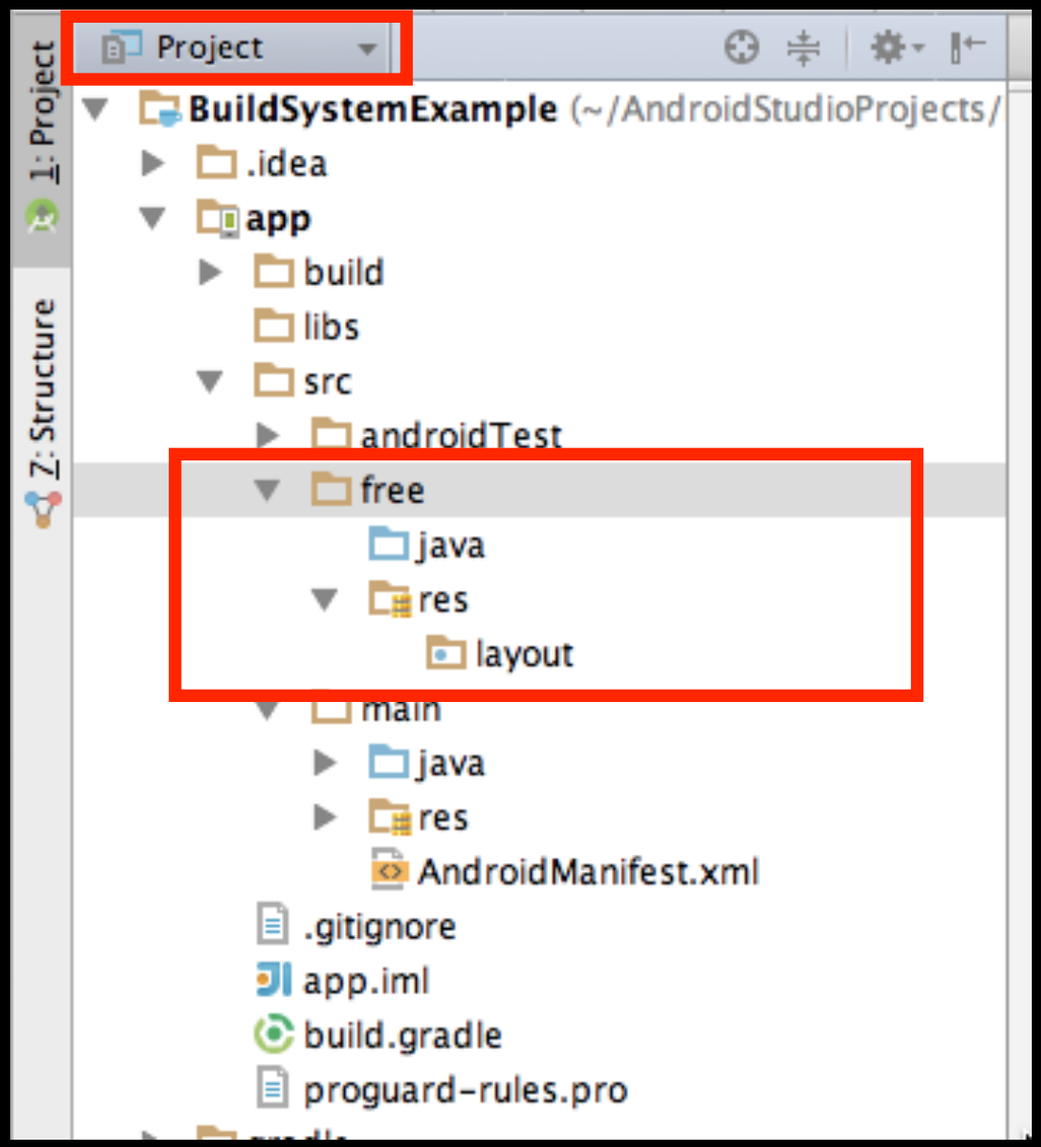
Step 2. Make sure grade performs a sync



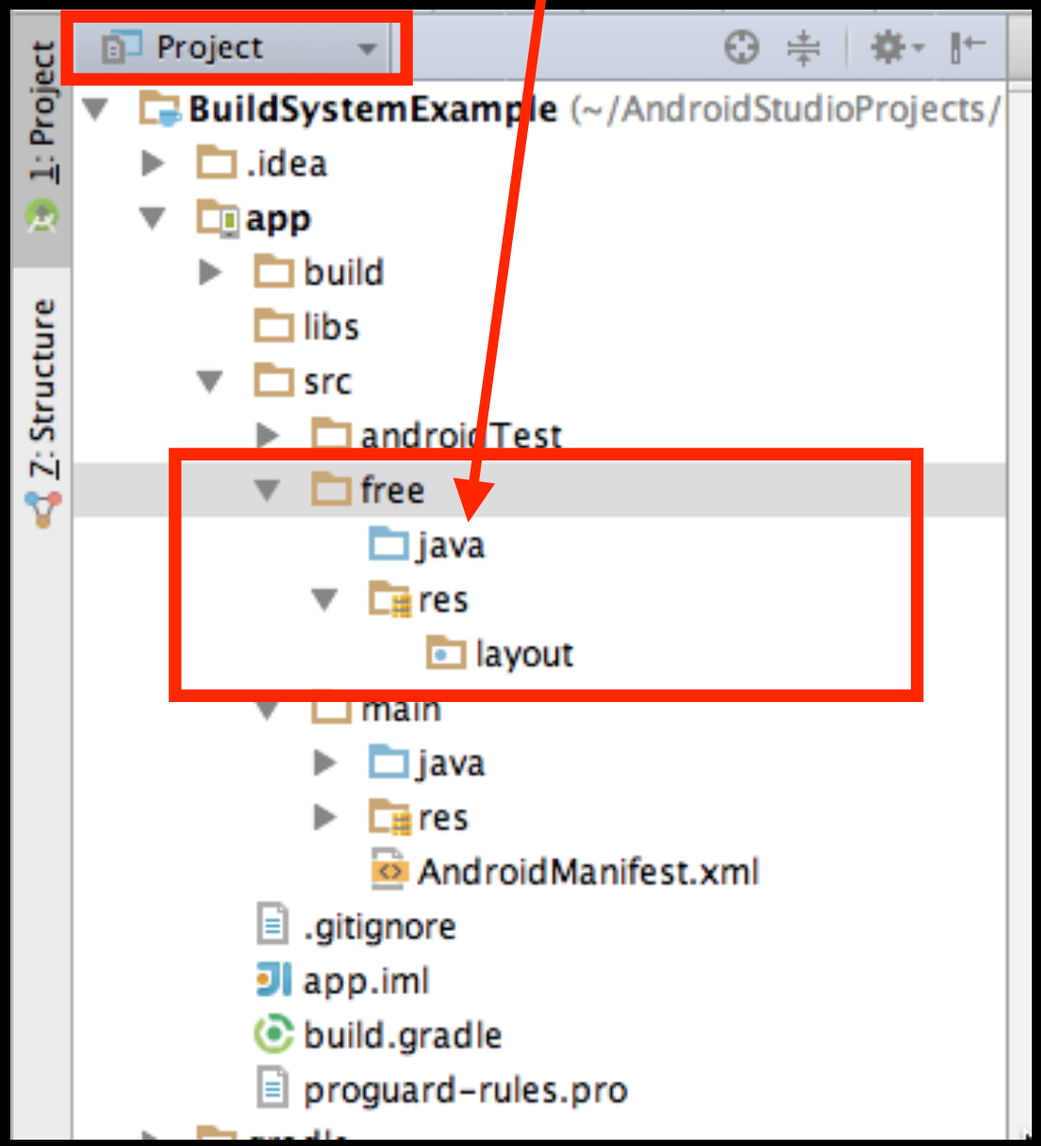
Step 3. Select the "freeDebug"



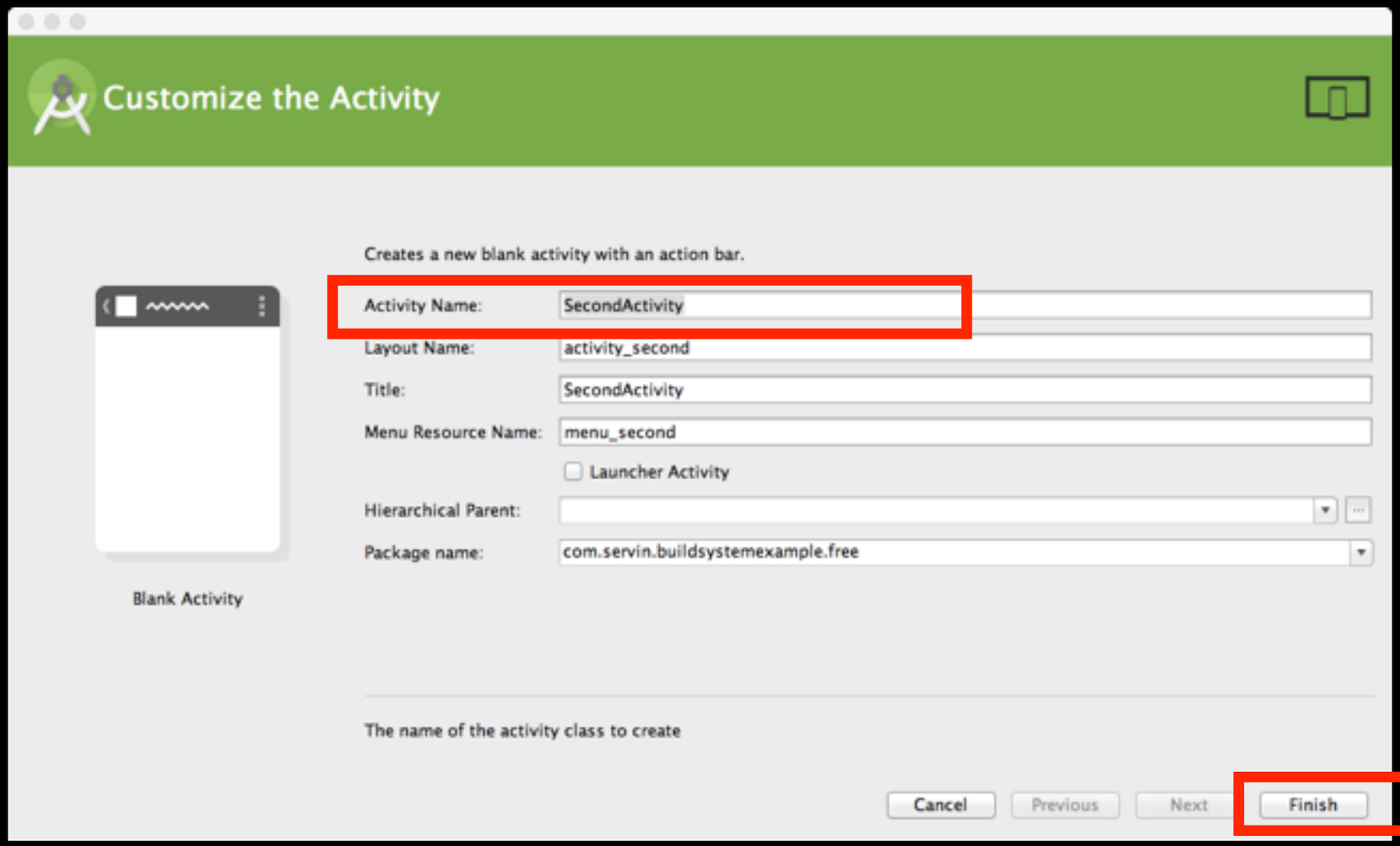
Step 4. Add Directories for “free” Product Flavor



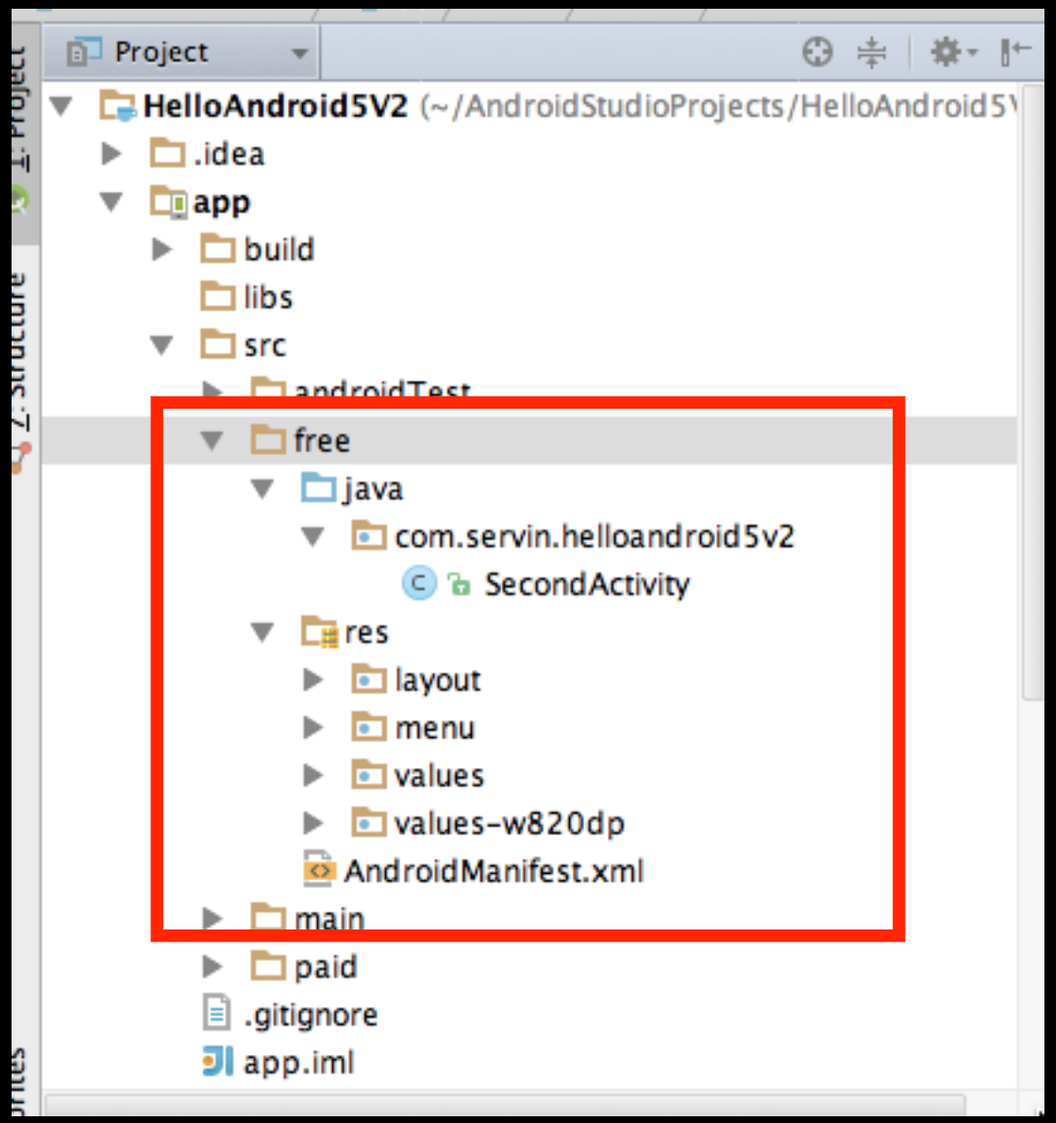
Step 5a. Right-Click on java, select New, Activity, Blank Activity



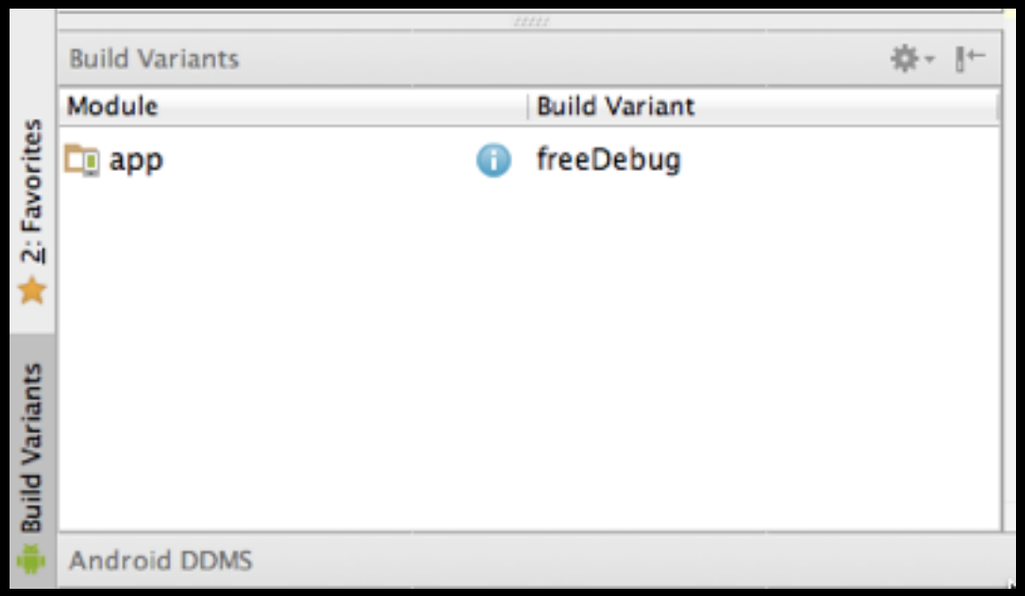
Step 5b. Add SecondActivity for “free” Product Flavor

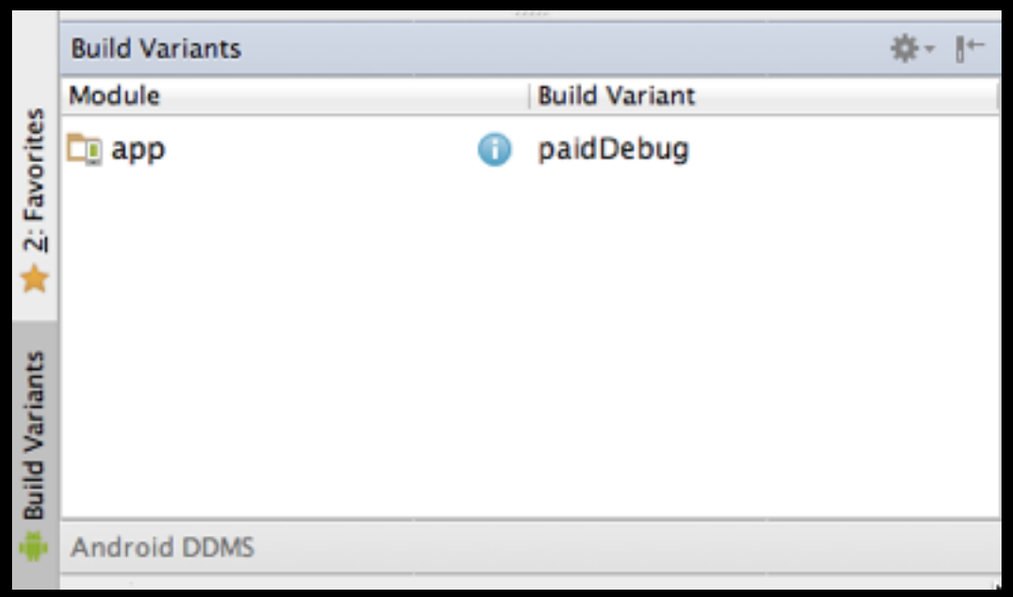


Directory Structure for “free” Product Flavor

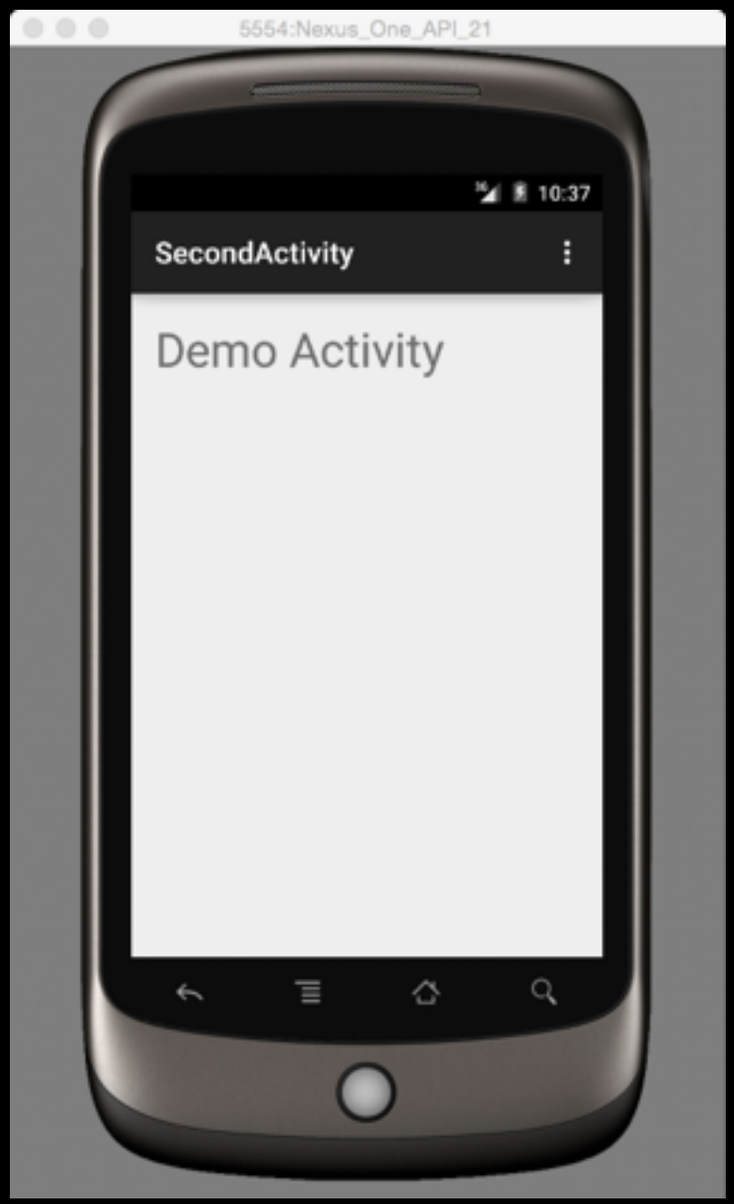


Selecting Different Product Flavors - freeDebug and paidDebug



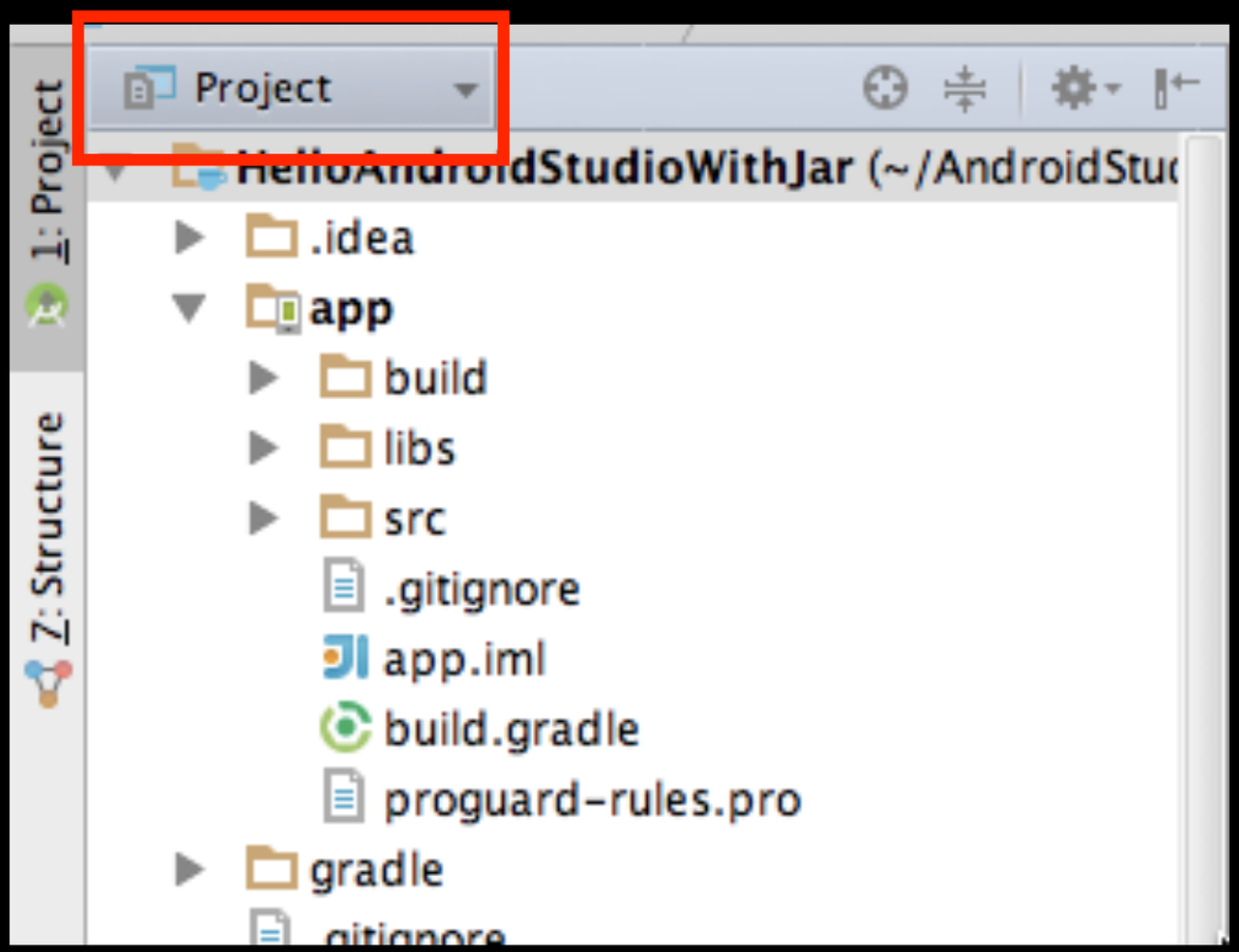


Running In Demo Mode

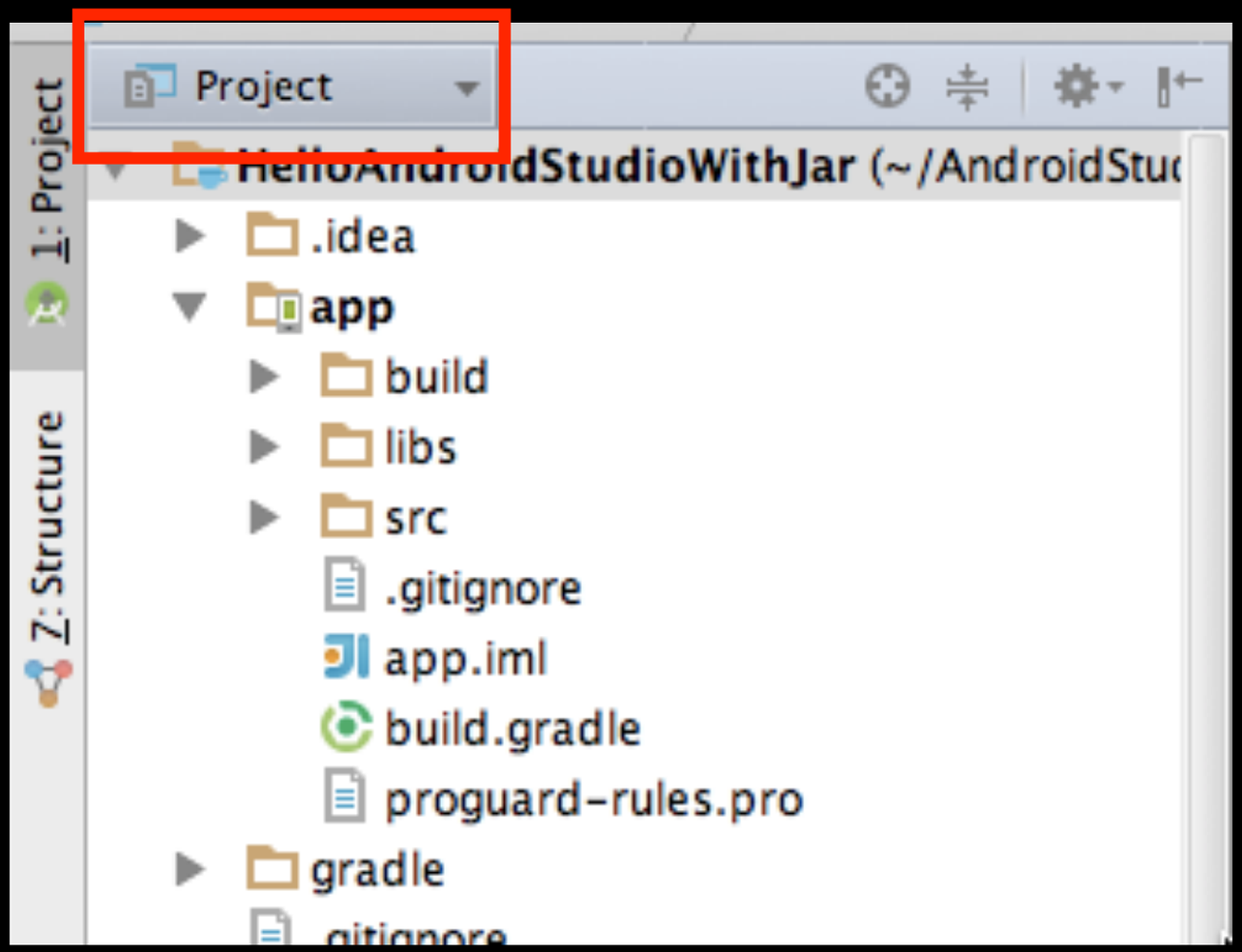


Adding a .jar file to your Android Studio Project

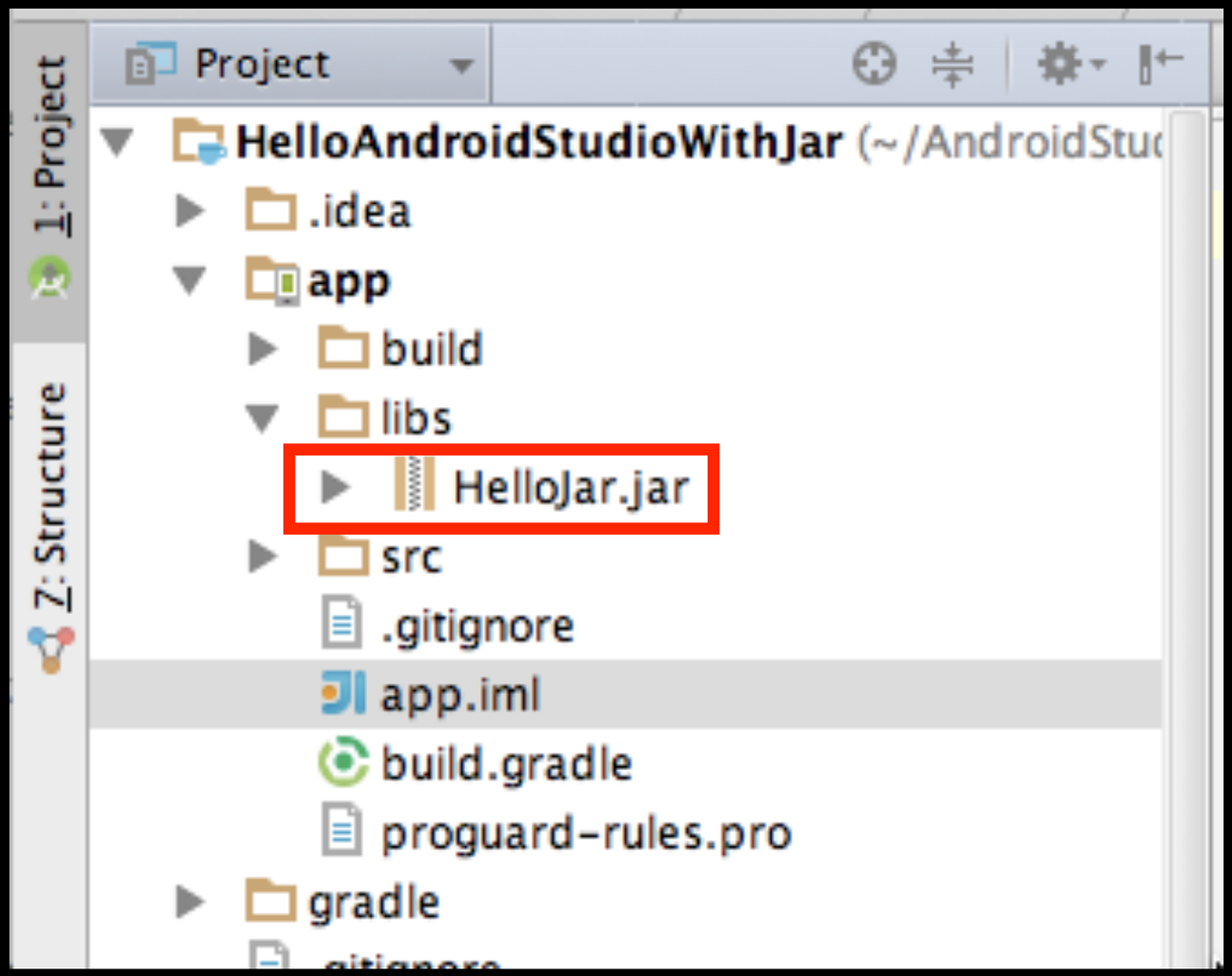
Step 1. Select **Project** View



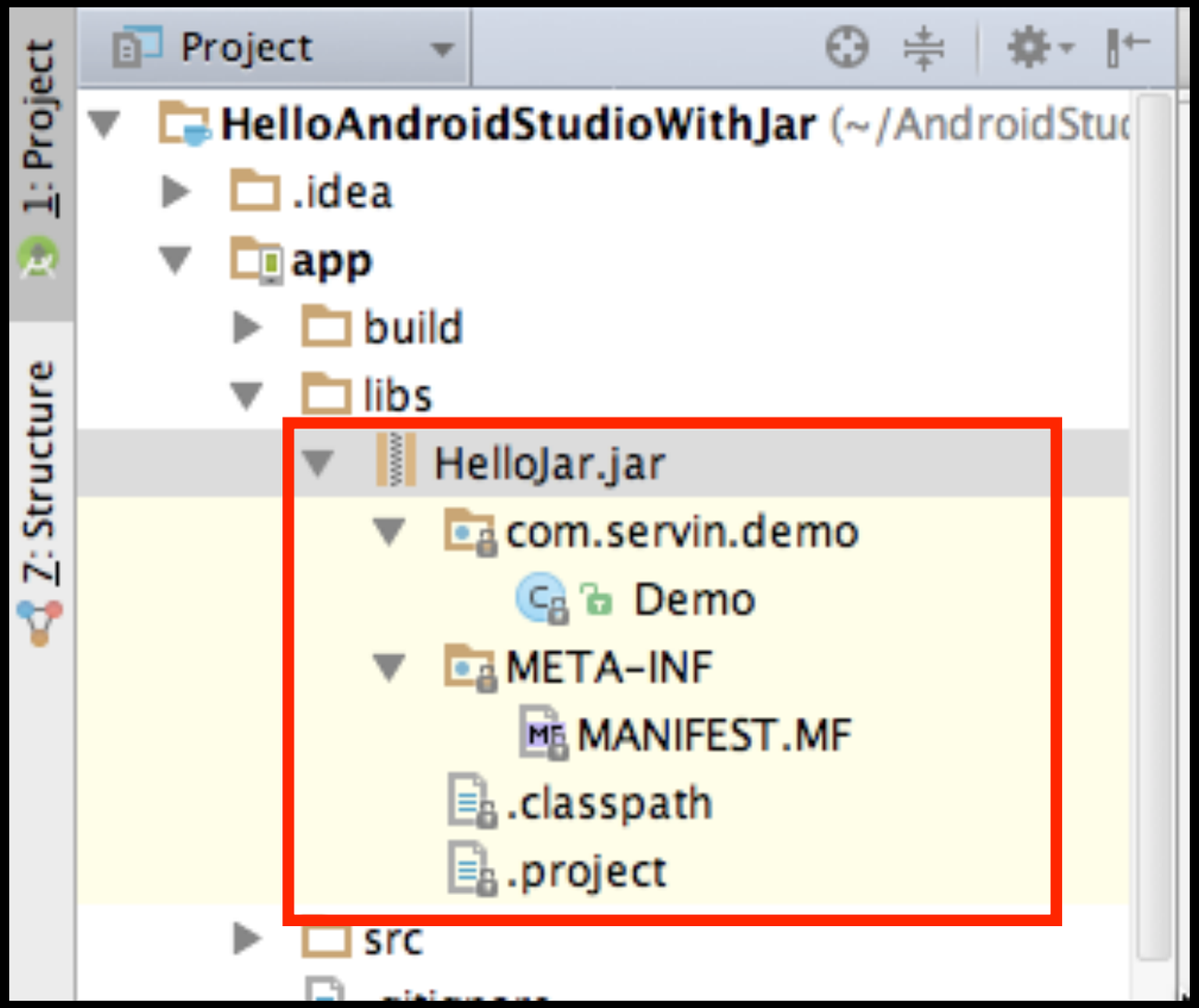
Step 2. Drag/Drop .jar file into **libs** directory



Step 3. Right-click on .jar file, select **Add as Library**



Step 4. Confirm Results



Step 5. Confirm gradle

```
dependencies {
    compile fileTree(dir: 'libs', include: ['*.jar'])
    compile 'com.android.support:appcompat-v7:21.0.3'
    compile files('libs/HelloJar.jar')
}
```

Step 5. Confirm In MainActivity

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

    Log.i("DEMO", "onCreate");

    Demo demo = new Demo();
    Log.i("DEMO", demo.sayHello());
}
```

Thank You!