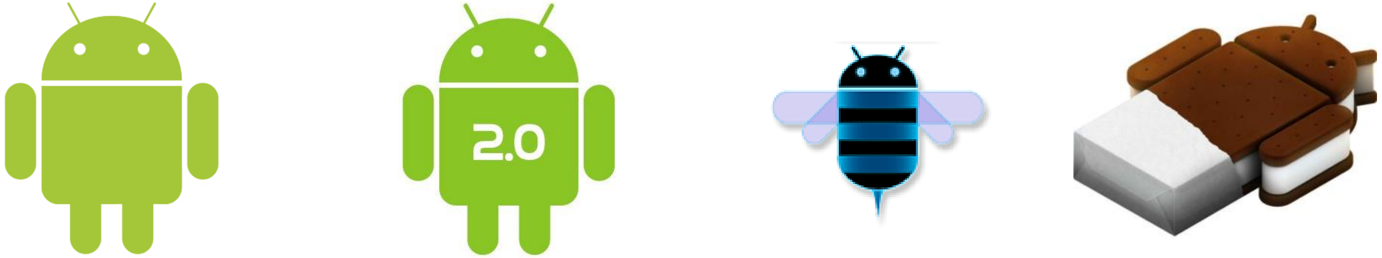


CHAPTER 2. SDK INSTALL AND INITIAL SETUP

Outline

- **Today:**
  - **How to setup a machine to start developing Android applications**
  - **An overview of an Android project**
  - **Some useful tools**
  - **Your first Android application**
    - **Maybe on a real device!**

Two options

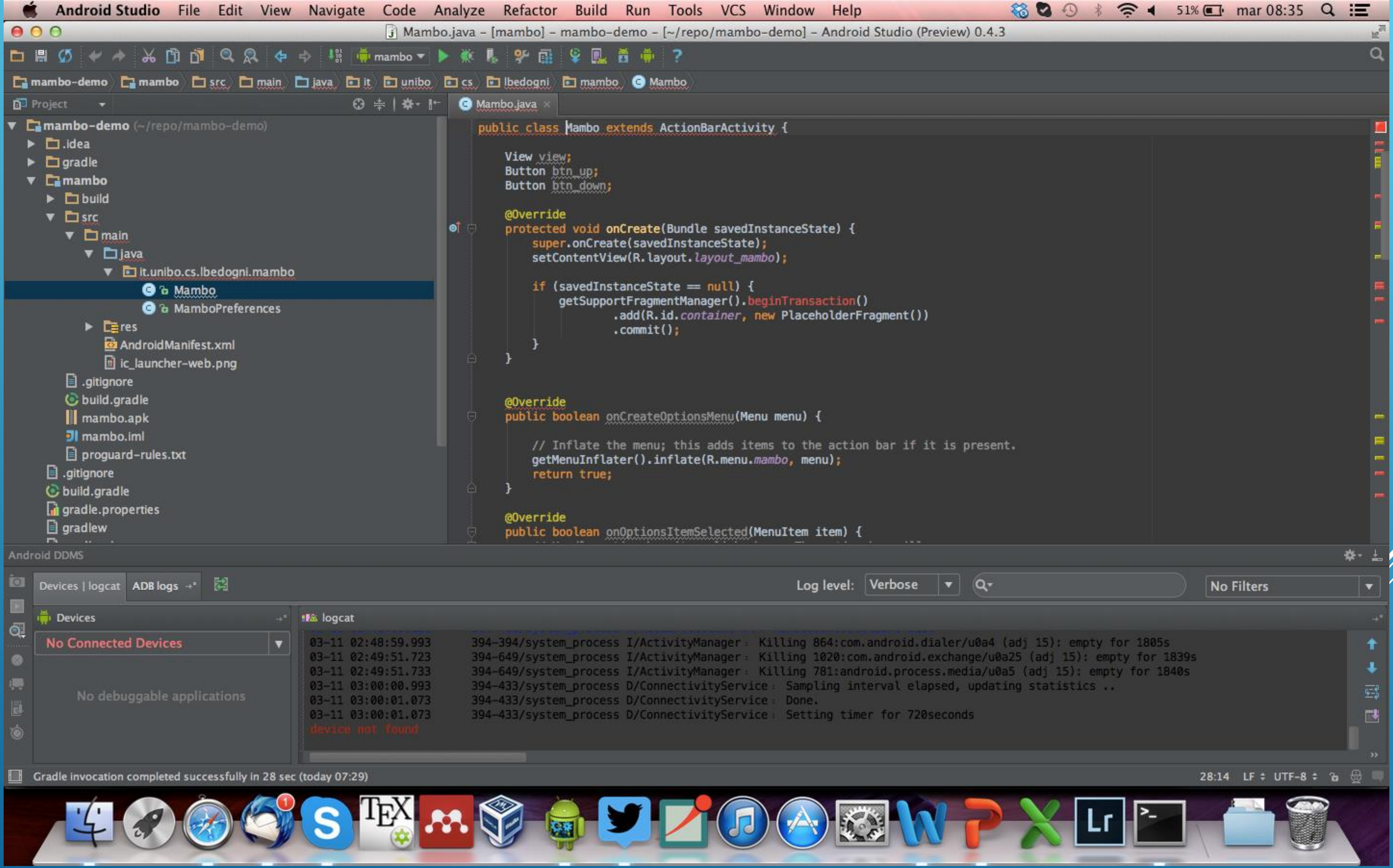
- Historically Android development was in Eclipse
  - Download the SDK
  - Install the Android plugin

- Now there is Android Studio
  - You still have to download the SDK
  - Customized environment

Which one?

- For the class, it’s the same
- If you already use and love Eclipse, go with it
- You might see examples done in both platforms
  - You can always import an Eclipse project into Android Studio
  - The inverse is difficult

Android Studio



```java
public class Mambo extends ActionBarActivity {

    View view;
    Button btn_up;
    Button btn_down;

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

        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.mambo, menu);
        return true;
    }

    @Override
    public boolean onOptionsItemSelected(MenuItem item) {
```

No Connected Devices

No debuggable applications

```
03-11 02:48:59.993     394-394/system_process I/ActivityManager﹕ Killing 864:com.android.dialer/u0a4 (adj 15): empty for 1805s
03-11 02:49:51.723     394-649/system_process I/ActivityManager﹕ Killing 1020:com.android.exchange/u0a25 (adj 15): empty for 1839s
03-11 02:49:51.733     394-649/system_process I/ActivityManager﹕ Killing 781:android.process.media/u0a5 (adj 15): empty for 1840s
03-11 03:00:00.993     394-433/system_process D/ConnectivityService﹕ Sampling interval elapsed, updating statistics ..
03-11 03:00:01.073     394-433/system_process D/ConnectivityService﹕ Done.
03-11 03:00:01.073     394-433/system_process D/ConnectivityService﹕ Setting timer for 720seconds
device not found
```

Gradle invocation completed successfully in 28 sec (today 07:29)

Where and how to get it

- Go to http://developer.android.com/sdk/installing/studio.html
- Download Android Studio and the SDK
- Install it and you're done!

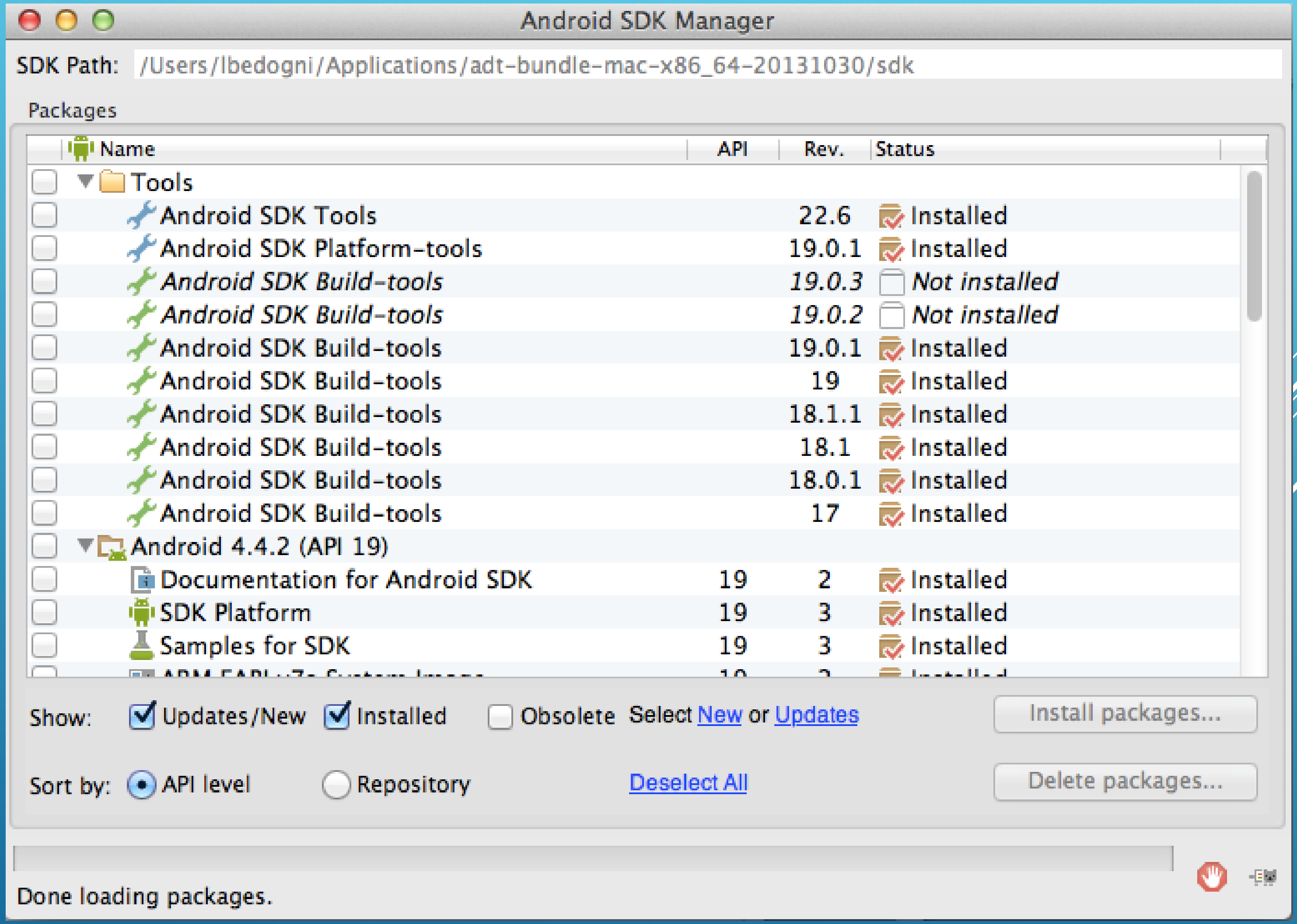
Basic requirements

- How to develop Android applications?
  - Linux/Mac OS X/Windows? Doesn’t matter
  - A real device is not mandatory

- Code your application in the IDE
  - Test it with the emulator
  - Deploy it on a real device

SDK tools

- Android tool
- Used to get APIs and add-ons
- You can also start it from Android Studio



Android SDK Manager

SDK Path: /Users/lbedogni/Applications/adt-bundle-mac-x86_64-20131030/sdk

Packages

	Name	API	Rev.	Status
☐	Tools			
☐	Android SDK Tools		22.6	Installed
☐	Android SDK Platform-tools		19.0.1	Installed
☐	*Android SDK Build-tools*		*19.0.3*	☐ *Not installed*
☐	*Android SDK Build-tools*		*19.0.2*	☐ *Not installed*
☐	Android SDK Build-tools		19.0.1	Installed
☐	Android SDK Build-tools		19	Installed
☐	Android SDK Build-tools		18.1.1	Installed
☐	Android SDK Build-tools		18.1	Installed
☐	Android SDK Build-tools		18.0.1	Installed
☐	Android SDK Build-tools		17	Installed
☐	Android 4.4.2 (API 19)			
☐	Documentation for Android SDK	19	2	Installed
☐	SDK Platform	19	3	Installed
☐	Samples for SDK	19	3	Installed
☐	ARM EABI v7a System Image	19	2	Installed

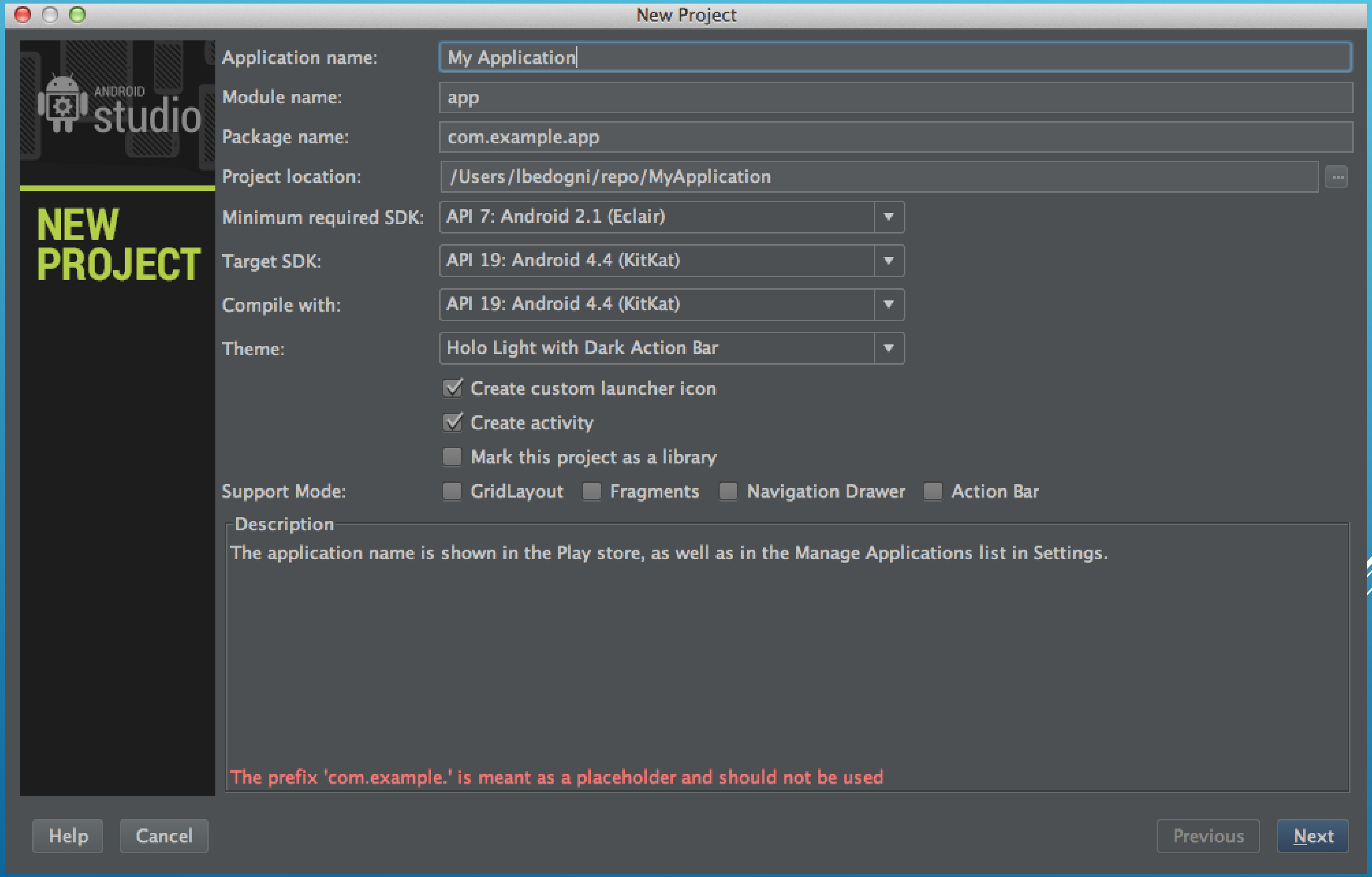
Show: ☑ Updates/New ☑ Installed ☐ Obsolete Select New or Updates

Sort by: ● API level ○ Repository

Done loading packages.

Your first Android application

- Go to File > New Project



Application name: My Application

Module name: app

Package name: com.example.app

Project location: /Users/lbedogni/repo/MyApplication

Minimum required SDK: API 7: Android 2.1 (Eclair)

Target SDK: API 19: Android 4.4 (KitKat)

Compile with: API 19: Android 4.4 (KitKat)

Theme: Holo Light with Dark Action Bar

☑ Create custom launcher icon

☑ Create activity

☐ Mark this project as a library

Support Mode: ☐ GridLayout ☐ Fragments ☐ Navigation Drawer ☐ Action Bar

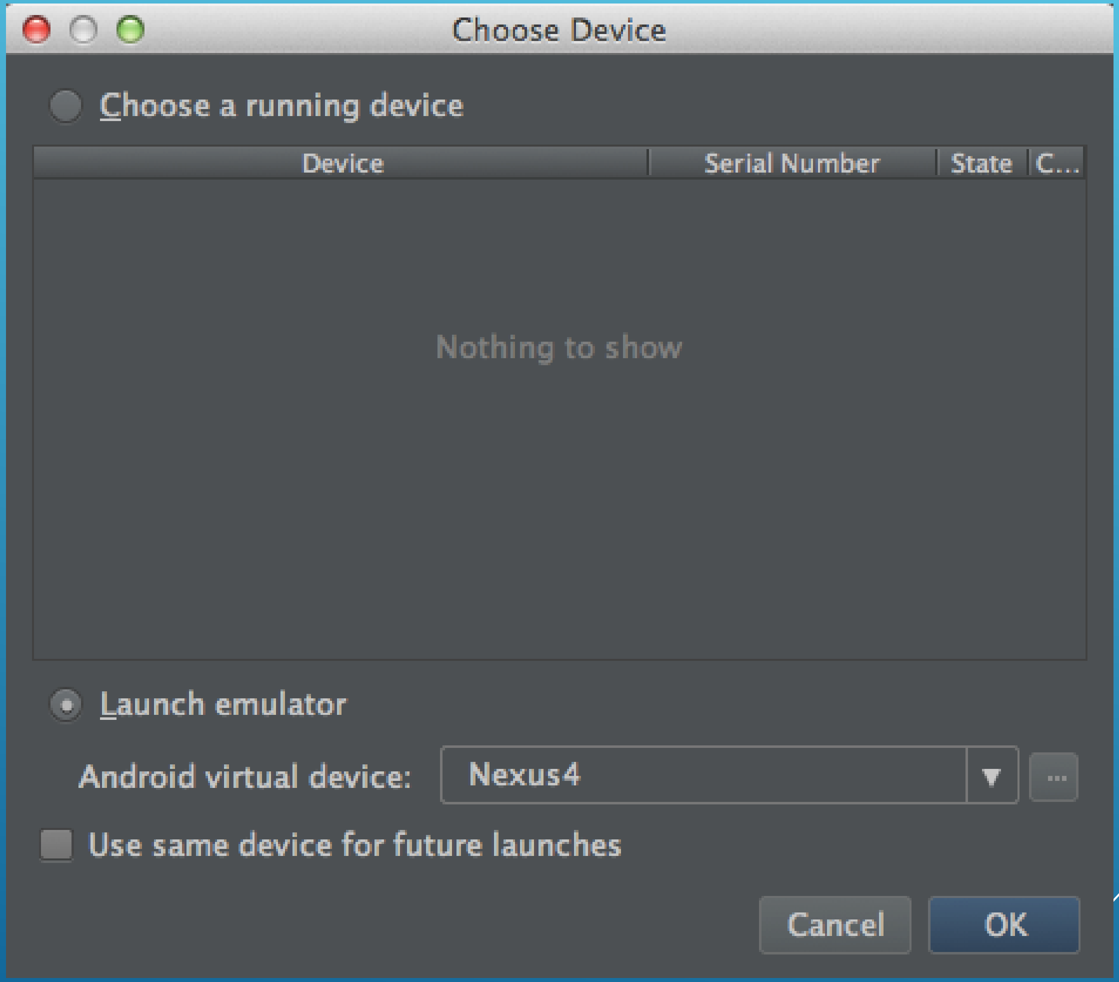
Description

The application name is shown in the Play store, as well as in the Manage Applications list in Settings.

The prefix 'com.example.' is meant as a placeholder and should not be used

Testing on the emulator

- Select "Launch Emulator"



Choose Device

○ Choose a running device

Device	Serial Number	State	C...

Nothing to show

● Launch emulator

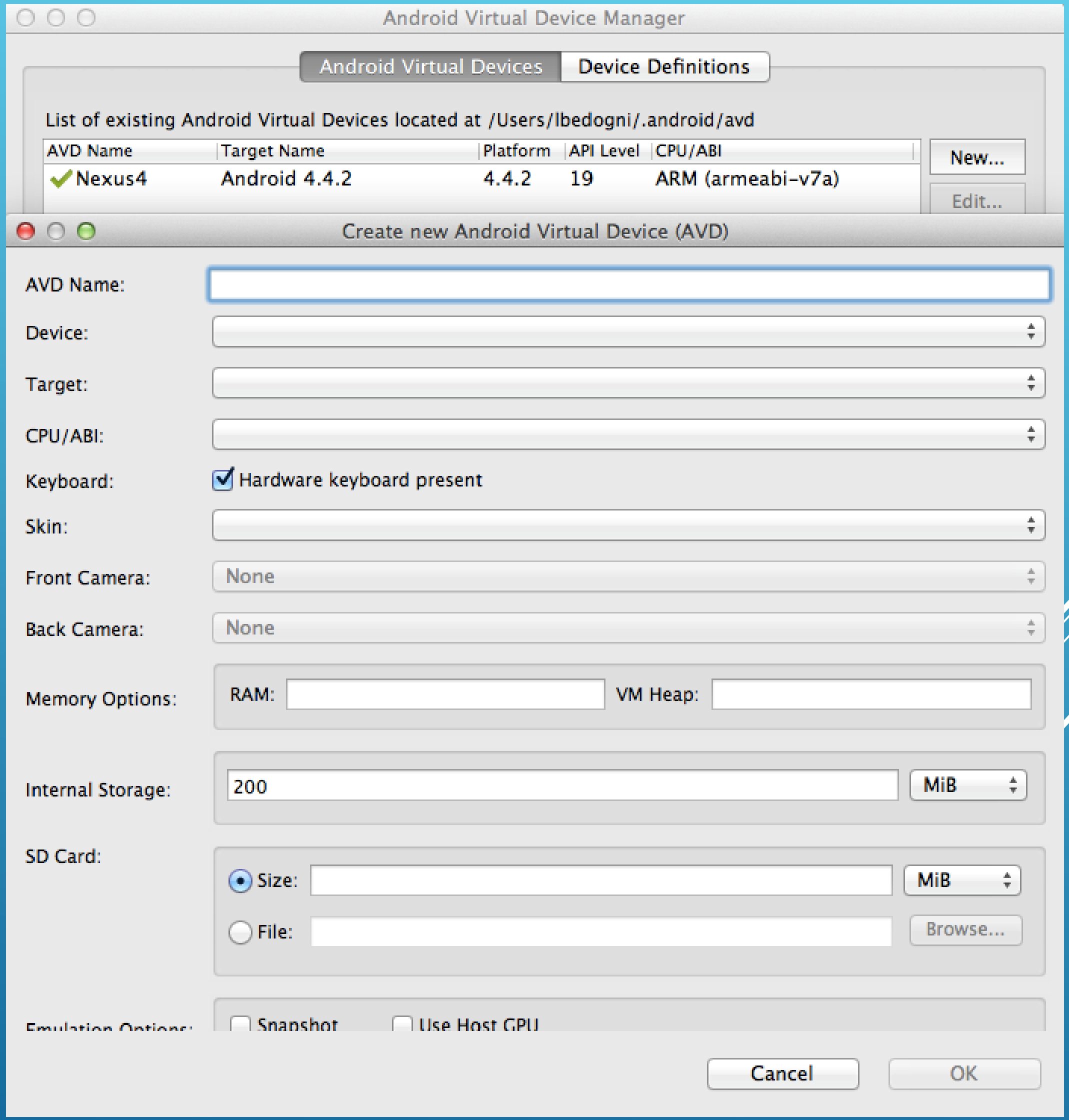
Android virtual device: Nexus4

□ Use same device for future launches

Creating an emulator

- Can configure several options

- You can create as many as you want



Android Virtual Device Manager

Android Virtual Devices | Device Definitions

List of existing Android Virtual Devices located at /Users/lbedogni/.android/avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ Nexus4	Android 4.4.2	4.4.2	19	ARM (armeabi-v7a)

New... Edit...

Create new Android Virtual Device (AVD)

AVD Name:

Device:

Target:

CPU/ABI:

Keyboard: ☑ Hardware keyboard present

Skin:

Front Camera: None

Back Camera: None

Memory Options: RAM: VM Heap:

Internal Storage: 200 MiB

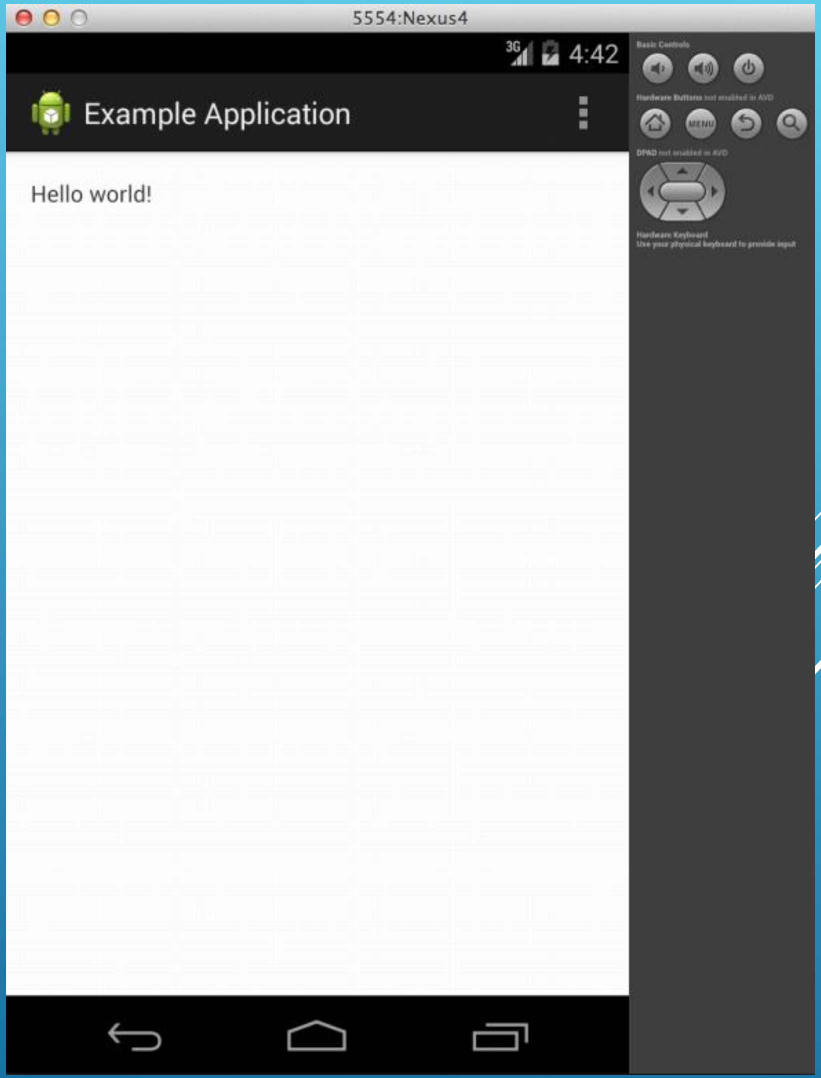
SD Card: ● Size: MiB ○ File: Browse...

Emulation Options: □ Snapshot □ Use Host GPU

Cancel OK

Did everything went fine?

- ➢ Test in on the emulator
- ➢ You should see something similar to this



```
Hello world!
```

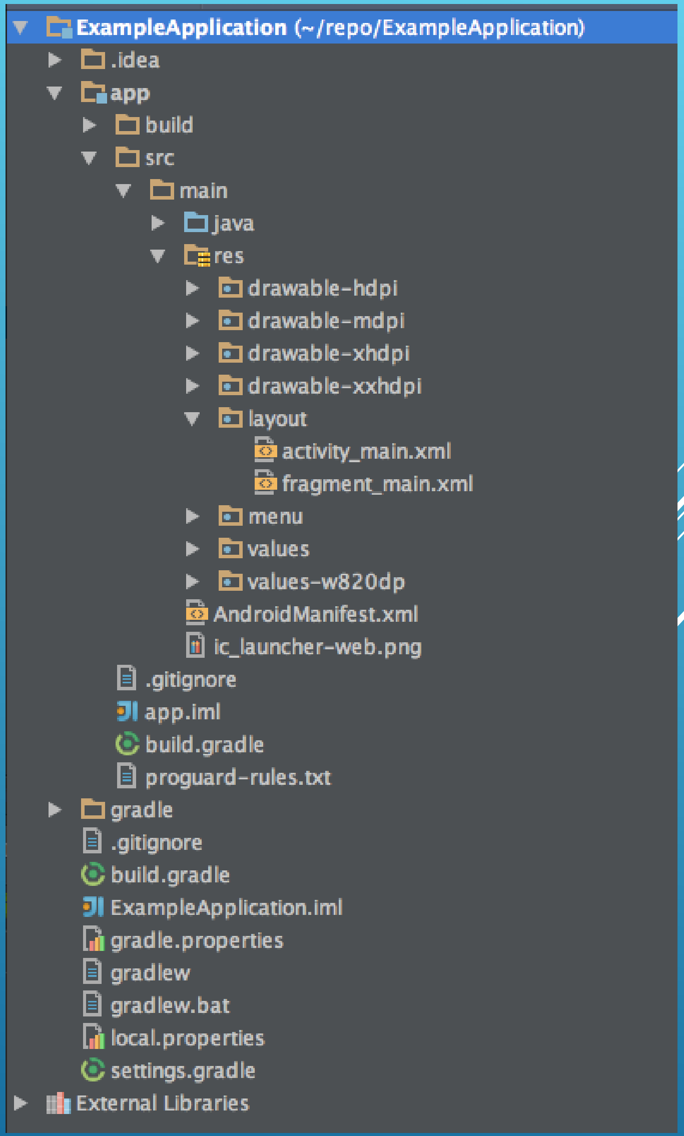
Hello world, Android!

- Anatomy of an application
  - Activity: what is started
  - View: what is seen
  - Intent: how to communicate with others

- Mix of XML and Java
  - Generally XML for layouts
  - Java for the app logic

Application structure

- Even the simplest application has a lot of files
- Some are autogenerated
- Each project contains
  - Activities
  - Layouts
  - XML files
  - AndroidManifest.xml



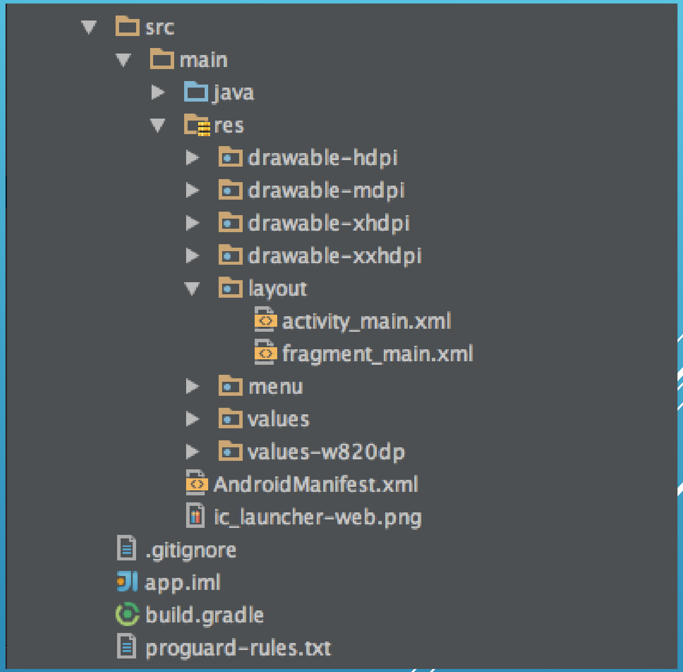
```
▼ ExampleApplication (~/repo/ExampleApplication)
    ▶ .idea
    ▼ app
        ▶ build
        ▼ src
            ▼ main
                ▶ java
                ▼ res
                    ▶ drawable-hdpi
                    ▶ drawable-mdpi
                    ▶ drawable-xhdpi
                    ▶ drawable-xxhdpi
                    ▼ layout
                          activity_main.xml
                          fragment_main.xml
                    ▶ menu
                    ▶ values
                    ▶ values-w820dp
                  AndroidManifest.xml
                  ic_launcher-web.png
          .gitignore
          app.iml
          build.gradle
          proguard-rules.txt
    ▶ gradle
      .gitignore
      build.gradle
      ExampleApplication.iml
      gradle.properties
      gradlew
      gradlew.bat
      local.properties
      settings.gradle
▶ External Libraries
```

build

- Autogenerated files
  - No need to modify them
- Some are autogenerated
- Each project contains
  - Activities
  - Layouts
  - XML files
  - AndroidManifest.xml

src

- All the code of your application
  - The java folder contains… the java code!
  - Inside res there are a lot of resources
    - Images
    - Layouts
    - Xml files
    - AndroidManifest.xml



```
▼ src
  ▼ main
    ▶ java
    ▼ res
      ▶ drawable-hdpi
      ▶ drawable-mdpi
      ▶ drawable-xhdpi
      ▶ drawable-xxhdpi
      ▼ layout
          activity_main.xml
          fragment_main.xml
      ▶ menu
      ▶ values
      ▶ values-w820dp
      AndroidManifest.xml
      ic_launcher-web.png
  .gitignore
  app.iml
  build.gradle
  proguard-rules.txt
```

AndroidManifest.xml

- Mandatory file for every application
  - Contains:
    - Application declaration
    - Permissions
    - Intent filters
    - Targets

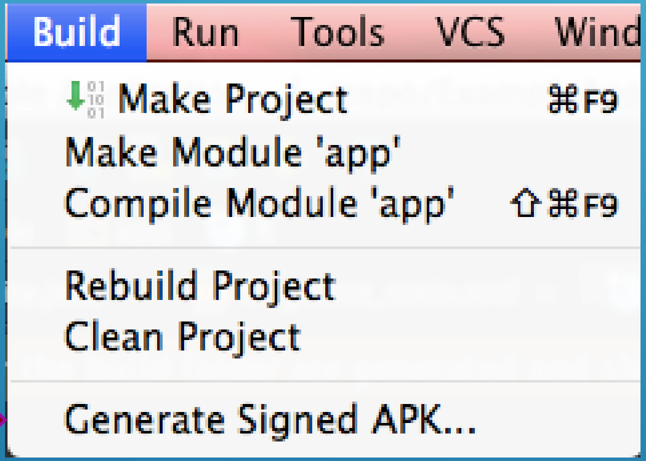
AndroidManifest.xml

How to test

- Via an AVD
  - Fast, possible to have different resolutions, APIs
  - Not a real device
- On a real device
  - You get exactly what is deployed
  - Must own a real device
- So?
  - Test in on AVDs.
  - When you feel that the application is ready, go with a real device

How to deploy

- Android applications must be signed before installing them on a real device



You need a key for this and you can generate one from the menu.

ECLIPSE

3 step setup

- Download Android SDK for your platform:

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

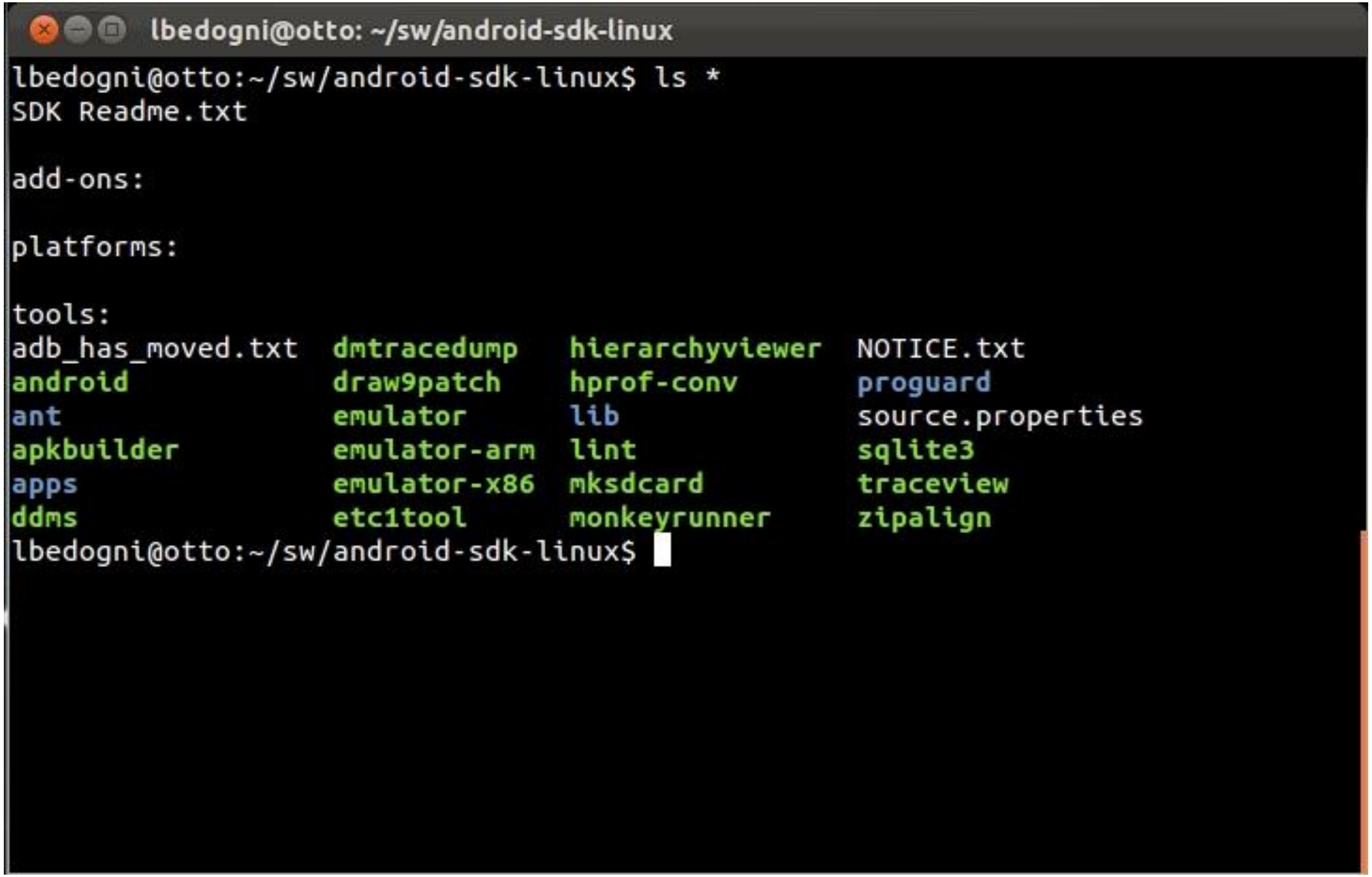
- Execute (and then select the Android API version):

android-sdk-xxx/tools/android

- Install the ADT plugin for Eclipse:

http://www.eclipse.org/downloads/
https://dl-ssl.google.com/android/eclipse

SDK



```
lbedogni@otto:~/sw/android-sdk-linux$ ls *
SDK Readme.txt

add-ons:

platforms:

tools:
adb_has_moved.txt  dmtracedump    hierarchyviewer  NOTICE.txt
android            draw9patch     hprof-conv       proguard
ant                emulator       lib              source.properties
apkbuilder         emulator-arm   lint             sqlite3
apps               emulator-x86   mksdcard         traceview
ddms               etc1tool       monkeyrunner     zipalign
lbedogni@otto:~/sw/android-sdk-linux$ 
```

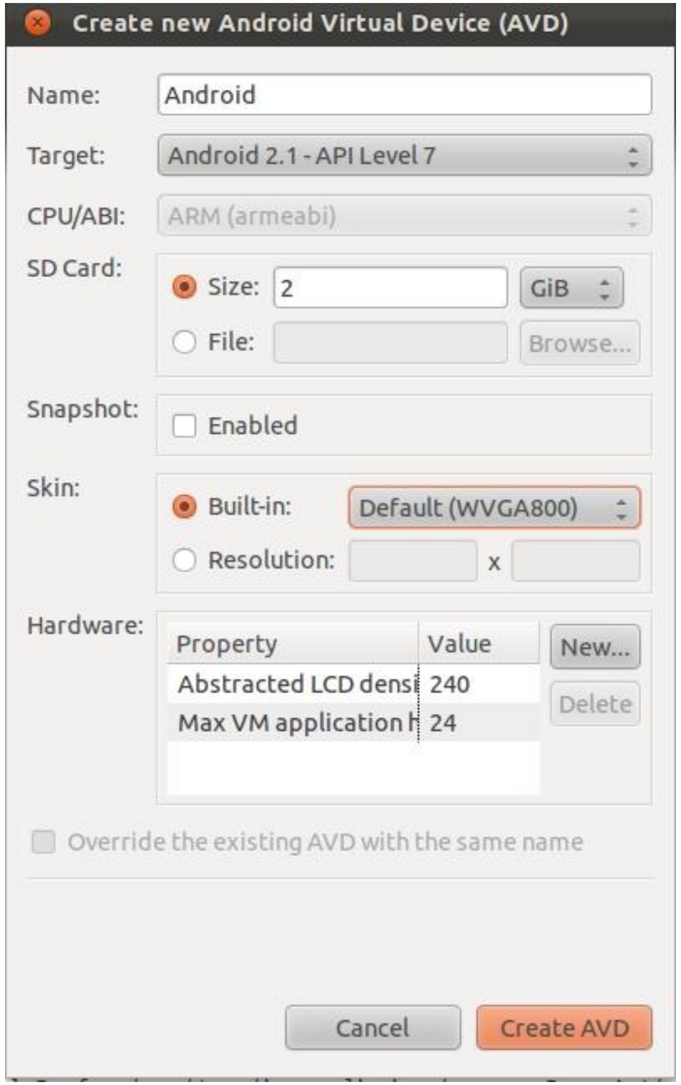
ADT ECLIPSE

- Useful to run applications via Eclipse, highly reccomended
- Go to http://developer.android.com/sdk/eclipse-adt.html
- Pick the repository (actually https://dl-ssl.google.com/android/eclipse/)
- Add a repository in Eclipse and download the "Developers Tools"
- This will make a virtual bridge between eclipse and the SDK

PROJECT

- Create it under Eclipse
  - Assign an SDK target
  - Choose an application name
  - Choose a package name
  - Create an activity and assign a name
- Run it to test that everything is ok

AVD



Create new Android Virtual Device (AVD)

Name: Android

Target: Android 2.1 - API Level 7

CPU/ABI: ARM (armeabi)

SD Card:
● Size: 2 GiB
○ File:

Snapshot: □ Enabled

Skin:
● Built-in: Default (WVGA800)
○ Resolution: x

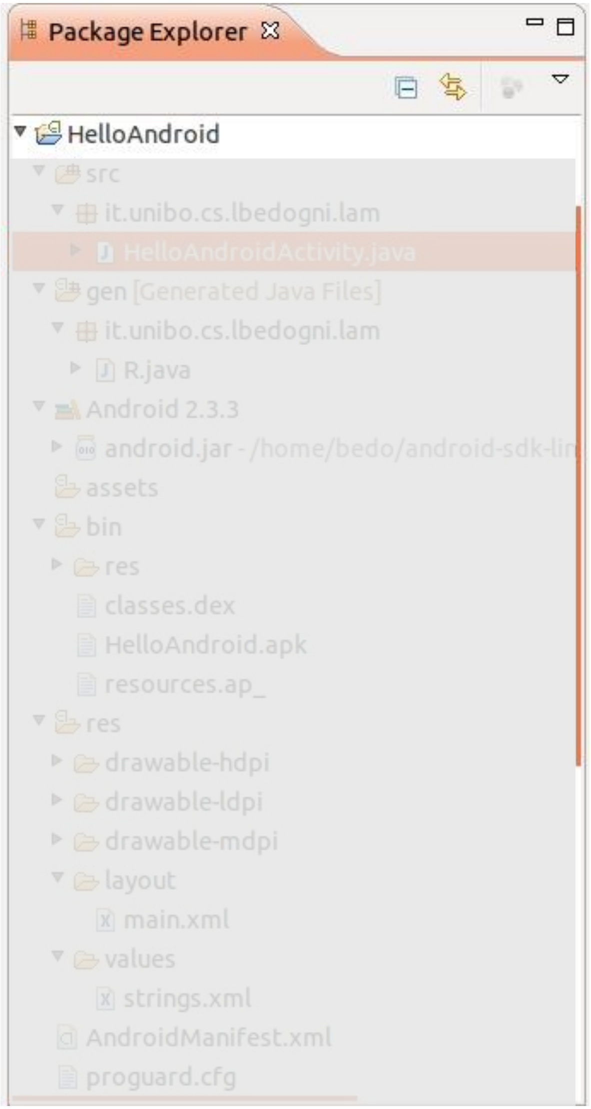
Hardware:

Property	Value
Abstracted LCD densi	240
Max VM application h	24

□ Override the existing AVD with the same name

- AVD means Android Virtual Device
- Test the application before running it on a device
- Multiple APIs → Multiple targets
- Makes it faster (and cheap) to test application on different configurations/resolutions/storage

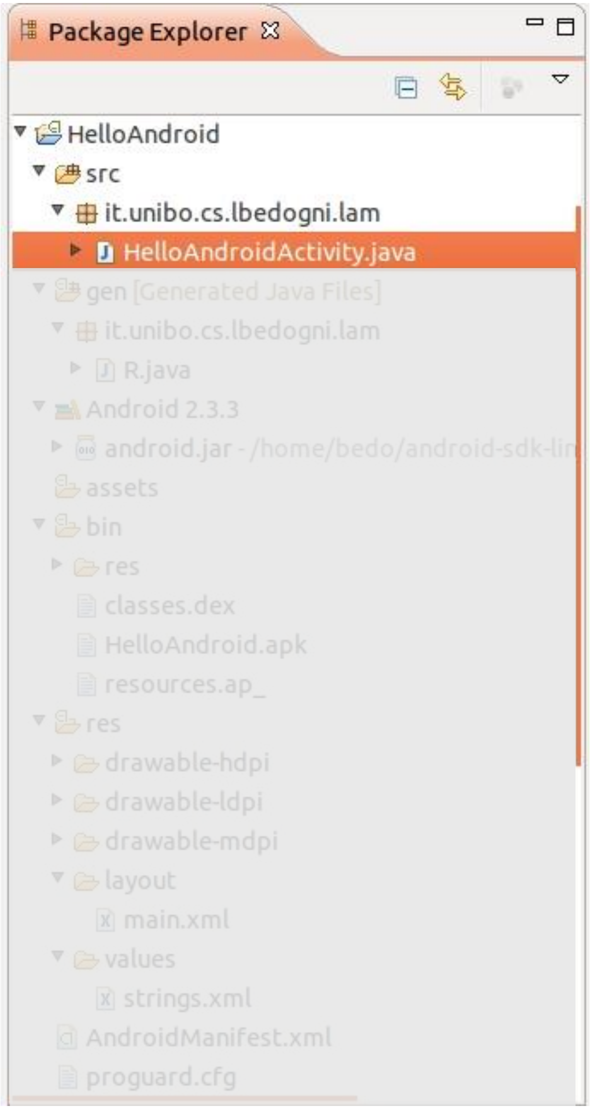
CONTENTS



```
HelloAndroid
  src
    it.unibo.cs.lbedogni.lam
      HelloAndroidActivity.java
  gen [Generated Java Files]
    it.unibo.cs.lbedogni.lam
      R.java
  Android 2.3.3
    android.jar - /home/bedo/android-sdk-lin
  assets
  bin
    res
    classes.dex
    HelloAndroid.apk
    resources.ap_
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  proguard.cfg
```

- Project name

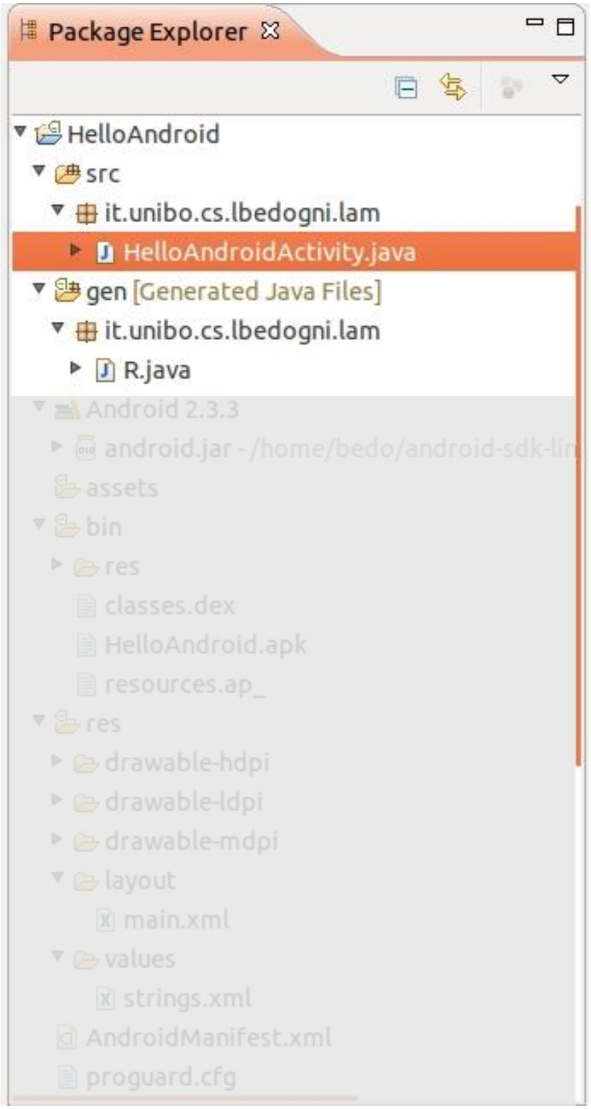
CONTENTS



```
Package Explorer
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
    assets
  ▼ bin
    ▶ res
      classes.dex
      HelloAndroid.apk
      resources.ap_
  ▼ res
    ▶ drawable-hdpi
    ▶ drawable-ldpi
    ▶ drawable-mdpi
    ▼ layout
        main.xml
    ▼ values
        strings.xml
    AndroidManifest.xml
    proguard.cfg
```

- Project name
- Src folder with java files

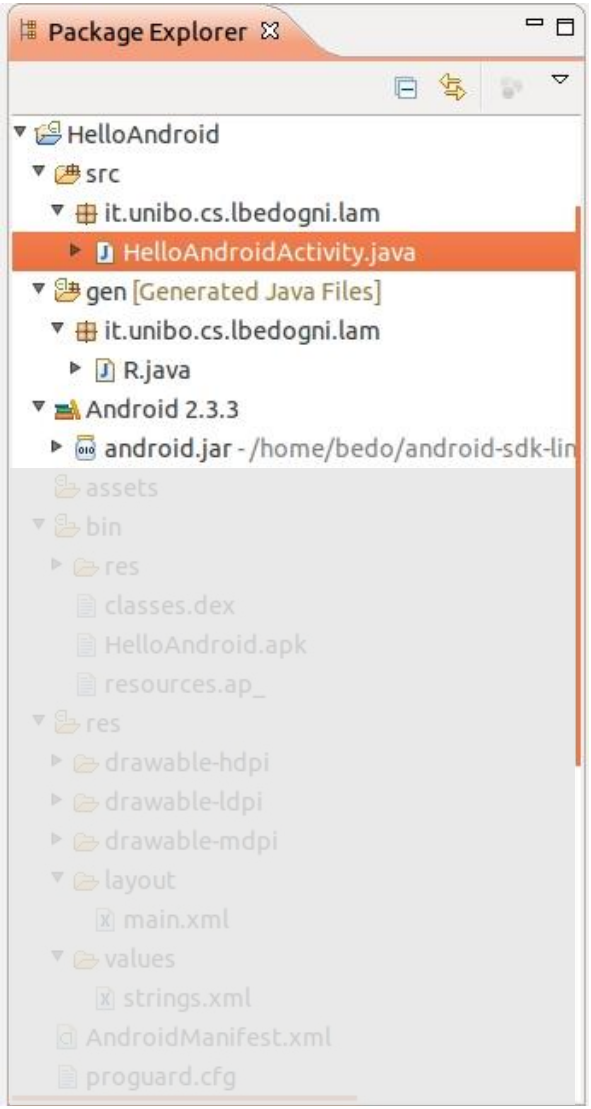
CONTENTS



```
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
    assets
  ▼ bin
    ▶ res
      classes.dex
      HelloAndroid.apk
      resources.ap_
  ▼ res
    ▶ drawable-hdpi
    ▶ drawable-ldpi
    ▶ drawable-mdpi
    ▼ layout
        main.xml
    ▼ values
        strings.xml
    AndroidManifest.xml
    proguard.cfg
```

- Project name
- Src folder with java files
- Auto-generated files

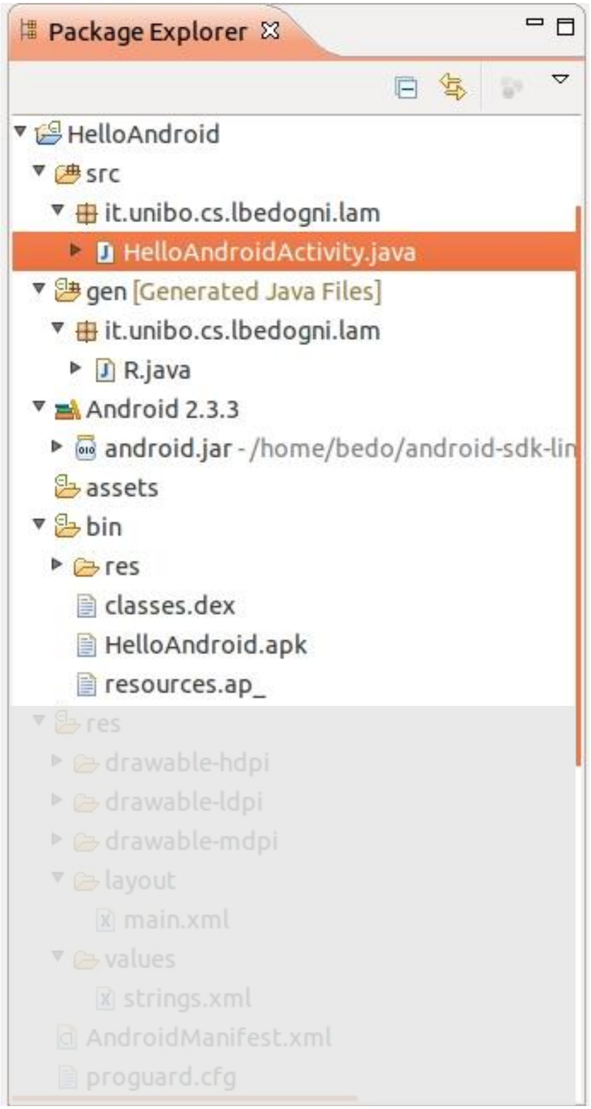
CONTENTS



```
HelloAndroid
  src
    it.unibo.cs.lbedogni.lam
      HelloAndroidActivity.java
  gen [Generated Java Files]
    it.unibo.cs.lbedogni.lam
      R.java
  Android 2.3.3
    android.jar - /home/bedo/android-sdk-lin
  assets
  bin
    res
    classes.dex
    HelloAndroid.apk
    resources.ap_
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  proguard.cfg
```

- Project name
- Src folder with java files
- Auto-generated files
- Android's base files

CONTENTS

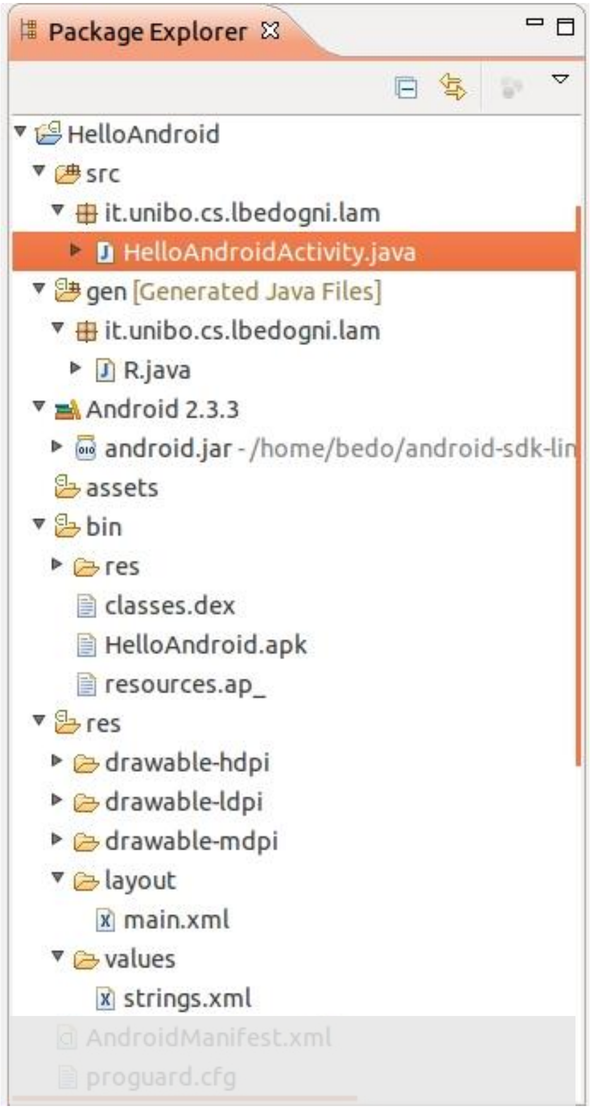


```
Package Explorer
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
  assets
  ▼ bin
    ▶ res
      classes.dex
      HelloAndroid.apk
      resources.ap_
  ▼ res
    ▶ drawable-hdpi
    ▶ drawable-ldpi
    ▶ drawable-mdpi
    ▼ layout
        main.xml
    ▼ values
        strings.xml
    AndroidManifest.xml
    proguard.cfg
```

- Project name
- Src folder with java files
- Auto-generated files
- Android's base files

- Compiled files

CONTENTS



Package Explorer

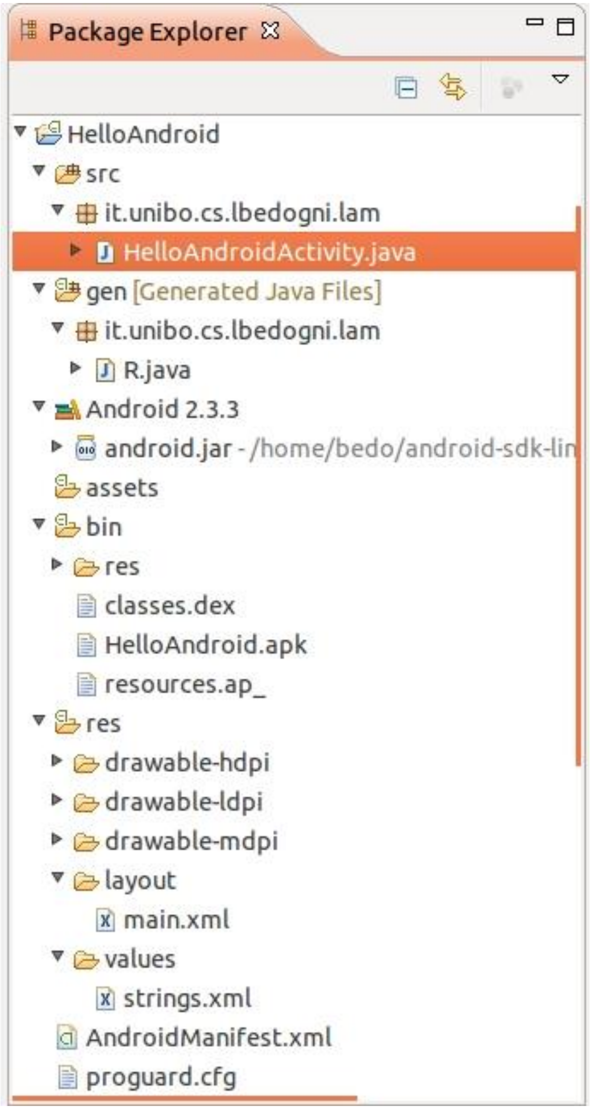
```
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
  assets
  ▼ bin
    ▶ res
      classes.dex
      HelloAndroid.apk
      resources.ap_
  ▼ res
    ▶ drawable-hdpi
    ▶ drawable-ldpi
    ▶ drawable-mdpi
    ▼ layout
        main.xml
    ▼ values
        strings.xml
  AndroidManifest.xml
  proguard.cfg
```

- Project name
- Src folder with java files
- Auto-generated files
- Android's base files

- Compiled files

- Resources files

CONTENTS



Package Explorer

```
HelloAndroid
  src
    it.unibo.cs.lbedogni.lam
      HelloAndroidActivity.java
  gen [Generated Java Files]
    it.unibo.cs.lbedogni.lam
      R.java
  Android 2.3.3
    android.jar - /home/bedo/android-sdk-lin
  assets
  bin
    res
    classes.dex
    HelloAndroid.apk
    resources.ap_
  res
    drawable-hdpi
    drawable-ldpi
    drawable-mdpi
    layout
      main.xml
    values
      strings.xml
  AndroidManifest.xml
  proguard.cfg
```

- Project name
- Src folder with java files
- Auto-generated files
- Android's base files

- Compiled files

- Resources files
- Android Manifest

DEPLOY

- Android applications must be signed before they can be installed on a device
- Eclipse can sign applications
  - Debug mode, just to test it on your device
  - Release mode, when it's ready for other users

SIGNING

- Eclipse has a tool called Export Wizard
  - File > Export
  - Export Android Application
  - Select your key and preferences
  - Application is compiled, signed and aligned, ready to be deployed
- Keep your private key safe
  - Use a strong password
  - Don't lend it to anyone