Programming with Android: SDK install and initial setup

Luca Bedogni Marco Di Felice

Dipartimento di Informatica: Scienza e Ingegneria

Università di Bologna

SDK 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!

SDK and initial setup: 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

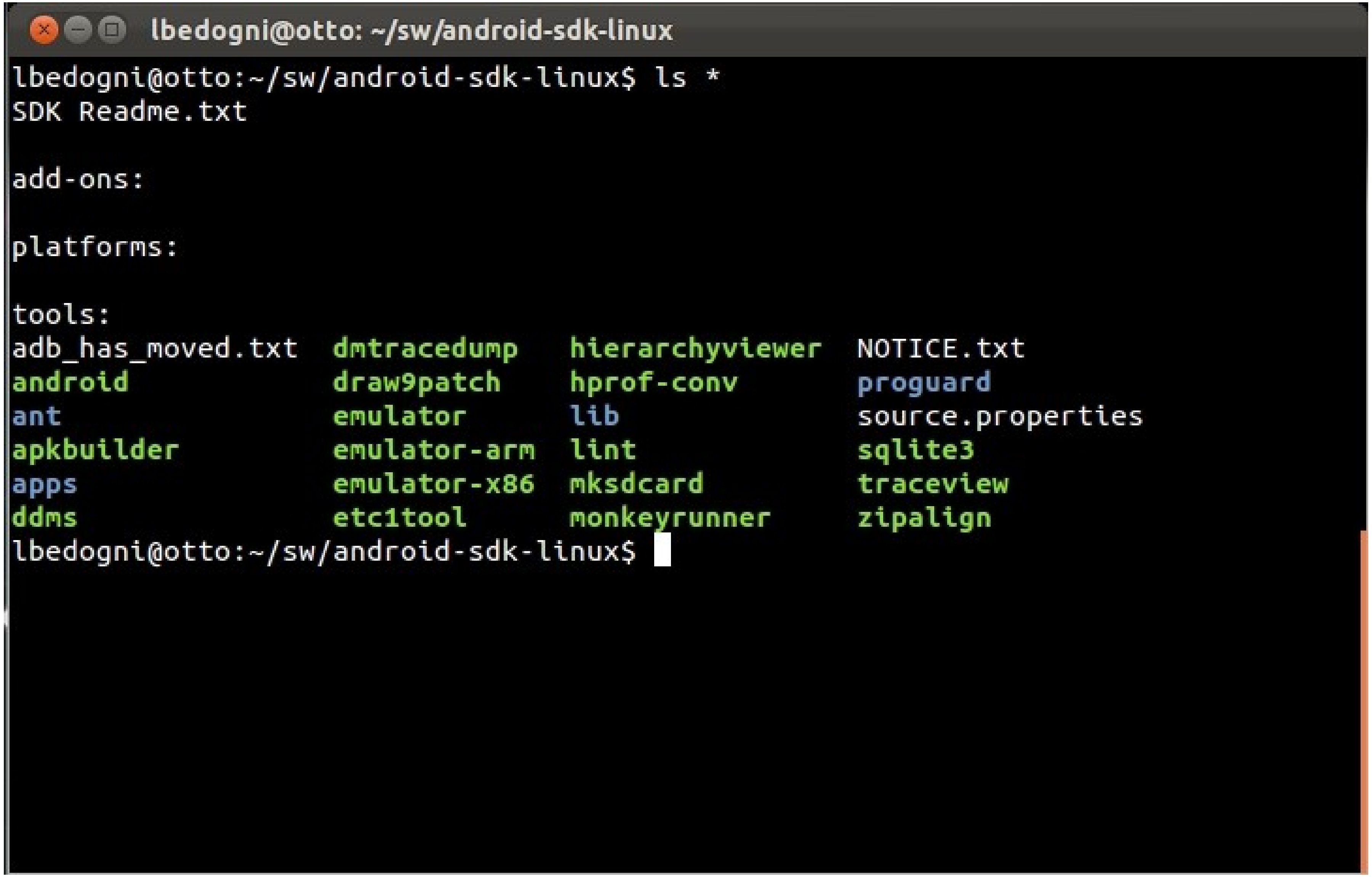
Basics: requirements

- How to develop Android Applications?
  - Linux/MacOS X/Windows? Doesn't matter
  - Android SDK
  - Eclipse, not mandatory but can help
  - Eclipse Plugin
  - An Android device is not required

SDK: download and unpack

- Go to http://developer.android.com/sdk/index.html
- Download the SDK according to your OS
- Unpack it
- Done!
- Let's see what's inside...

Inside the SDK



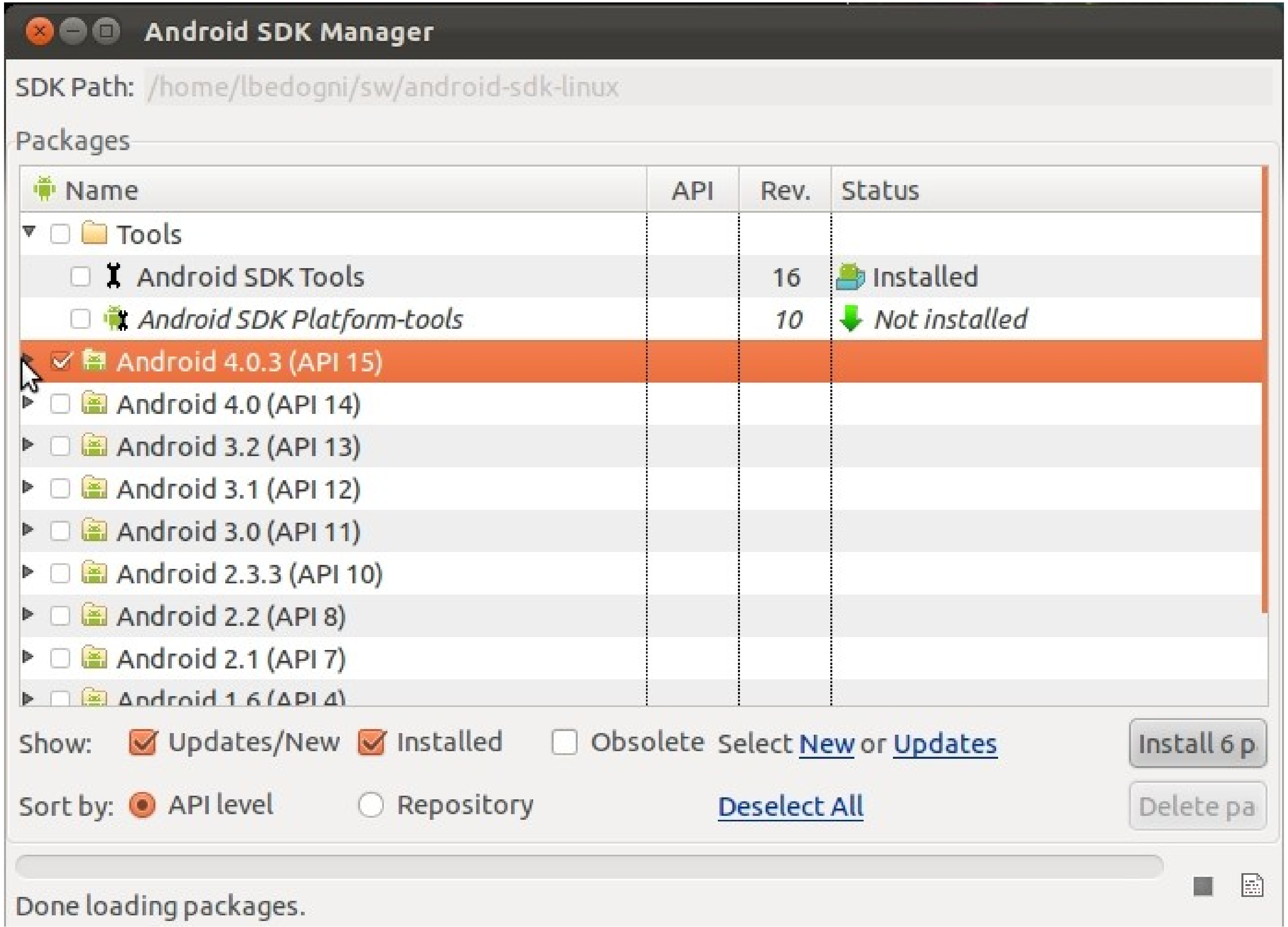
```
lbedogni@otto:~/sw/android-sdk-linux$ ls *
SDK Readme.txt

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$
```

SDK tools: android



Android SDK Manager

SDK Path: /home/lbedogni/sw/android-sdk-linux

Packages

Name	API	Rev.	Status
▼ □ Tools			
□ Android SDK Tools		16	Installed
□ *Android SDK Platform-tools*		*10*	*Not installed*
▶ ☑ Android 4.0.3 (API 15)			
▶ □ Android 4.0 (API 14)			
▶ □ Android 3.2 (API 13)			
▶ □ Android 3.1 (API 12)			
▶ □ Android 3.0 (API 11)			
▶ □ Android 2.3.3 (API 10)			
▶ □ Android 2.2 (API 8)			
▶ □ Android 2.1 (API 7)			
▶ □ Android 1.6 (API 4)			

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

Sort by: ● API level ○ Repository Deselect All

Done loading packages.

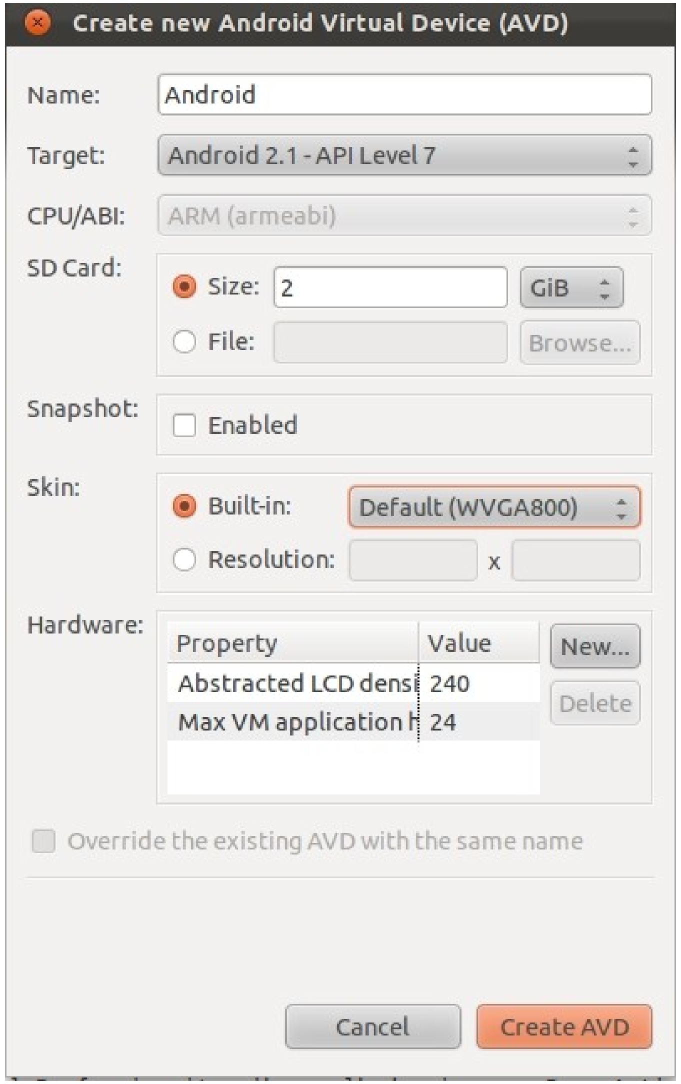
ADT plugin for Eclipse

- 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

Create a 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

Create an 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: Browse...

Snapshot: □ Enabled

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

Hardware:

Property	Value
Abstracted LCD densi	240
Max VM application h	24

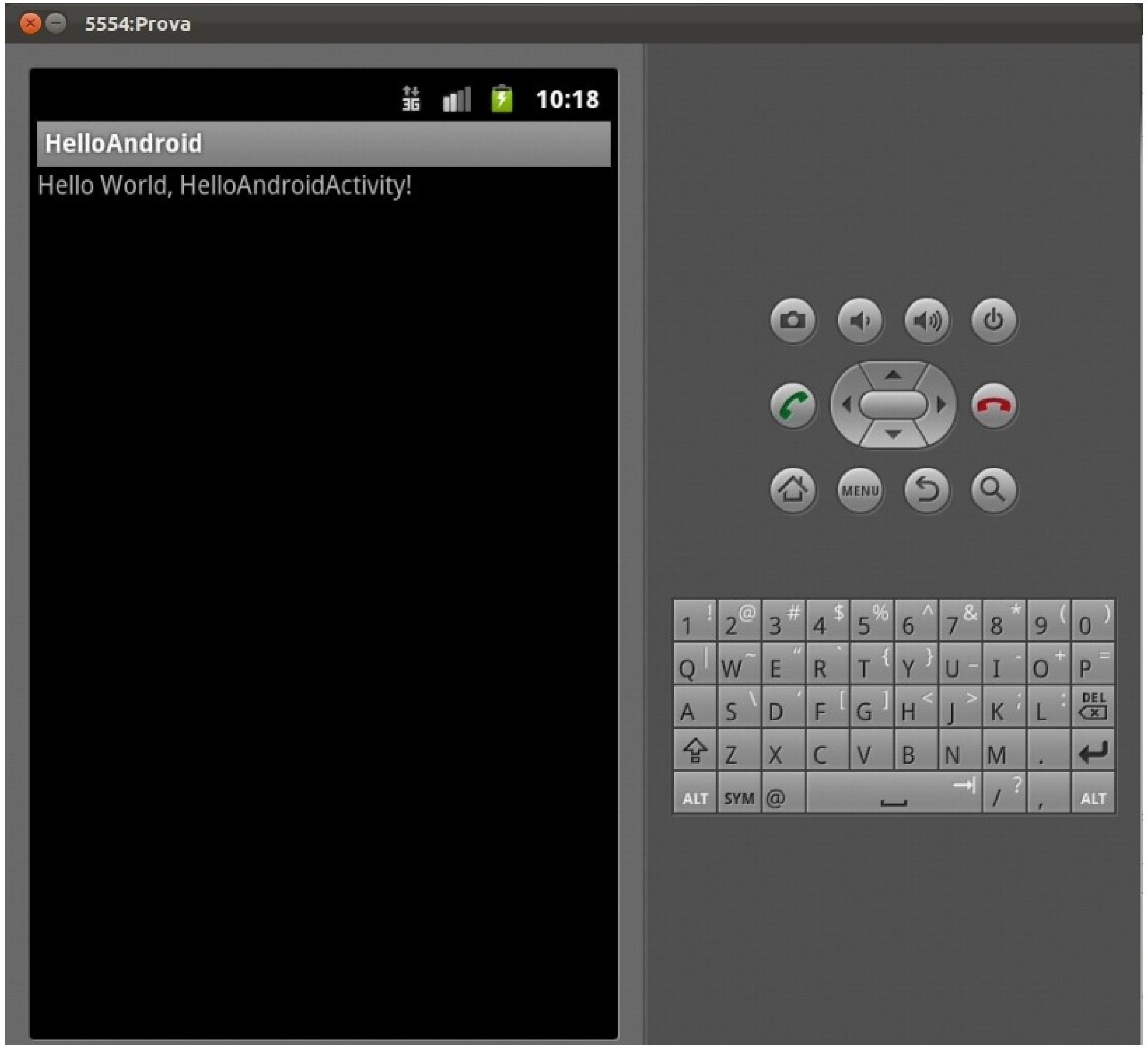
□ Override the existing AVD with the same name

- 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

Hello World, Android!

- Anatomy of an application:
  - Activity – what is started
  - View – what is seen
  - Intent – how to communicate with others
- R.java - Auto-generated file containing:
  - Layouts
  - Values
  - Strings
  - …

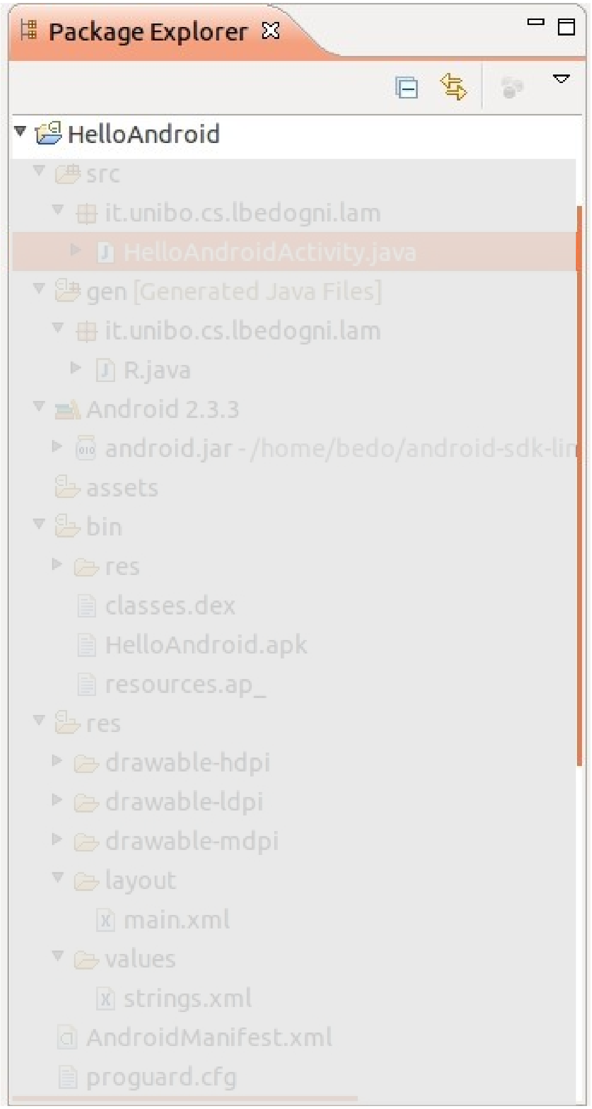
Hello World, Android!



HelloAndroid

Hello World, HelloAndroidActivity!

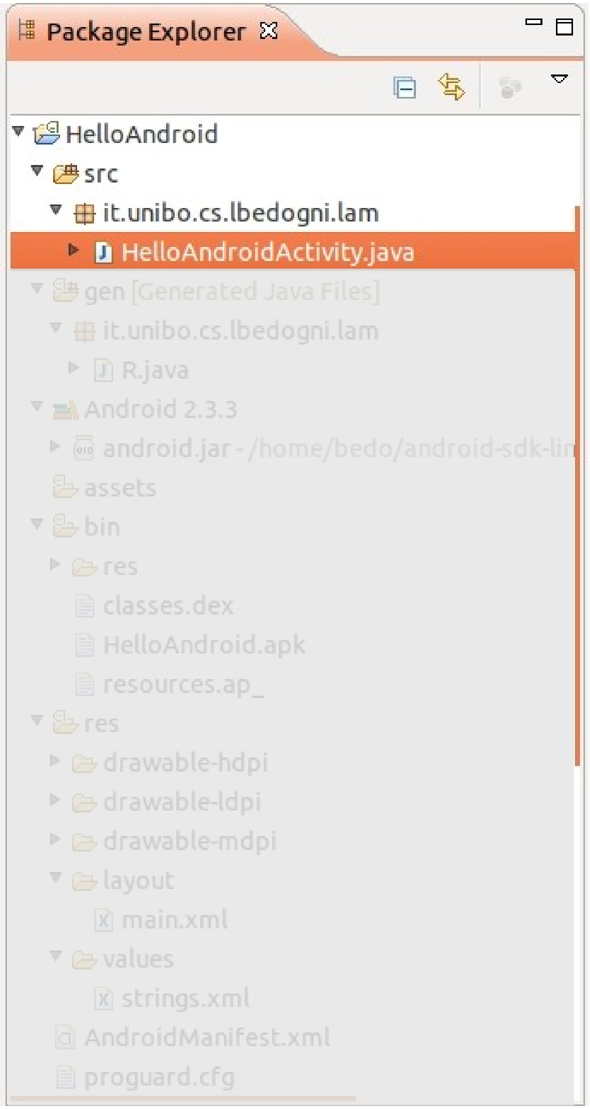
Project contents



```
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

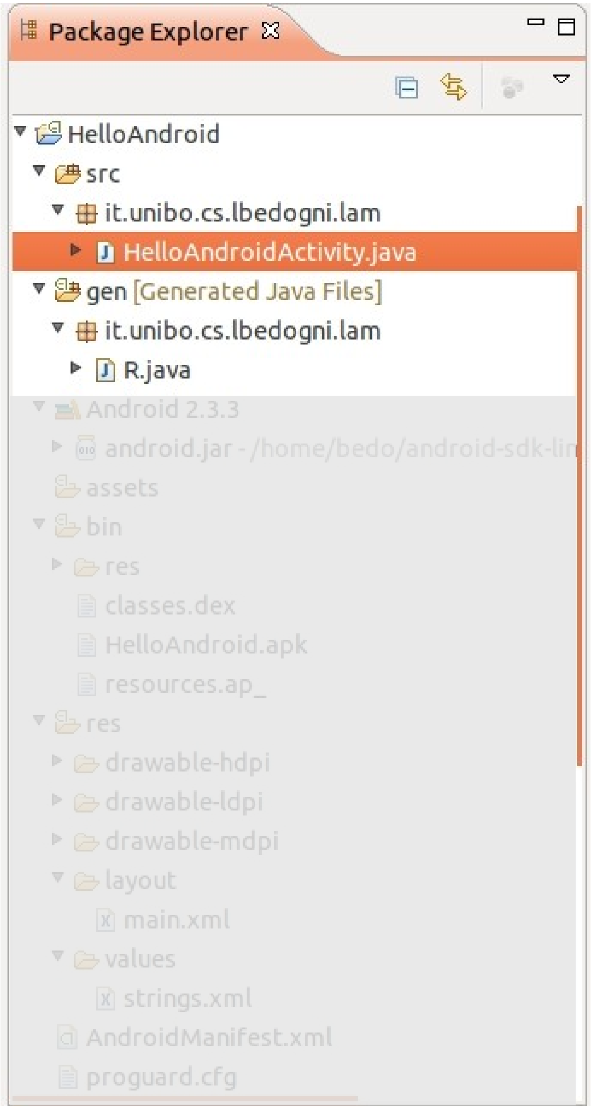
Project contents



```
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

Project contents

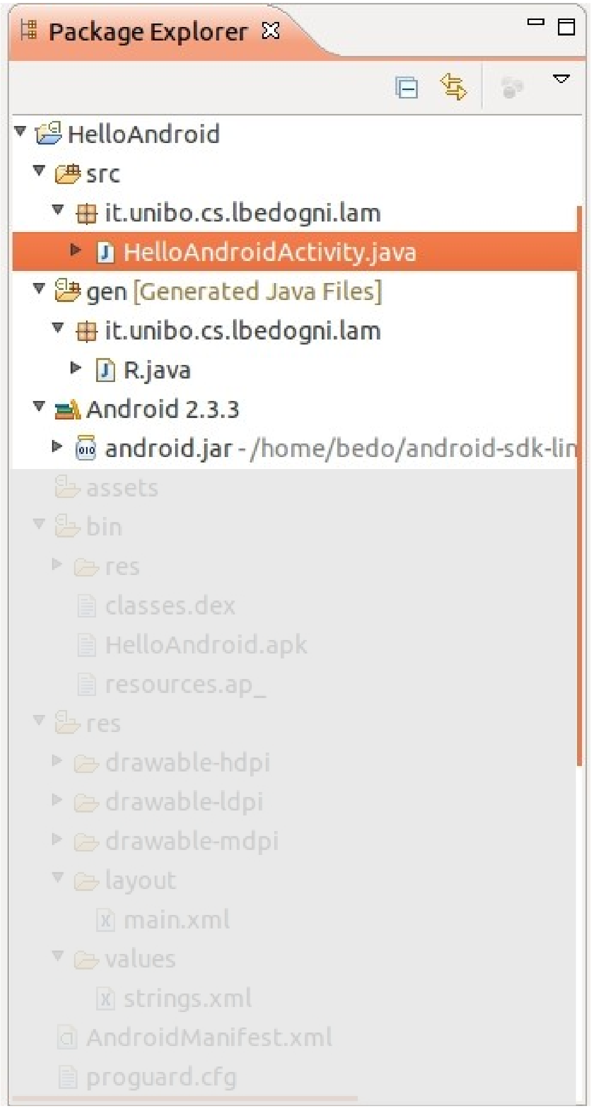


Package Explorer

```
▼ 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

Project **contents**

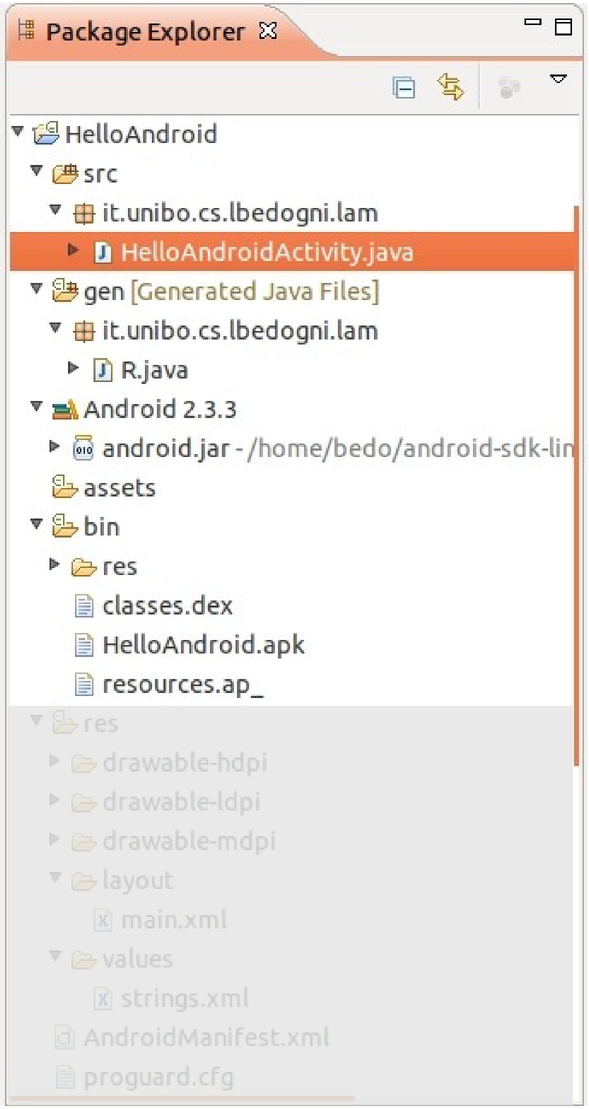


Package Explorer

```
▼ 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

Project contents

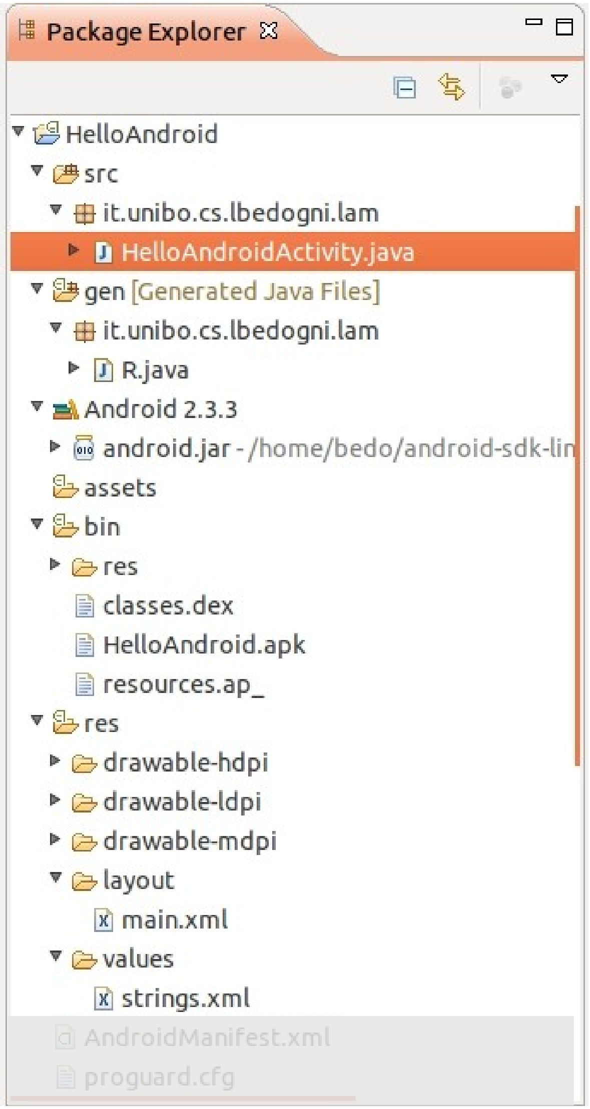


```
Package Explorer
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
  assets
  ▼ bin
    ▶ res
      classes.dex
      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

Project contents



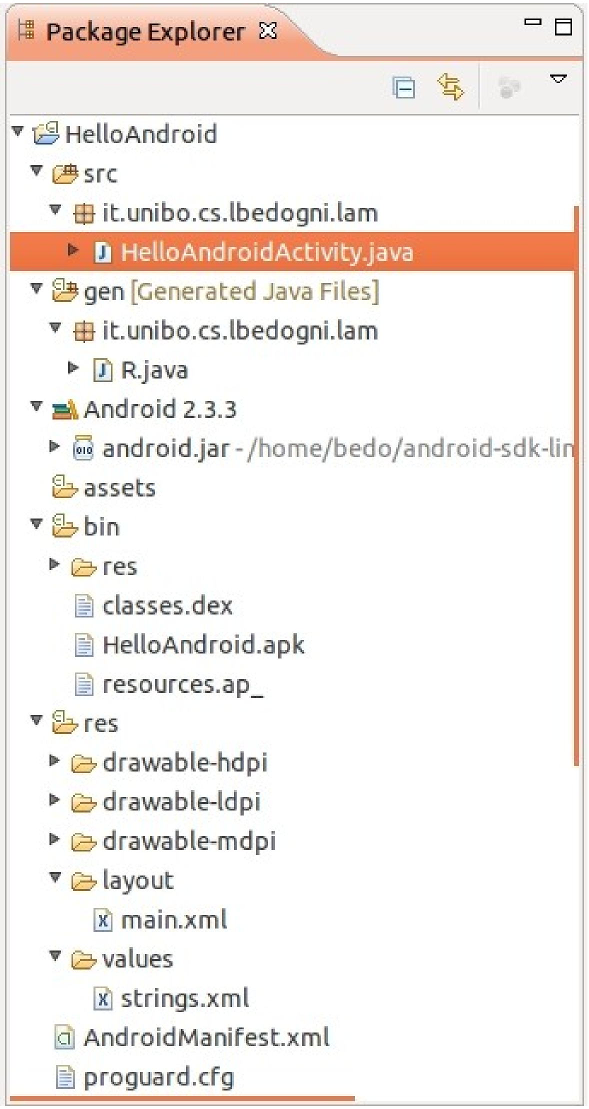
```
Package Explorer
▼ HelloAndroid
  ▼ src
    ▼ it.unibo.cs.lbedogni.lam
      ▶ HelloAndroidActivity.java
  ▼ gen [Generated Java Files]
    ▼ it.unibo.cs.lbedogni.lam
      ▶ R.java
  ▼ Android 2.3.3
    ▶ android.jar - /home/bedo/android-sdk-lin
    assets
  ▼ bin
    ▶ res
      classes.dex
      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

Project 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

AndroidManifest.xml

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

How to **test**

- Via an AVD
  - Fast, possible to have different resolutions/APIs/...
  - Not a real device
- On a real device
  - You feel exactly what it will get deployed
  - Must own a real device
- So?
  - Test it on AVD, and when you feel the application is ready, test it on a real device

How to 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 in release mode

- 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