What else can you do with Android?

Inside Android

Chris Simmonds

Embedded Linux Conference Europe 2010

Overview

- Some background on Android
- Quick start
  - Getting the SDK
  - Running and emulated environment

What is Android?

- Started in 2007 with formation of Open Handset Alliance
  - http://www.openhandsetalliance.com/
  - A group of (currently) 76 mobile network operators, handset manufacturers, silicon vendors and software companies
  - Google is the key player, of course
- The OHA sponsors the development of the Android operating system

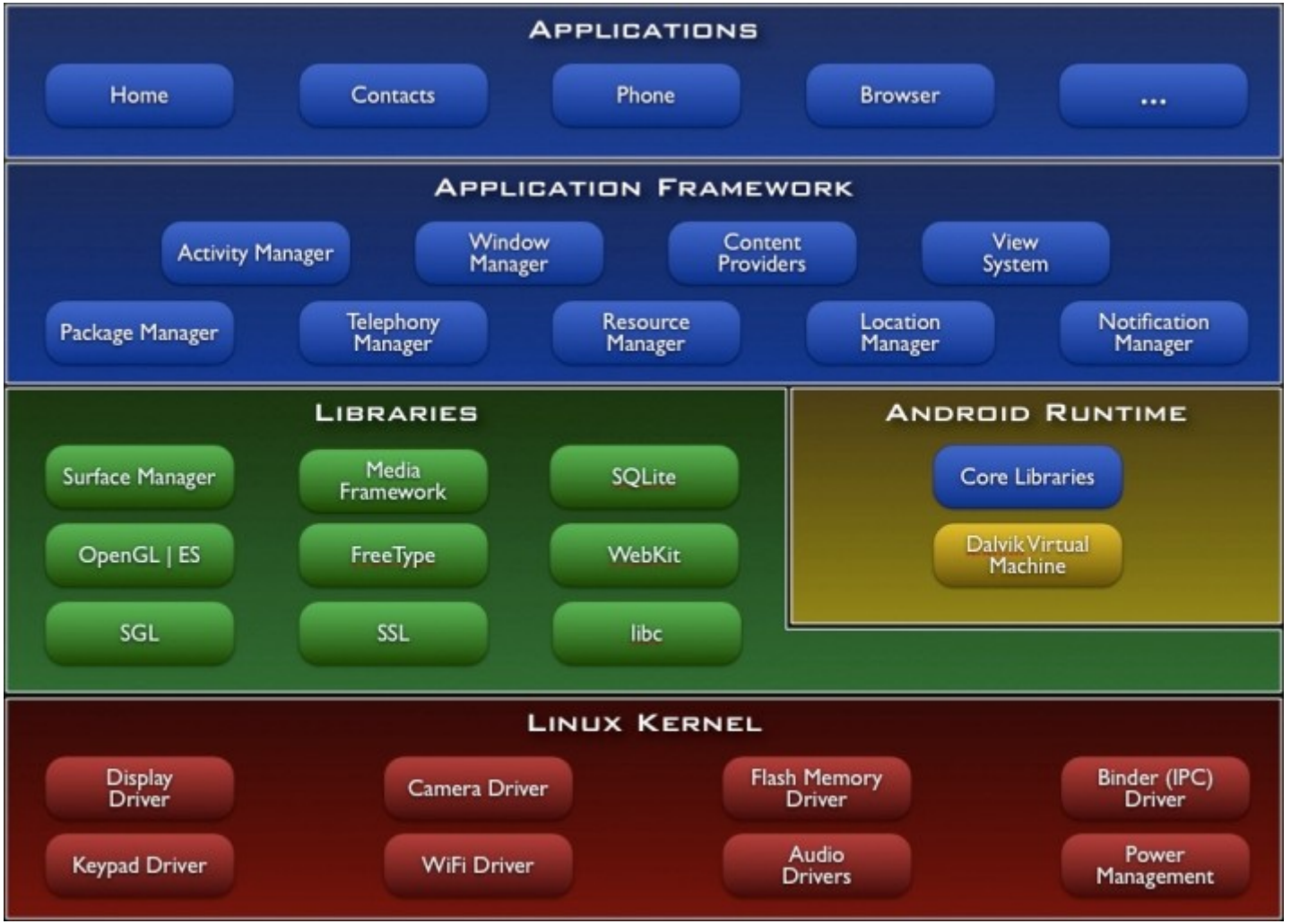
Really, what is Android?

“It's Linux, Jim, but not as we know it”

No, but really?

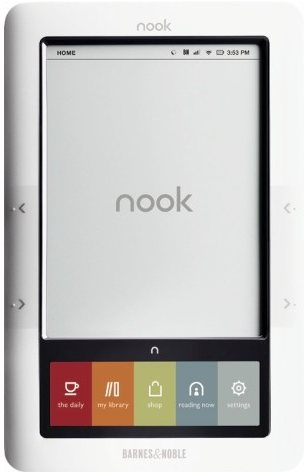
- An open source software stack
- Linux kernel – with patches
- Apache/BSD licensed user space, mostly
- Java application framework
  - Using Dalvik run-time
- A set of Java apps (calendar, clock, etc.)
- Plus many more free and non-free apps
  - Google Market

Android internals



Stuff using Android

- More than 50 hand sets
- Tablets
- eReaders
- Set top boxes



Why Android is important

- It is a standard implementation across all devices
- Good and well-documented application framework
- Good tools for developing and debugging
- Active development from OHA and community

Proposition

- Android is not just for smart phones and tablets
- Also suitable for a wide range of devices
  - Battery-powered portable devices
  - Devices with a touch screen
  - Devices which require fast 2D and 3D graphics
  - Test and Medical equipment
  - Printers, building access, weighing scales ...

Challenges when porting Android

- Non-mainline kernel
  - need to merge Android patches
- The framework (written in Java) is not compatible with Sun/Oracle Java SE or ME
- The C/C++ support is not full ANSI/POSIX
  - problems porting existing code
- Operating system start-up and configuration
  - completely different to “traditional” Linux practices

Downloads from Google

- The SDK
  - For developing Java apps
  - http://developer.android.com
- The NDK
  - For writing native (C/C++) code
  - http://developer.android.com
- Android Open Source Project
  - All the open source components
  - http://source.android.com

Versions

Version	Code name	Framework API level	Release date
3	Gingerbread	?	?Oct 2010
2.2	Froyo	8	May 2010
2.1	Eclair	7	Jan 2010
2	Eclair	5	Dec 2009
1.6	Donut	4	Sept 2009
1.5	Cupcake	3	Apr 2009

Quick start: the Android SDK

- The SDK contains everything you need to write apps and test them using the emulator
- In this section I will show you how to
  - Download the SDK
  - Create an AVD
  - Start the emulator
  - Log on to the emulated session using adb
  - Install and use the Eclipse ADT plug-in

Installing the SDK

- Download SDK “starter pack” from http://developer.android.com/sdk/index.html
  e.g. android-sdk_r06-linux_86.tgz
- Extract files
- Add the “tools” directory to your path
- Run “android”
- Select and install the platform(s) you will be developing for

Creating a virtual device

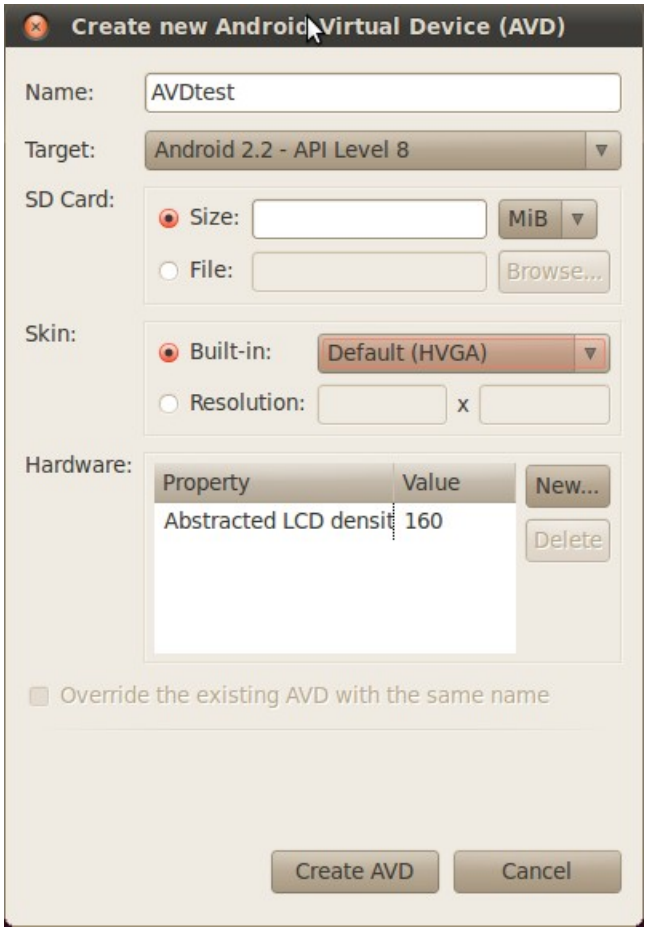
An Android Virtual Device (AVD) defines the hardware the emulator
will emulate

Create from the command line using

```
android create avd -n AVDtest -t 1
```

Or graphically using the SDK and AVD manager (shown)

Note the “Skin” section – basically the screen resolution. The default is HVGA, 320 x 480



Run the emulator

```
emulator -avd AVDtest
```

You can also add:

-show-kernel
to show Linux boot messages

-shell
to give a root shell prompt

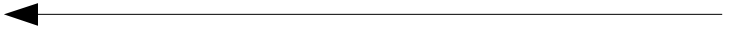
Using adb

- “Android Debug Bridge”
- A tool to connect to and interact with Android devices, including emulated ones
- Useful options for now
  - list devices
  - connect to a device
  - run a shell

Adb example

```
$ adb devices
List of devices attached
emulator-5554    offline

$ adb -s emulator-5554 shell
# cat /proc/cpuinfo
Processor    : ARM926EJ-S rev 5 (v5l)
BogoMIPS : 242.48
Features : swp half thumb fastmult vfp edsp java
CPU implementer  : 0x41
CPU architecture: 5TEJ
CPU variant  : 0x0
CPU part : 0x926
CPU revision : 5
Hardware : Goldfish
Revision : 0000
Serial       : 0000000000000000
```



Note: Goldfish is the name of the emulator platform

Since there is only one device you can miss off the -s

```
$ adb shell
#
```

System log: logcat

- logcat is a replacement for syslog
- Three logs: 'main' (default), 'radio', 'events'
  - each is a ring-buffer of 64 KiB (default)

```
$ adb logcat -b main
I/DEBUG   (   30): debuggerd: Jun 30 2010 13:59:20
D/qemud   (   37): entering main loop
I/Vold    (   28): Vold 2.1 (the revenge) firing up
D/Vold    (   28): Volume sdcard state changing -1 (Initializing) -> 0 (No-Media)
I/Netd    (   29): Netd 1.0 starting
...
```

Real hardware

- You can use adb to connect to real devices via
  - USB
  - Network
- Note:
  - for production hardware you will need to enable Settings->Applications->Development->USB debugging and tick Settings->Applications->Development->Unknown sources

Connecting via a network

- If your target hardware is networked you can use adb in this way

```
export ADBHOST=192.168.1.101
adb shell
#
```

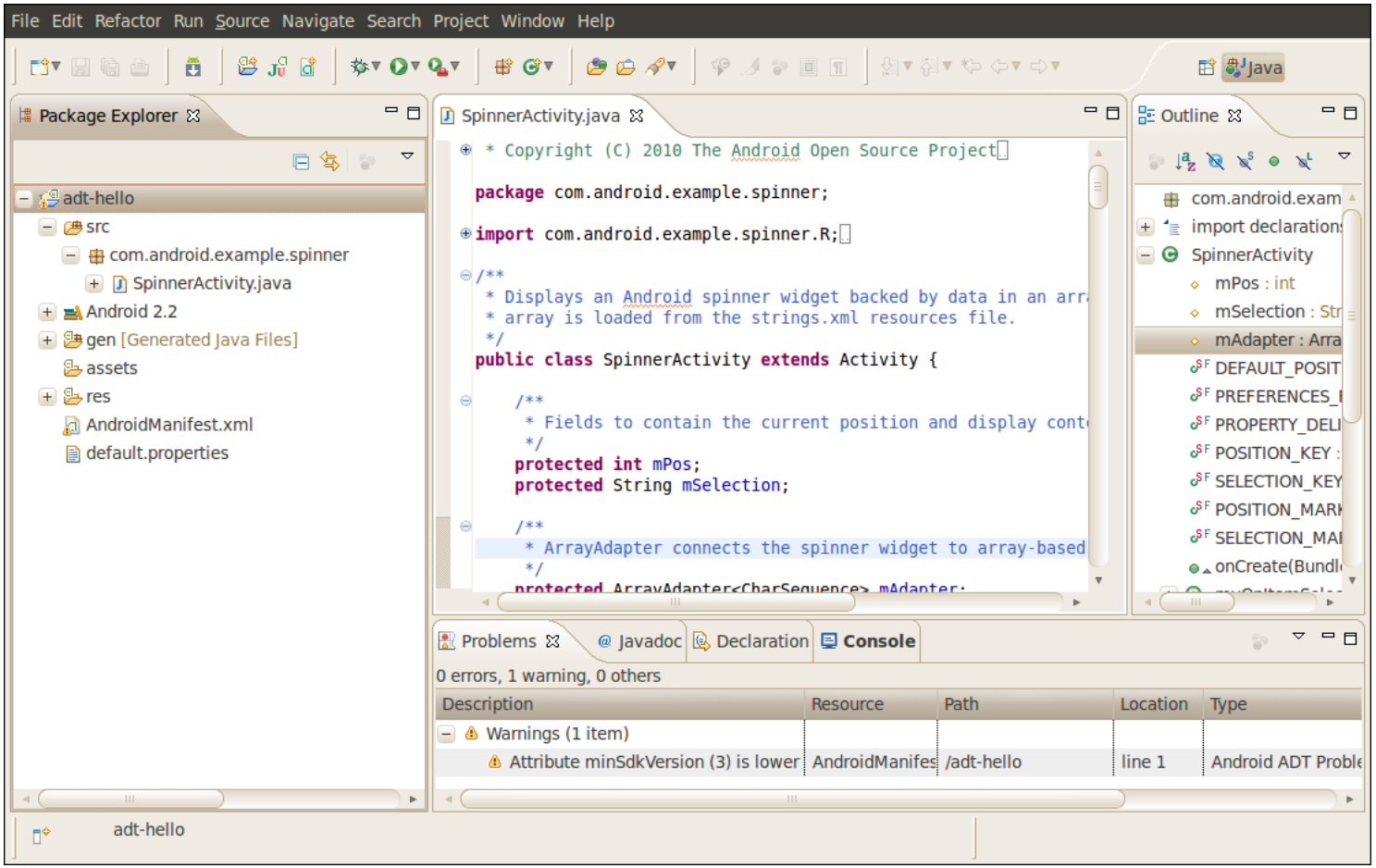
Android Development Tool for Eclipse

- The ADT allows you to do everything mentioned so far in the Eclipse environment
  - Create AVDs
  - Run the emulator
  - Create projects
  - Write and debug code

Installing ADT

- Begin by installing Eclipse
- Then go to Help->Install new software
- Set the URL to “Work with” to
  - https://dl-ssl.google.com/android/eclipse/
- Select “Developer Tools” and click Install

ADT in action



Summary

- Android is a re-imagining of Linux for devices
  - for phones and more..
- The SDK contains everything you need to write apps and test them in the emulator or on real hardware