I will try to collect here useful tips for making Android development work for command-line loving person like me. See Android G1 if you are interested in hardware.

t=0x8dcb5b8 [0,0]

Contents: [Dobrica PavlinuÅ¡iÄ 's random unstructured stuff]

- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (ADB)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (Remote adb access)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (provision device)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (CyanogenMod)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (Google apps)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (Build)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (flashing)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (SL4A)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (x86)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (froyo 2.2)
  - Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (eclair 2.1)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (Application build process)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (repo)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (repack applications)
- Dobrica PavlinuÅ¡iÄ 's random unstructured stuff (repository with scripts)

ADB

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

Remote adb access

```
dpavlin@t61p:~/Downloads/android$ adb shell
# setprop service.adb.tcp.port 5555



dpavlin@t61p:~$ ssh -R 5555:192.168.1.40:5555 klin



dpavlin@klin:~$ adb connect 127.0.0.1:5555

dpavlin@klin:~$ adb devices
List of devices attached
127.0.0.1:5555  device
```

provision device

- http://balloonboard.org/balloonwiki/AndroidBalloon

```
adb shell
su
cd /data/data/com.android.providers.settings/databases
sqlite3 settings.db
INSERT INTO system (name, value) VALUES ('device_provisioned', 1);
.exit
```

CyanogenMod

- http://wiki.cyanogenmod.com/index.php?title=Compile_CyanogenMod_for_Dream_%26_Sapphire
- enable swap with 99swapon in /system/etc/init.d/

Google apps

```
dpavlin@android:/srv/cyanogen$ wget \
--referer=http://developer.htc.com/google-io-device.html \
-O dream_sapphire_update.zip \
http://member.america.htc.com/download/RomCode/ADP/signed-google_ion-ota-14721.zip?
--2010-10-01 17:42:10--  http://member.america.htc.com/download/RomCode/ADP/signed-google_ion-ota
Resolving member.america.htc.com... 216.139.227.226
Connecting to member.america.htc.com|216.139.227.226|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 53965203 (51M) [application/x-zip-compressed]
Saving to: `dream_sapphire_update.zip'

100%[======================================================================>] 53,965,203   376

2010-10-01 17:44:39 (354 KB/s) - `dream_sapphire_update.zip' saved [53965203/53965203]

dpavlin@android:/srv/cyanogen$ cd device/htc/dream_sapphire
dpavlin@android:/srv/cyanogen/device/htc/dream_sapphire$ ./unzip-files.sh
```

Build

A **lot** of host dependencies to install and build.

```
dpavlin@android:/srv/cyanogen$ . build/envsetup.sh
dpavlin@android:/srv/cyanogen$ lunch
dpavlin@android:/srv/cyanogen$ make -j8
```

flashing

Transfer files you just built:

```
dpavlin@x200:/virtual/android/cyanogen$ rsync -v android:/srv/cyanogen/out/target/product/dream_s
```

Download fastboot from http://developer.htc.com/adp.html

Start G1 by pressing camera+power button

```
dpavlin@x200:/virtual/android/cyanogen$ sudo ./fastboot devices
HT840GZ00000    fastboot

dpavlin@x200:/virtual/android/cyanogen$ sudo sh -c "ANDROID_PRODUCT_OUT=2010-09-25/ ./fastboot fl
-----------------------------------------
Bootloader Version...: 1.33.2005
Baseband Version.....: 2.22.19.26I
Serial Number........: HT840GZ33728
-----------------------------------------
checking product... OKAY
checking version-bootloader... OKAY
sending 'boot' (2234 KB)... OKAY
writing 'boot'... OKAY
sending 'recovery' (3308 KB)... OKAY
writing 'recovery'... OKAY
sending 'system' (92239 KB)... OKAY
writing 'system'... OKAY
rebooting...
```

- review changes: http://github.com/CyanogenMod/android_vendor_cyanogen/blob/froyo/CHANGELOG.mkdn

SL4A

- http://code.google.com/p/android-scripting/wiki/RemoteControl

x86

- http://www.android-x86.org/getsourcecode

froyo 2.2

```
dpavlin@android:/srv/android-x86$ repo init -u git://git.android-x86.org/manifest.git -b froyo-x8
dpavlin@android:/srv/android-x86$ repo sync

# configure build
dpavlin@android:/srv/android-x86$ . build/envsetup.sh
dpavlin@android:/srv/android-x86$ lunch

# build
dpavlin@android:/srv/android-x86$ make -j8
```

audio doesn't work

alternativly, you can build `generix_x86`

We need `/sbin` in `PATH` to access `e2fsprogs` utilities needed to build image

```
dpavlin@android:/srv/android-x86$ export PATH=/sbin:$PATH
dpavlin@android:/srv/android-x86$ make -j8 iso_img TARGET_PRODUCT=generic_x86
```

eclair 2.1

```
dpavlin@android:/srv/android-x86$ repo init -u git://git.android-x86.org/manifest.git -b eclair-x
dpavlin@android:/srv/android-x86$ repo sync
dpavlin@android:/srv/android-x86$ make -j8 iso_img TARGET_PRODUCT=generic_x86
# doesn't work

dpavlin@android:/srv/android-x86$ make -j8 iso_img TARGET_PRODUCT=eeepc
```

Application build process

- http://www.alittlemadness.com/2010/06/07/understanding-the-android-build-process/

repo

- http://source.android.com/source/git-repo.html
- enviroment vars: REPO_PROJECT REPO_PATH REPO_REMOTE
- see git command output: `export REPO_TRACE=1`
- export md5 hashes of verything (create tag): `repo manifest -r -o tag.xml`

repack applications

- http://code.google.com/p/android-apktool/

repository with scripts

fetchatom: http://github.com/dpavlin/android-command-line/commits/master.atom

- There was an error: 500 SSL negotiation failed: