Android SDK + Toolkit

- SDK installieren
- Dalvik-Cache leeren
- Screenshots vom Gerät machen

SDK installieren

Diese Anleitung ist für die Linux-Version des SDKs, Windows ist sicher ähnlich wird hier aber nicht behandelt.

1. Download des Starter-Pakets von android.com, momentan ist dieser Link aktuell (android-sdk_r04-linux_86.tgz), bitte nach neuen Versionen direkt hier schauen.

2. MD5 Summe mit den Angaben auf der Downloadseite vergleichen:

```
# md5sum android-sdk_r04-linux_86.tgz
ef84b08fd9da84f4c4ae77564fe4eaee  android-sdk_r04-linux_86.tgz
```

3. Entpacken und Installer ausführen:

```
tar xvfz android-sdk_r04-linux_86.tgz
cd android-sdk-linux_86
tools/android
```

4. Wer nur das Tool „adb“ benötigt muss keines der Packages installieren und kann direkt bei Punkt 5 weitermachen.

Wenn der grafische Installer gestartet ist, hier unter „Available Packages“ die gewünschte Version bzw. die Google-APIs auswählen und über „Install“ einspielen. Sollte beim Zugriff auf das Repository ein Fehler auftauchen, kann man mal probieren, den Zugriff per https zu deaktivieren. Dazu in den „Settings“ das Häckchen bei „Force https://.. sources to be fetched using http://..“

5. Es empfiehlt sich den Installpfad in die PATH-Variable aufzunehmen, ich hab das z.B. in ~/apps/android-sdk-linux_86 abgelegt:

```
PATH=$PATH:$HOME/bin:$HOME/apps/android-sdk-linux_86/tools
```

6. Im Handy muss jetzt das USB-Debugging aktiviert werden. Der Schalter dazu versteckt sich in Einstellungen ? Anwendungen ? Entwicklung ? USB-Debugging.

7. Damit das Handy auch korrekt erkannt wird, muss unter Linux noch UDEV konfiguriert werden. Dazu als root eine neue Datei anlegen und rebooten:

```
SUBSYSTEM==”usb”, SYSFS{idVendor}==”0bb4?, MODE=”0666?
```

8. wenn jetzt das Gerät per USB verbunden wird, sollten etwa folgende Meldungen im dmesg auftauchen

```
# dmesg
usb 2-2: new high speed USB device using ehci_hcd and address 33
usb 2-2: New USB device found, idVendor=0bb4, idProduct=0c02
usb 2-2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
usb 2-2: Product: Android Phone
usb 2-2: Manufacturer: HTC
usb 2-2: SerialNumber: HT93YKF04289
usb 2-2: configuration #1 chosen from 1 choice
scsi27 : SCSI emulation for USB Mass Storage devices
usb-storage: device found at 33
usb-storage: waiting for device to settle before scanning
usb-storage: device scan complete
```

```
scsi 27:0:0:0: Direct-Access     HTC      Android Phone    0100 PQ: 0 ANSI: 2
sd 27:0:0:0: Attached scsi generic sg4 type 0
sd 27:0:0:0: [sdd] Attached SCSI removable disk
```

9. jetzt nur noch den adb-Server starten und den Zugriff testen (der Server muss als root gestartet werden, zumindest hats bei mir nicht anders funktioniert):

```
# cd ~/apps/android-sdk-linux_86/tools
# sudo ./adb start-server
```

Falls alles geklappt hat ist das Handy nun per adb ansprechbar:

```
# ./adb devices
List of devices attached
HT93YKF04289    device
```

Dalvik-Cache leeren

Einleitung

Erstmal ein wenig zur Erklärung: Die Dalvik virtual machine ist eine „Java Virtual Machine“ und wurde speziell für die Android Plattform entworfen. Normalerweise habe diese Geräte wenig RAM, langsamen internen Flash-Speicher und verhalten sich allgemein wie uralte Desktopsysteme. Allerdings laufen sie unter Linux, was virtuellen Speicher, Prozesse und Threads und UID-basierte Sicherheitsmechanismen zur Verfügung stellt.

Um diese Nachteile etwas auszugleichen werden in der Dalvik virtual machine mehrere .class (Java-Programmierklassen) mit einem speziellen Programm in einer .dex-Datei zusammengefasst und einige Optimierungen bzgl. des Speicherbedarfs vorgenommen. Bei Android läuft jede Applikation in einer eigenen Dalvik VM.

Das Verzeichnis, in dem die .dex-Dateien abgelegt werden, heißt üblicherweise dalvik-cache und liegt unter /system/sd/dalvik-cache oder /data/dalvik-cache.

Cache leeren

Wenn man viel mit verschiedenen ROMs experimentiert oder allgemein das Handy etwas langsamer als gewohnt wirkt, kann es nötig sein, den Dalvik-Cache neu aufzubauen.

Zuerst muss das SDK installiert werden und ADB eingerichtet sein. Der Zugriff aufs Gerät muss natürlich möglich sein.

Jetzt das Handy rebooten und im Recovery-Mode starten (Home-Button und rote Taste gleichzeitig drücken). Die Recovery Console starten und mit ADB verbinden:

```
$ ./adb shell
/ # mount -o rw /dev/block/mmcblk0p2 /system/sd
/ # rm -r /system/sd/dalvik-cache/*
/ # reboot
```

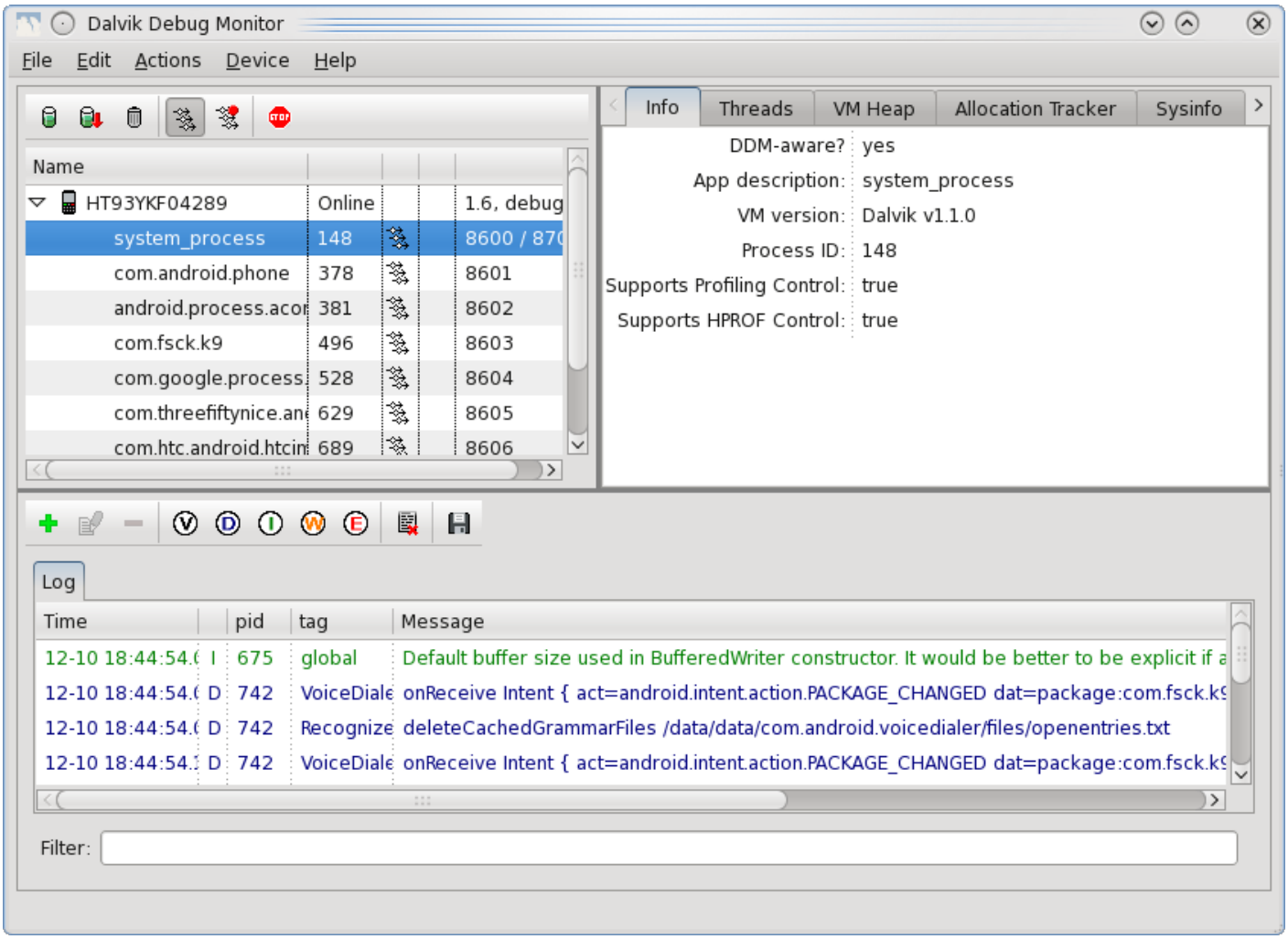
Weitergehende Infos für Programmierer und Leute, dies wirklich wissen wollen gibts hier und hier.

Screenshots vom Gerät machen

zuerst muss das SDK installiert werden und ADB eingerichtet sein. Der Zugriff aufs Gerät muss natürlich möglich sein.

Um Screenshots von der aktuellen Anzeige zu machen muss folgendes Tool gestartet werden: android-sdk-linux_86/tools/ddms

Der „Dalvik Debug Monitor“ setzt ein laufendes X voraus. Sofern er gestartet und ein Handy verbunden ist, siehts erstmal so aus:



Im Menü Device den Eintrag „Screen Capture“ auswählen oder über „STRG+S“ direkt starten. Nun wird der aktuelle Screen gegrabbt und läßt sich auch gleich speichern:


Device Screen Capture
Refresh
Rotate
Save
Copy
Done
Captured image:
7°
18:52
Google
Listen
Log Collect...
Lumby
Mail
Maps
Market
Maths Worko...
Metal Detect...
Mobento
Musical Lite
Musik
My Maps E...
My Tracks
MyMarke
Nav4All
NetMeter
OI Dateim...
OI Notizbl...
OI Safe
OpenSud