

# **Smart Home Control**

## **Installation des Android SDK's mit Eclipse**

Prof.Dr.Ing.Birgit Wendholt

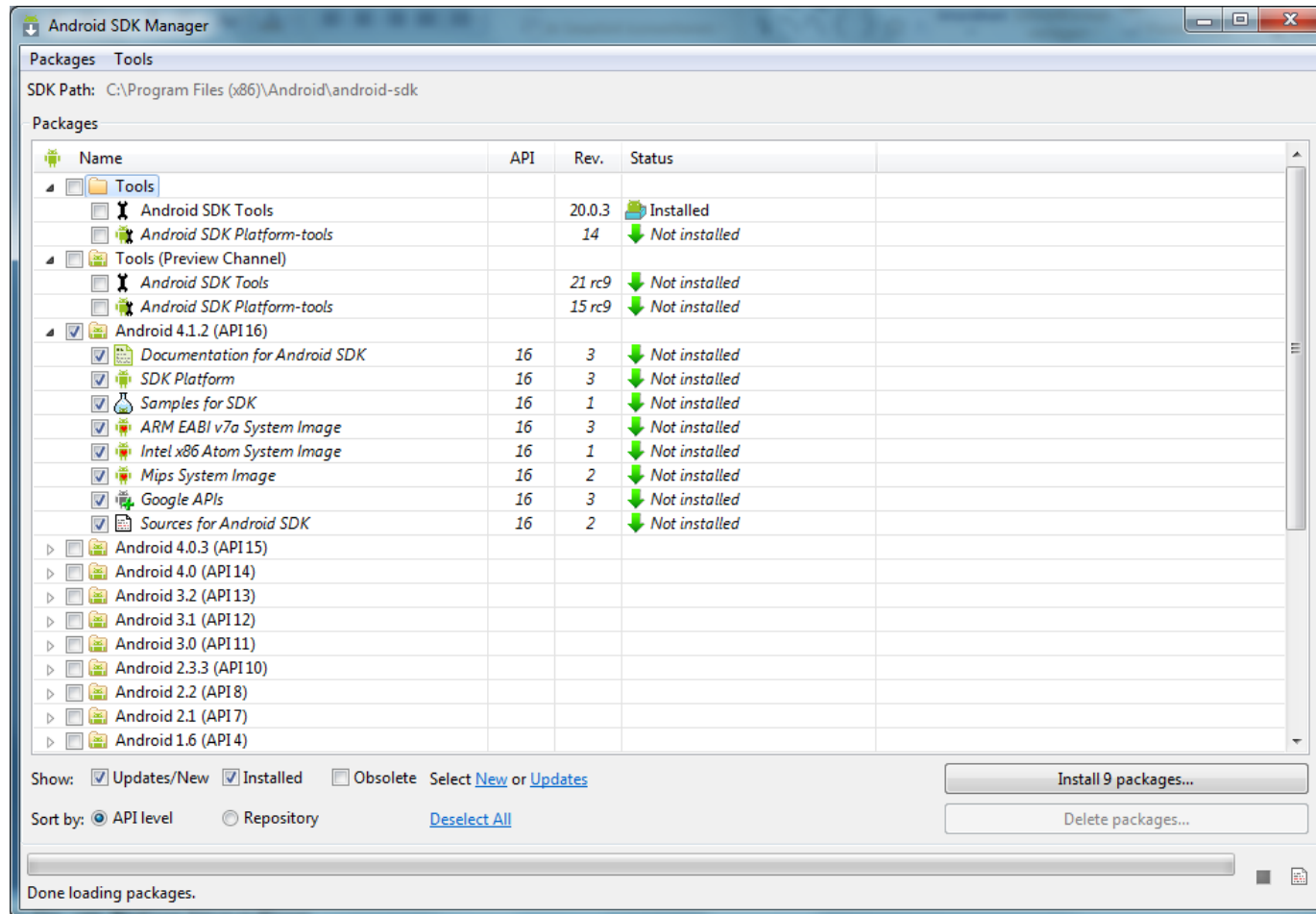
# Schritte

1. JDK 6 (JRE nicht ausreichend) installieren (32 Bit Version)
2. Eclipse installieren (→ <http://www.eclipse.org/downloads/>) Eclipse Classic wählen. Mindestens Eclipse Helios.
3. Android SDK installieren (→ <http://developer.android.com/sdk/index.html#Requirements>)
4. Eclipse Plugin installieren (→ <http://developer.android.com/sdk/installing/installing-adt.html> )
5. Android virtuelles Device (AVD) einrichten
6. Android App auf AVD installieren
7. Android App Samsung Galaxy Tab 2 installieren
8. Project SmartHome Controller laden

### 3. Android Plugin installieren

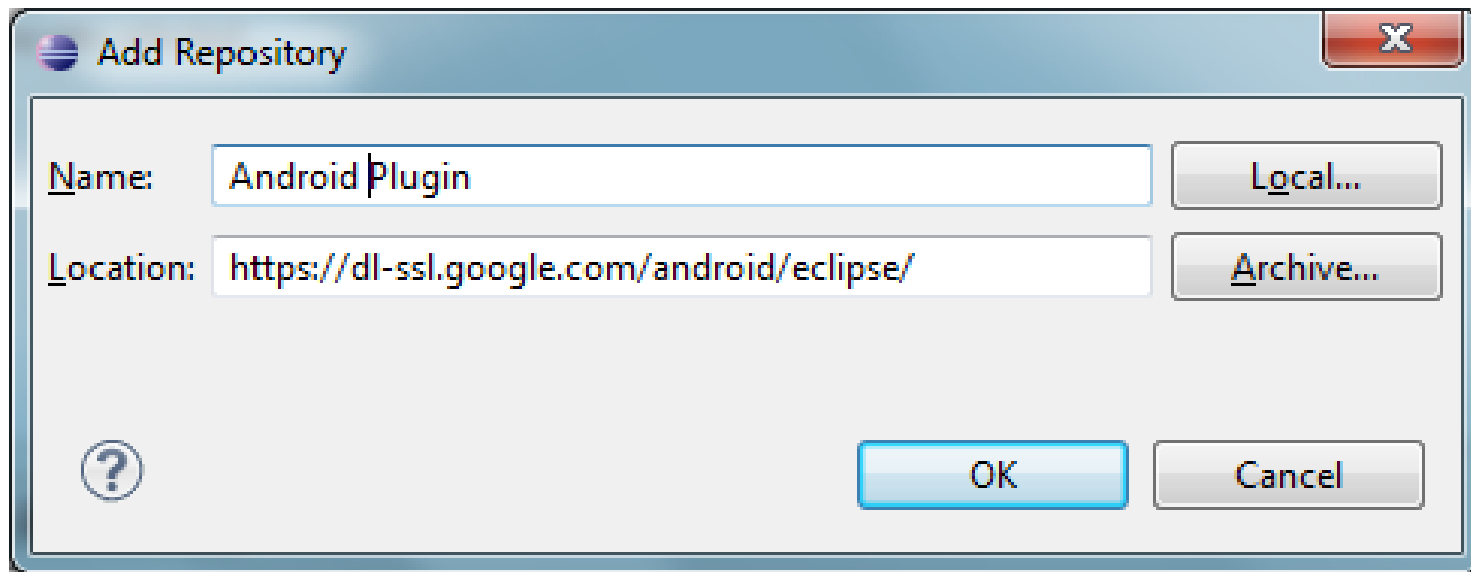
- Download SDK  
<http://developer.android.com/sdk/index.html#Requirements>
- installer\_r20.0.3-windows.exe ausführen. Dabei muss eine JAVA SDK gefunden werden!
- Android SDK Manager starten und die korrekten Packages installieren (Geduld es dauert etwas):
  - Android 4.1.2 API 16
  - Android 4.0.3 API 15 (für Beispiel Projekt SmartHomeController)
- **Hinweis:** Bei Verwendung eigener SmartPhones oder Tablets werden ggf. nur frühere Versionen des SDK's unterstützt

# Android SDK Manager



## 4 Eclipse Plugin Installieren

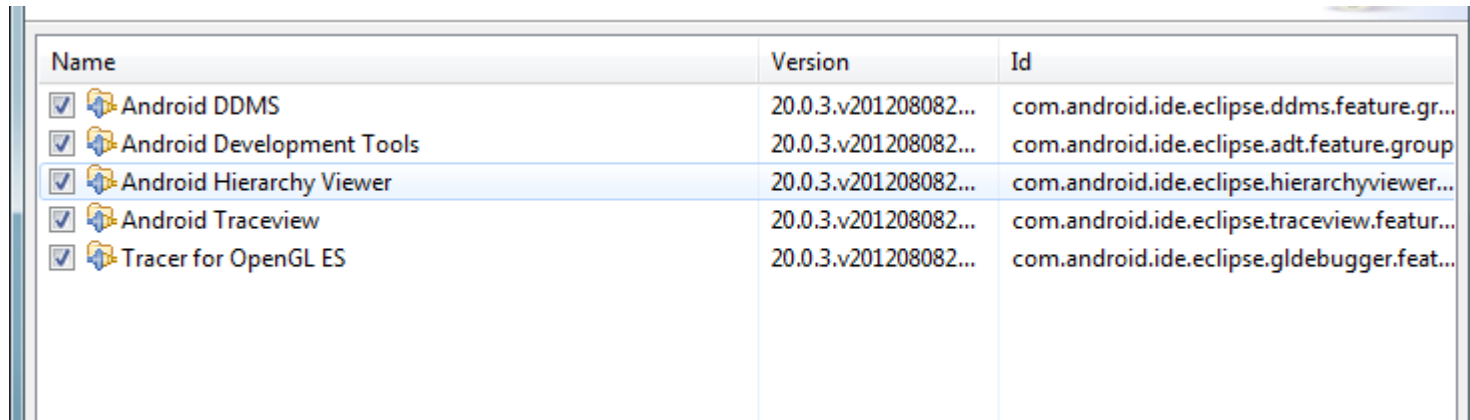
- In Eclipse: Help → Install New Software dann Add



- In dem Dialog: Android Plugin als Name und <https://dl-ssl.google.com/android/eclipse/> als URL angeben

## 4 Eclipse Plugin Installieren

- **Dialog** Available Software → Developer Tools wählen, dann **Next**

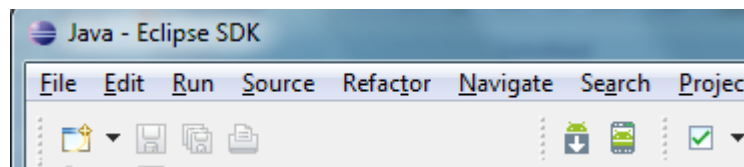


Name	Version	Id
<input checked="" type="checkbox"/> Android DDMS	20.0.3.v201208082...	com.android.ide.eclipse.ddms.feature.gr...
<input checked="" type="checkbox"/> Android Development Tools	20.0.3.v201208082...	com.android.ide.eclipse.adt.feature.group
<input checked="" type="checkbox"/> Android Hierarchy Viewer	20.0.3.v201208082...	com.android.ide.eclipse.hierarchyviewer...
<input checked="" type="checkbox"/> Android Traceview	20.0.3.v201208082...	com.android.ide.eclipse.traceview.featur...
<input checked="" type="checkbox"/> Tracer for OpenGL ES	20.0.3.v201208082...	com.android.ide.eclipse.gldebugger.featur...

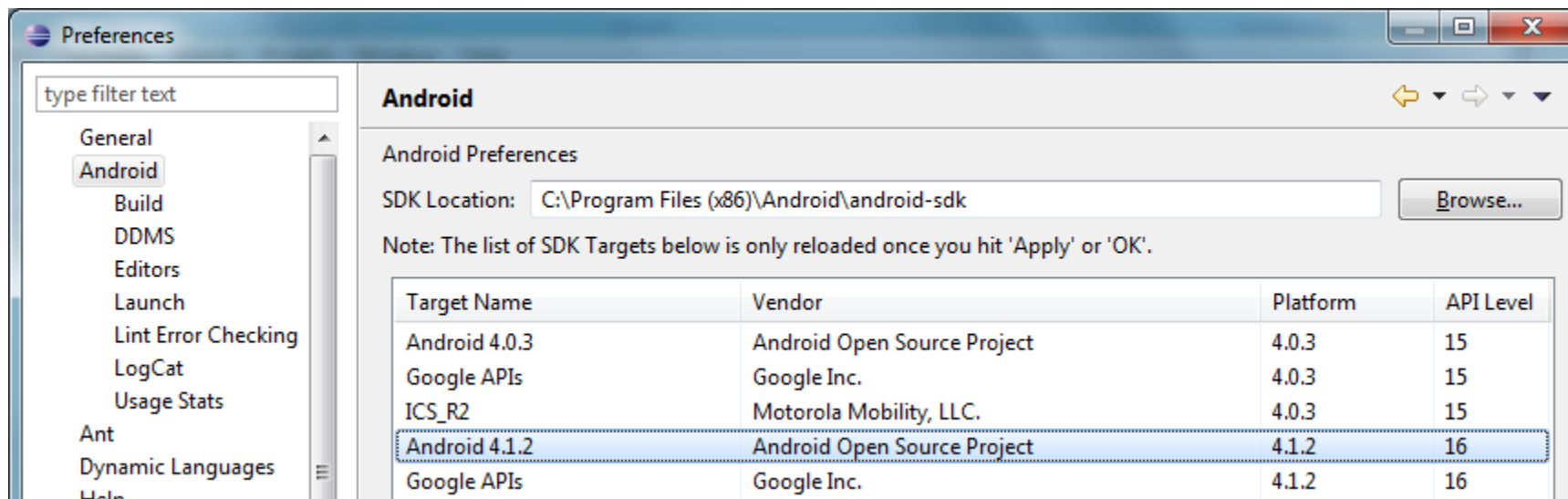
- Liste der angebotenen Tools mit **Next** bestätigen
- License Agreements lesen, akzeptieren und mit **Finish** Installation anstoßen
- Eclipse neu starten

## 4 Eclipse Plugin Installieren

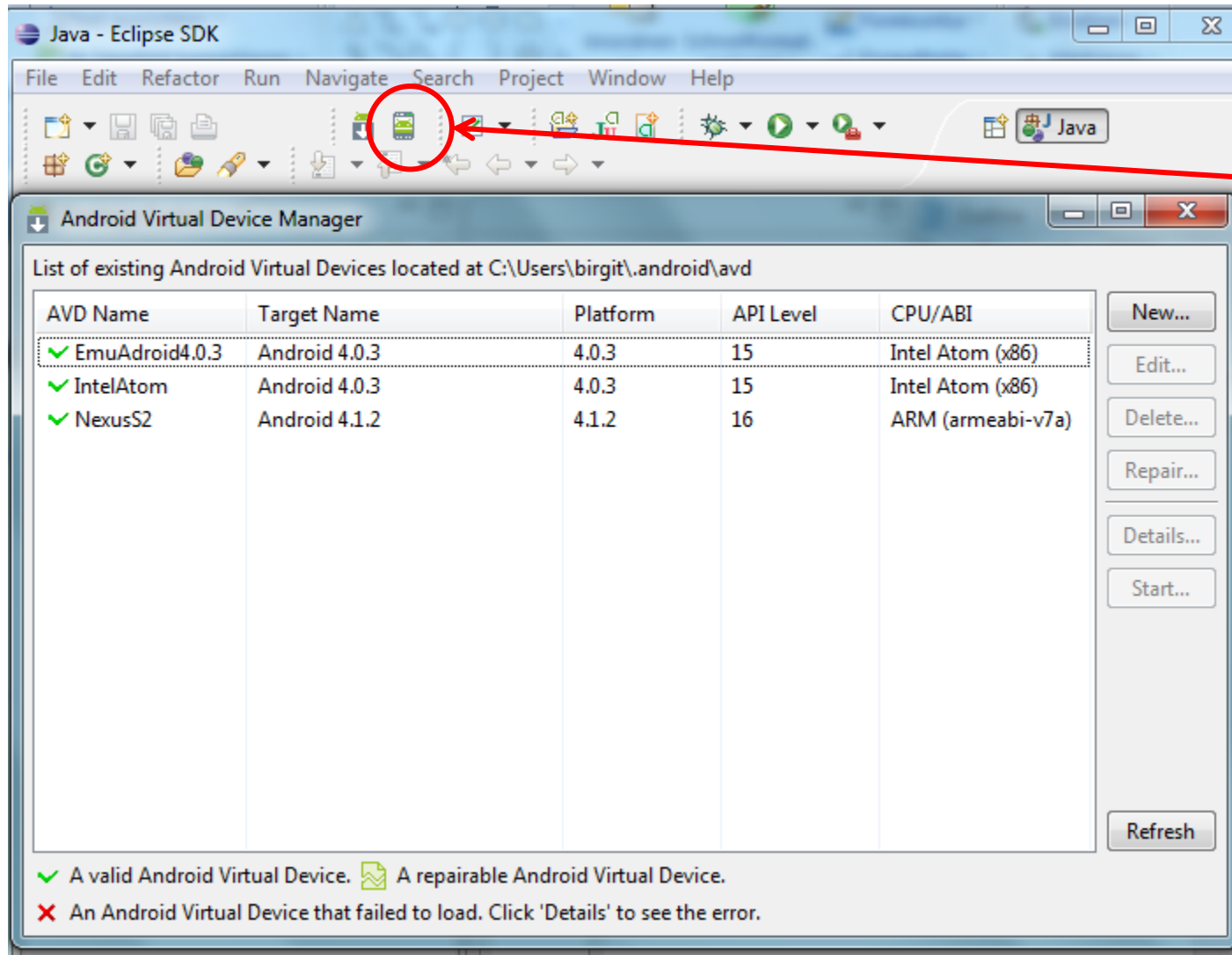
- nach erfolgreicher Installation zeigt die Eclipse Toolbar Icons für Android SDK Manager/ AVD Manager



- Android SDK einrichten: **Windows** → **Preferences** **Android**. Dort den Pfad zur Android Installation eingeben. (<InstVerzeichnis>/**android-sdk**)



# 5 Virtuelles Device (AVD) einrichten



Virtual  
Device  
Manager  
starten

Virtual  
Device  
Manager

mit **New**  
neues  
virtuelles  
Device  
hinzufügen





## 5 (AVD) einrichten

Create new Android Virtual Device (AVD)

Name: EmulAndroid4.1.2

Target: Android 4.1.2 - API Level 16

CPU/ABI: Intel Atom (x86)

SD Card:

☒ Size: 512 MiB

☐ File: Browse...

Snapshot:

☒ Enabled

Skin:

☒ Built-in: Default (WVGA800)

☐ Resolution: x

Hardware:

Property	Value
Abstracted LCD density	240
Max VM application heap size	48
Device ram size	512

☐ Override the existing AVD with the same name

Create AVD Cancel

Android Version

CPU (für Intel Atom (x86) sind  
noch paar zusätzliche Installationen  
Notwendig)

Größe der SD-Card  
(512 MB reicht für erste Tests)

Starten des virtuellen Devices  
ohne Booten

Oberfläche des virtuellen Devices  
und Voreinstellungen für Hardware

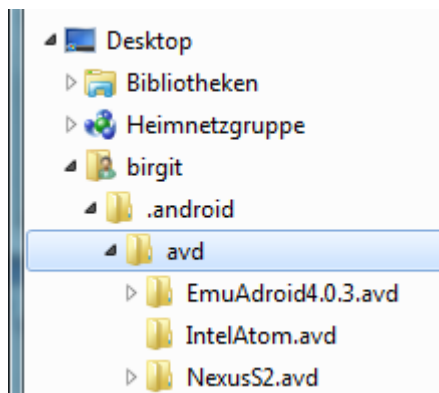
Erweitern der Hardware-  
eigenschaften

Erzeugen des virtuellen Devices  
(kann ein wenig dauern)

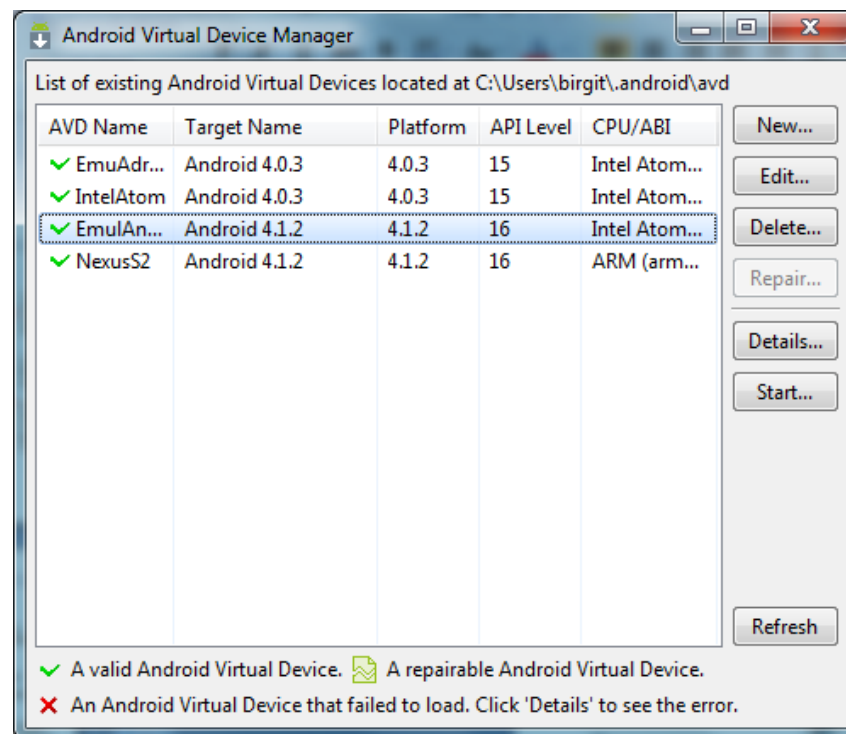
# 5 Virtuelles Device (AVD) einrichten

## Ergebnis (Filesystem)

- <user\_home\_verz>/.android  
enthält die Images der virtuellen  
Devices in separaten  
Verzeichnissen



## Ergebnis AVD Manager



# 5 Virtuelles Device (AVD) einrichten



Ergebnis



(Starten kann dauern)

Infos zur Bedienung,  
Tastenbelegung etc.

<http://developer.android.com/tools/devices/emulator.html>

# 5 Virtuelles Device (AVD) einrichten

## Hinweise

- **Device Ram Size**
  - plus maximale Ram-Size für das Emulator Image bestimmt den maximalen Arbeitsspeicher, den der Emulator auf dem Hostrechner benötigt
  - wird dieser > 2 GB startet der Emulator nicht
  - guter Mittelwert: 512 MB (1024 MB verursacht häufig Probleme)
- **Skin mit hoher Bildschirm-Auflösung**
  - Emulator startet auf einigen Systemen nicht, wenn der Speicher nicht ausreicht.

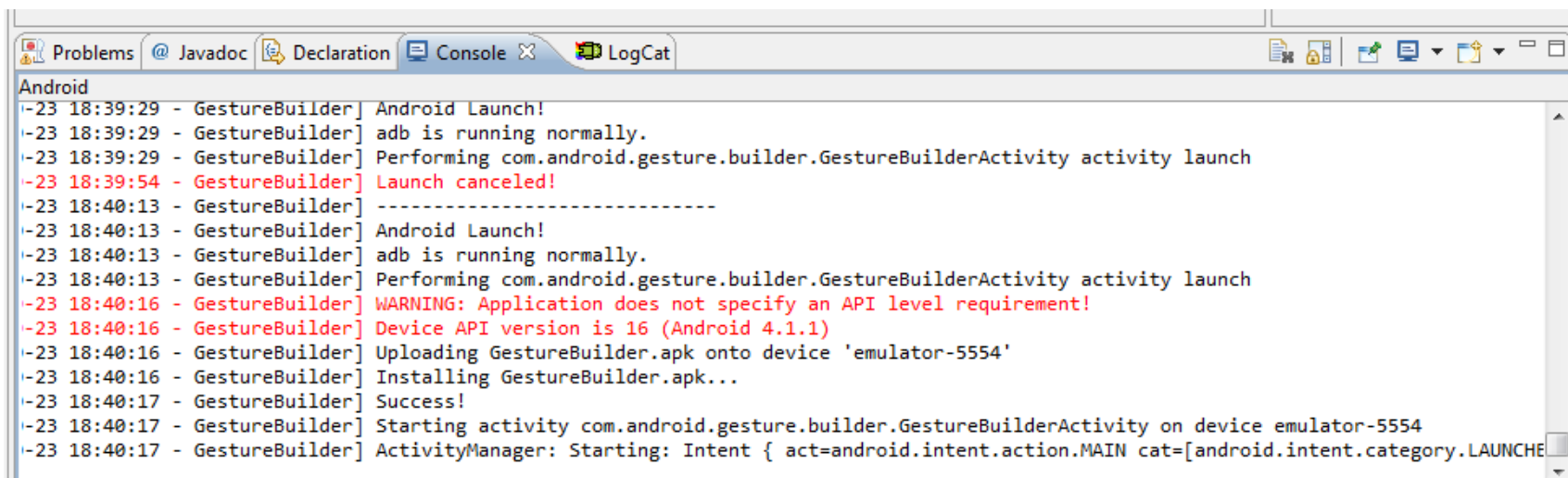
Arbeitsspeicher						
0 Seitenfehler/s			76% Verwendeter phys. Spei...			
Abbild	PID	Seitenf...	Zugesi...	Arbeits...	Freiga...	Privat (...)
emulator-x86.exe	7084	0	830.780	587.264	10.884	576.380
explorer.exe	1532	0	278.348	117.988	54.416	63.572
eclipse.exe	2800	0	271.512	246.528	18.204	228.324
SavService.exe	1196	0	177.468	184.832	15.176	169.656
sqlservr.exe	4688	0	129.968	1.920	284	1.636
firefox.exe	1636	0	118.596	137.700	31.684	106.016
SimpleTap.exe	4504	0	104.604	18.912	7.276	11.636
dwm.exe	1224	0	80.028	67.064	33.408	33.656
Skype.exe	3264	0	70.484	80.556	18.872	61.684
PrivateClient.exe	8140	0	55.348	38.716	16.348	13.468

# Target für Intel Atom (x86) installieren

- Eclipse als Admin starten
- Android Plugin installieren
- im SDK Manager unter Extras „Intel Hardware Accelerated Execution Manager“ auswählen
  - lädt IntelHAXM.exe. in das Verzeichnis  
    <sdk>extras/intel/Hardware\_Accelerated\_Execution\_Manager/
  - exe als Admin ausführen
  - im Bios VT bzw. VT-x für Intel GPU Prozessor aktivieren
- danach lässt sich als Target für ein virtuelles Device Intel Atom (x86) nutzen (schneller als ARM Target)
- bei Problemen die Farbtiefe mit Hilfe von Intel Tools auf 32 Bit einstellen
- Gilt nur für Windows für MacOS und Linus siehe  
(<http://developer.android.com/tools/devices/emulator.html>)

## 6 Android App auf AVD installieren

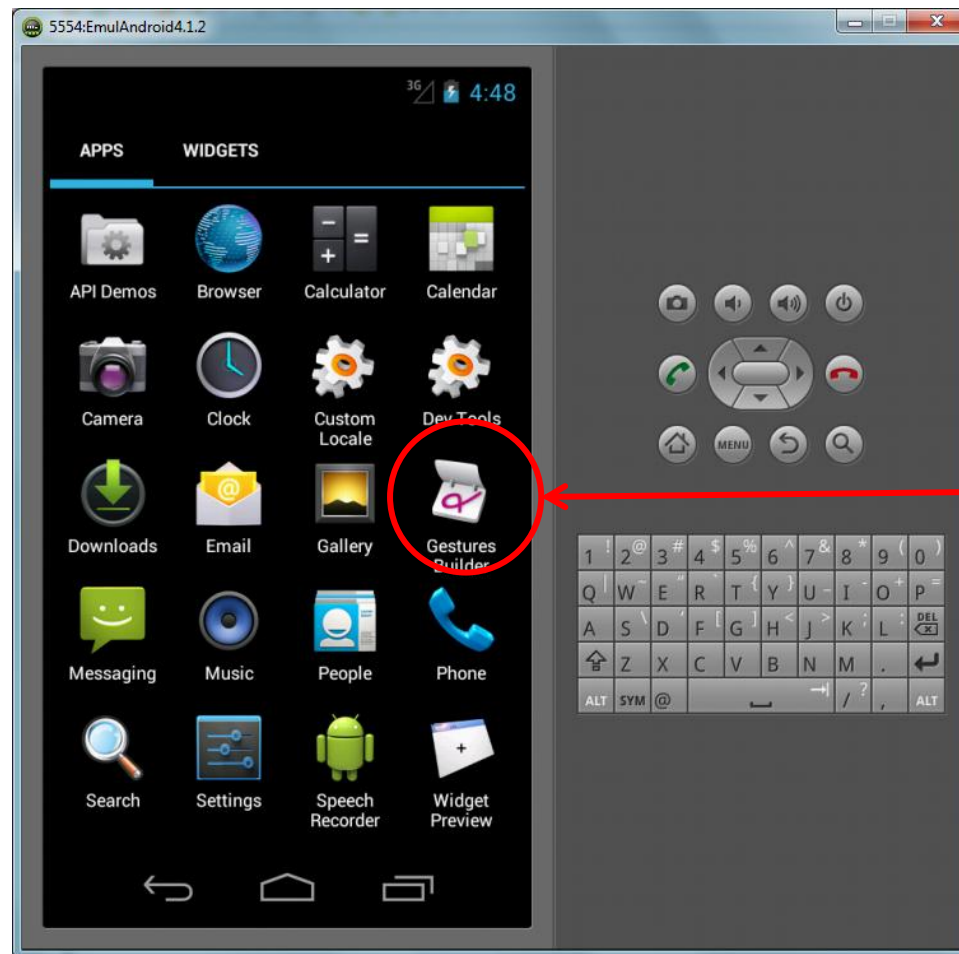
- File → New → Other → Android : Android Sample Project
- Richtiges Target auswählen (z.B. 4.1.2)
- GestureBuilder Projekt auswählen
- Emulator starten
- GestureBuilder Projekt selektieren → Run Configuration → Reiter Target  
CheckBox „Always Prompt To Pick Device“



```

Android
-23 18:39:29 - GestureBuilder] Android Launch!
-23 18:39:29 - GestureBuilder] adb is running normally.
-23 18:39:29 - GestureBuilder] Performing com.android.gesture.builder.GestureBuilderActivity activity launch
-23 18:39:54 - GestureBuilder] Launch canceled!
-23 18:40:13 - GestureBuilder] -----
-23 18:40:13 - GestureBuilder] Android Launch!
-23 18:40:13 - GestureBuilder] adb is running normally.
-23 18:40:13 - GestureBuilder] Performing com.android.gesture.builder.GestureBuilderActivity activity launch
-23 18:40:16 - GestureBuilder] WARNING: Application does not specify an API level requirement!
-23 18:40:16 - GestureBuilder] Device API version is 16 (Android 4.1.1)
-23 18:40:16 - GestureBuilder] Uploading GestureBuilder.apk onto device 'emulator-5554'
-23 18:40:16 - GestureBuilder] Installing GestureBuilder.apk...
-23 18:40:17 - GestureBuilder] Success!
-23 18:40:17 - GestureBuilder] Starting activity com.android.gesture.builder.GestureBuilderActivity on device emulator-5554
-23 18:40:17 - GestureBuilder] ActivityManager: Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHED]
  
```

# 6 Android App auf AVD installieren



Gesture Builder App

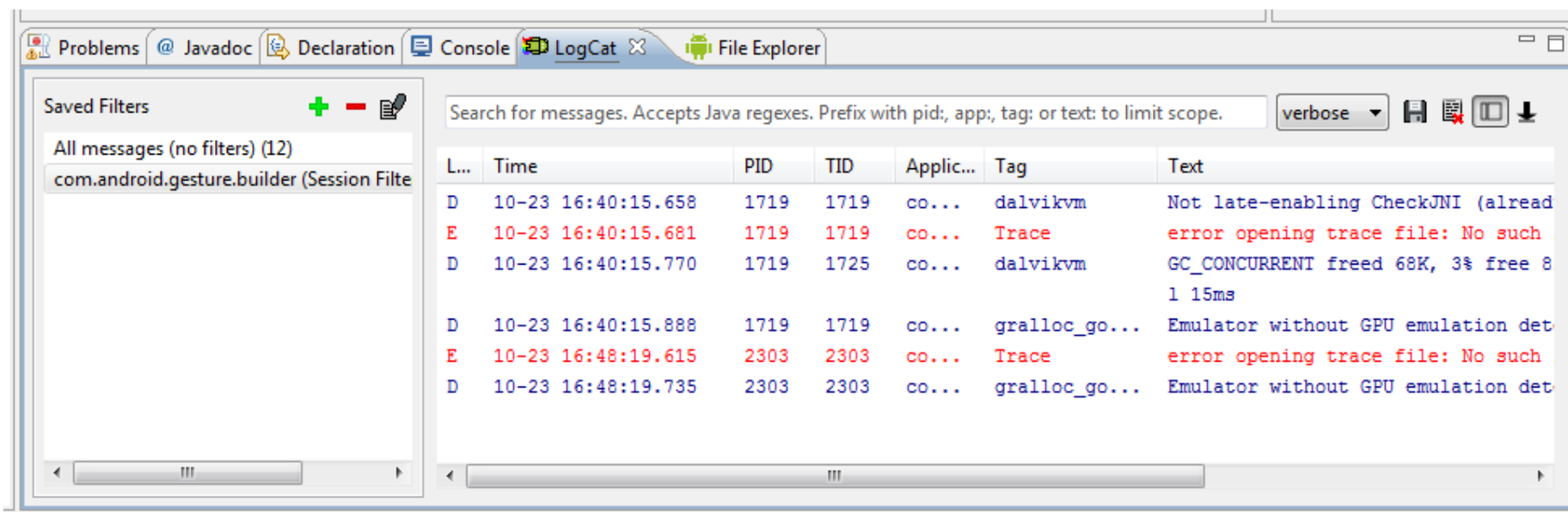
## 7 Android App auf Samsung Galaxy Tab 2 installieren

1. Kies installieren (dauert ein paar Minuten installiert u.a. Treiber für das Galaxy Tab) (<http://www.samsung.com/de/support/usefulsoftware/KIES/JSP>)
2. Galaxy Tab anschließen
3. Rechner neu starten
4. **Eclipse:** GestureBuilder als Android Application starten, dann das Samsung Device auswählen und installieren
5. Weiterer Test: Android Beispiel Projekt „BasicGLSurfaceView“ auf dem Galaxy Tab installieren



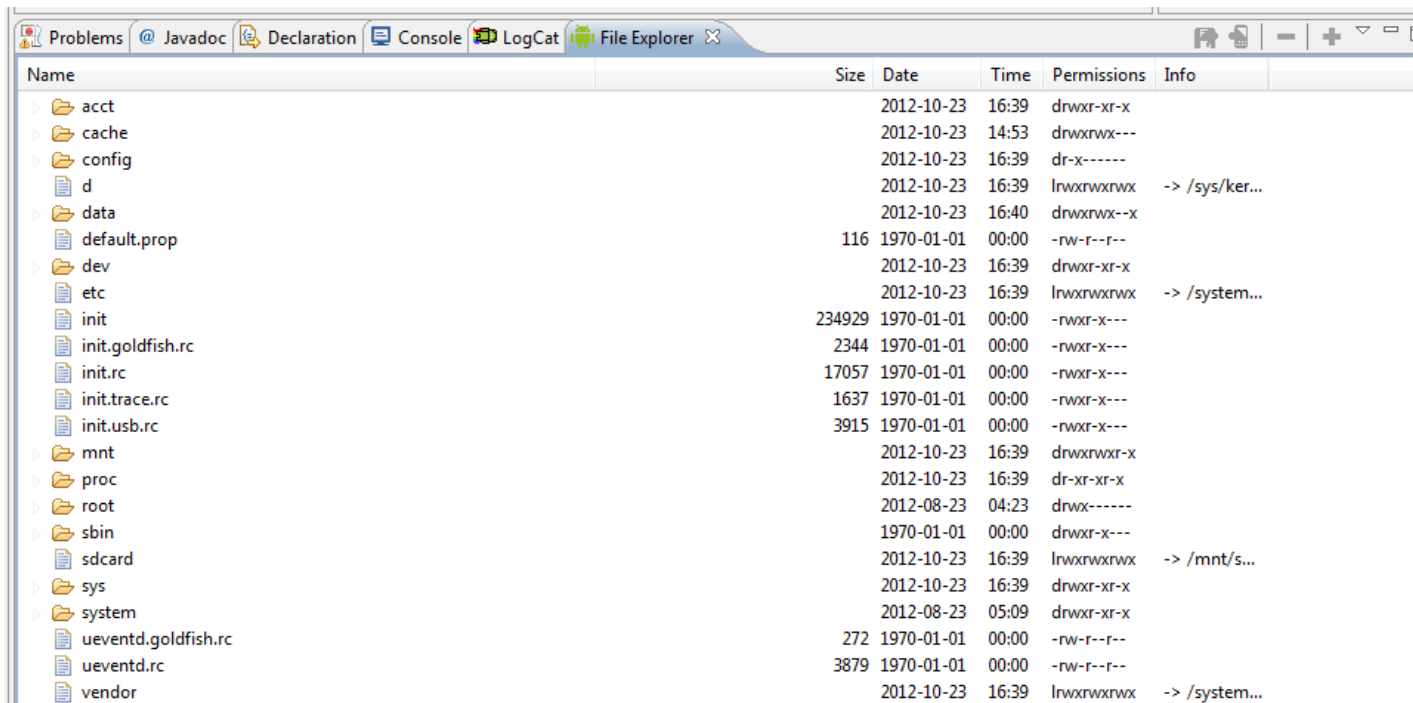
# Nützliche Tools

- LogCat logged Systemmeldungen der Android VM
  - **Einrichten:** Window → ShowView → Other → Android → LogCat
  - **Filter:** Einschränken der Meldungen auf Apps, Prozesse etc.



# Nützliche Tools

- DDMS für Inspektion und Management des Android BS
  - z.B. zum Laden von Daten auf die SD-Karte
  - **Einrichten:** Window → ShowView → Other → Android → File Explorer



Name	Size	Date	Time	Permissions	Info
acct		2012-10-23	16:39	drwxr-xr-x	
cache		2012-10-23	14:53	drwxrwx---	
config		2012-10-23	16:39	dr-x-----	
d		2012-10-23	16:39	lrwxrwxrwx	-> /sys/ker...
data		2012-10-23	16:40	drwxrwx--x	
default.prop	116	1970-01-01	00:00	-rw-r--r--	
dev		2012-10-23	16:39	drwxr-xr-x	
etc		2012-10-23	16:39	lrwxrwxrwx	-> /system...
init	234929	1970-01-01	00:00	-rwxr-x---	
init.goldfish.rc	2344	1970-01-01	00:00	-rwxr-x---	
init.rc	17057	1970-01-01	00:00	-rwxr-x---	
init.trace.rc	1637	1970-01-01	00:00	-rwxr-x---	
init.usb.rc	3915	1970-01-01	00:00	-rwxr-x---	
mnt		2012-10-23	16:39	drwxrwxr-x	
proc		2012-10-23	16:39	dr-xr-xr-x	
root		2012-08-23	04:23	drwx-----	
sbin		1970-01-01	00:00	drwxr-x---	
sdcard		2012-10-23	16:39	lrwxrwxrwx	-> /mnt/s...
sys		2012-10-23	16:39	drwxr-xr-x	
system		2012-08-23	05:09	drwxr-xr-x	
ueventd.goldfish.rc	272	1970-01-01	00:00	-rw-r--r--	
ueventd.rc	3879	1970-01-01	00:00	-rw-r--r--	
vendor		2012-10-23	16:39	lrwxrwxrwx	-> /system...

## 8 Project SmartHomeController installieren

- SmartHomeController extrahieren
- **Eclipse:** File → New → Other → Android Project From Existing Source (Verzeichnis angeben und Option „Copy Project into Workspace“ wählen)
- **Fehler:** de.ba.helper.FileWalker.findPluginFiles im LogCat
- **Ursache:** auf der SDCard wird im Verzeichnis /SmartHomeController/Plugins nach den Plugins \*.json gesucht, die nicht gefunden werden.
- **Lösung:** Erzeugen des Verzeichnisse und Kopieren der \*.json Dateien aus dem Android Projekt auf die SDCard
  - Windows Explorer „GT-P5110“ öffnen.
  - Das Tablet Device ist die SDCard.
  - Verzeichnis anlegen und Dateien kopieren, dann erneut die App installieren