

Apps-Entwicklung mit Netbeans

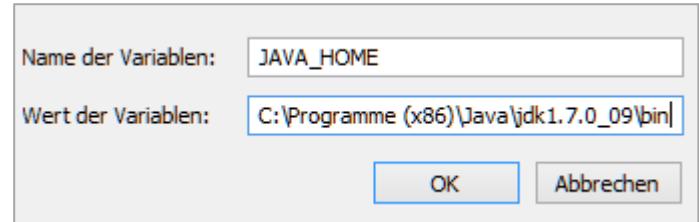
Version 2.2, 30. April 2013

Vorbereitungen:

1. **JDK SE** neuste Version installieren,

(<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)

Environment variable: JAVA_HOME
auf home Verzeichnis
Pfad auf <home>\bin



2. **Android-SDK** installieren

<http://developer.android.com/sdk/index.html>

Wählen: **DOWNLOAD FOR OTHER PLATFORMS**, dann SDK Tools Only (kein Bundle)

zip/tgz auspacken und in ein lokales Verzeichnis kopieren, z.B. unter Windows:

c:\programs\android-sdk (im Folgenden <sdkhome> genannt).

Pfad auf <sdkhome>\tools und auf <sdkhome>\platform-tools setzen

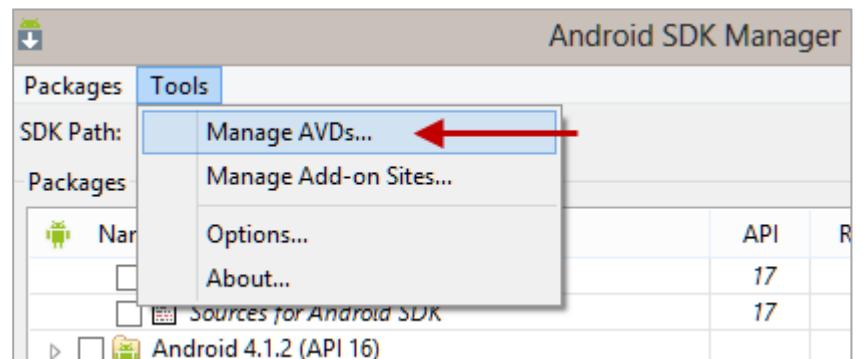
3. **SDK Manager starten** (im <sdkhome>) Choose package to install: Minimale Version: Alles wegwählen. Dann wieder dazuwählen:

- Android SDK-Tools
- Android SDK Platform-tools
- Android 2.2 (API 8)

Lizenz akzeptieren und Pakete installieren lassen

4. Smartphone Emulator

Im Menu **Tools** |
Manage AVDs wählen und
New klicken.



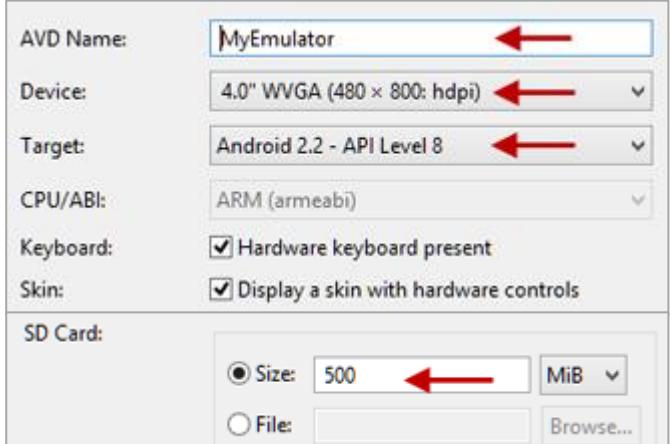
AVD Name: MyEmulator

Device: 4.0" WVGA

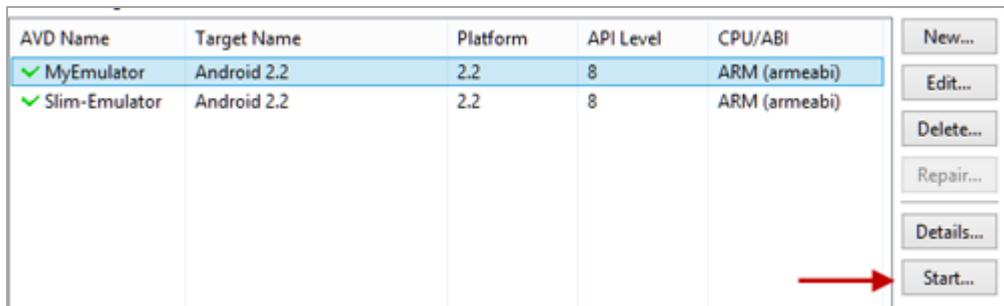
Target: Android 2.2

SD Card: Size: 500 MiB

Dann OK drücken (warten)



Dann im SDK-Manager MyEmulator klicken und Button Start drücken. Der Emulator wird gestartet (sehr lange warten).



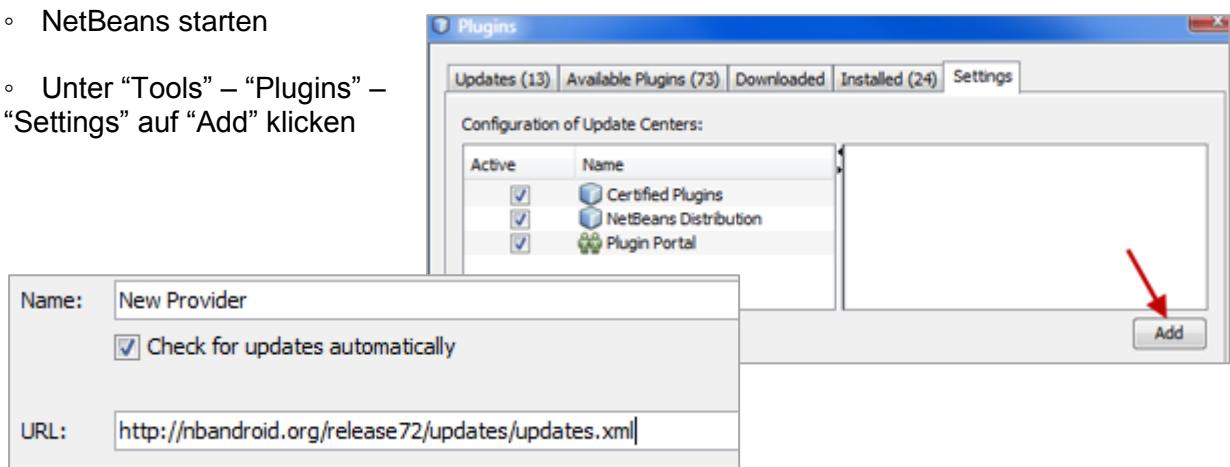
(Falls der Emulator ohne SDK-Manager gestartet werden soll, eine Batch-Datei schreiben mit Inhalt: emulator @MyEmulator und einen Link darauf erstellen.

5. Netbeans installieren

Netbeans neuste Version installieren, siehe <http://netbeans.org/downloads>, Version Java SE genügt

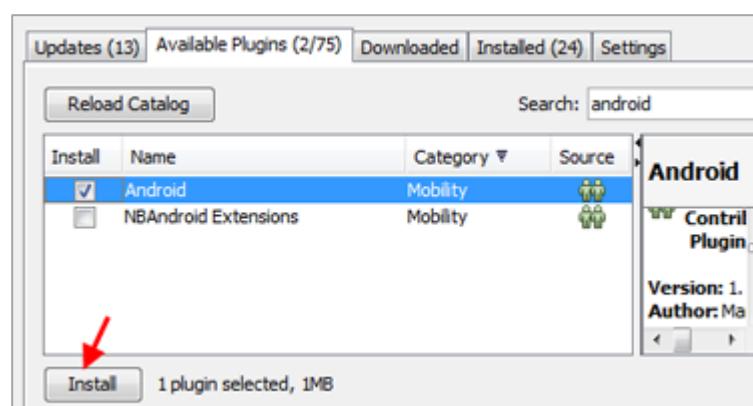
6. In Netbeans Android-Plugin installieren:

- NetBeans starten
- Unter "Tools" – "Plugins" – "Settings" auf "Add" klicken



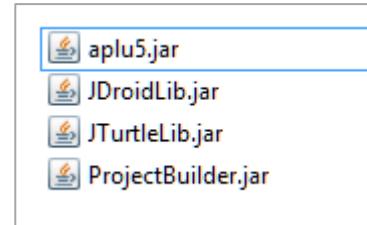
- Als URL <http://nandroid.org/release72/updates/updates.xml> hineinkopieren
- Der Name für die Update Center URL ist frei wählbar.
- Mit OK bestätigen

- Unter "Plugins" auf "Available Plugins" klicken
- Bei Search "android" eingeben und mit der Enter-Taste danach suchen
- Android auswählen
- Install, alles bestätigen



7. JDroid-Distribution

JDroidLib Distribution von <http://www.aplu.ch/android> downloaden und auspacken. **JDroidLib.jar** in irgend ein Verzeichnis mit anderen Jars kopieren, z.B. in c:\jars. **ProjectBuilder.jar** ebenfalls in c:\jars kopieren und einen Startlink darauf erstellen. Falls auch Turtle-Apps erstellt werden sollen, ebenfalls die Distribution von <http://www.aplu.ch/jturtlelib> downloaden und **JTurtleLib.jar** ins gleiche Verzeichnis kopieren



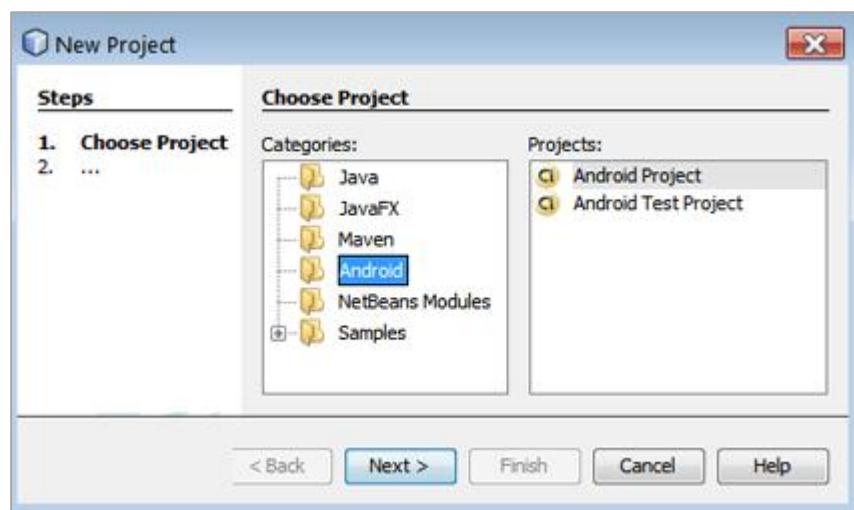
8. Neues Android Project erstellen

File | New Project

Choose Project: Android

Projects: Android Project

Next



Project Name: MyApp (Beispiel)

Project Location : e:\MyNetbeans

Set as Main Project wählen

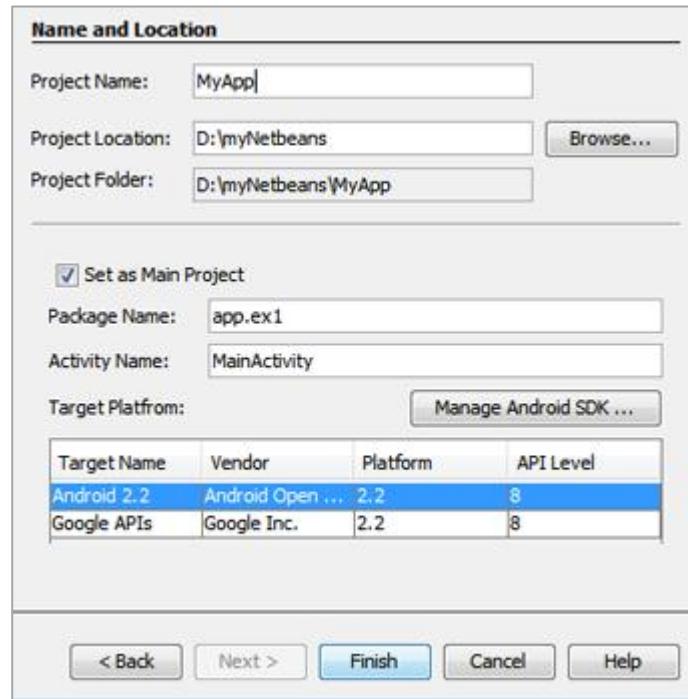
Package Name: app.ex1 (Beispiel)
muss mindestens einen Punkt enthalten

Activity Name: MainActivity

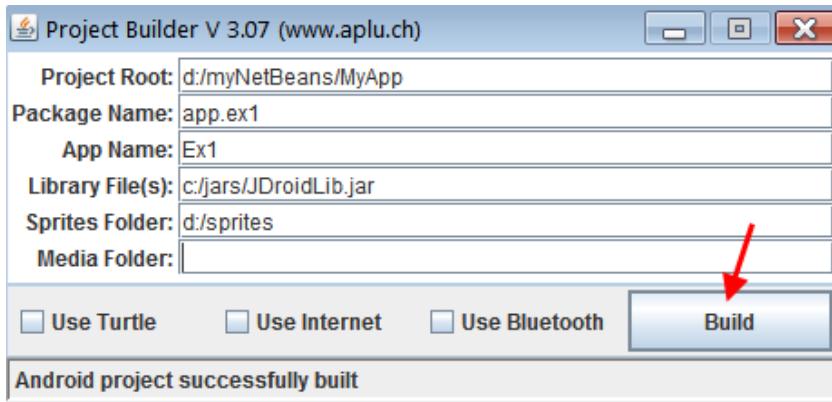
Bei erster Verwendung muss der Button: **Manage Android-SDK** geklickt werden und Android SDK home gewählt werden. Ok

Target Platform Android 2.2 für die meisten heutigen Handys

Finish

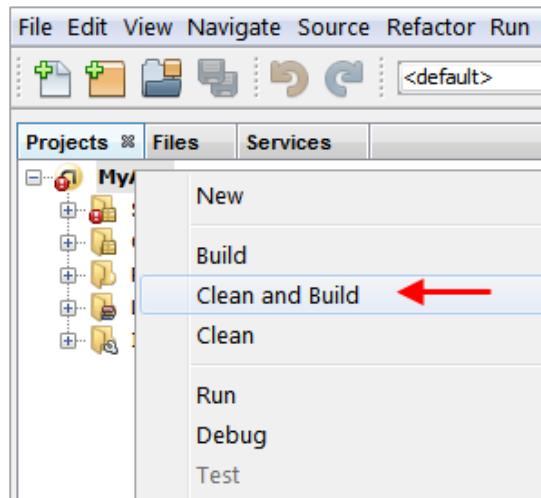


9. ProjectBuilder starten. Ausfüllen (gemäss Beispiel, anpassen!)

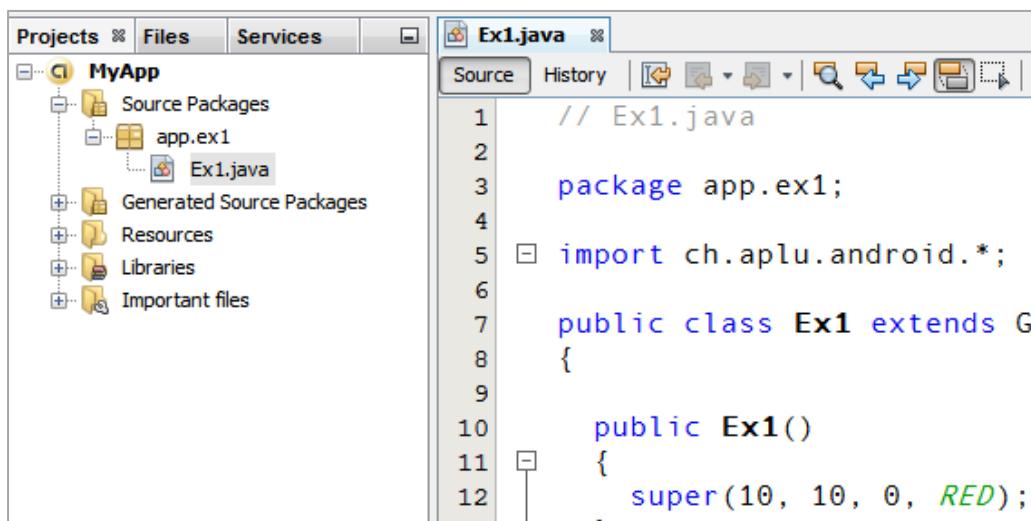


Der Package Name wird jetzt festgelegt. Sprites Folder und Media Folder können leer bleiben, andere Felder müssen ausgefüllt sein. Achtung im Feld *Library Folder* muss der qualifizierte Pfad zu den strichpunkt-getrennten Libraries angegeben werden. Für JDroid-Apps muss die Bibliothek **JDroidLib.jar** angegeben werden, für JTurtle-Apps die Bibliothek **JTurtleLib.jar** (Download von <http://www.aplu.ch/jturtlelib>). Zudem muss *Use Turtle* selektiert sein.

Klicken Sie mit der rechten Maustaste auf den Projektnamen und wählen Sie *Clean und Build* im Kontextmenü der rechten Maustaste



Die neue Applikation erscheint unter
Source Packages



Was macht der ProjectBuilder:

Der ProjektBuilder ersetzt

- AndroidManifest.xml
- <projectroot>\res\layout\main.xml
- <projectroot>\res\values\strings.xml
- <projectroot>\src\<packagepath>\<ApplicationName>.java erzeugt
- Library file in <projectroot>\libs kopiert (falls libs nicht existiert, wird der Folder erzeugt)
- Alle Files aus <SpritesFolder> in <projectroot>\res\drawable kopiert (folder drawable erzeugt, falls nicht bereits vorhanden). Files jdroid_gglogo.png und turtle_default.png in <projectroot>\res\drawable kopiert.
- In build.xml wird der App-name eingetragen
- Event. unter Menu Window Output wählen. Es wird im unteren Teil ein Fenster geöffnet, wo Android-spezifische Angaben sichtbar sind.
- Auf dem Projectnamen eine Rechtsklick und Run wählen. Das Project wird kompiliert/gepackt und nachfolgend auf ein angeschlossenes Smartphone kopiert und dort gestartet. Falls keine Smartphone angeschlossen ist, wird der Emulator automatisch gestartet und die App darauf installiert und gestartet. Falls das Project als Main Project bezeichnet wird, kann auch der Run-Pfeil in der Titelzeile geklickt werden.

10. Installation/Starten der App auf Emulator:

Den grünen **Run** Button in der Menuleiste oder Run aus dem Kontextmenü der rechten Maustaste wählen.

Das Projekt wird kompiliert/gepackt. Ist kein Smartphone angeschlossen, wird die App nachfolgend auf den Emulator kopiert und dort gestartet.

Da es lange dauert, bis der Emulator gestartet wird, lässt man ihn am besten offen, solange man entwickelt. Um die Erstellung des APK zu verfolgen, sollte man mit Menü Window | Show View | Console die Ausgabekonsole sichtbar machen. App erneut kompilieren/packen und installieren mit Ctrl-F11.

11. Installation/Starten der App auf Smartphone über USB

Modellspezifische Synchronisationssoftware installieren, z.B. bei HTC: *HTC Sync*, bei Samsung *Kies*.

Einstellungen auf dem Smartphone:

- Anwendungen: Unbekannte Quellen OK wählen
- Entwicklung: USB Debugging OK
- Wach bleiben
- Smartphone NICHT als externe Disk ansprechen (auf HTC: Einstellungen | Mit PC verbinden | Standardtypen NUR LADEN !)

In einer Command Shell kann mit **adb devices** überprüft werden, ob ein Device erkannt wird.. Das Smartphone muss mit seiner ID erscheinen, sonst ist der Smartphone-USB-Treiber nicht installiert/nicht funktionsfähig.

Smartphone über USB-Kabel anschliessen. Den grünen **Run** Button in der Menuleiste klicken. und Android Application auswählen.

Das Project wird kompiliert/gepackt und nachfolgend wird die App auf das Smartphone kopiert und dort gestartet.

12. Manuelle Installation einer App:

In einer Command Shell im Verzeichnis der Applikation

adb install -r Apps-debug.apk (Apps-debug.apk ist ein Beispiel)

ausführen. Eventuelle Fehlermeldung beachten.

Ergänzungen:

- Falls man die sichtbare Applikationsikone ändern will, im Verzeichnis res/drawable die Icone jdroid_gglogo.png ersetzen
- Falls die App auf dem Smartphone nicht startet, kontrollieren, ob es nicht bereits eine Applikation mit demselben Package oder sogar mit demselben Namen gibt. Falls ja, mit Smartphone-Tool unter (Einstellungen/Anwendungen) **deinstallieren** (da es einen Konflikt wegen der Signatur geben kann, falls man sich auf einem anderen Rechner befindet.)
- Falls man mehrere Apps im gleichen Projekt hat, muss immer wieder der ProjectBuilder ausgeführt werden werden, damit die App mit dem richtigen App-Name gestartet wird.
- Falls man im angegebenen Sprites-Folder ein Sprite ändert/hinzufügt, muss man den ProjectBuilder wieder ausführen, damit die Änderung übernommen wird