

Bauen eines Chromium-Android-Paketes (apk)

Nachfolgend soll das Bauen eines Chromium-Android-Paketes (apk) beschrieben werden. Es wird dabei die Anleitung der Seite [Googlesource](#) abgearbeitet. Alle nachfolgenden Schritte werden in einer CHROOT-Umgebung ausgeführt.

Voraussetzungen

Es wird ein DEBIAN-Linux-System vorausgesetzt, sowie eine Verbindung zum Internet.

Depot-Werkzeuge

Im ersten Schritt werden die Depot-Werkzeuge mit „git“ geklont:

```
~# git clone  
https://chromium.googlesource.com/chromium/tools/depot_tools.git
```

Damit diese geklonten Werkzeuge später ohne Angabe eines Pfades aufgerufen werden können, sollte der Pfad in die Pfadumgebungsvariable aufgenommen werden:

```
export PATH="$PATH:<Pfad zu Depot-Werkzeuge>/depot_tools"
```

Quelldaten herunterladen

Im nächsten Schritt können dann die Chromium-Quellen heruntergeladen werden:

```
~# mkdir ~/chromium && cd ~/chromium  
~$ fetch --nohooks android  
Running: gclient root  
Running: gclient config --spec 'solutions = [  
  {  
    "url": "https://chromium.googlesource.com/chromium/src.git",  
    "managed": False,  
    "name": "src",
```

```
"deps_file": ".DEPS.git",
"custom_deps": {},
},
]
target_os = ["android"]
'

Running depot tools as root is sad.
Running: gclient sync --nohooks
Running depot tools as root is sad.

_____ running 'git -c core.deltaBaseCacheLimit=2g clone --no-checkout --
progress https://chromium.googlesource.com/chromium/src.git
/root/chromium/_gclient_src_FK9Vzn' in '/root/chromium'
Cloning into '/root/chromium/_gclient_src_FK9Vzn'...
remote: Sending approximately 9.69 GiB ...
remote: Counting objects: 66611, done
remote: Finding sources: 100% (55/55)
Receiving objects:  0% (41226/10393332), 17.06 MiB | 617.00 KiB/s
[0:01:00] Still working on:
[0:01:00]   src

...

[9:47:43] Still working on:
[9:47:43]   src/third_party/android_tools/ndk
Syncing projects: 100% (110/110), done.
Running: git submodule foreach 'git config -f $toplevel/.git/config
submodule.$name.ignore all'
Running: git config --add remote.origin.fetch '+refs/tags/*:refs/tags/*'
Running: git config diff.ignoreSubmodules all
```

Dieser Vorgang kann, je nach Internetanbindung, eine Weile dauern.

Dann bitte in das neu erstellte Verzeichnis „src“ wechseln:

```
~# cd src
```

Bauabhängigkeiten

Im vorliegenden Fall mussten noch abhängige Pakete nachinstalliert werden, bevor die eigentlichen Bauabhängigkeiten ermittelt werden konnten:

```
~# apt install lsb-release
```

Jetzt kann das Skript gestartet werden, welches die Bauabhängigkeiten ermittelt:

```
~$ build/install-build-deps-android.sh
Skipping debugging symbols.
Including 32-bit libraries for ARM/Android.
Skipping ARM cross toolchain.
Skipping NaCl, NaCl toolchain, NaCl ports dependencies.
Get:1 http://security.debian.org/debian-security stretch/updates InRelease
[63.0 kB]
Ign:2 http://ftp.uni-hannover.de/debian/debian stretch InRelease
Get:3 http://ftp.uni-hannover.de/debian/debian stretch-updates InRelease
[91.0 kB]

...

install-build-deps-android.sh complete.
```

Hooks ausführen

Jetzt können die Hooks ausgeführt werden:

```
~# gclient runhooks
Running depot tools as root is sad.

_____ running '/usr/bin/python src/build/download_nacl_toolchains.py --
mode nacl_core_sdk sync --extract' in '/root/chromium'
INFO: --Syncing arm_trusted to revision
9f62c3ead52424e10ebd520213a8f36e02b84377--
INFO: Downloading package archive: sysroot_arm_trusted_precise.tgz (1/1)
INFO: --Syncing nacl_arm_glibc to revision
053185d68ed3b96640f15c4ae457b1ff373c6cac--
INFO: Downloading package archive: binutils_arm_x86_64_linux.tgz (1/6)
INFO: Downloading package archive: gcc_arm_x86_64_linux.tgz (2/6)

...

14.7 The License Agreement, and your relationship with Google under the
License Agreement, shall be governed by the laws of the State of California
without regard to its conflict of laws provisions. You and Google agree to
submit to the exclusive jurisdiction of the courts located within the county
of Santa Clara, California to resolve any legal matter arising from the
License Agreement. Notwithstanding this, you agree that Google shall still
be allowed to apply for injunctive remedies (or an equivalent type of urgent
legal relief) in any jurisdiction.
```

December 9, 2016

```
Do you accept the license for version 11.2.0 of the Google Play services
client library? [y/n]:
y

...

Running hooks: 96% (58/60) doclava
_____ running '/usr/bin/python src/build/android/download_doclava.py' in
'/root/chromium'
0> Downloading src/buildtools/android/doclava.tar.gz...
0> Extracting 7 entries from src/buildtools/android/doclava.tar.gz to
src/buildtools/android/doclava
Downloading 1 files took 35.700493 second(s)
Hook '/usr/bin/python src/build/android/download_doclava.py' took 35.75 secs
Running hooks: 100% (60/60), done.
```

Ninja-Bauverzeichnis

Chromium nutzt „Ninja“ zum Bauen der APK-Datei:

```
~# gn gen --args='target_os="android"' out/Default
Done. Made 17134 targets from 1345 files in 11041ms
```

Bauen

Jetzt kann die Chromium-APK-Datei gebaut werden:

```
~# ninja -C out/Default chrome_public_apk
ninja: Entering directory `out/Default'
[35827/35827] STAMP obj/chrome/android/chrome_public_apk.stamp
```

Die fertige APK-Datei ist dann unter „out/Default/apks/“ zu finden.

— [Steffen Bornemann](#) 13.02.2018

[Google](#), [Chromium](#), [APK](#), [Android](#), [GIT](#), [DEBIAN](#)

From:

<https://looper.de/wiki/> - **Linux4Ever**

Permanent link:

<https://looper.de/wiki/doku.php?id=android:apk-google-chromium>

Last update: **2025/12/11 15:00**

