

# dpkg / APT / Aptitude

- Alle durch Benutzer geänderten paketierte Konfigurationsdateien ausgeben
- Alle installierten Pakete auf einen anderen Server übertragen
- dpkg-Cheat-Sheet
- dpkg-rc Pakete entfernen
- Was bedeuten die verschiedenen flags bei dpkg -l?
- APT / Aptitude - Tipps und Tricks
- GPG-Key eines Repositorys in Apt hinzufügen

# Alle durch Benutzer geänderten paketierte Konfigurationsdateien ausgeben

```
dpkg-query -W -f='${Conffiles}\n' '*' | awk 'OFS=" " {print $2,$1}' | LANG=C md5sum -c 2>/dev/nu
```

# Alle installierten Pakete auf einen anderen Server übertragen

Vorher sollten als entfernt markierte Pakete gelöscht werden, siehe [hier](#).

Quelle:

```
dpkg --get-selections > selection.txt
```

dann die Datei per scp oder Copy&Paste auf das Ziel übertragen. Dort:

```
dpkg --set-selections < selection.txt
```

Falls hier Warnings auftreten siehe unten.

Danach können die Pakete installiert werden:

```
aptitude install  
# oder  
apt-get -u dselect-upgrade
```

Sollte es bei „-set-selections“ Warnings gegeben haben, muss noch dselect installiert werden:

```
apt-get install dselect  
dselect  
-> Update  
-> Install  
-> Quit
```

Folgende Alternative funktioniert auch, wenn man Pakete nur selektiv übertragen möchte. Mit diesem Beispiel erhalte ich eine einfach zu übertragende Liste aller Pakete, die `perl` im Namen oder der Beschreibung haben (läßt sich dann einfach an `aptitude install` übergeben):

```
dpkg -l | grep perl | awk '{ printf "%s ", $2 }'
```

# dpkg-Cheat-Sheet

## Pakete auf Version festhalten / hold

ein Paket auf „hold“ setzen

```
echo <paketname> hold | dpkg --set-selections
```

ein Paket wieder fürs Updaten freigeben

```
echo <paketname> install | dpkg --set-selections
```

gehaltene Pakete anzeigen

```
dpkg --get-selections | awk '$2 == "hold" { print $1 }'
```

## Fremdpakete finden

Die folgenden Tipps sind nicht immer zuverlässig

Pakete finden, die aus keinem der konfigurierten Repositories stammen (auf einem sauberen System sollte diese Liste leer sein):

```
aptitude search '?narrow(?not(?archive("^[^n][^o].*$")),?version(CURRENT))'
```

Pakete finden, die nicht aus einem Debian-Repo stammen

```
dpkg -l | awk '/^i/ {print $2}' | xargs apt-cache policy | awk '/^[a-z0-9.\-]+:/ {pkg=$1}; /\*\'
```



# Was bedeuten die verschiedenen flags bei dpkg -l?

Im Header von „dpkg -l“ steht die Erklärung dazu. Hier eine kurze Übersicht:

```
$ dpkg -l | head -6
```

```
Desired=Unknown/Install/Remove/Purge/Hold
```

```
| Status=Not/Inst/Conf-files/Unpacked/halF-conf/Half-inst/trig-aWait/Trig-pend
```

```
|/ Err?=(none)/Reinst-required (Status,Err: uppercase=bad)
```

```
||/ Name
```

```
Version
```

```
Architecture Description
```

```
ii accountsservice
```

```
0.6.35-0ubuntu7.2
```

```
amd64
```

```
query and manipulate
```

```
user account information
```

Hier die Beschreibung der einzelnen Felder:

Erster Buchstabe	gewünschter Zustand („selection state“)
u	unknown
i	install
r	remove/deinstall
p	purge (Paket soll inklusive der Konfigurationsdateien entfernt werden)
h	hold

Zweiter Buchstabe	aktueller Zustand
n	not-installed
i	installed
c	config-files (nur Konfigurationsdateien sind installiert)
u	unpacked
f	half-configured (Konfiguration ist aus irgendeinem Grund fehl geschlagen)
h	half-installed (Installation ist aus irgendeinem Grund fehl geschlagen)
w	triggers-awaited (Paket wartet auf Trigger durch ein anderes Paket)
t	triggers-pending (Paket wurde getriggert)

Zweiter Buchstabe	aktueller Zustand
B	broken (Installation/Update fehlgeschlagen)

Dritter Buchstabe	Fehler (da sollte normalerweise nichts stehen)
r	reinst-required (Paket ist kaputt und sollte neu installiert werden)

# APT / Aptitude - Tipps und Tricks

- Verfügbare Updates anzeigen, aber nicht installieren:

```
# aptitude -F%p --disable-columns search ~U
# oder:
# apt-get --just-print dist-upgrade
```

- durch „aptitude hold“ gehaltene Pakete:

```
aptitude search ~ahold
```

- Abhängigkeiten, Vorschläge, Empfohlen, Konflikte eines Paketes anzeigen:

```
# apt-cache depends tar
tar
PreDepends: libc6
Suggests: bzip2
Suggests: ncompress
Suggests: xz-utils
Conflicts: cpio
Breaks: dpkg-dev
Replaces: cpio
```

- warum ist ein bestimmtes Paket installiert worden (required by):

```
# aptitude why tar
i dpkg PreDepends tar (>= 1.23)
```

- Installierte Pakete anzeigen die kein anderes Paket voraussetzt (depends on):

```
aptitude search '~i ! ~R ~i'

# erweitert auf "Empfohlen" oder "Vorgeschlagen" (recommends / suggests):
aptitude search '~i ! ~R ~i ! ~Rrecommends:~i ! ~Rsuggests:~i'

# für alle Pakete, installiert oder nicht:
aptitude search '! ~R .'
```

# GPG-Key eines Repositorys in Apt hinzufügen

Beim Einbinden eines neuen Repositorys erscheint diese Meldung, wenn der GPG-Key, mit dem die Pakete signiert wurden, nicht in der APT-Datenbank enthalten ist:

```
W: GPG error: http://packages.linuxmint.com sarah Release:
The following signatures couldn't be verified because the
public key is not available: NO_PUBKEY A6616109451BBBBF2
```

Das lässt sich ganz einfach beheben:

```
# apt-key adv --recv-keys --keyserver keyserver.ubuntu.com A6616109451BBBBF2

Executing: gpg --ignore-time-conflict --no-options --no-default-keyring --homedir /tmp/tmp.in8AoFu6Qi --no-auto-
check-trustdb --trust-model always --keyring /etc/apt/trusted.gpg --primary-keyring /etc/apt/trusted.gpg --keyring
/etc/apt/trusted.gpg.d/steam.gpg --recv-keys --keyserver keyserver.ubuntu.com A6616109451BBBBF2
gpg: requesting key 451BBBBF2 from hkp server keyserver.ubuntu.com
gpg: key 451BBBBF2: public key "Linux Mint Repository Signing Key <root@linuxmint.com>" imported
gpg: Total number processed: 1
gpg:          imported: 1 (RSA: 1)
```