

# lokale Paketverwaltung

## dpkg und rpm

Funktion	Debian und Ubuntu	RedHat, Fedora und SuSE <sup>1)</sup>
Anzeigen welche Pakete installiert sind:	<code>dpkg -l</code> <code>COLUMNS=150 dpkg -l</code>	<code>rpm -qa</code>
Anzeigen zu welchem Paket die Datei <code>/usr/bin/top</code> gehoert:	<code>dpkg -S /usr/bin/top</code>	<code>rpm -qf /usr/bin/top</code>
Beschreibung zum Paket <code>procps</code> anzeigen:	<code>dpkg -s procps</code>	<code>rpm -qi procps</code>
Anzeigen welche Dateien zum Paket <code>procps</code> gehoeren:	<code>dpkg -L procps</code>	<code>rpm -ql procps</code>
Paket installieren <sup>2)</sup> :	<code>dpkg -i xskat_4.0-1_i386.deb</code>	<code>rpm -i datei.rpm</code>
Paket loeschen: <sup>3)</sup>	<code>dpkg --remove paket</code> <code>dpkg --purge paket</code>	<code>rpm -e &lt;paket&gt;</code>
Funktion	Debian und Ubuntu	RedHat, Fedora und SuSE <sup>4)</sup>
Pakete nach Größe sortiert ausgeben:	<code>dpkg-query -W --showformat='\${Installed-Size} \\${Package}\n'   sort -n</code>	<code>rpm -qa --queryformat '%{SIZE} %{NAME}\n'   sort -n</code>
Geaenderte Dateien suchen	<code>debsums -e procps</code>	<code>rpm -V procps</code>
Geänderte Konfigurationsdateien suchen	<code>debsums -ec</code>	<code>rpm -qac --dump   perl -naF'\s+' -e 'print \$F[0], \$/ if -f \$F[0] and not -l \$F[0] and (split /\s+/, `md5sum \$F[0]`)[0] ne \$F[3]'</code>
Konfigurationsdatei wiederherstellen	<code>cd /var/cache/apt/archives/</code> <code>dpkg -i --force-confmiss</code> <code>dokuwiki_0.0.20120125b-2_all.deb</code>	
Zuletzt installierte Pakete anzeigen:	<code>grep 'status installed'</code> <code>/var/log/dpkg.log</code>	<code>rpm -qa --last   tac</code>
Datenverzeichnis für Dienst <code>apache</code> finden:	<code>dpkg -S var/   grep apache</code>	

## Abhängigkeiten (rpm)

Von welchen Eigenschaften hängt das Paket `cups` ab?

```
rpm -q --requires cups
```

Welche Eigenschaften stellt das Paket `systemd` anderen Paketen zur Verfügung?

```
rpm -q --provides systemd
```

Welche Pakete hängen von *cups* ab?

```
rpm -q --whatrequires cups
```

Welches Paket stellt die Eigenschaft *syslog* zur Verfügung?

```
rpm -q --whatprovides syslog
```

Bei *dpkg* sind Pakete direkt von anderen Paketen abhängig.

## Repository basierte Paketverwaltung

TODO:

- [apt](#) ist neu in debian 8.0 (jessie). Dieses Kapitel überarbeiten, sobald debian 7 nicht mehr supportet wird.
- [dnf](#) statt yum, sobald das in CentOS ist.

### yum changelog

Changelog-Plugin installieren

```
yum install yum-plugin-changelog
```

Changelog abfragen

```
yum changelog 2017-10-01 "*openssh*"
```

oder bei Update

```
yum update <paket> --changelog
```

### apt, yum und zypper

Funktion	Debian	RedHat	SuSE
Repositories einbinden	<code>/etc/apt/sources.list</code> <code>/etc/apt/sources.list.d/*.list</code>	<code>/etc/yum.repos.d/*.repo</code>	<code>/etc/zypp/repos.d/*.repo</code>
Repository-Daten aktualisieren	<code>aptitude update</code>	<code>yum check-update</code>	<code>zypper refresh</code>
Updates einspielen	<code>aptitude safe-upgrade</code> oder <code>apt-get upgrade</code>	<code>yum update</code>	<code>zypper update</code>

Funktion	Debian	RedHat	SuSE
Distributions upgrade	aptitude full-upgrade oder apt-get dist-upgrade Ubuntu: do-release-upgrade	yum upgrade	zypper dup
Nach Paketen suchen	aptitude search suchwort apt-cache search suchwort	yum search suchwort yum list   grep paketname	zypper search suchwort
Beschreibung zu einem Paket anzeigen:	aptitude show paketname	yum info paketname	zypper info paketname
Paket installieren	aptitude install paketname apt-get install paketname	yum install paketname	zypper install paketname
Paket erneut installieren	apt-get install --reinstall paketname	yum reinstall paketname	TODO
Paket entfernen	aptitude remove paketname aptitude purge paketname apt-get [--purge] remove paketname	yum remove paketname	zypper remove paketname
Paket für Datei samba.schema im Repository suchen	apt-file update apt-file search samba.schema	yum provides */samba.schema	

### Abhängigkeiten (dpkg und apt)

Von welchen Paketen ist das Paket cups abhängig?

```
apt-cache depends cups
```

Welche Pakete sind vom Paket cups abhängig?

```
apt-cache rdepends cups
```

### yast, y2pmsh, rug und zypper

YaST, y2pmsh, rug und zypper sind nur für SuSE verfügbar.

Funktion	y2pmsh ( SuSE 9.2 bis 10.1 / SLES 9 & 10 ) <sup>5)</sup>	rug ( SuSE 10.1 bis 10.2 / SLES 10 )	zypper ( ab SuSE 10.2)
Repositories einbinden	y2pmsh source --add http://suse/i386/update/10.1 oder y2pmsh source -a http://suse/i386/update/10.1	rug service-add -t zypp http://suse/i386/update/10.1 oder rug sa -t zypp http://suse/i386/update/10.1	zypper service-add http://suse/i386/update/10.3 oder zypper service-add http://suse/i386/update/10.3
Repository-Daten aktualisieren		rug refresh	zypper refresh
Updates einspielen	y2pmsh upgrade y2pmsh commit	rug update	zypper update
Nach Paketen suchen	y2pmsh search suchwort	rug search suchwort	zypper search suchwort
Paketbeschreibung anzeigen:	y2pmsh show paketname	rug info paketname	zypper info paketname

Funktion	y2pmsh ( SuSE 9.2 bis 10.1 / SLES 9 & 10 ) <sup>5)</sup>	rug ( SuSE 10.1 bis 10.2 / SLES 10 )	zypper ( ab SuSE 10.2)
Paket installieren	y2pmsh <code>isc paketname</code> oder y2pmsh <code>install paketname</code> y2pmsh <code>solve</code> y2pmsh <code>commit</code>	rug <code>install paketname</code>	zypper <code>install paketname</code>
Paket entfernen	y2pmsh <code>remove paketname</code> y2pmsh <code>solve</code> y2pmsh <code>commit</code>	rug <code>remove paketname</code>	zypper <code>remove paketname</code>

Die Paketverwaltung von YaST baut bis SuSE 10.0 / SLES 9 auf dem Programm y2pmsh auf, daß man hier auch an Stelle von yast benutzen kann. y2pmsh hat auch einen interaktiven Modus: einfach starten, und die oben angegebenen Befehle funktionieren incl. Tab-Completion

y2pmsh/YaST/rug/zypper bietet die Möglichkeit verschiedene Repositories einzubinden. Eine Liste verfügbarer Repositories findet sich hier:

[http://www.opensuse.org/Additional\\_YaST\\_Package\\_Repositories](http://www.opensuse.org/Additional_YaST_Package_Repositories)

Hier ist beschrieben, wie man ein lokales Repository für YaST erstellt:

[http://en.opensuse.org/Network\\_Installation\\_Source](http://en.opensuse.org/Network_Installation_Source)

## Neues Repository samt Schlüssel zu Debian hinzufügen

Adresse des Repositories zu `/etc/apt/sources.list.d` hinzufügen.

Aus diesem Repository das Paket `repositoryname-keyring` installieren, darin sind die Schlüssel enthalten.

Oder: `apt-key add dateiname`, fügt die Schlüssel aus `dateiname` zum Schlüsselbund hinzu.

Mit `apt-key export keyId` wird ein Schlüssel aus einem bestehenden Schlüsselbund von APT auf stdout exportiert.

Mit `gnupg --export --armor keyId` wird ein Schlüssel aus einem privaten Schlüsselbund auf stdout exportiert.

## Dokumentation

[Fedora](#) [RedHat](#) [Debian](#) [openSuSE](#)

[yum](#)

## Zusätzliche Repositories

## Debian

- <http://backports.org> - neuere Versionen vieler Pakete für Debian stable
- <http://deb-multimedia.org/> - Software und Codecs für Multimedia

## Ubuntu

- <http://getdeb.net>
- <http://www.medibuntu.org/>

## RedHat / Centos / Scientific Linux

- <http://wiki.centos.org/Repositories>

## openSuSE

- [https://de.opensuse.org/Paket\\_Repositories](https://de.opensuse.org/Paket_Repositories)

## rpm

### Defragmentieren der RPM-Datenbank

```
rpm --rebuilddb
```

#### Links

- [Tips zu rpm und zypper](#)

### Einzelne Dateien aus einem .rpm-Paket rausfischen

Anzeigen welche Dateien drin sind: <sup>6)</sup>

```
rpm2cpio nxclient-1.5.0-113.i386.rpm | cpio -i -t
```

Datei extrahieren:

```
rpm2cpio nxclient-1.5.0-113.i386.rpm | cpio -id  
./usr/NX/share/keys/server.id_dsa.key
```

# Paketbeschreibungen

Die Beschreibungen von Paketen bei Debian sind im Original auf Englisch. Deutsche (und anderssprachige) Übersetzungen werden mit Hilfe des Debian Description Translation Project <http://ddtp.debian.net/> organisiert und bei nächtlichen Durchläufen mit beachtet.

1) , 4)

OK, rpm ist auch für debian verfügbar, aber sollte da nur in besonderen Fällen eingesetzt werden

2)

das macht man normalerweise mit *apt-get*, *yum*, *zypper* oder *yast*

3)

auch das geht mit *apt-get*, *yum*, *zypper* oder *yast*

5)

nicht offiziell unterstützt - aber brauchbar

6)

mehr zu [cpio](#)

From:

<https://wiki.lab.linuxhotel.de/> - **Linuxhotel Wiki**

Permanent link:

<https://wiki.lab.linuxhotel.de/doku.php/lpi1:paketverwaltung?rev=1511191121>

Last update: **2017/11/20 15:18**

