

# RPM / YUM

- [RPM Kurzübersicht](#)
- [RPM-Datenbank abfragen](#)
- [RPM-Datenbank defekt](#)
- [RPM-Paket entpacken](#)
- [YUM - Tipps und Tricks](#)

# RPM Kurzübersicht

Einige Tips zum Umgang mit dem Redhat Package Manager (RPM)

\* Alle installierten Pakete anzeigen (in dieser Liste lässt sich auch suchen, wenn man in der Ausgabe greift)

```
rpm -qa
```

\* Informationen zu einem installierten Paket ausgeben:

```
rpm -qi paketname
```

\* Herausfinden zu welchem Paket eine bestimmte Datei gehört

```
rpm -qf /bin/cp
```

\* Liste aller in einem Paket enthaltenen Dateien anzeigen

```
rpm -ql paketname
```

\* Mit der zusätzlichen Angabe des Parameters p kann man auch ein nicht installiertes RPM-Paket abfragen

```
rpm -qlp programm-1.0.rpm
```

\* ein installiertes Paket auf Veränderungen prüfen

```
rpm -V paketname
```

\* ein heruntergeladenes Paket überprüfen

```
rpm -K paketname
```

\* Die Dateirechte aller RPM-Pakete auf den Installationszustand zurücksetzen (für Dateien, die NICHT im Päckchen enthalten sind, gilt dies nicht)

```
for package in `rpm -qa` ; do rpm --setperms --setugids "${package}"; done
```

# RPM-Datenbank abfragen

Mit `rpm -q` lassen sich detaillierte Informationen aus der RPM-Datenbank ziehen. Welche Tags einem hier zur Verfügung stehen zeigt dieser Befehl:

```
[root@ov ~]# rpm --querytags | sort
```

```
ARCH
ARCHIVESIZE
BASENAMES
BUILDARCHS
BUILDHOST
BUILDTIME
C
CAPABILITY
CHANGELOGNAME
CHANGELOGTEXT
CHANGELOGTIME
CLASSDICT
CONFLICTFLAGS
CONFLICTNAME
CONFLICTS
CONFLICTVERSION
COOKIE
DEPENDSDICT
DESCRIPTION
DIRINDEXES
DIRNAMES
DISTRIBUTION
DISTTAG
DISTURL
DSAHEADER
E
EPOCH
EXCLUDEARCH
EXCLUDEOS
EXCLUSIVEARCH
EXCLUSIVEOS
FILECLASS
FILECOLORS
```

FILECONTEXTS  
FILEDEPENDSN  
FILEDEPENDSX  
FILEDEVICES  
FILEDIGESTALGO  
FILEDIGESTS  
FILEFLAGS  
FILEGROUPNAME  
FILEINODES  
FILELANGS  
FILELINKTOS  
FILEMD5S  
FILEMODES  
FILEMTIMES  
FILENAMES  
FILEPROVIDE  
FILERDEVS  
FILEREQUIRE  
FILESIZES  
FILESTATES  
FILEUSERNAME  
FILEVERIFYFLAGS  
FSCONTEXTS  
FSNAMES  
FSSIZES  
GIF  
GROUP  
HDRID  
HEADERI18NTABLE  
HEADERIMAGE  
HEADERIMMUTABLE  
HEADERREGIONS  
HEADERSIGNATURES  
ICON  
INSTALLCOLOR  
INSTALLTID  
INSTALLTIME  
INSTPREFIXES  
LICENSE  
LONGARCHIVESIZE

LONGFILESIZES  
LONGSIGSIZE  
LONGSIZE  
N  
NAME  
O  
OBSOLETEFLAGS  
OBSOLETENAME  
OBSOLETES  
OBSOLETEVERSION  
OLDFILENAMES  
OPTFLAGS  
ORIGBASENAMES  
ORIGDIRINDEXES  
ORIGDIRNAMES  
ORIGFILENAMES  
OS  
P  
PACKAGER  
PATCH  
PATCHESFLAGS  
PATCHESNAME  
PATCHESVERSION  
PAYLOADCOMPRESSOR  
PAYLOADFLAGS  
PAYLOADFORMAT  
PKGID  
PLATFORM  
POLICIES  
POSTIN  
POSTINPROG  
POSTTRANS  
POSTTRANSPROG  
POSTUN  
POSTUNPROG  
PREFIXES  
PREIN  
PREINPROG  
PRETRANS  
PRETRANSPROG

PREUN  
PREUNPROG  
PROVIDEFLAGS  
PROVIDENAME  
PROVIDES  
PROVIDEVERSION  
PUBKEYS  
R  
RECONTEXTS  
RELEASE  
REMOVETID  
REQUIREFLAGS  
REQUIRENAME  
REQUIRES  
REQUIREVERSION  
RHNPLATFORM  
RPMVERSION  
RSAHEADER  
SHA1HEADER  
SIGGPG  
SIGMD5  
SIGPGP  
SIGSIZE  
SIZE  
SOURCE  
SOURCEPACKAGE  
SOURCEPKGID  
SOURCERPM  
SUMMARY  
TRIGGERCONDS  
TRIGGERFLAGS  
TRIGGERINDEX  
TRIGGERNAME  
TRIGGERSCRIPTPROG  
TRIGGERSCRIPTS  
TRIGGERTYPE  
TRIGGERVERSION  
URL  
V  
VENDOR

```
VERIFYSRIPT
VERIFYSRIPTPROG
VERSION
XPM
```

Nun lässt sich mit Hilfe dieser Tags und dem Parameter `--queryformat` eine ordentliche Ausgabe erzeugen. Will man z.B. wissen für welche Architektur ein Paket gebaut wurde empfiehlt sich etwa dieser Befehl:

```
[root@ov ~]# rpm -q --queryformat "%{NAME}-%{VERSION}-%{ARCH}\n" libxml2
libxml2-2.7.3.x86_64
libxml2-2.7.3.i386
```

dies bedeutet, dass das Paket libxml in der i386 und in der 64bit Version (x86\_64) installiert ist.

Eine praktische Anwendung des Queryformat bietet sich auch an. Vor allem auf Systemen, auf denen 64- und 32-bit Pakete installiert sind. Einfach folgenden Code in `/etc/profile` oder in eine Datei in `/etc/profile.d` eintragen. Nach erneuter Anmeldung oder sourcen von `/etc/profile` steht das „rpmqa“-Kommando zur Verfügung, das folgende Ausgabeformat hat:

```
acl-2.2.39-6.el5.x86_64
acpid-1.0.4-9.el5_4.2.x86_64
alsa-lib-1.0.17-1.el5.i386
alsa-lib-1.0.17-1.el5.x86_64
amavisd-new-2.4.5-1.el5.noarch
amtu-1.0.6-1.el5.x86_64
anacron-2.3-45.el5.centos.x86_64
ant-1.6.5-2jpp.2.x86_64
antlr-2.7.6-4jpp.2.x86_64
apr-1.2.7-11.el5_3.1.i386
apr-1.2.7-11.el5_3.1.x86_64
apr-devel-1.2.7-11.el5_3.1.x86_64
apr-util-1.2.7-11.el5.i386
apr-util-1.2.7-11.el5.x86_64
apr-util-devel-1.2.7-11.el5.x86_64
[...]
```

# RPM-Datenbank defekt

Wenn rpm oder yum beim Aufruf hängen und keine Abfragen mehr gehen ist mit ziemlicher Wahrscheinlichkeit die RPM-Datenbank defekt. Die Datenbank kann folgendermaßen wiederhergestellt werden:

zuerst alle laufenden rpm-Prozesse abwürgen:

```
ps -ax | grep rpm  
kill -9 <pid>
```

mit folgenden Befehlen läßt sich die rpm-Datenbank neu aufbauen:

```
rm /var/lib/rpm/_*  
rpm --rebuilddb  
rpm --initdb
```

# RPM-Paket entpacken

Alle Dateien aus einem RPM entpacken:

```
$ rpm2cpio RPM_file | cpio -idv
```

Ein bestimmtes File aus einem RPM entpacken:

```
$ rpm2cpio RPM_file | cpio -id individual_file(s)
```

z.B. libcrypto.so.0.9.7a und libssl.so.0.9.7a aus openssl-0.9.7a-2.i386.rpm extrahieren:

```
$ rpm2cpio openssl-0.9.7a-2.i386.rpm | cpio -it egrep "libcrypto.so.0.9.7a|libssl.so.0.9.7a"

-rwxr-xr-x  1 root    root      992092 Feb 27 12:10 ./lib/libcrypto.so.0.9.7a
-rwxr-xr-x  1 root    root      216004 Feb 27 12:10 ./lib/libssl.so.0.9.7a

$ rpm2cpio openssl-0.9.7a-2.i386.rpm | cpio -idv ./lib/libssl.so.0.9.7a
./lib/libcrypto.so.0.9.7a
```

Eine cpio Datei entpacken:

```
$ cpio -iv < cpio_file
```

Den Inhalt einer cpio Datei anzeigen:

```
$ cpio -itv < cpio_file
```

Ein cpio file mit den Dateien im aktuellen Verzeichnis erstellen:

```
$ ls | cpio -o > cpio_file
```

# YUM - Tipps und Tricks

- Pakete nach Namen und Beschreibung suchen:

```
yum search blabla
```

- Info zu einem bestimmten Paket anzeigen:

```
yum info bind
```

- Pakete nur nach Namen suchen:

```
yum list blabla
```

- Suchen welches Paket eine bestimmte Datei zur Verfügung stellt:

```
yum whatprovides 'libstdc++.so.5'
```

- kürzlich neu hinzugekommene Pakete eines Repos anzeigen

```
yum list recent
```

- Alle Abhängigkeiten eines Paketes anzeigen:

```
yum deplist bind
```

- lokal heruntergeladenes Paket mit yum installieren und eventuelle Abhängigkeiten gleich aus dem Repository nachziehen, z.B.:

```
yum --nogpgcheck localinstall opera opera-9.24-20071015.6-shared-qt.i386-en.rpm
```