

## HOWTO zu RPM (RedHat Package Manager)

(C) 2008–2023 T.Birnthaler/H.Gottschalk <howtos(at)ostc.de>  
OSTC Source Training and Consulting GmbH  
<http://www.ostc.de>

\$Id: unix-rpm-HOWTO.txt,v 1.17 2025/02/23 20:14:55 tsbirn Exp \$

Dieses Dokument beschreibt die Installation und Verwaltung von RPM-Software-Paketen (RedHat Package Manager).

---

### INHALTSVERZEICHNIS

- 1) Einführung
  - 2) RPM-Haupt-Optionen
  - 3) Zusätzliche Angaben zu Optionen "--install/freshen/upgrade/erase"
  - 4) Zusätzliche Angaben zu Optionen "--verify/query"
  - 5) Änderungsstati/Dateihinweise der Option "--verify"
  - 6) Aufruf-Beispiele
  - 7) Hinweise
  - 8) UNIX Standard Software-Installationsmethode
  - 9) Standard-UNIX-Programm-Pfade für Anwenderprogramme
- 

#### 1) Einführung

---

RPM ist ein Paketmanager, mit dem unter Linux Softwarepakete verwaltet werden. Er pflegt dazu eine RPM-Datenbank "/var/lib/rpm/\*" der installierten Softwarepakete. Die Aufgaben eines Paketmanagers sind:

- \* Überprüfen ob Platz für Paket-Installation ausreicht
- \* Anwendung aus Quellen erzeugen
- \* Anwendung installieren
- \* Anwendung konfigurieren
- \* Anwendung updaten (ohne alte Konfiguration/Anpassungen zu zerstören)
- \* Auf Konflikte mit anderer Software prüfen
- \* Voraussetzungen und Abhängigkeiten überprüfen
- \* Korrekte Besitzverhältnisse, Zugriffsrechte, Datum und Inhalt überprüfen
- \* Anwendung deinstallieren

RPM arbeitet bei der Installation eines Pakets in mehreren Phasen:

Nr	Beschreibung	Typ
1	Preinstall-Skript ausführen	Optional
2	Archiv auspacken und Inhalt in Dateisystem kopieren	Muss
3	Postinstall-Skript ausführen	Optional
4	Deinstall-Skript ausführen	Optional

Mögliche Fehler beim Installieren eines RPM-Pakets sind:

- \* Abhängigkeiten mit anderen Paketen sind nicht auflösbar
- \* Konflikte mit bereits installierten Paketen (gleichnamige Dateien)
- \* Fehler bei der Konfiguration eines Pakets

#### 2) RPM-Haupt-Optionen

---

Option	Bedeutung
--------	-----------

-? --help	Usage-Meldung ausgeben
-q --query	Installierte Pakete oder Informationen zu Paketen abfragen
-i --install	Paket installieren
-U --upgrade	Paket aktualisieren (falls nicht inst. oder älteres inst.)
-F --freshen	Paket aktualisieren (NUR falls bereits älteres inst.)
-e --erase	Paket entfernen
-V --verify	Integrität eines inst. Pakets überprüfen ("SM5DLUGT", s.u.)
-K --checksig	RPM-DATEI original und unbeschädigt

### 3) Zusätzliche Angaben zu Optionen "--install/upgrade/freshen/erase"

Option	Bedeutung
-h --hash	Fortschrittsanzeige einschalten (###...)
--ignoresize	Platzbedarf vor Install. nicht prüfen (Netzlaufwerk!)
--nodeps	Paketabhängigkeiten ignorieren [no dependencies]
--noscripts	Ausführung von Paket-Skripten unterdrücken
--percent	Prozentanzeige während Installation einschalten
--prefix=DIR	Pfad dem Inst.verz. voransetzen (falls relocierbar)
--relocate=OLD=NEW	Verz.pfad umsetzen (mehrfach angebbbar)
--replacefiles	Dateikonflikte zw. Paketen ignorieren
--test	Erfolg der Paketinst. prüfen (aber nicht durchführen)

### 4) Zusätzliche Angaben zu Optionen "--verify/query"

Option	Bedeutung
-a --all	Alle installierten Pakete
-i --info	Information über Paket
-l --list	Dateiliste des Pakets, eingeschränkt durch:
-c --configfiles	Nur Konfigurationsdateien
-d --docfiles	Nur Dokumentationsdateien
-s --state	Nur Status ("normal", "not installed", "replaced")
-R --requires	Was ein Paket benötigt (Pakete oder Dateien)
--provides	Was ein Paket enthält (Pakete oder Dateien)
--scripts	(De)Installations-Skripte auflisten
-p --package	Noch nicht installiertes Paket untersuchen
-f FILE ...	Zu welchem Paket gehört Datei FILE
--qf=FORMAT	Ausgabeformat festlegen (queryformat)
--querytags	Mögliche Formatangaben für --qf auflisten

Formatangabe für -qf:

\* Eine Formatangabe enthält Header-Tag-Namen der Form "%{TAGNAME}", die durch eine Breitenangabe und die Angabe linksbündig erweitert sein können (z.B. %{FILENAMES}, %{FILESIZES} "%-20{LICENSE}, %-10{SIZE}, %-30{NAME})

\* Die Liste der Header-Tags erhält man mit "rpm --querytags".

### 5) Änderungsstati/Dateihinweise der Option "--verify"

Folgende Änderungsstati werden bei der Option "--verify" in Form einer Flagleiste "SM5DLUGTP" zu jeder Datei angezeigt:

Flg	Eigenschaft	Bedeutung
S	Size	Dateigröße unterschiedlich
M	Mode	Dateityp oder -rechte unterschiedlich
5	MD5	MD5-Checksumme unterschiedlich
D	Device	Gerätenummer(n) unterschiedlich (major/minor)
L	Link	Symbolischer Link passt nicht (Zielpfad)
U	User	Benutzer unterschiedlich
G	Group	Gruppe unterschiedlich
T	Time	Zeitpunkt der letzten Änderung unterschiedlich
P	Capabilities	Inhalt des Pakets (Name, Version, Bibliotheken, Files)

#### HINWEIS:

- \* Status "." statt Buchstabe: Test bestanden.
- \* Status "?" statt Buchstabe: Test nicht durchführbar (wg. Zugriffsrechten)
- \* Dateien, die ALLE Tests bestanden haben, werden NICHT angezeigt.
- \* Fehlende Dateien haben "missing" als Wert
- \* Für Sicherheitschecks sehr sinnvoll!

Folgende Dateihinweise werden bei Option "--verify" angezeigt:

Typ	Typ	Bedeutung
c	config	Konfigurationsdatei
d	doc	Dokumentationsdatei
g	ghost	Nicht im Paket enthalten
l	licence	Lizenzdatei
r	readme	Readmedatei
	(LEER)	Sonstige Datei

#### 6) Aufruf-Beispiele

- \* Alle installierten Pakete auflisten (-q=query, -a=all):  
rpm -qa
- \* Prüfen ob PAKET installiert ist und in welcher Version (-q=query, -a=all):  
rpm -q PAKET  
rpm -qa | grep -i PAKET
- \* Dateiliste eines installierten PAKETs auflisten (-q=query):  
rpm -q --list PAKET
- \* Beschreibung eines installierten PAKETs ausgeben (-q=query):  
rpm -q --info PAKET
- \* Pre/Postinstallations-Skripte eines installierten PAKETs ausgeben (-q=query):  
rpm -q --scripts PAKET
- \* Zu welchem installierten RPM-Paket gehört DATEI (-q=query, -f=file):  
rpm -q -f /PFAD/ZU/DATEI
- \* Dateien eines RPM-PAKETs auf Änderung seit Installation prüfen (-V=verify):  
rpm -V PAKET
- \* ALLE installierten Pakete auf Änderung seit Installation prüfen (-V=verify):  
rpm -Va

- \* Was ist in RPM-DATEI enthalten (-q=query, -p=package):  
rpm -q -p DATEI.rpm
- \* Ist RPM-DATEI original und unbeschädigt (-K=--checksig):  
rpm -K DATEI.rpm
- \* RPM-DATEI installieren (falls fehlend, -i=install -v=verbose -h=Fortschritt):  
rpm -ivh DATEI.rpm
- \* RPM-DATEI inst. (nur falls ältere Version vorhanden, -F=freshen):  
rpm -Fvh DATEI.rpm
- \* RPM-DATEI inst. (falls nicht inst. oder ältere Version vorhanden, -U=update):  
rpm -Uvh DATEI.rpm
- \* RPM-Paket deinstallieren (falls nicht von anderem Paket benötigt, -e=erase):  
rpm -e PAKET
- \* RPM-Paket deinstallieren (ohne Abhängigkeitsüberprüfung):  
rpm -e --nodeps PAKET
- \* In allen Paketen auf einer Installations-CD nach einer bestimmte Datei suchen:  
LIB="libgif"  
for PAKET in /media/cdrom/RedHat/RPMS/\*.rpm # RedHat  
for PAKET in /media/cdrom/suse/{i586,i686,noarch,nosrc,src}/\*.rpm # SuSE  
do  
rpm -qp --provides \$PAKET | grep -iq \$LIB && echo "\$PAKET contains \$LIB"  
done
- \* Größe der Dateien in Paket "XFree86" ausgeben:  
rpm -q --qf "[%{FILENAMES} %{FILESIZES} bytes\n]" XFree86
- \* Größe der Dateien in Paket "xf86" aufsummieren:  
echo \$((`rpm -q --qf "[%{FILESIZES+}]" xf86`0)) # ohne Overhead!
- \* Lizenzen der installierten RPM-Pakete ausgeben:  
rpm -qa --qf "%-20{LICENSE}\n" | sort | uniq -c | sort -n
- \* Größtes installiertes RPM-Paket ausgeben (--qf=queryformat):  
rpm -qa --qf "%-10{SIZE} %-30{NAME}\n" | sort -n | tail -1

## 7) Hinweise

---

- \* Der Name von RPM-Paketen sagt leider wenig über ihren konkreten Inhalt aus (besonders problematisch bei "Provides"-Namen). Die verschiedenen Distributionen verwenden hier auch kein einheitliches Namensschema.
- \* Die verschiedenen Distributionen unterscheiden sich in ihrem "init"-Konzept. D.h. "Normale Anwendungen" können fast immer aus einem RPM-Paket einer anderen Distribution installiert werden, für "Systemnahe Programme" müssen hingegen häufig erst Anpassungen gemacht werden.
- \* Für viele RPM-Operationen muss man "root"-Rechte haben.
- \* Unter LINUX ist nach dem Installieren im allgemeinen kein Reboot notwendig (evtl. mit "rc... reload/restart" Konfigurationsdatei neu lesen lassen).
- \* RPM sortiert mehrere zu installierende Pakete automatisch gemäß ihren Abhängigkeiten.
- \* Viele Optionen gibt es in Kurzform (-X) und in POSIX-Notation (--XXXXXX).
- \* Bei "--update" werden veränderte Konfigurations-Dateien als "\*.rpm.save" gesichert (anschließend per "diff" vergleichen und Anpassungen übernehmen).

- \* Bei allen RPM-Aufrufen kann der Paketname auch als URL angegeben werden (Standard: anonymous ftp, "ftp" kann durch "http" ersetzt werden)

```
ftp://USER:PASSWORD@HOST:PORT/PATH/T0/PACKAGE.rpm
```

Zugehörige Optionen:

```
--ftp proxy HOST
--ftp port PORT
```

- \* Der "Midnight Commander" ("mc") kann mit RPM-Archiven direkt umgehen (Verzeichnisse INFO, BUILDTIME, BUILDHOST, PACKAGER, DESCRIPTION, REQUIRENAME, PROVIDES, ...).
- \* Der "Konqueror" kann RPM-Pakete direkt installieren (Doppelklick oder Rechte Maustaste + Installieren...).
- \* Alternative: Debian Package Manager (dpkg, \*.deb), Stampede, Slackware.
- \* Das Programm "alien" kann andere Paketformate von/nach RPM konvertieren.
- \* Das Programm "rpm2cpio" erstellt aus einem RPM-Paket ein cpio-Archiv, das mit dem Befehl "cpio" entpackt werden kann.
- \* Es gibt einige grafische Frontends für RPM: "kpackage", "GnoRPM", "xrpm", SuSE-Skripte (html)
- \* Der Paketname kann eine Architektur-Information enthalten (".i386.rpm", ".noarch.rpm", ".src.rpm", ".spm").

## 8) UNIX Standard Software-Installationsmethode

Die altehrwürdige UNIX-Standard-Methode zur Installation von Anwendungen aus dem Quellcode besteht im Auspacken und Übersetzen eines sogenannten "tar-Balls" (komprimiertes tar-Archiv). Die darin befindlichen Quellcode-Dateien werden anschließend übersetzt (dazu ist ein C/C++-Compiler und weitere Hilfsprogramme der "Tool-Chain" notwendig). Die erstellten Binärprogramme + Dokus + Konfigurationsdateien werden abschließend in die typischen Verz. "sbin", "bin", "etc", usw. kopiert. Anschließend muss noch die Konfigurationsdatei angepasst werden.

```
gunzip PAKET.gz          # oder
bzunip2 PAKET.bz2       # oder
uncompress PAKET.Z

tar xzf PAKET.tar        # oder
tar xzf PAKET.tgz        # oder
tar xzf PAKET.tar.gz     # oder
tar xjf PAKET.tar.bz2   # oder
tar xZf PAKET.tar.Z

cd PAKET                 # In beim Auspacken erstelltes Verz. wechseln

more README              # oder
more INSTALL             # oder
more CONFIG              # oder
more LIESMICH

makefile anpassen        # oder
Makefile anpassen       # oder
MAKEFILE anpassen

make config              # oder
./configure              # oder
./configure --prefix=/usr/local/F00
```

```

make all          # oder
make

make doc         # Dokumentation erzeugen

make install    # (als root) oder
./setup.sh     # (als root)

make clean      # Zwischendateien aufräumen

make deinstall  # (als root) oder
make uninstall  # (als root)

```

## 9) Standard-UNIX-Programm-Pfade für Anwenderprogramme

---

Verzeichnis	Inhalt
/usr/...	Basisverz. (UNIX System Resources)
/usr/local/...	Basisverz. lokal installierte Software
.../bin	Ausführbare Programme (Benutzer)
.../sbin	Ausführbare Programme (root, System)
.../lib	Bibliotheken
.../etc	Konfigurationsdateien
.../var/lib	Skripte
.../share	Dokumentation
.../share/man	Manualeseiten (man)
.../share/doc	Dokumentation (info)
.../X11R6	X11-Anwendungen