

# Debian-Paket-Management

- am Anfang war das .tar.gz
- seit 03/1995 gibt es die Paketverwaltung mit dpkg für Debian 0.93R5 (zeitlich vor rpm)

## Stabilität und Sicherheit (Release, Archiv, Distribution)

die Release-Varianten sind auf den Servern durch „Links“ mit den jeweiligen Release-Namen verbunden

- unstable: unfertig, tägliche Änderungen, zur Entwicklung genutzt (unstable -> sid = still in development)
- testing: Sammelstelle für nächstes Release; gute Stabilität, kleinere Fehler möglich (testing -> etch)
- stable: zuletzt veröffentlichte Version; stabil, fehlerfrei, zeitnahe Sicherheitsupdates garantiert; ca. alle zwei Jahre eine neue stable-Version (stable -> sarge, Vorgänger: potato, woody)
- experimental: neuere Softwareversionen mit sehr große Änderungen (experimental -> ../project/experimental)
- frozen: temporär; eingefrorener Zustand einer testing-Distribution, kurz bevor diese zu „stable“ wird

## Verzeichnis der Releases (<http://www.de.debian.org/releases/>)

- Das nächste Release von Debian heißt »Etch« – bisher noch kein Release-Termin
- **Debian GNU/Linux 3.1 (»Sarge«)** – **aktuelles stable Release**
- Debian GNU/Linux 3.0 (»Woody«) – veraltetes stable Release
- Debian GNU/Linux 2.2 (»Potato«) – veraltetes stable Release
- Debian GNU/Linux 2.1 (»Slink«) – veraltetes stable Release

## Anordnung der Pakete (Kategorie, Sektion)

- main: nach den Regeln DFSG des Debian Projektes freie Pakete (offizielle Distribution)
- contrib: freie Pakete, die aber in Abhängigkeit von unfreien Paketen stehen
- non-free: nicht-freie Pakete nach den Debian-Regeln (zB. Netscape); zukünftig nicht mehr unterstützt
- non-US: export-beschränkte Pakete (Verschlüsselung); seit sarge-Release entfallen

## Namenskonventionen:

Programmname\_Versionsnummer-Auflage\_Architektur.deb

## Aufbau eines binären Debian-Pakets

- Ein Debian Paket besteht aus einem ar-Archiv, das zwei tar-Archive und eine Versionsdatei beinhaltet.  
**debian:~# ar -t [-v] /media/cdrom/pool/main/d/diffutils/diff\_2.8.1-11\_i386.deb**  
debian-binary enthält die Versionsnummer des Paketformats  
control.tar.gz enthält die Metainformationen über das Paket, die Scripts sowie eine Prüfsumme  
data.tar.gz enthält die zu installierenden Dateien

## /var/lib/dpkg als Systemdatenbank der installierten Pakete

- /var/lib/dpkg/available enthält die Paketinformationen aller zur Verfügung stehender (installierbarer) Pakete
- /var/lib/dpkg/status enthält die Informationen über den Status der Installation, so dass unterschieden werden kann zwischen korrekt oder nur teilweise installierten Paketen.
- Im Verzeichnis /var/lib/dpkg/info/ liegen zu jedem installierten Paket die vier Scripte (\*.preinst, \*.postinst, \*.prepm, \*.postpm), eine Liste aller enthaltenen Dateien (\*.list), die md5-Prüfsummendatei (\*.md5sums) und evt. noch andere Informationen wie die zur Verfügung gestellten Libraries (\*.shlibs).

## Zustände von Paketen

diese Stati und Flags werden üblicherweise mit dselect gesetzt:

- Status eines Paketes bezüglich des Installations-Fortschritts = PACKAGE STATES
  - installed Das Paket ist vollständig installiert
  - half-installed Die Installation wurde nicht vollständig ausgeführt
  - not-installed Das Paket ist nicht installiert
  - unpacked Das Paket ist zwar ausgepackt, aber nicht konfiguriert
  - half-configured Das Paket ist ausgepackt, die Konfiguration wurde begonnen, aber nicht korrekt abgeschlossen
  - config-files Nur die Konfigurationsdateien des Paketes existieren auf dem System (nach Deinstallation mit -R)
- aktueller Status eines Paketes = PACKAGE SELECTION STATES
  - install The package is selected for installation.
  - deinstall The package is selected for deinstallation (i.e. to remove all files, except configuration files).
  - purge The package is selected to be purged (i.e. we want to remove everything, even configuration files).
- Paket-Flags = PACKAGE FLAGS
  - hold This package is not handled by dpkg, unless forced to do that with option --force-hold.
  - reinst-required A package marked reinst-required is broken and requires reinstallation. These packages cannot be removed, unless forced with option --force-remove-reinstreq.

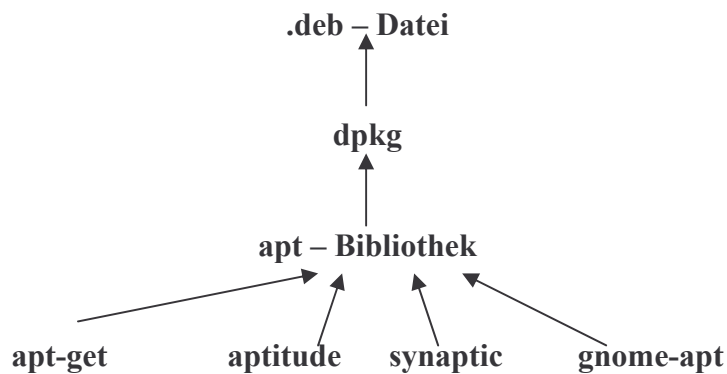
## Werkzeuge zum Paketmanagement

dpkg	low level Verwaltungswerkzeug für Softwarepakete
dselect	konsolenbasierte ncurses-Oberfläche für dpkg, sehr alt, nicht besonders intuitiv, abgelöst durch apt
apt	Fortschrittliches Paketverwaltungswerkzeug (apt-get, apt-cache, apt-listchanges, ...)
aptitude	konsolenbasierte Benutzerschnittstelle für den Paketmanager
tasksel	Frontend zur Installation von themenbezogenen Paketgruppen
gnome-apt	GTK+-basiertes grafisches Frontend für den GNOME-Desktop
synaptic	apt-nahes grafisches Frontend
kpackage	grafisches Frontend

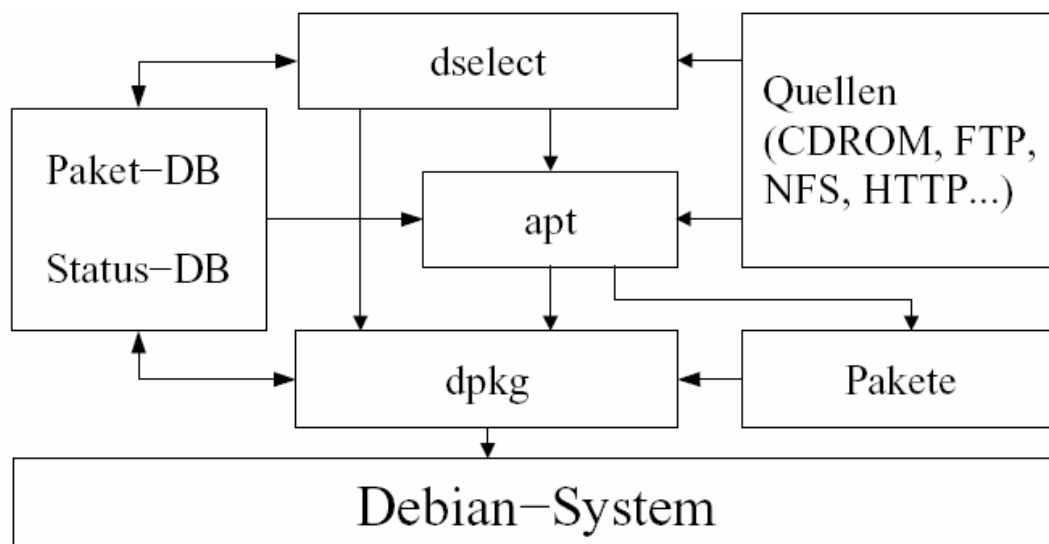
## interessante Urls

<a href="ftp://ftp.de.debian.org">ftp://ftp.de.debian.org</a>	FTP-Server von Debian
<a href="http://security.debian.org">http://security.debian.org</a>	Sicherheits-Update-Server
<a href="http://www.us.debian.org/distrib/archive">http://www.us.debian.org/distrib/archive</a>	ältere Debian-Releases (z. B. für Inst. auf älterer Hardware)
<a href="http://snapshot.debian.net">http://snapshot.debian.net</a>	Pakete auch in ehemaligen älteren Paketversionen

## die verschiedenen Ebenen der Paketinstallation:



## Zusammenhang von Back- und Frontends sowie möglichen Installationsquellen



## 1. dpkg (medium-level package manager for Debian)

- bietet wesentliche Funktionen der Paketverwaltung,
- keine Auflösung von Abhängigkeiten!!!
- keine Paketsuche zur Konfliktlösung

### Installieren eines Paketes

**dpkg -i** | **--install Paketdatei** Installation des Pakets  
**dpkg -R** | **--recursive Verzeichnis** alle Pakete dieses Verzeichnisses werden installiert

Die Installation besteht aus folgenden Einzelschritten:

- Die Kontrolldateien (Skripte) des neuen Paketes werden entpackt aus control.tar.gz.
- Wenn eine ältere Version des selben Paketes bereits installiert war, wird das PreRemove-Script der älteren Version ausgeführt.
- Das PreInst-Script des neuen Paketes wird ausgeführt.
- Die Dateien des neuen Paketes werden entpackt aus data.tar.gz und gleichzeitig werden die Dateien einer eventuell existierenden älteren Version gesichert, so dass im Falle eines Fehlers die Installation des neuen Paketes rückgängig gemacht werden kann.
- Wenn eine ältere Version existierte, dann wird das PostRemove-Script der älteren Version ausgeführt.
- Das neue Paket wird durch das Abarbeiten des PostInstall-Scripts konfiguriert.

### Deinstallieren eines Paketes

**dpkg --unpack Paketdatei** Installation der Paketdateien ohne Konfiguration  
**dpkg --configure Paketname** separate Konfiguration des Pakets  
**dpkg --configure -a** | **--pending -a** oder **--pending** anstelle des Paketnamens für alle ausgepackten Pakete  
**dpkg -i Paketdatei --force-depends** Installation ohne das Prüfen von Abhängigkeiten (gefährlich)  
**dpkg -i Paketdatei --force-overwrite** überschreibt gleichnamige Dateien anderer Pakete

**dpkg -r** | **--remove Paketname** Löschen der Paketdateien, außer den Konfigurationsdateien werden

**dpkg -P** | **--purge Paketname** entfernt alle Dateien, auch die Konfigurationsdateien

Das Entfernen der Pakete geht in folgenden Schritten vor sich:

- Das PreRemove-Script des Paketes wird abgearbeitet.
- Die zu entfernenden Dateien werden gelöscht.
- Das PostRemove-Script wird abgearbeitet.

### Abfrage von Versionsinformationen

**dpkg -p|--print-avail Paketname** Auslesen der Informationen aus der Datei /var/lib/dpkg/available

### Abfrage des Installationsstatus

**dpkg -s** | **--status Paketname** Anzeige der Stati von Paketen

Paketnamen (Package:)

Status und die Priorität des Pakets (von required über important, standard und optional bis extra)

Themenbereich (Section:)

Installationsfortschritt: installed | half-installed | not-installed | unpacked | half-configured | config-files

Angaben über Paketabhängigkeiten:

- Depends Die angegebenen Pakete müssen konfiguriert sein, damit das Paket konfiguriert werden kann.
- Pre-Depends Die angegebenen Pakete müssen fertig installiert sein, bevor überhaupt mit der Installation des Pakets begonnen werden kann. Diese Form von Abhängigkeit wird verwendet, wenn zum Beispiel die Installationsskripte des Pakets zwingend Software aus dem anderen Paket benötigt.
- Recommends Eine nicht absolute, aber sehr naheliegende Abhängigkeit. Die erwähnten Pakete würde man fast immer gemeinsam mit dem Paket installieren und nur in sehr ungewöhnlichen Umständen darauf verzichten.
- Suggests Die genannten Pakete sind im Zusammenhang mit dem aktuellen Paket nützlich, aber nicht erforderlich.
- Enhances Wie Suggests, aber andersherum – dieses Paket ist nützlich für das oder die genannten Pakete.
- Conflicts Das Paket kann nicht gleichzeitig mit den genannten Paketen installiert sein.

Die Informationen werden der Datei /var/lib/dpkg/status entnommen.

### Auflisten aller installierten Pakete

**dpkg -l** | **--list ["Paketnamensmuster"]** erzeugt Liste aller oder bestimmter installierter Pakete

**dpkg -l** | **grep '^ii'**

**dpkg -l** | **grep Paketname** Namen eines Pakets herausfinden

**COLUMNS=132 dpkg -l** | **grep Paketname** wichtige Infos (3) vorgezogen gegen Zeilenumbruch

In der ersten Spalte steht die zuletzt erwünschte Aktion; Spalte 2 enthält den aktuellen Zustand. Steht hier ii, ist die letzte Aktion deren Installation (i). Dass diese abgeschlossen wurde, markiert das weitere i der zweiten Spalte.

### Auflisten aller Dateien eines Paketes

**dpkg -L | --listfiles Paketname** welche Dateien liefert ein bestimmtes Paket

Die Dateien, die von Installationsscripts angelegt wurden, werden hier nicht angezeigt.

### Aus welchem Paket stammt eine bestimmte Datei?

**dpkg -S | --search Dateinamemuster** Durchsuchen der `/var/lib/dpkg/info`

Ausgabe aller Paketnamen, die entsprechende Dateien enthalten. Zur Musterbildung stehen alle Shell-Wildcards zur Verfügung.

### Optionen für dpkg

Vor jedem der vorgenannten Befehle können verschiedene Optionen gesetzt werden. Diese Optionen können entweder direkt an der Kommandozeile eingegeben, oder in die Datei `/etc/dpkg/dpkg.cfg` (ohne Minuszeichen) eingetragen werden. Dann werden die Optionen für jeden dpkg Befehl angewandt.

**--abort-after=Zahl** Gibt an, nach wievielen Fehlern dpkg beendet werden muß. Voreingestellt ist 50.

**-B|--auto-deconfigure** Wenn ein Paket deinstalliert wird, so ist es möglich, daß ein anderes installiertes Paket dieses Paket gebraucht hätte. Mit dieser Option werden automatisch alle anderen Pakete auch deinstalliert, die das zu installierende Paket benötigen hätten.

**--ignore-depends=Paket** Abhängigkeiten für das angegebene Paket werden ignoriert. Es werden solche Abhängigkeiten zwar überprüft, aber es werden nur Warnungen statt Fehlermeldungen angezeigt, wenn nicht erfüllte Abhängigkeiten auftreten.

**--refuse-downgrade | -G** Ein Paket, von dem eine neuere Version bereits installiert ist, wird nicht installiert

### Welche Datei gehört zu welchem Paket

Server:/dists/{Version}/Contents-{Architektur}.gz → wget ftp://ftp.de.debian.org/debian/dists/sarge/Contents-i386.gz  
zgrep {Pfad/Dateiname} Contents-{Architektur}.gz → zgrep /bin/ls /root/ Contents-i386.gz

### Paketauswahl auslesen, Änderung des Paket-Zustandes

Paket-Status auslesen und neu setzen oder Paket-Auswahl sichern, wiederherstellen, auf mehrere Systeme verteilen

<b>dpkg --get-selections</b>	Sichten der Paketstatus
<b>dpkg --get-selections [*] [Zeichenkette]   [Paketname]} &gt; selections</b>	Schreiben in eine Datei
Paketname install   deinstall   purge   hold	Ändern des Dateiinhalts
<b>dpkg --set-selections &lt; selections</b>	Einlesen in deb-Datenbank

### installierte Pakete von einem Update ausschließen

**echo <paketname> "hold" | dpkg --set-selections** im Zustand "hold" wird ein Paket nicht aktualisiert

### Verifikation von Paketen

- Im Gegensatz zu rpm erledigt dpkg diese Aufgabe nicht.
- Falls überhaupt, findet man md5-Summen im Verzeichnis `»/var/lib/dpkg/info«` unter Namen `»Paketname.md5sum«`. Die Überprüfung muss man von Hand vornehmen:

**md5sum -c /var/lib/dpkg/info/Paketname.md5sums**

- Weitere Prüfmechanismen (Rechte, Eigentümer, Modifikationszeit) kennt die Debian-Paketverwaltung nicht
- Die Integrität eines installierten Pakets kann auch mit dem Kommando `debsum` überprüft werden:

**debsum Paketname**

### Konfiguration von Paketen mit debconf

- allgemeine Syntax: `dpkg-[p]reconfigure -f{Frontend} -p{Priorität} Paketname`

**dpkg-reconfigure Paketname** erneute Abfragen zur Paketkonfiguration nach der Installation

**dpkg-preconfigure Paketname** Abfragen zur Paketkonfiguration vor der Installation

- nur dann bedeutsam, wenn ein Paket ein debconf-Script enthält (post-install).
- die Antworten werden in der debconf-Datenbank unterhalb `/var/cache/debconf/` gespeichert und bei einer Paket-Neuinstallation ausgelesen (interessant für unattended Installation)
- Konfiguration in `/etc/debconf.conf`
- Parameter für die Priorität von debconf gesetzt mit `--priority=` oder Variable `DEBIAN_PRIORITY`
  - low Sehr einfache Fragen, die mit Voreinstellungen belegt sind, die in den meisten Fällen sinnvoll sind.
  - medium Normale Fragen mit sinnvollen Vorgaben.
  - high Fragen, die keine Vorgaben haben.
  - critical Fragen, die unbedingt beantwortet werden müssen. --> z. B. **dpkg-reconfigure -plow ssh**
- Frontend gesetzt mit `--frontend=readline` oder Variable `DEBIAN_FRONTEND`
  - mögliche Werte sind `dialog`, `readline`, `noninteractive`, `gnome`, `kde`, `editor` und `web`
  - z. B. **DEBIAN\_FRONTEND=readline apt-get install ssh**

**debconf-show Paketname** Auslesen der Paket-Konfiguration aus der Datenbank

**configure-debian** zeigt alle mittels debconf konfigurierbaren Pakete an und ermöglicht die erneute Konfiguration

## Erweitertes Paket-Management mit apt (Advance Package Tool)

- steuert Paket-Manager dpkg
- nutzt die libapt-pkg fuer nicht-triviale Aufgaben, fuer die dpkg "zu dumm" ist
- verwaltet Paket-Quellen
- lädt benötigte anhand der Aktualität Pakete von der entsprechenden Quelle (cdrom, floppy, file, ftp, http)
- anhand einer Paketdatenbank wird das System in einem stabilen Zustand gehalten und Konflikte gelöst
- installiert Pakete in der richtigen Reihenfolge

### Konfiguration von apt

- Optionen aller apt-Programme in `/etc/apt/apt.conf` oder in `/etc/apt/apt.conf.d/`
- Quellen (cdrom, floppy, file, ftp, http) stehen in `/etc/apt/sources.list`

### Installieren eines Paketes

**apt-get install Paketname1 [Paket2]** Installation der ausgewählten Pakete  
**apt-get remove Paketname+** Installation der ausgewählten Pakete  
**apt-get --reinstall install Paketname** Neuinstallation eines z.B. beschädigten Programms  
**apt-get install Paketname/Distribution** gezielte Installation eines Pakets aus z. B. testing|unstable|stable  
**apt-get install package=version**

- APT durchsucht seine Datenbank nach der aktuellsten Version dieses Paketes und holt es aus dem entsprechenden Archiv, welches in der sources.list spezifiziert ist.
- Dabei werden alle in Depends:-Abhängigkeiten erwähnten Pakete ebenfalls installiert oder auf den erforderlichen Versionsstand aktualisiert, nachdem vorher die Bestätigung abgefragt wurde
- Pakete, die über das Netzwerk oder Internet heruntergeladen wurden, werden im Verzeichnis `/var/cache/apt/archives` für spätere Installationen gespeichert.
- Optionen:
  - d Nur herunterladen - Nicht installieren oder entpacken
  - f Versuche fortzufahren, auch wenn der integrity check fehlschlägt (von Hand mit **apt-get check**)
  - s Nichts wirklich tun. Simulation durchführen.
  - y Beantworte alle Fragen mit Ja anstatt sie zu stellen.
  - t Bevorzugung eines Pakets aus dem angegebenen Release testing|unstable|stable
  - u Zeige eine Liste der Pakete die geupgraded werden.

### Deinstallieren eines Paketes

**apt-get remove Paketname** Entfernen des ausgewählten Pakets ohne Konfigurationsdateien  
**apt-get install Paketname-** Entfernen des ausgewählten Pakets ohne Konfigurationsdateien  
**apt-get --purge remove Paketname** Entfernen des ausgewählten Pakets mit Konfigurationsdateien

- APT entfernt hierbei auch alle Pakete, die vom zu deinstallierenden Paket abhängen. Das lässt sich mit apt nicht umgehen, nur mit dpkg.
- Ein '\*' hinter den Namen der Pakete (in den Ausgaben bei Ausführung von apt), die vom zu entfernenden Paket abhängen, bedeutet, dass deren Konfigurationsdateien ebenso entfernt werden.

### Upgrade der installierten Pakete

**apt-get [-u] upgrade** Update der wichtigsten Pakete der gleichen Distribution auf die neueste verfügbare V.  
**apt-show-versions -u** zeigt die verfügbaren Versionen installierter Pakete, die upgradebar sind (-u)  
**apt-get install `apt-show-versions -u -b | grep unstable`** nur unstable-Pakete aktualisieren

- vor jedem Aktualisieren der Pakete sollte apt-get update ausgeführt werden
- es werden aber keine installierten Pakete entfernt und keine neuen Pakete installiert
- Oft werden deshalb einige Pakete von Upgrade zurückgehalten und nicht installiert (kept back), weil sich Abhängigkeiten geändert haben. Problem nur behebbar mit apt-get dist-upgrade.
- Optionen: -u Anzeige der kompletten Liste der Pakete, die aktualisiert werden sollen

### Upgrade auf eine neue Debian-Version (Distribution)

**apt-get dist-upgrade** Upgrade aller Pakete auf die neue Distribution  
**apt-get -o Debug::pkgProblemResolver=yes dist-upgrade** Problemsuche bei zurückgehaltenen Paketen

- erlaubt das Aktualisieren eines Debian-Systems entweder über das Internet oder von einer neuen CD
- dazu muss die neue Quelle der `/etc/apt/sources.list` hinzugefügt werden (z.B. stable gegen ein testing austauschen)  
Achtung: APT sucht sich von allen Quellen immer die neueste Version aus (also eher von ftp als CD-ROM)
- nachfolgend müssen die Paket-Listen erneuert werden mit "apt-get update"
- erst jetzt erfolgt der dist-upgrade
- hierbei werden Konflikte intelligent aufgelöst, indem geänderte Abhängigkeiten durch das gezielte Entfernen und Installieren von Paketen entsprechend ihrer Priorität erfüllt werden.

- Oft kommt es vor, dass zwar alle Pakete heruntergeladen, aber nicht korrekt installiert werden, weil voneinander abhängende Pakete noch nicht installiert sind. In dem Fall sollte man "apt-get dist-upgrade" einfach immer wieder neu starten, bis alle Pakete ordnungsgemäß installiert sind.

#### Pflege des lokalen Archivs /var/cache/apt/archives/

**apt-get clean** entfernt alles bis auf Lock-Dateien aus /var/cache/apt/archives/

**apt-get autoclean** entfernt nur Pakete, die nicht mehr heruntergeladen werden können

- in /var/cache/apt/archives/ liegen die aus dem Netzwerk geladenen Pakete, die installiert wurden.
- wird dieses Verzeichnis gecleant, muss APT ein Paket, das erneut installiert werden soll, auch erneut herunterladen.

#### Wartung eines gemischten Systems

**apt-get -t unstable install Paketname** Installation eines unstable-Pakets auf einem testing-System

- wenn nicht das ganze System auf "unstable" umgestellt werden soll, kann man "testing" und "unstable" mischen
- dazu muss die folgende Zeile in die /etc/apt/apt.conf eingefügt werden: **APT::Default-Release "testing";**
- Einfügen der Quellzeilen für die "unstable"-Distribution in die /etc/apt/sources.list und apt-get update

#### Abfrage von Paketinformationen

dazu wird der apt-Cache nach Paket-Metadaten mit dem Tool **apt-cache** durchsucht

**apt-cache stats** Statistik über den aktuellen Cache

#### Abfrage von Versionsinformationen

**apt-cache show Paketname(n)** detaillierte Anzeige der Paketinformationen, ähnlich dpkg --print-avail

- Ist das Paket bereits installiert und gibt es eine neuere Version, bekommt man Informationen über beide Versionen

#### Auflisten aller installierten Pakete mit Kurzbeschreibung

**apt-cache search regex** Auflisten aller Pakete mit folgendem Muster, durchsucht Kdo. und Metadaten

**apt-cache -full search regex** Ausgabe wie apt-cache show, zeigt durchsuchte Metadaten

**apt-cache --names-only search regex** durchsucht nur die Kommandonamen nach regex

**apt-cache search .** Listen aller installierten Pakete

#### Auflisten aller verfügbaren Pakete

**apt-cache dumpavail** erzeugt Liste aller verfügbaren Pakete

**apt-cache policy Paketname** listet die installierte Version sowie die verfügbaren Versionen

#### Welche installierten Pakete benützt ein bestimmtes Paket? (Abhängigkeiten)

**apt-cache depends** von welchen Paketen hängt dieses Paket ab

**apt-rdepends Paketname** listet rekursive Abhängigkeiten "Depends:"

**apt-cache showpkg** Welche Pakete die Installation von Paketname bedingen, beschreibt der Eintrag "Reverse Depends:", die vorausgesetzten Pakete beschreibt "Dependencies:".

- Anmerkung: Apt arbeitet analog zu Rpm's --requires, --provides und --whatrequires, nicht jedoch zu --whatprovides.

#### Auflisten aller Dateien eines installierten oder nicht installierten Paketes

**apt-file update** Pflege der Paketdateien-Datenbank für apt-file

**apt-file search Dateinamen** Suche des Paketnamens zur zugehörigen Datei; funktioniert genau wie dpkg -S, es zeigt aber auch nicht installierte Pakete, die diese Datei enthalten.

**apt-file list Paketname** den Inhalt von Paketen auflisten

#### Informationen über Änderungen in Paketen

**apt-listchanges** Anzeige der Änderungen installierter Pakete, die heruntergeladen werden, bevor Sie installiert werden

- analysiert in /usr/share/doc/Paketname die Datei changelog.Debian.gz, welche die Liste der Änderungen gegenüber der letzten Version enthält. Auch lesbar mit Hilfe von zless.

#### benötigte Pakete installieren während eines ./configure-Durchlaufs beim Kompilieren

**auto-apt run ./configure** von ./configure benötigte Pakete werden nachgeladen

- Es wird fragen, ob die benötigten Pakete installiert werden sollen, und apt-get automatisch aufrufen.
- Auto-apt funktioniert mit einer Datenbank welche gepflegt werden muss mit den Kommandos **auto-apt update**, **auto-apt updatedb** und **auto-apt update-local**.

#### Quellpakete

**apt-get source Paketname** kompiliert und installiert Quellpakete

### weitere nützliche apt-Tools

<b>apt-get moo</b>	einfach mal eintippen!
<b>apt-cacher</b>	Caching-System für Debian-Pakete als CGI in Apache
<b>apt-proxy</b>	Aufbau cachender Proxy-Server für Debian-Pakete (effizienter als Squid)
<b>apt-move</b>	Verschieben von Debian-Paketen in eine Verzeichnisstruktur wie bei einem Debian-Archiv
<b>apt-ftparchive</b>	erzeugt Index-Dateien Packages.gz analog dpkg-scanpackages
<b>apt-ftp</b>	Pakete auf Rechnern installieren ohne Internetzugang
<b>apt-listbugs</b>	vor der Paket-Installation eine Liste der für das Paket gemeldeten Fehler anzeigen

### anderes:

**localepurch**                   Deinstallieren nicht benötigter Sprachunterstützungen

**dpkg-repack**                   packt bereits installierte Pakete wieder zu einem Debian-Paket zusammen  
Wenn irgendwelche Veränderungen, beispielsweise an den Konfigurationsdateien, vorgenommen wurden, so werden diese in das Paket übernommen. Dies ermöglicht es, auf einfache Weise Pakete mit spezifischen Konfigurationen auf mehreren Systemen zu verteilen oder auch längst aus der Distribution verschwundene Pakete wiederzubeleben (solange noch auf einem anderen erreichbaren System die notwendige Software installiert ist). Auf diesem Wege kann auch der momentane Zustand eines Pakets gesichert werden, bevor Sie es upgraden.

**debian-updates**               erstellt einen Report über verfügbare Sicherheitsupdates für das System auf dem es installiert ist. Die Informationen dazu werden von einem Debian-Server bezogen auf dem alle verfügbaren Updates gelistet sind. Für jedes in der Liste aufgeführte Paket wird geprüft, ob eine veraltete Version installiert ist. Ist dies der Fall, so wird eine Information auf der Konsole ausgegeben. Alternativ können diese Informationen auch per E-Mail verschickt werden.

**deborphan**                   sucht auf einem Debian-System nach verwaisten Paketen. Wird ein Paket gefunden, von dem keine anderen Pakete abhängig sind, so wird der Name ausgegeben. Dies ist hauptsächlich sinnvoll, um installierte Bibliotheken zu finden, die nicht mehr benötigt werden.

## Konfiguration der apt-Quellen und Herunterladen der Paketlisten

### Aufbau der /etc/apt/sources.list

#### Typ URI Distribution [Kategorie] [Kategorie]

Typ	deb, deb-src
URI	Uniform Resource Identifier mit Basisverzeichnis der Distribution (file, http, ftp, ssh oder cdrom).
Distribution	Ort der Quelle relativ zum Basisverzeichnis mit mindestens einer Bezeichnung (z. B. potato, woody oder stable)
Kategorie	auch mehrere, welche Pakete der Distribution können zur Installation gewählt werden können (z. B. contrib, non-free oder non-US).

### Einrichten der Sourcen für apt - Erstellen der Datei /etc/apt/sources.list

- **apt-setup** fragt Quellen (cdrom, http, ftp und filesystem), Kategorie, Länder und Debian-Mirrors ab
- **apt-cdrom ident|add**
- **vi**

### Beispieleinträge fuer /etc/apt/sources.list:

```
# See sources.list(5) for more information
##### Debian Sarge (stable):
# vorkompilierte Binärpakete (*.deb)
deb ftp://ftp.de.debian.org/debian sarge main contrib non-free
deb ftp://ftp.de.debian.org/debian-non-US sarge/non-US main contrib non-free
# Quellpakete (*.deb-src), mit originalen Programmquellen, der Debian-Kontrolldatei .dsc und diff.gz
deb-src ftp://ftp.de.debian.org/debian sarge main contrib non-free
deb-src ftp://ftp.de.debian.org/debian-non-US sarge/non-US main contrib non-free
##### Debian Sid:
#deb ftp://ftp.de.debian.org/debian sid main contrib non-free
#deb ftp://ftp.de.debian.org/debian-non-US sid/non-US main contrib non-free
#deb-src ftp://ftp.de.debian.org/debian sid main contrib non-free
#deb-src ftp://ftp.de.debian.org/debian-non-US sid/non-US main contrib non-free
##### Security Updates:
deb http://security.debian.org/ stable/updates main contrib non-free
##### Java 1.3
deb ftp://ftp.informatik.hu-berlin.de/pub/Java/Linux/debian woody non-free
##### Eigenes Verzeichnis:
deb file:/verzeichnis/ ./
```

### Auswahl eines Debian-Mirrors

- Die Liste der Debian-Mirrors gibt es auf <http://www.debian.org/mirror/list>
- Auswahl des performantesten Servers aus einer mitgegebenen Liste mit:  
**debian:~# netselect ftp.debian.org ftp.de.debian.org http.us.debian.org ftp.at.debian.org ftp.ch.debian.org**
- Auswahl des performantesten Servers aus der Debian-Liste aller Mirrors und Erzeugen einer sources.list im Arbeitsverzeichnis:  
**debian:~# netselect-apt stable|testing|unstable|experimental|woody|sarge|etch|sid # Default ist stable**

### Deb-Pakete die sich in einem lokalen Verzeichnis befinden in die /etc/apt/sources.list mit einbinden

Dazu muss in dem Verzeichnis in dem sich die Pakete befinden die Datei mit den Paket-Informationen erstellt werden:

```
cd /verzeichnis/mit/deb/paketen/
```

```
dpkg-scanpackages ./ /dev/null | gzip > Packages.gz
```

/dev/null bedeutet, dass keine override-Datei zum Überschreiben der Paketeigenen Optionen verwendet wird

Nun kann das Verzeichnis mit folgendem Eintrag in die Datei /etc/apt/sources.list aufgenommen werden:

```
deb file:/verzeichnis/mit/deb/paketen/ ./
```

das ./ bezeichnet den Ort der zugehörigen Packages.gz mit den Paketinformationen für apt

### Aktualisieren der Paket-Datenbank

**apt-get update** lädt die Paketlisten der eingetragenen Quellen herunter (Scan der Packages.gz-Dateien der Quellen) und erzeugt die lokalen apt-Caches

- sollte regelmäßig ausgeführt werden, um das System auf dem neusten Stand über mögliche Paket- bzw. Sicherheitsupdates zu halten.
- erzeugt z.B. /var/lib/apt/lists/ftp.de.debian.org\_debian\_dists\_sarge\_main\_binary-i386\_Release  
/var/lib/apt/lists/ftp.de.debian.org\_debian\_dists\_sarge\_main\_binary-i386\_Packages

**dpkg-query** Abfrage der Paketdatenbank des Systems; berücksichtigt alle Pakete aus der Datei /var/lib/dpkg/available (installierte und nicht-installierte Pakete).



## Paketmanagement mit dem Frontend dselect

- menügeführtes Frontend für dpkg, aber auch für apt einstellbar
- bis zur Version 2.2 von Debian das wichtigste Werkzeug
- installiert und deinstalliert Pakete, Verwaltung der verfügbaren Pakete und der Abhängigkeiten
- nicht besonders intuitiv, nur noch für Debian-Urgesteine

### mögliche Paketquellen (Zugriffsmethoden):

- cdrom Die CDROM kann, muss aber nicht gemountet sein und
- nfs Installiert über einen (noch nicht gemounteten) NFS-Server. Der Server muss die Paketbeschreibungsdateien (Packages.gz) jeder Distributionsabteilung (stable, contrib und non-free) zur Verfügung stellen sowie die entsprechenden .deb Dateien.
- harddisk Installiert von einer noch nicht gemounteten Festplattenpartition. Inhalte wie bei nfs
- mounted Installation von einem bereits gemounteten Dateisystem.
- floppy Installation über einen Stapel Disketten, von dem die erste Disk die Paketinformationen enthält.
- apt Benutzt die neue, empfohlene Methode via /etc/apt/sources.list

### angebotene Menüpunkte:

die Menüpunkte 1 bis 3 (nicht 0) sollten beim ersten Aufruf in genau dieser Reihenfolge aufgerufen werden:

0. Zugriff (access) Auswahl der Zugriffsmethode, bevorzugt apt (siehe oben)  
weiter mit ENTER
1. Erneuern (update) Erneuert die Liste der verfügbaren Pakete, wenn möglich.
2. Auswählen (select) Menügeführte Auswahl aller zu installierender oder zu entfernender Pakete.  
Hilfe mit SPACE verlassen  
Hilfemenü mit ?  
Auswahl von Gruppen oder einzelnen Paketen mit +  
Abwahl mit -  
Vorschlag von dselect rückgängig machen mit R  
Sortierung beeinflussen mit o  
Ausblenden der Paketinformationen mit I  
Suchen mit /string, Weitersuchen mit \  
Beenden mit ENTER
3. Install Installiert die Pakete, die ausgewählt wurden.
4. Konfig Konfiguriert Pakete, die bei der letzten Installation nicht vollständig konfiguriert wurden. (entspricht dget --pending --configure)
5. Löschen (remove) Entfernt die zum Löschen markierten Pakete.

## Paketmanagement mit dem Frontend aptitude

- ncurses-Frontend für apt
- unterscheidet streng nach installierten, nicht installierten, virtuellen Paketen und Paketen in einer neueren Version
- die Baumstruktur entspricht der Verzeichnisstruktur innerhalb des Debian-Archives

### Tastenbelegung

PFEIL-UNTEN	Bewegt den Auswahlbalken zum nächsten Eintrag.
PFEIL-OBEN	Bewegt den Auswahlbalken zum vorherigen Eintrag.
RETURN	Klappt ein Verzeichnis auf/zu.
^	Springt zum Verzeichnis, zu dem das Paket gehört.
+	Markiert ein Paket zur Installation.
-	Markiert ein Paket zum Löschen.
i	Zeigt die Beschreibung des Pakets an.
d	Zeigt die Abhängigkeiten des Pakets an.
v	Zeigt die verfügbaren Versionen des Pakets an.
u	Aktualisiert die Liste der verfügbaren Pakete.
g	Startet die Installation der ausgewählten Pakete.

### Hintergrundfarben

schwarz	Normalzustand eines Pakets. Beim nächsten Installationsdurchlauf wird dieses Paket nicht verändert.
Fett	Pakete sind bereits installiert.
rot	Das Paket ist in einem unbrauchbaren Zustand oder kann nicht installiert werden.
blau	Das Paket wird mit einer neueren Programmversion aktualisiert.
weiß	Dieses Paket könnte aktualisiert werden, es wurde aber auf dem aktuellen Stand fixiert (hold).
grün	Das Paket wird installiert.
magenta	Das Paket wird gelöscht.

### Aktionen auf der Kommandozeile

**aptitude [Optionen] <Aktion> ...** → ohne Aktion Start im interaktiven Modus

install Paket	Installiere/aktualisiere Pakete
remove Paket	Entferne Pakete
purge Paket	Entferne Pakete und ihre Konfiguration
hold Paket	Pakete auf den Status „halten“ setzen
unhold Paket	Entfernt den Status „halten“ des angegebenen Paketes
markauto Paket	Pakete als automatisch installiert markieren
unmarkauto Pak.	Pakete als manuell installiert markieren
update	Neue Listen neuer/aktualisierbarer Paketen laden
upgrade	Sicheres Upgrade durchführen
dist-upgrade	Upgrade durchführen, dabei wenn notwendig Pakete installieren/entfernen
search	Pakete nach Namen oder Ausdruck suchen
show	Gibt detaillierte Informationen zu dem Paket aus
clean	Heruntergeladene Pakete löschen
autoclean	Alte heruntergeladene Pakete löschen
download	Die .deb-Datei eines Pakets herunterladen

### Optionen beim Aufruf auf der Kommandozeile

-h	Eine kurze Hilfe zu aptitude
-s	Aktionen simulieren, aber nicht wirklich durchführen
-d	Pakete nur herunterladen, nichts installieren oder entfernen
-P	Zur Bestätigung nachfragen
-y	Annehmen, dass die Antwort zu einfachen Ja/nein-Fragen „ja“ ist
-F format	Format für die Suchergebnisse angeben
-O order	Wie die Suchergebnisse sortiert werden sollen
-w breite	Die Breite für die Formatierung der Suchergebnisse
-f	Aggressiv versuchen, kaputte Pakete trotzdem zu installieren
-V	Anzeigen, welche Version eines Pakets installiert wird
-D	Abhängigkeiten automatisch veränderter Pakete anzeigen
-Z	Größenveränderung der einzelnen Pakete anzeigen
-t release	Release, aus dem Pakete installiert werden sollen
--with(out)-recommends	Beeinflusst, ob oder ob nicht Empfehlungen wie starke Abhängigkeiten gehandhabt werden
-S fname	die erweiterten Aptitude-Statusinformationen aus der Datei fname laden
-u	Neue Paketlisten beim Start laden
-i	Installationslauf am Anfang durchführen

## Paketkonvertierung mit alien

alien konvertiert verschiedene Paketformate in jeweils andere um. Das Programm kann mit folgenden Formaten umgehen:

- \* RedHat (rpm)
- \* Debian (deb)
- \* Stampede (slp)
- \* Slackware (tgz)
- \* Solaris (pkg)

alien erkennt das Format einer angegebenen Paketdatei automatisch und konvertiert es dann in das angegebene Format. Das gewünschte Ausgabeformat wird durch die folgenden Parameter angegeben:

--to-deb	Aus dem angegebenen Paket wird ein Debian-Paket erstellt
--to-rpm	Aus dem angegebenen Paket wird ein RedHat-Paket erstellt
--to-tgz	Aus dem angegebenen Paket wird ein Slackware-Paket erstellt
--to-slp	Aus dem angegebenen Paket wird ein Stampede-Paket erstellt

Wenn also das Paket foo-1.2.34.i386.rpm existiert, und daraus ein Debian-Paket erstellt werden soll, so genügt der Befehl

```
alien --to-deb foo-1.2.34.i386.rpm
```

und alien erzeugt daraus das Paket foo\_1.2.34\_i386.deb

Jetzt kann dieses neu erstellte Paket mit dpkg -i installiert werden und wird so nahtlos in die Debian-Paketverwaltung integriert.

## rpm und dpkg im Vergleich

RPM	DEB	Aktion
rpm -i <i>name</i>	dpkg -i <i>name</i>	Installiert das Paket <i>name</i>
rpm -i --	nodepsnamedpkg --install --force-depends <i>name</i>	Installiert das Paket <i>name</i> ohne Abhängigkeiten zu berücksichtigen
rpm -i --replacefiles <i>name</i>	dpkg --install --force-overwrite <i>name</i>	Installiert das Paket <i>name</i> und überschreibt bereits vorhandene Dateien
rpm -Va	debsums -a	Überprüft die Checksummen aller installierten Dateien auf dem System
rpm -qf /etc/syslog.conf	dpkg -S /etc/syslog.conf	Zeigt an zu welchem Paket die Datei /etc/syslog.conf gehört
rpm -ql <i>name</i>	dpkg -L <i>name</i>	Listet alle Dateien des Paketes <i>name</i> auf
rpm -qpi <i>name.rpm</i>	dpkg -I <i>name.deb</i>	Zeigt die Informationen zum Paketarchiv <i>name.xxx</i>
rpm -qpl <i>name.rpm</i>	dpkg -c <i>name.deb</i>	Listet alle Dateien des Paketarchives <i>name.xxx</i> auf
rpm -qia	dpkg -l	Listet alle Pakete mit Informationen zu den Paketen. Das rpm-Kommando zeigt alle Paketinformationen, die dpkg-Variante nur die Kurzinformationen. Kurzinformationen können mit rpm über das Kommando rpm -qa --qf '%-20{NAME} %-10{VERSION} %{SUMMARY}\n' ausgegeben werden.
rpm: Funktion nicht implementiert	dpkg -r <i>name</i>	Entfernt das Paket <i>name</i> , Konfigurationsdateien werden nicht gelöscht
rpm -e <i>name</i>	dpkg --purge <i>name</i>	Entfernt das Paket <i>name</i> inklusive aller Konfigurationsdateien
rpm -qi <i>name</i>	dpkg -s <i>name</i>	Zeigt die Paketinformationen (Name, Status, Version, Abhängigkeiten usw.) zu dem Paket <i>name</i> an.
rpm -q --qf '%-20{NAME} %-10{VERSION} %{SUMMARY}\n' <i>name</i>	dpkg --list <i>name</i>	Zeigt eine Kurzbeschreibung zu dem Paket <i>name</i> an.
rpm -qa --qf '['%{=NAME}:%{FILENAMES}\n]' <i>string</i>	dpkg -S <i>string</i>	Sucht nach dem String <i>string</i> in den Paketinformationen/Dateilisten aller installierten Pakete
rpm -q --whatrequires <i>name</i>	nicht implementiert	Welche Pakete hängen von dem Paket <i>string</i> ab.
rpm -q -requires <i>name</i>	nicht implementiert	Welche Abhängigkeiten hat das Paket <i>string</i> .
rpm -q --scripts <i>name</i>	nicht implementiert	Zeigt die Scripte des Paketes <i>string</i> an.