

Debian-Schnellreferenz

Osamu Aoki <osamu\#at\#debian.org>
Übersetzer: Jens Seidel <tux-master\#at\#web.de>
'Autoren' auf Seite **27**

CVS, Don 18. Jan 2007, 11:54:34 UTC

Zusammenfassung

Diese Debian-Schnellreferenz (<http://qref.sourceforge.net/>) ist als kurze Einführung in das Debian-System gedacht, als eine **Schnellreferenz**. Dies ist ein Auszug der Debian-Referenz (<http://qref.sourceforge.net/>).

Copyright-Hinweis

Copyright © 2001–2005 by Osamu Aoki <osamu#@#debian.org>.

Dieses Dokument kann unter den Bedingungen der GNU General Public License Version 2 oder höher genutzt werden. (<http://www.gnu.org/copyleft/gpl.html>)

Es ist erlaubt, unveränderte Kopien dieses Dokuments zu erstellen und zu vertreiben, vorausgesetzt der Copyright-Hinweis und diese Genehmigung bleiben auf allen Kopien erhalten.

Es ist erlaubt, veränderte Kopien dieses Dokuments unter den Voraussetzungen für unverändertes Kopieren, zu erstellen und zu vertreiben, sofern die gesamte resultierende Arbeit unter den Bedingungen einer Genehmigung identisch zu dieser, vertrieben wird.

Es ist erlaubt, Übersetzungen dieses Dokuments in eine andere Sprache, unter den obigen Bedingungen für veränderte Versionen zu kopieren und zu verteilen, mit der Ausnahme, dass diese Genehmigung übersetzt, statt im ursprünglichem Englisch, eingebunden werden kann, sofern diese Übersetzung (des Copyrights) von der Free Software Foundation genehmigt ist.

Inhaltsverzeichnis

1	Vorwort	1
1.1	Konventionen in diesem Dokument	1
1.2	Grundlagen der Debian-Distributionen	1
2	Aktualisieren einer Distribution auf <i>stable</i>, <i>testing</i> oder <i>unstable</i>	3
2.1	Aktualisierung von Potato auf Woody	3
2.2	Vorbereitung einer Aktualisierung	3
2.3	Aktualisierung	4
2.3.1	Verwendung von <code>dselect</code>	4
3	Debian-Paketverwaltung	7
3.1	Einführung	7
3.1.1	Grundlegende Werkzeuge	7
3.1.2	Nützliche Tools	8
3.2	Grundlagen der Debian-Paketverwaltung	8
3.2.1	<i>Aufgaben-Profil</i> mit <code>tasksel</code> oder <code>aptitude</code> installieren	8
3.2.2	Einrichten des APT-Systems	9
3.2.3	<code>dselect</code>	9
3.2.4	<code>aptitude</code>	10
3.2.5	Die <code>apt-cache-</code> und <code>apt-get-</code> Kommandos	11
3.2.6	Verwenden einer speziellen Version der Debian-Distribution	12
3.2.7	Downgrad aller Pakete zu <i>stable</i>	13
3.2.8	Überblick über <code>/etc/apt/preferences</code>	13
3.3	Debian-Überlebensbefehle	14

3.3.1	Überprüfen von Bugs in Debian und Suche nach Hilfe	14
3.3.2	APT-Upgrade-Fehlersuche	14
3.3.3	Wiederherstellung mit <code>dpkg</code>	15
3.3.4	Wiederherstellung von Paketauswahldaten	16
3.3.5	Systemrettung nach Löschen von <code>/var</code>	16
3.3.6	Installation eines Pakets in ein nicht bootfähiges System	17
3.3.7	Der <code>dpkg</code> -Befehl funktioniert nicht mehr – was nun?	17
3.4	Debian-Nirwana-Befehle	18
3.4.1	Informationen über eine Datei	18
3.4.2	Informationen über ein Paket	19
3.4.3	Nicht-Interaktive Installation mit APT	19
3.4.4	Installierte Pakete erneut konfigurieren	20
3.4.5	Deinstallation von Paketen	20
3.4.6	“Halten“ älterer Pakete	21
3.4.7	Gemischtes <code>stable/testing/unstable</code> -System	21
3.4.8	Löschen von zwischengespeicherten Paketdateien	22
3.4.9	Speichern/Kopieren der Systemkonfiguration	22
3.4.10	Portierung eines Pakets auf die <code>stable</code> -Distribution	22
3.4.11	Lokale Paketarchive	23
3.4.12	Installation und Konvertierung eines fremden Binärpakets	24
3.4.13	Verifizierung installierter Paketdateien	24
3.5	Weitere Debian-Besonderheiten	25
3.5.1	Der <code>dpkg-divert</code> -Befehl	25
3.5.2	Das <code>equivs</code> -Paket	25
3.5.3	Alternative Befehle	25
3.5.4	<code>init</code> und Runlevel	26
3.5.5	Deaktivierung von Daemon-Diensten	26
A	Anhang	27
A.1	Autoren	27
A.2	Haftung	29
A.3	Rückmeldungen	29

Kapitel 1

Vorwort

Dieses Dokument war ursprünglich eine "Schnellreferenz" wuchs jedoch schnell. Dennoch ist **Halte es kurz und einfach** mein Leitmotiv.

1.1 Konventionen in diesem Dokument

Diese Debian-Schnellreferenz enthält Informationen in Form von kurzen `bash`-Shell-Kommandos.

Verweise auf:

- eine UNIX-artige **Handbuchseite** erfolgen in der Form: `bash(1)`.
- eine GNU-TEXINFO-Seite erfolgen in der Form: `info libc`.

1.2 Grundlagen der Debian-Distributionen

Debian betreut gleichzeitig drei verschiedene Distributionen. Diese sind:

- `stable` – Am nützlichsten für einen produktiven Server, da nur Sicherheitsaktualisierungen eingespielt werden.
- `testing` – Die bevorzugte Distribution für eine Workstation, da sie aktuelle Ausgaben von Desktop-Software enthält, die schon ein wenig getestet wurden.
- `unstable` – Topaktuell. Die Wahl der Debian-Entwickler.

Wenn Pakete in `unstable` keine veröffentlichungskritischen Fehler innerhalb der ersten Woche oder so haben, gelangen sie automatisch nach `testing`.

Debian-Distributionen haben auch Kodennamen. Bevor Woody im August 2002 veröffentlicht wurde, entsprachen den drei Distributionen Potato, Woody und Sid. Nachdem Woody veröffentlicht wurde, entsprachen sie den drei Distributionen Woody, Sarge und Sid. Wenn Sarge

veröffentlicht wurde, werden die `stable`- und `unstable`-Distributionen Sarge und Sid sein; eine neue `testing`-Distribution wird dann erzeugt (zunächst als Kopie von `stable`) und ihr ein neuer Kodename zugeordnet.

Abonnieren Sie die Mailingliste `debian-devel-announce@lists.debian.org` (geringes Mail-Aufkommen) für wichtige Ankündigungen über Debian.

Möchte man Versionen von Paketen verwenden, die aktueller sind als die Versionen der Distribution die man nutzt, kann man entweder auf eine neuere Distribution, wie in 'Aktualisieren einer Distribution auf `stable`, `testing` oder `unstable`' auf der nächsten Seite beschreiben, aktualisieren, oder nur die ausgesuchten Pakete aktualisieren. Wenn das Paket nicht einfach aktualisiert werden kann, kann man es wie in 'Portierung eines Pakets auf die `stable`-Distribution' auf Seite [22](#) erläutert, zurückportieren.

Kapitel 2

Aktualisieren einer Distribution auf `stable`, `testing` oder `unstable`

2.1 Aktualisierung von Potato auf Woody

Dies wird separat beschrieben, da Potato's APT nicht über alle in der aktuellen `apt_preferences(5)`-Handbuchseite beschriebenen Fähigkeiten verfügt.

Nach dem Einbinden von Woody-Verweisen (und keinen anderen) zu `/etc/apt/sources.list` werden APT und benötigte Kernpakete auf Woody-Versionen wie folgt aktualisiert:

```
# apt-get update
# apt-get install libc6 perl libdb2 debconf
# apt-get install apt apt-utils dselect dpkg
```

Danach wird der Rest des Systems auf Woody aktualisiert:

```
# apt-get upgrade
# apt-get dist-upgrade
```

2.2 Vorbereitung einer Aktualisierung

Die Aktualisierung einer Distribution auf eine andere kann durch Herunterladen der Pakete aus dem Netzwerk wie folgt durchgeführt werden.

Erzeugen einer sauberen Liste von Paket-Depots für `stable`:

```
# cd /etc/apt
# cp -f sources.list sources.list.old
```

```
# :>sources.list
# apt-setup noprobe
```

Soll auf `testing` aktualisiert werden, so sind `testing`-Quellen zu dieser neuen Liste hinzuzufügen. Soll auf `unstable` aktualisiert werden, so sind auch `unstable`-Quellen hinzuzufügen.

```
# cd /etc/apt
# grep -e "^deb " sources.list >srcs
# :>sources.list
# cp -f srcs sources.list
# sed -e "s/stable/testing/" srcs >>sources.list
# sed -e "s/stable/unstable/" srcs >>sources.list
# apt-get update
# apt-get install apt apt-utils
```

Vergleiche ‘Grundlagen der Debian-Paketverwaltung’ auf Seite 8 für die Kunst des Tunings von `/etc/apt/sources.list` und `/etc/apt/preferences`.

2.3 Aktualisierung

Nach den korrekten Anpassungen von `/etc/apt/sources.list` und `/etc/apt/preferences`, wie oben beschrieben, kann mit der Aktualisierung begonnen werden.

Das Verwenden der `testing`-Distribution von Debian kann den Nebeneffekt haben, dass Sicherheitsupdates nur langsam eingespielt werden, da solche Pakete nach `unstable` hochgeladen werden und erst später nach `testing` gelangen.

Man vergleiche mit ‘Debian-Paketverwaltung’ auf Seite 7 für die Grundlagen und sollten dabei Fehler auftreten mit ‘APT-Upgrade-Fehlersuche’ auf Seite 14.

2.3.1 Verwendung von `dselect`

Wenn ein System zahlreiche Pakete enthält, die `-dev`-Pakete, etc. nutzen, wird die im Folgenden beschriebene Vorgehensweise mittels `dselect` empfohlen, da hier eine feinere Kontrolle der Pakete möglich ist.

```
# dselect update # muss vor jeder Aktualisierung ausgeführt werden
# dselect select # Auswahl zusätzlicher Pakete
```

All Ihre aktuellen Pakete werden ausgewählt, wenn `dselect` startet. `dselect` verlangt dabei eventuell nach zusätzlichen Paketen, basierend auf Abhängigkeiten, Vorschlägen und Empfehlungen. Möchten Sie keine Pakete hinzufügen, drücken Sie einfach `Q`, um `dselect` erneut zu beenden.


```
# dselect install
```

Sie werden einige Paket-Konfigurationsfragen während dieses Teils des Prozesses beantworten müssen, halten Sie also Ihre Notizen bereit und planen Sie genug Zeit dafür ein. Vergleichen Sie 'dselect' auf Seite 9.

Nutzen Sie `dselect`. **Dies funktioniert immer :)**

Kapitel 3

Debian-Paketverwaltung

Dieses Kapitel basiert auf älterer Version des englischen ursprünglichen Textes. Die englische Version bitte auch überprüfen.

Um die Netzlast in den Debian-Archiven zu reduzieren, sollten Sie einen lokalen HTTP-Proxy mit `squid` für das Herunterladen von Paketen mittels APT einrichten. Wenn nötig muss die `http_proxy`-Umgebungsvariable oder der `http`-Wert in `/etc/apt/apt.conf` gesetzt werden. Dies erhöht die Leistung im Netzwerk enorm, insbesondere dann, wenn viele Debian-Rechner in einem LAN vorhanden sind.

Obwohl die Pinning-Eigenschaft von `apt_preferences(5)` sehr leistungsfähig ist, löst sie nicht alle Abhängigkeitsprobleme, da die Voraussetzungen für Abhängigkeiten an neueren Versionen anderer fundamentaler Programmpakete zerren.

Die Verwendung der `chroot` Methode ist geeignet, um gleichzeitig die Stabilität beider Systeme zu sichern und auf die neuesten Versionen von Software zuzugreifen.

Dieses Kapitel bezieht sich auf ein Woody-System, jedoch sollten die Informationen hier auch auf ein Potato-System (mit Ausnahme von `apt_preferences(5)` und Themen die Bezug auf die `/etc/apt/preferences`-Datei nehmen) zutreffen.

3.1 Einführung

Wenn das Lesen der Entwicklerdokumentation zu viel für Sie ist, dann lesen Sie zuerst dieses Kapitel um dann die ganze Vielfalt von Debian (`testing/unstable`) zu nutzen :-)

3.1.1 Grundlegende Werkzeuge

<code>dselect</code>	- menügesteuertes Paketverwaltungs-Werkzeug (top level)
<code>dpkg</code>	- installiert Pakete (Paketdatei orientiert)
<code>apt-get</code>	- installiert Pakete (Paketarchiv orientiert, CLI-APT)
<code>tasksel</code>	- installiert Aufgaben-Profile (Verbund von Paketen)

```
aptitude - installiert Pakete (Pakete & Profile, ncurses APT)
deity    - alternatives ncurses APT
synaptic, gsynaptic - GUI-APT-Alternativen
```

Diese Tools arbeiten nicht alle auf dem gleichen Level. `dselect` arbeitet ein Level über APT (der Kommandozeilenbefehl ist `apt-get`) und `dpkg`.

APT nutzt `/var/lib/apt/lists/*` um verfügbare Pakete zu erfassen, `dpkg` jedoch nutzt `/var/lib/dpkg/available`. Wenn Sie Pakete direkt mit `apt-get` oder mit einem ähnlichen Programm, wie z.B. `aptitude` installiert haben, dann stellen Sie sicher, dass Sie die Datei `/var/lib/dpkg/available` mittels `[U]pdate` aus dem `dselect`-Menü oder von der Kommandozeile mit `"dselect update"` updaten, bevor Sie `dselect select, tasksel` oder `dpkg -l` ausführen.

Was die Paketabhängigkeiten angeht, so berücksichtigt `apt-get` automatisch alle **benötigten** Pakete, ignoriert aber **empfohlende** und **vorgeschlagene** Pakete. `dselect` bietet jedoch eine menübasierte Kontrolle über die Auswahl von solchen Paketen und fragt zur Auswahl von **benötigten, empfehlenden** und **vorgeschlagenen** Paketen. `aptitude` bietet die Möglichkeit, alle **benötigten, empfehlenden** und **vorgeschlagenen** Pakete automatisch zu berücksichtigen.

3.1.2 Nützliche Tools

```
apt-cache          - durchsucht das Paketarchiv im lokalen Cache
dpkg-reconfigure  - erneutes Konfigurieren von installierten
                   Paketen (sofern sie debconf benutzen)
dpkg-source       - verwaltet Quellpaketdateien
dpkg-buildpackage - automatisiert den Prozess der Erstellung von
                   Paketdateien
...
```

3.2 Grundlagen der Debian-Paketverwaltung

Man kann eine Menge von Paketen genannt *Aufgaben-Profil* oder auch *Task* installieren oder individuelle Pakete. Eine Aktualisierung des Systems kann auch mittels Paketverwaltungswerkzeugen wie weiter unten beschrieben erfolgen.

3.2.1 Aufgaben-Profil mit `tasksel` oder `aptitude` installieren

`tasksel` ist das **Debian-Aufgaben-Profil-Installationsprogramm**, das auch während der Debian-Installation unter dem Punkt "einfach" angeboten wird.

Wenn man eine bestimmte Funktion benötigt, die verschiedene Pakete erfordert, so ist das der einfachste Weg dies zu erledigen. Stellen Sie sicher, dass Sie die Befehle wie folgt starten:

```
# dselect update
# tasksel
```

`aptitude` bietet auch Zugang zu *Aufgaben-Profilen*. Dies erlaubt nicht nur die Auswahl von *Aufgaben-Profilen*, sondern ermöglicht es auch, Pakete in diesem *Profil* selektiv durch ein Menü abzuwählen.

3.2.2 Einrichten des APT-Systems

Versuchen Sie ein System mittels verschiedener Distributionen, wie im Folgenden beschrieben, einzurichten, so ist es wahrscheinlich, dass Sie auf einige Paketabhängigkeitskonflikte stoßen. Es ist eine gute Idee, verschiedene Ausgaben nicht zu vermengen. Das Folgende ist für Personen, die gerne experimentieren und dabei Risiken in Kauf nehmen.

Um selektive Upgrades durchzuführen, während man die `testing`-Distribution installiert hat, muss das APT-System (>Woody) wie in 'Aktualisierung von Potato auf Woody' auf Seite 3 beschrieben eingerichtet werden, damit die `apt_preferences(5)` Eigenschaften genutzt werden können.

Fügen Sie zuerst die Quellen für `stable`, `testing` und `unstable` in Ihre `/etc/apt/sources.list`-Datei ein. Editieren Sie danach `/etc/apt/preferences`, um `Pin-Priority` korrekt einzurichten.¹

```
Package: *
Pin: release a=stable
Pin-Priority: 500

Package: *
Pin: release a=testing
Pin-Priority: 600

Package: *
Pin: release a=unstable
Pin-Priority: 50
```

3.2.3 dselect

Wenn `dselect` startet, wählt es automatisch alle Pakete aus, die in den Kategorien "Required", "Important" und "Standard" enthalten sind. In der Potato-Distribution waren etliche große Pakete wie z.B. `teTeX` und `Emacs` in diesen Kategorien enthalten und wurden am besten

¹Ich behielt die Schilderung hier bei wie sie ist, um Konsistenz mit anderen Teilen zu gewährleisten. Wenn Sie wirklich `testing` oder `unstable` verwenden möchten, sollten Sie Verweise auf `stable` in `/etc/apt/sources.list` und `/etc/apt/preferences` entfernen. Dies ist empfehlenswert, da `testing` als eine Kopie von `stable` startet.

für die Erstinstallation manuell abgewählt (mittels '_'). In der Woody-Distribution sind diese Pakete nun in der "Optional"-Paketkategorie.

dselect hat eine einigermaßen gewöhnungsbedürftige Benutzerschnittstelle. Es gibt gleich vier doppeldeutige Befehle (Großbuchstaben sind gemeint!):

Taste	Aktion
Q	Beenden. Momentane Auswahl bestätigen und beenden. (Abhängigkeiten vernachlässigen)
R	Rückgängig! Es war nicht so gemeint.
D	Verdammt! Es ist uninteressant was dselect meint. Mach es!
U	Alles auf den vorgeschlagenen Status setzen.

Mit D und Q kann eine Auswahl die Konflikte enthält, auf eigenes Risiko gewählt werden. Hiermit sollte vorsichtig umgegangen werden. Zurzeit ist dselect das ausgereifteste menü-basierte Werkzeug, das fein gegliederte Kontrolle über **Empfehlungen** und **Vorschläge** bietet.

Fügen Sie eine Zeile in `/etc/dpkg/dselect.cfg` ein, die die Option "expert" enthält, um die Ausgabe unnötiger Meldungen zu reduzieren. Für langsame Rechner, ist es empfehlenswert dselect auf einem anderen, schnelleren Rechner zu starten um Pakete zu finden und diese mit `apt-get install` zu installieren.

dselect erlaubt keinen Zugriff auf Pakete, die nicht durch Pin-Priority begünstigt sind.

3.2.4 aptitude

aptitude ist ein menübasiertes Paketinstallationprogramm ähnlich zu dselect. Dies kann auch als alternativer Kommandozeilen-Befehl an Stelle von `apt-get` genutzt werden. Vergleichen Sie `aptitude(1)`.

aptitude akzeptiert folgende Tastenkürzel, die meist Kleinbuchstaben sind.

Tastenkürzel	Aktion
F10	Menü
?	Hilfe zu Tastenkürzel
u	Aktualisiere Paketarchivinformation
g	Herunterladen und installieren gewählter Pakete
q	Beenden und Sichern der Änderungen
x	Beenden und Verwerfen der Änderungen
Enter	Informationen zu einem Paket anzeigen

aptitude erlaubt es, alle Pakete, je nachdem, ob sie **benötigt**, **empfohlen** oder **vorgeschlagen** werden, auszuwählen. Dieses Verhalten kann durch Auswahl von F10 -> Optionen -> Abhängigkeitsbehandlung im Menü geändert werden.

aptitude erlaubt den Zugriff auf alle Versionen eines Pakets.

3.2.5 Die `apt-cache`- und `apt-get`-Kommandos

Beim Einrichten von `testing` wie im obigen Beispiel beschrieben, kann das System durch folgende Kommandos angepasst werden:

- `apt-get -u upgrade`
Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System, wobei die durch Abhängigkeiten benötigten Pakete von `testing` stammen.
- `apt-get -u dist-upgrade`
Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System, wobei alle Abhängigkeiten aufgelöst werden und die dadurch benötigten Pakete von `testing` stammen.
- `apt-get -u dselect-upgrade`
Dies verwendet die `testing`-Distribution und aktualisiert alle Pakete im System entsprechend der Auswahl durch `dselect`.
- `apt-get -u install Paket`
Dies installiert `Paket` und dessen Abhängigkeiten aus der `testing`-Distribution.
- `apt-get -u install Paket/unstable`
Dies installiert `Paket` aus der `unstable`-Distribution, während dessen Abhängigkeiten aus der `testing`-Distribution installiert werden.
- `apt-get -u install -t unstable Paket`
Dies installiert `Paket` aus der `unstable`-Distribution, während dessen Abhängigkeiten auch aus der `testing`-Distribution mit einer Pin-Priority von 990 für `unstable` installiert werden.
- `apt-cache policy foo bar ...`
Dies gibt den Status der Pakete `foo bar ...` aus.
- `apt-cache show foo bar ... | less`
Dies gibt die Informationen zu den Paketen `foo bar ...` aus.
- `apt-get install foo=2.2.4-1`
Dies installiert die spezielle Version `2.2.4-1` des Pakets `foo`.
- `apt-get -u install foo bar-`
Dies installiert das Paket `foo` und entfernt das `bar` Paket.
- `apt-get remove bar`
Dies entfernt das Paket `bar`, die angepasste Konfiguration wird jedoch beibehalten.

- `apt-get remove --purge bar`

Dies entfernt das Paket *bar* zusammen mit allen Konfigurationsdateien.

Die Option `-u` in den obigen Beispielen sorgt dafür, dass `apt-get` eine Liste aller Pakete ausgibt, die aktualisiert werden können und fragt den Nutzer bevor es beginnt. Das Folgende setzt die Option `-u` standardmäßig:

```
$ cat >> /etc/apt/apt.conf << .
// Immer zu aktualisierende Pakete anzeigen (-u)
APT::Get::Show-Upgraded "true";
.
```

Die Option `-s` kann zum Simulieren eines Upgrades verwendet werden, ohne dass eine Aktualisierung erfolgt.

3.2.6 Verwenden einer speziellen Version der Debian-Distribution

Abhängig von der von Ihnen bevorzugten Debian-Version kann die Datei `/etc/apt/preferences` in 'Einrichten des APT-Systems' auf Seite 9 entsprechend Ihren Wünschen angepasst werden:

```
für stable:           Pin-Priority von testing auf 50 setzen
für testing:         Einstellungen wie oben gezeigt beibehalten
für testing(unstable): Pin-Priority von unstable auf 500 setzen
für unstable(testing): Pin-Priority von unstable auf 800 setzen
```

Eine Richtlinie für die Wahl der Pin-Priority ist das Wechseln von oben nach unten in der obigen Tabelle, so wie die Zeit von unmittelbar nach einer Veröffentlichung bis zum Einfrieren für die nächste Version vergeht.

Achtung: Das Verwenden der `testing`-Version von Debian hat den Nebeneffekt, dass Sicherheitsupdates nur langsam eingespielt werden.

Wenn verschiedene Versionen von Debian verwendet werden, so wie `testing` mit `stable` oder `unstable` mit `stable`, so werden höchstwahrscheinlich wichtige Pakete unabsichtlich von `testing` oder `unstable` genutzt, die fehlerhaft sein können. Sie wurden gewarnt.

Beispiele von `/etc/apt/preferences`, in welchen einige wichtige Pakete auf ausgereifte Versionen gesetzt werden, während die nicht ganz ausgereiften für nichtessentielle Pakete verwendet werden, sind im Beispielverzeichnis (<http://www.debian.org/doc/manuals/debian-reference/examples/>) als `preferences.testing` und `preferences.unstable` verfügbar. Auf der anderen Seite erzwingt `preferences.stable` einen Versionssprung aller Pakete zu `stable` (Downgrad).

3.2.7 Downgrad aller Pakete zu stable

Um alle Pakete auf die `stable`-Distribution zurückzustufen, editieren Sie die Datei `/etc/apt/preferences` wie folgt:

```
Package: *
Pin: release a=stable
Pin-Priority: 1001
```

und führen Sie `apt-get upgrade` aus. Dieser Befehl stuft aufgrund der `Pin-Priority > 1000` sämtliche Pakete herunter. Aber seien Sie gewarnt, es könnte kleinere Abhängigkeitsprobleme geben.

3.2.8 Überblick über `/etc/apt/preferences`

In dem in 'Einrichten des APT-Systems' auf Seite 9 aufgeführten Beispiel von `/etc/apt/preferences` bedeutet `"Package: *"` in der ersten Zeile, dass sich der Abschnitt auf alle Pakete bezieht. Man kann auch einen Paketnamen anstatt von `"*"` verwenden, um die `Pin-Priority` für das Paket festzulegen.

`"Pin: release a=stable"` in der zweiten Zeile bedeutet, dass `apt-get` die `Package.gz`-Datei aus dem Archiv nimmt, das die Zeile `"Archive: stable"` in seiner `Release`-Datei enthält.

Die letzte Zeile `"Pin-Priority: 500"` setzt `Pin-Priority` auf 500.

Wenn es mehrere Pakete mit dem selben Namen gibt, so wird normalerweise das mit der höchsten `Pin-Priority` installiert.

Die Bedeutung der `Pin-Priority` ist:

- 1001 und höher: Zurückstufungsprioritäten.
Ein Zurückstufen (Downgrade) des Pakets ist erlaubt für diesen Bereich der `Pin-Priority`.
- 100 bis 1000: Standardprioritäten.
Ein Zurückstufen des Pakets ist nicht erlaubt. Einige wichtige `Pin-Priority` Werte:
 - 990: die Priorität, die durch die `--target-release` oder `-t` Option von `apt-get (8)` gesetzt wird.
 - 500: die Priorität für alle Standardpaketdateien.
 - 100: die Priorität der zurzeit installierten Paketdateien.
- 0 bis 99: Nicht-automatische Prioritäten. (Diese werden nur genutzt, wenn das Paket nicht installiert ist und keine andere Version verfügbar ist.)
- kleiner als 0: Die Version wird niemals gewählt.

Die selben Effekte wie `--target-release` können durch Setzen von `/etc/apt/apt.conf` mittels

```
# echo 'APT::Default-Release "testing";' >> /etc/apt/apt.conf
```

erreicht werden. Die Kommandozeilenoption `--target-release` und der Wert in `/etc/apt/apt.conf` ersetzen die Einstellung in `/etc/apt/preferences`. Seien Sie vorsichtig damit, wenn Sie gleichzeitig `/etc/apt/preferences` editieren.

3.3 Debian-Überlebensbefehle

Mit diesem Wissen kann jeder ein Leben des ewigen "Aktualisierens" führen :-)

3.3.1 Überprüfen von Bugs in Debian und Suche nach Hilfe

Wenn Sie auf Probleme bezüglich eines bestimmten Pakets stoßen, stellen Sie sicher, dass Sie zuerst diese Seiten überprüfen, bevor Sie um Hilfe fragen oder einen Bug-Report erstellen. (`lynx`, `links` und `w3m` sind hier gleichwertig):

```
$ lynx http://bugs.debian.org/  
$ lynx http://bugs.debian.org/Paketname # wenn der Name bekannt ist  
$ lynx http://bugs.debian.org/Bugnummer # wenn Bugnr. bekannt ist
```

Suchen Sie auch bei Google (www.google.de) mit Suchbegriffen wie "site:debian.org".

Wenn Sie nicht weiter kommen, lesen Sie die guten Handbücher. Setzen Sie `CDPATH` wie folgt:

```
export CDPATH=./usr/local:/usr/share/doc
```

und dann

```
$ cd Paketname  
$ pager README.Debian # falls dies existiert  
$ mc
```

3.3.2 APT-Upgrade-Fehlersuche

Es können Paketabhängigkeitsprobleme beim Aktualisieren auf `unstable/testing` entsprechend 'Aktualisierung' auf Seite 4 entstehen. Meistens liegt das daran, dass ein Paket aktualisiert wird, das neue Abhängigkeiten hat, die nicht erfüllt sind. Diese Probleme können unter Verwendung folgender Befehle gelöst werden:

```
# apt-get dist-upgrade
```

Wenn dies nicht hilft das Problem zu lösen, dann wiederholen Sie folgende Befehle solange, bis sich das Problem auflöst:

```
# apt-get upgrade -f          # selbst bei Fehlern mit Upgrade ...  
... oder  
# apt-get dist-upgrade -f    # ... bzw. Dist-Upgrade fortfahren
```

Einige sehr fehlerhafte Upgrade-Skripte könnten permanente Schwierigkeiten bereiten. In solch einer Situation ist es gewöhnlich besser, wenn man sich die `/var/lib/dpkg/info/Paketname.{post,pre}{inst,rm}`-Skripte der betreffenden Pakete ansieht und anschließend

```
# dpkg --configure -a        # konfiguriert alle nicht vollständig  
                             # installierten Pakete
```

ausführt.

Wenn eines der Skripte sich über eine fehlende Konfigurationsdatei beschwert, dann schauen Sie in `/etc/` nach der entsprechenden Datei. Wenn eine mit der Dateiendung `.new` (oder so ähnlich) existiert, entfernen Sie diese Dateiendung einfach (z.B. mit `mv`).

Paketabhängigkeitsprobleme können auftreten, wenn Pakete in `unstable/testing` installiert werden. Diese können auf unterschiedliche Art und Weise umgangen werden.

```
# apt-get install -f Paket # ignoriere falsche Abhängigkeiten
```

Eine Alternative dazu wäre, das `equivs`-Paket zu verwenden. Siehe `/usr/share/doc/equivs/README.Debian`.

3.3.3 Wiederherstellung mit dpkg

Sollte das System einmal wirklich schwer geschädigt worden sein, beispielsweise wenn `dselect` (APT) abstürzt, kann man es auch nur mit Hilfe von `dpkg` wiederherstellen:

```
# cd /var/cache/apt/archives  
# dpkg -i libc6* libdb2* perl*  
# dpkg -i apt* dpkg* debconf*  
# dpkg -i *          # solange, bis keine Fehler mehr auftreten
```

Sollte einmal ein Paket fehlen, können Sie es von Mirror-Seiten (<http://www.debian.org/mirror/>) herunterladen:

```
# mc      # verwenden Sie "FTP-Verbindung" auf Debian-FTP-Server
```

Neue Pakete auf den HTTP/FTP-Servern befinden sich nicht mehr in dem klassischen `dist`-Verzeichnis, sondern im neuen `pool`-Verzeichnis.

Die Installation des Pakets erfolgt dann folgendermaßen:

```
# dpkg -i /var/cache/apt/archives/Paketdatei.deb
```

Sollte ein Problem mit einer Abhängigkeit auftreten, kann dieses wie folgt gelöst werden:

```
# dpkg --ignore-depends=Paket1 ... -i Paketdatei.deb
# dpkg --force-depends -i Paketdatei.deb
# dpkg --force-depends --purge Paket
# dpkg --force-confmiss -i Paketdatei.deb # installiert fehlende
                                           # Konfigurationsdateien
```

3.3.4 Wiederherstellung von Paketauswahldaten

Wenn `/var/lib/dpkg/status` aus irgendeinem Grund fehlerhaft ist, so verliert das Debian-System die Paketauswahldaten und nimmt Schaden. Schauen Sie nach der alten `/var/lib/dpkg/status-Datei` `/var/lib/dpkg/status-old` oder `/var/backups/dpkg.status.*`.

Eine eigene Partition für `/var/backups/` zu erstellen, könnte eine gute Idee sein, insbesondere da sich dort wichtige Systemdaten befinden.

Wenn die alte `/var/lib/dpkg/status-Datei` nicht verfügbar ist, kann man dennoch Informationen von Verzeichnissen in `/usr/share/doc/` wiederherstellen.

```
# ls /usr/share/doc | \
  grep -v [A-Z] | \
  egrep -v '^(debian|texmf)$' | \
  awk '{print $1 " install"}' | \
  dpkg --set-selections
... neuinstallieren des Systems, abwählen je nach Bedarf:
# dselect --expert
```

3.3.5 Systemrettung nach Löschen von `/var`

Da das `/var`-Verzeichnis regelmäßig aktualisierte Daten wie Mails enthält, ist es für einen Ausfall anfällig. Das Auslagern auf eine unabhängige Partition reduziert die Risiken. Wenn ein Unglück geschieht, muss das `/var`-Verzeichnis wiederhergestellt werden, um das Debian-System zu retten.

Besorgen Sie sich den wichtigsten Inhalt des `/var`-Verzeichnisses von einem kleinen noch funktionsfähigem Debian-System, basierend auf der selben oder einer älteren Debian-Version, z.B. `var.tar.gz` (<http://people.debian.org/~osamu/pub/>) und kopieren Sie dies in das Stammverzeichnis des defekten Systems. Führen Sie nun Folgendes aus:

```
# cd /
# mv var var-alt      # wenn noch sinnvolle Daten vorhanden sind
# tar xvzf var.tar.gz # Woody's Minimaldatei verwenden
# aptitude            # oder dselect
```

Dies sollte zu einem funktionierenden System führen. Die Wiederherstellung der Paketauswahl kann beschleunigt werden, wenn man die in 'Wiederherstellung von Paketauswahldaten' auf der vorherigen Seite beschriebene Technik anwendet. (FIXME: Dieses Vorgehen benötigt mehr Versuche zur Bestätigung.)

3.3.6 Installation eines Pakets in ein nicht bootfähiges System

Booten Sie Linux von einer der Debian-Rettungs-Disketten/CDs oder von einer alternativen Partition in einem Multi-Boot-System. Mounten Sie nun die nicht bootfähige Partition nach `/target` und verwenden Sie die chroot-Installationsmethode von `dpkg`.

```
# dpkg --root /target -i Paketdatei.deb
```

Anschließend sollten Sie alles nötige konfigurieren und Probleme beheben.

Übrigens, wenn `lilo` alles ist, was Sie davon abhält das System zu booten, können Sie auch ganz einfach von einer Debian-Rettungsdiskette booten. Am Boot-Prompt, vorausgesetzt Ihre root-Partition befindet sich auf `/dev/hda12` und Sie möchten im Run-Level 3 starten, geben Sie Folgendes ein:

```
boot: rescue root=/dev/hda12 3
```

Daraufhin startet ein nahezu vollständiges System, mit dem Kernel von der Bootdiskette. (Möglicherweise werden nicht alle Hardwarekomponenten erkannt oder Module nicht geladen, dies liegt aber an dem Kernel von der Bootdiskette.)

3.3.7 Der `dpkg`-Befehl funktioniert nicht mehr – was nun?

Ein funktionsgestörtes `dpkg` kann es verständlicherweise unmöglich machen, beliebige `.deb`-Dateien zu installieren. Ein Vorgehen wie das unten aufgeführte wird Ihnen dabei helfen, diese Situation wieder in den Griff zu bekommen. (Natürlich können Sie `links` durch Ihrem favorisierten Browser ersetzen.)

```
$ links http://http.de.debian.org/debian/pool/main/d/dpkg/  
... herunterladen von dpkg_version_arch.deb  
$ su  
password: *****  
$ ar x dpkg_version_arch.deb  
# mv data.tar.gz /data.tar.gz  
# cd /  
# tar xzfv data.tar.gz
```

Für die i386-Architektur kann auch <http://packages.debian.org/dpkg> als URL genutzt werden.

3.4 Debian-Nirwana-Befehle

Das **Verständnis** der folgenden Befehle ist eine essentielle Voraussetzung, um zum einem eventuellen Upgrade-Problemen gewachsen zu sein und zum anderen um das **Debian-Nirwana** zu erreichen. :-)

3.4.1 Informationen über eine Datei

Sie möchten den Namen des Pakets in Erfahrung bringen, welchem eine bestimmte Datei angehört:

```
... suche nach Muster in den installierten Paketdateien:  
$ dpkg {-S|--search} Muster  
... finde das Dateinamen-Muster in den Debian-Archivdateien:  
$ wget http://ftp.us.debian.org/debian/dists/sarge/Contents-i386.gz  
$ zgrep -e pattern Contents-i386.gz
```

Sie können auch spezialisierte Paketkommandos verwenden:

```
... dlocate:  
# apt-get install dlocate  
... schnellere Alternative zu dpkg -L and dpkg -S:  
$ dlocate filename  
... installiert Pakete automatisch auf Anfrage:  
# apt-get install auto-apt  
... erzeugt eine Datenbank für auto-apt:  
# auto-apt update  
... sucht nach einem Muster in allen Paketen, ob sie installiert  
... sind oder nicht:  
$ auto-apt search Muster
```

3.4.2 Informationen über ein Paket

Die Debian-Paketarchive können durchsucht und angezeigt werden. Stellen Sie jedoch vorher sicher, dass APT die richtigen Archive in `/etc/apt/sources.list` findet. Wenn Sie sehen wollen, wie sich Pakete in `testing/unstable` gegen die aktuell installierten verhalten, so verwenden Sie `apt-cache policy`. Dies ist sehr nett.

```
# apt-get check                # aktualisiert den Cache und schaut nach
                                # fehlerhaften Paketen
$ apt-cache search Muster    # sucht Pakete anhand Paketbeschreibung
$ apt-cache policy Paket     # Paketprioritäten/Distributions-Infos
$ apt-cache show -a Paket    # zeigt die Paketbeschreibung in allen
                                # Distributionen
$ apt-cache showsrc Paket    # zeigt die Paketbeschreibung des
                                # Quellpakets
$ apt-cache showpkg Paket    # zeigt Paketinformationen für Debugging
# dpkg --audit|-C              # sucht nach nicht vollständig
                                # installierten Paketen
$ dpkg {-s|--status} Paket ... # Beschreibung eines installierten
                                # Pakets
$ dpkg -l Paket ...          # Status des installierten Pakets
$ dpkg -L Paket ...          # zeigt die Dateien an, die durch das
                                # Paket installiert wurden
```

`apt-cache showsrc` ist in Woody noch nicht dokumentiert, funktioniert aber :)

Sie können ebenfalls anstelle der oben genannten Befehle Paketinformationen in folgenden Dateien finden (verwenden Sie beispielsweise `mc` zum Suchen):

```
/var/lib/apt/lists/*
/var/lib/dpkg/available
```

Ein Vergleich folgender Dateien kann Aufschluss darüber geben, was genau bei der letzten Installation passiert ist.

```
/var/lib/dpkg/status
/var/backups/dpkg.status*
```

3.4.3 Nicht-Interaktive Installation mit APT

Um Pakete nicht-interaktiv zu installieren, fügen Sie folgende Zeile zu `/etc/apt/apt.conf` hinzu:

```
Dpkg::Options {"--force-confold";}
```

Dies ist äquivalent zu `apt-get -q -y Paket`. Da dies alle Fragen automatisch mit "Ja" beantwortet, kann dies zu Problemen führen. Nutzen Sie diese Option deshalb mit Vorsicht. Lesen Sie `apt.conf(5)` und `dpkg(1)`.

Sie können jedes einzelne Paket auch später wie in 'Installierte Pakete erneut konfigurieren' auf dieser Seite beschrieben konfigurieren.

3.4.4 Installierte Pakete erneut konfigurieren

Verwenden Sie folgende Befehle, um bereits installierte Pakete erneut zu konfigurieren.

```
# dpkg-reconfigure --priority=medium Paket [...]
# dpkg-reconfigure --all # Neukonfiguration aller Pakete
# dpkg-reconfigure locales # generiere zusätzliche locales
# dpkg-reconfigure --p=low xserver-xfree86 # konfiguriere den
# X-Server neu
```

Führen Sie dies für `debconf` aus, wenn Sie den Dialogmodus von `debconf` permanent ändern möchten.

Einige Programme sind mit speziellen Konfigurationsskripten ausgestattet.²

```
apt-setup      - erstellt die Datei /etc/apt/sources.list
install-mbr    - installiert einen Master-Boot-Record-Manager
tzconfig       - setzt die lokale Zeitzone
gpmconfig      - konfiguriert den gpm-Maus-Daemon
eximconfig     - konfiguriert Exim (MTA)
texconfig      - konfiguriert teTeX
apacheconfig   - konfiguriert Apache (httpd)
cvsconfig      - konfiguriert CVS
sndconfig      - konfiguriert das Soundsystem
...
update-alternatives - definiert Standardbefehle, z.B. vi für vim
update-rc.d     - System-V-init-Skript-Management
update-menus    - das Debian-Menüsystem
...
```

3.4.5 Deinstallation von Paketen

Entfernen eines Pakets, die Konfiguration wird nicht gelöscht:

```
# apt-get remove Paket ...
# dpkg --remove Paket ...
```

²Einige `*config`-Skripte verschwinden in der neueren Sarge-Ausgabe und die Paketkonfigurationsfunktionen wurden nach `debconf` verschoben.

Entfernen eines Pakets und dessen Konfiguration:

```
# apt-get remove --purge Paket ...
# dpkg --purge Paket ...
```

3.4.6 "Halten" älterer Pakete

Zunächst ein Beispiel: Möchte man, dass die Pakete `libc6` und `libc6-dev` beim Ausführen von `dselect` und `apt-get -u upgrade` nicht durch neuere Versionen ersetzt werden, so kann dies folgendermaßen erreicht werden:

```
# echo -e "libc6 hold\nlibc6-dev hold" | dpkg --set-selections
```

`apt-get -u install Paket` bleibt davon jedoch unbeeinflusst. Um diese Pakete andererseits bei einem automatischen Systemdowngrade mittels `apt-get -u upgrade` oder `apt-get -u dist-upgrade` zurück zu halten, ist die Datei `/etc/apt/preferences` wie folgt zu erweitern:

```
Package: libc6
Pin: release a=stable
Pin-Priority: 2000
```

An dieser Stelle kann der Eintrag `"Package:"` keine Paketnamen wie z.B. `"libc6*"` enthalten. Wenn Sie jedoch alle Pakete, die im Zusammenhang mit dem `glibc`-Quellpaket stehen, auf dem selben Stand halten möchten, müssen Sie diese alle explizit auflisten.

Folgender Befehl zeigt alle Pakete an, die den Status `"hold"` haben:

```
dpkg --get-selections "*" | grep -e "hold$"
```

3.4.7 Gemischtes `stable/testing/unstable`-System

`apt-show-versions` kann alle verfügbaren Paketversionen und deren Zugehörigkeit zu den einzelnen Distributionen anzeigen.

```
$ apt-show-versions | fgrep /testing | wc
... Anzahl der installierten Pakete aus testing
$ apt-show-versions -u
... Liste aller Pakete für die es aktuellere Versionen gibt
$ apt-get install `apt-show-versions -u -b | fgrep /unstable`
... aktualisiert alle Paket aus unstable auf die aktuellste Version
```

3.4.8 Löschen von zwischengespeicherten Paketdateien

APT speichert alle heruntergeladenen Paketdateien in `/var/cache/apt/archives/`. Diese können folgendermaßen gelöscht werden:

```
# apt-get autoclean # löscht nur nicht benötigte Paketdateien
# apt-get clean     # löscht alle zwischengespeicherten Paketdateien
```

3.4.9 Speichern/Kopieren der Systemkonfiguration

Erstellen einer lokalen Kopie der aktuellen Paketauswahl:

```
# debconf-get-selections > debconfsel.txt
# dpkg --get-selections "*" >meineAuswahl # oder \*
```

"*" veranlasst, dass auch Pakete mit einem Status von "purge" in *meineAuswahl* gespeichert werden.

Diese Datei kann auf einen anderen Computer kopiert werden, um dort die gleichen Pakete zu installieren:

```
# dselect update
# debconf-set-selections < debconfsel.txt
# dpkg --set-selections <meineAuswahl
# apt-get -u dselect-upgrade # oder dselect install
```

3.4.10 Portierung eines Pakets auf die stable-Distribution

Um selektive Upgrades in der stabilen Distribution zu machen, ist es möglich, ein Quellpaket in dieser Umgebung zu kompilieren. Dies verhindert viele Paketaktualisierungen, die durch Abhängigkeiten verursacht würden. Zuerst müssen folgende Zeilen der `/etc/apt/sources.list`-Datei hinzugefügt werden:

```
deb-src http://http.us.debian.org/debian testing \
main contrib non-free
deb-src http://http.us.debian.org/debian unstable \
main contrib non-free
```

Aus Formatierungsgründen sind die Zeilen hier in je zwei Zeilen umgebrochen. Natürlich muss aber jede mit "deb-src" beginnende Zeile für sich alleine in `sources.list` stehen.

Nun ist der Quellcode zu beziehen und ein lokales Paket zu erstellen:

```
$ apt-get update # aktualisiere die Quellpaketsuchliste
$ apt-get source Paket
$ dpkg-source -x Paket.dsc
$ cd paket-version
... schauen Sie nach benötigten Paketen (Build-depends in der
... .dsc-Datei) und installieren Sie diese ebenso wie fakeroot

$ dpkg-buildpackage -rfakeroot

... oder (keine Signatur erstellen)
... verwenden Sie später "debsign" wenn nötig
$ dpkg-buildpackage -rfakeroot -us -uc

... Anschließend die Installation des Pakets
$ su -c "dpkg -i Paketdatei.deb"
```

Gewöhnlich ist es nötig, einige Pakete mit der "-dev"-Erweiterung zu installieren, um Paketabhängigkeiten auflösen zu können. `debsign` ist im Paket `devscripts` enthalten. `auto-apt` kann die Auflösung von Abhängigkeiten erleichtern. Mittels `fakeroot` lässt sich unnötige Benutzung des `root`-Accounts vermeiden.

In der Woody-Distribution lassen sich diese Abhängigkeitsprobleme vereinfachen. Als ein Beispiel kompilieren wir `pine`, das nur im Quellcode vorliegt.

```
# apt-get build-dep pine
# apt-get source -b pine
```

3.4.11 Lokale Paketarchive

Möchte man ein lokales Paketarchiv erzeugen, welches mit APT und `dselect` kompatibel ist, so muss die Datei `Packages` erstellt werden und Paketdateien müssen in einem bestimmten Verzeichnis erstellt werden.

Ein lokales `deb`-Archiv (Repository) das ähnlich zum offiziellen Debian-Archiv ist, kann wie folgt erstellt werden:

```
# apt-get install dpkg-dev
# cd /usr/local
# install -d pool # die Paketdateien werden hier gespeichert
# install -d dists/unstable/main/binary-i386
# ls -1 pool | sed 's/_.*$/ priority section/' | uniq > override
# editor override # anpassen von priority und section
# dpkg-scanpackages pool override /usr/local/ \
  > dists/unstable/main/binary-i386/Packages
# cat > dists/unstable/main/Release << EOF
```

```
Archive: unstable
Version: 3.0
Component: main
Origin: Local
Label: Local
Architecture: i386
EOF
# echo "deb file:/usr/local unstable main" \
  >> /etc/apt/sources.list
```

Alternativ kann auch folgende, schnellere aber unsaubere, Methode verwendet werden:

```
# apt-get install dpkg-dev
# mkdir /usr/local/debian
# mv /irgend/ein/Paket.deb /usr/local/debian
# dpkg-scanpackages /usr/local/debian /dev/null | \
  gzip - > /usr/local/debian/Packages.gz
# echo "deb file:/usr/local/debian ." >> /etc/apt/sources.list
```

Auf ein solches Archiv kann auch von entfernten Computern zugegriffen werden, indem der Zugriff auf diese Verzeichnisse durch HTTP oder FTP ermöglicht wird. Die Datei `/etc/apt/sources.list` muss dementsprechend angepasst werden.

3.4.12 Installation und Konvertierung eines fremden Binärpakets

Mit Hilfe des `alien`-Pakets lassen sich binäre Pakete, wie sie in vielen Distributionen wie Red Hat rpm, Stampede slp, Slackware tgz und Solaris pkg verwendet werden, in das Debian deb-Paketformat konvertieren. Wenn Sie ein Paket von einer anderen Distribution verwenden möchten, können Sie es mit `alien` in Ihr bevorzugtes Paketformat konvertieren und anschließend installieren. `alien` unterstützt ebenfalls LSB-Pakete.

3.4.13 Verifizierung installierter Paketdateien

`debsums` verifiziert installierte Paketdateien anhand deren MD5 Prüfsummen. Jedoch gibt es auch Pakete die keine MD5-Prüfsummen enthalten. Eine mögliche temporäre Lösung für Systemadministratoren ist folgende:

```
# cat >>/etc/apt/apt.conf.d/90debsums
DPkg::Post-Install-Pkgs {"xargs /usr/bin/debsums -sg";};
^D
```

von Joerg Wendland <joergland@debian.org> (nicht getestet).

3.5 Weitere Debian-Besonderheiten

3.5.1 Der `dpkg-divert`-Befehl

Dateiumleitungen ist eine Methode, um `dpkg` dazu zu bringen, Dateien nicht in ihre Standardverzeichnisse zu installieren, sondern in **umgeleitete** Verzeichnisse. **Umleitungen** können durch Debian-Skripte dazu genutzt werden, Dateien zu verschieben, wenn sie einen Konflikt verursachen. Systemadministratoren können auch eine Umleitung verwenden, um die Konfigurationsdatei eines Pakets zu überschreiben, oder wann immer einige Dateien (welche nicht als **conffiles** markiert sind) von `dpkg` nicht überschrieben werden sollen, wenn eine neuere Version eines Pakets installiert wird, das solche Dateien enthält.

```
# dpkg-divert [--add] Datei # erstellen der "Umleitung"
# dpkg-divert --remove Datei # entfernen der "Umleitung"
```

Für gewöhnlich ist es ratsam `dpkg-divert` nicht zu verwenden wenn es nicht absolut notwendig ist.

3.5.2 Das `equivs`-Paket

Wenn Sie ein Quellpaket kompilieren ist es empfehlenswert, es zu einem echten lokalen Debianpaket (`*.deb`) zu machen. Benutzen Sie `equivs` als letzten Ausweg.

```
Package: equivs
Priority: extra
Section: admin
Description: Circumventing Debian package dependencies
 This is a dummy package which can be used to create Debian
 packages, which only contain dependency information.
```

3.5.3 Alternative Befehle

Möchte man dass der Befehl `vi` `vim` ausführt, benutzen Sie `update-alternatives`:

```
# update-alternatives --display vi
...
# update-alternatives --config vi
  Selection      Command
-----
          1      /usr/bin/elvis-tiny
          2      /usr/bin/vim
*+       3      /usr/bin/nvi
```

Enter to keep the default[*], or type selection number: 2

Einstellungen im Debian-Alternativsystem werden in `/etc/alternatives/` als symbolische Links gespeichert.

Um Ihre bevorzugte X-Window-Umgebung einzustellen, führen sie `update-alternatives` für `/usr/bin/x-session-manager` und `/usr/bin/x-window-manager` aus.

`/bin/sh` ist ein symbolischer Link auf `/bin/bash` oder `/bin/dash`. Aus Kompatibilitätsgründen zu alten Bash-verunreinigten Skripten ist es sicherer, `/bin/bash` zu verwenden. Um die POSIX-Konformität einzuhalten sollte jedoch `/bin/dash` verwendet werden. Mit dem Upgrade auf den 2.4er-Linux-Kernel geht die Tendenz zugunsten zu `/bin/dash`.

3.5.4 `init` und Runlevel

Der Standardrunlevel in dem gebootet wird, lässt sich in `/etc/inittab` festlegen.

In Debian ist es dem Systemadministrator möglich, einzustellen, welche Dienste in welchem Runlevel laufen sollen. Das `update-rc.d`-Werkzeug kann verwendet werden, um diese Einstellungen zu ändern, unabhängig davon, ob die `sysv-rc`- oder die `file-rc`-Methode zum Kontrollieren von Diensten verwendet wird.

Um zum Beispiel den Dienst *name* (mit dem Init-Skript `/etc/init.d/name`) mit der Sequenznummer 20 in den Runlevel 2, 3 und 4 zu starten und in Runlevel 5 mit der Sequenznummer 80 zu stoppen, ist Folgendes auszuführen:

```
# update-rc.d name start 20 2 3 4 . stop 80 5 .
```

Wenn Sie das `sysv-rc` Paket benutzen, gibt es einen anderen Weg, solche Änderungen durchzuführen: Benennen Sie einfach die symbolischen Links in den `/etc/rc?.d` Verzeichnissen um.

```
# mv S99xdm K01xdm # xdm deaktivieren (X-Display-Manager)
```

3.5.5 Deaktivierung von Daemon-Diensten

Die Debian-Distribution nimmt die Systemsicherheit sehr ernst und erwartet, dass der Systemadministrator kompetent ist. Eine einfache Administration steht so oft nur an zweiter Stelle und viele Daemon-Dienste sind in der Standard-Installation mit dem höchsten Sicherheitslevel vorkonfiguriert oder mit den wenigsten Diensten (oder keinen) ausgestattet.

Führen Sie `ps aux` aus oder überprüfen Sie die Inhalte von `/etc/init.d/*` und `/etc/inetd.conf`, wenn Sie irgendetwas Zweifel (zu Exim, DHCP, ...) haben. Überprüfen Sie auch `/etc/hosts.deny`. Der `pidof`-Befehl ist ebenfalls hilfreich (siehe `pidof(8)`).

X11 erlaubt standardmäßig keine (entfernten) TCP/IP-Verbindungen in neueren Debian-Versionen. X-Weiterleitung mittels SSH ist ebenfalls deaktiviert.

Anhang A

Anhang

A.1 Autoren

Die Debian-Schnellreferenz wurde ursprünglich von Osamu Aoki <osamu\#at\#debian.org> als eine persönliche Installationsanleitung geschrieben, die “Schnellreferenz ...” hieß. Ein großer Teil des Inhalts kam vom Archiv der “debian-user“-Mailingliste. Auch “Debian Installation Manual” und “Debian Release Notes” waren Referenzen.

Einem Vorschlag von Josip Rodin folgend, der sehr aktiv im Debian-Dokumentations-Projekt (<http://www.debian.org/doc/ddp>) (DDP) ist und der der aktuelle Betreuer der “Debian-FAQ” ist, wurde dieses Dokument in “Debian-Referenz” umbenannt und einige Kapitel der “Debian-FAQ” mit verweisartigem Inhalt hinzugefügt. Danach wurde “Debian-Schnellreferenz” als ein Auszug geschaffen.

Dieses Dokument wurde geschrieben, übersetzt und erweitert durch die folgenden QREF-Teammitglieder:

- Englisch Original für die ursprüngliche “Schnellreferenz ...”
 - Osamu Aoki <osamu\#at\#debian.org> (Koordinator des gesamten Inhalts)
- Englisch Korrekturlesen und zusätzliche Beiträge
 - David Sewell <dsewell\#at\#virginia.edu> (umfangreiche Arbeit am Stil)
 - Thomas Hood <jdthood\#at\#yahoo.co.uk> (Netzwerk-bezogen)
 - Brian Nelson <nelson\#at\#bignachos.com> (besonders X-bezogen)
 - Jan Michael C Alonzo <jmalonzo\#at\#spaceants.net>
 - Daniel Webb <webb\#at\#robust.colorado.edu>
 - Rückmeldungen von allen Übersetzern
- Französische Übersetzung
 - Guillaume Erbs <gerbs\#at\#free.fr> (Koordinator fr)
 - Rénaud Casagraude <rcasagraude\#at\#interfaces.fr>
 - Jean-Pierre Delange <adeimantos\#at\#free.fr>
 - Daniel Desages <daniel\#at\#desages.com>
- Italienische Übersetzung
 - Davide Di Lazzaro <mc0315\#at\#mclink.it> (Koordinator it)
- Portugiesische Übersetzung (brasilianisches Portugiesisch)

- Paulo Rogério Ormenese <pormenese\#at\#uol.com.br> (Koordinator pt-br)
- Andre Luis Lopes <andrelap\#at\#ig.com.br>
- Marcio Roberto Teixeira <marciotex\#at\#pop.com.br>
- Rildo Taveira de Oliveira <to_rei\#at\#yahoo.com>
- Raphael Bittencourt Simoes Costa <raphael-bsc\#at\#bol.com.br>
- Gustavo Noronha Silva <kov\#at\#debian.org> (Koordinator)
- Spanische Übersetzung
 - Walter Echarri <wecharri\#at\#infovia.com.ar> (Koordinator es)
 - José Carreiro <ffx\#at\#urbanet.ch>
- Deutsche Übersetzung
 - Jens Seidel <tux-master\#at\#web.de> (Koordinator de)
 - Willi Dyck <wdyck\#at\#gmx.net>
 - Stefan Schröder <stefan\#at\#fkip.uni-hannover.de>
 - Agon S. Buchholz <asb\#at\#kefk.net>
- Polnische Übersetzung – Mitglieder von PDDP (<http://debian.linux.org.pl>):
 - Marcin Andruszkiewicz
 - Mariusz Centka <mariusz.centka\#at\#debian.linux.org.pl>
 - Bartosz Feński <fenio\#at\#debian.linux.org.pl> (Koordinator pl)
 - Radosław Grzanka <radekg\#at\#debian.linux.org.pl>
 - Bartosz 'Xebord' Janowski
 - Jacek Lachowicz
 - Rafał Michaluk
 - Leonard Milcin, Jr.
 - Tomasz Z. Napierała <zen\#at\#debian.linux.org.pl>
 - Oskar Ostafin <cx\#at\#debian.linux.org.pl>
 - Tomasz Piękoś
 - Jacek Politowski
 - Mateusz Prichacz <mateusz\#at\#debian.linux.org.pl>
 - Marcin Rogowski
 - Paweł Różański
 - Mariusz Strzelecki
 - Krzysztof Ścierański
 - Przemysław Adam Śmiejek <tristan\#at\#debian.linux.org.pl>
 - Krzysztof Szynter
 - Mateusz Tryka <uszek\#at\#debian.linux.org.pl>
 - Cezary Uchto
 - Krzysztof Witkowski <tjup\#at\#debian.linux.org.pl>
 - Bartosz Zapałowski <zapal\#at\#debian.linux.org.pl>
- Chinesische Übersetzung (vereinfachtes Chinesisch)
 - Hao "Lyoo" LIU <iamlyoo\#at\#163.net>
 - Ming Hua <minghua\#at\#rice.edu>
 - Xiao Sheng Wen <atzlinux\#at\#163.com> (Koordinator zh-cn)
 - Haifeng Chen <optical.dlz\#at\#gmail.com>
 - Xie Yanbo <xieyanbo\#at\#gmail.com>
 - easthero <easthero\#at\#gmail.com>
- Chinesische Übersetzung (traditionelles Chinesisch)

- Asho Yeh <asho\#at\#debian.org.tw> (Koordinator zh-tw)
- Tang Wei Ching <wctang\#at\#csie.nctu.edu.tw> (ehemaliger Koordinator zh-tw)
- Japanische Übersetzung
 - Shinichi Tsunoda <tsuno\#at\#ngy.1st.ne.jp> (Koordinator ja)
 - Osamu Aoki <osamu\#at\#debian.org>

A.2 Haftung

Da ich kein Experte bin, gebe ich nicht vor, ein volles Wissen über Debian oder Linux im Allgemeinen zu besitzen. Die Sicherheitsüberlegungen die ich aufführe sind eventuell nur für private Anwendungen anwendbar.

Dieses Dokument ersetzt nicht die ultimativen Anleitungen.

Jegliche Haftung wird abgelehnt. Alle Handelsmarken sind das Eigentum der entsprechenden Markeninhaber.

A.3 Rückmeldungen

Kommentare und Ergänzungen zu diesem Dokument sind immer willkommen. Senden Sie bitte eine E-Mail an das Debian-BTS-System (<http://bugs.debian.org/>) unter dem `debian-reference` Paket oder unter dem entsprechend übersetzten Paket. Das Verwenden von `reportbug` vereinfacht das Erstellen eines vollständigen Fehlerberichts. Sie können auch in Englisch eine E-Mail an Osamu Aoki (<http://people.debian.org/~osamu/>) nach <osamu\#at\#debian.org> oder zu jedem Übersetzer in dessen Sprache senden.