

> Smart <

Alle Angaben ohne "" eingeben!

1. Smart installieren

[Alt+F2] ⇒ "yast" ⇒ [Enter] ⇒ "root-Passwort" ⇒ [Enter] ⇒ "Software" ⇒ "Software installieren oder löschen" ⇒ "Suchen" ⇒ "smart" ⇒

smart
smart-gui (nur für die GUI notwendig)

auswählen ⇒ "übernehmen"

(für SUSE 10.0 muss man smart hier Downloaden und dann mit "rpm -Uhv /Pfad/zu/rpm.rpm" installieren: <http://linux01.gwdg.de/~pbleser/rpm-navigation.php?cat=/System/smart>)

Smart ist nun installiert

2. Smart einrichten

Alt+F2 ⇒ "Konsole" ⇒ Enter

IN DER KONSOLE EINGEBEN:

"su" ⇒ [Enter] ⇒ "root-Passwort" ⇒ [Enter] ⇒

SuSE 10.3 32bit:

```
smart channel --add http://divine160281.di.funpic.de/channel/suse103x86.txt
```

SuSE 10.3 64bit:

```
smart channel --add http://divine160281.di.funpic.de/channel/suse103x86_64.txt
```

SuSE 11.0 32bit:

```
smart channel --add http://divine160281.di.funpic.de/channel/suse110x86.txt
```

SuSE 11.0 64bit:

```
smart channel --add http://divine160281.di.funpic.de/channel/suse110x86_64.txt
```

[Enter] ⇒ Channels mit [j] bestätigen ⇒ [Enter] ⇒ [j] ⇒ [Enter] ⇒ ... ⇒ ... ⇒ ...

bis fertig ist

danach: "smart update" ⇒ [Enter] und laufen lassen bis fertig.

Jetzt ist Smart einsatzbereit

3. Verwenden von Smart

Es gibt 2 Möglichkeiten:

1. Smart auf der Konsole:

```
su  
root-Passwort  
smart install "paketname"
```

2. Smart mit GUI

[Alt+F2] ⇒ "kdesu smart --gui" ⇒ [Enter] ⇒ "root-Passwort" ⇒ [Enter]

EIN BEISPIEL:

```
smart install vlc
```

Smart wählt nun noch zusätzliche Pakete aus und fragt dann "Änderungen übernehmen?"

[Enter] (wer unbedingt will kann auch [j] ⇒ [Enter])

Warten (und Zuschauen)

Und Fertig!

4. Update mit Smart

als root auf der Konsole:

```
smart update
```

Durchlaufen lassen

```
smart upgrade
```

(Wenn Smart mal ein Paket nicht auf dem Server findet bricht er ab, obwohl er die anderen installieren könnte. Man kann dann einfach das entsprechende Paket sperren (*siehe Punkt 6*), und smart upgrade nochmals starten, dann sollte es die restlichen Pakete installieren.)

Das System ist jetzt wieder aktuell

5. Automatisches Update mit Smart

Script von http://www.benjaminneff.ch/download/automatisches_update.sh herunterladen und irgendwo speichern.

Script öffnen und ganz am Anfang hinter EMAIL= den Benutzernamen zwischen "" eintragen, an den der tägliche Bericht gehen sollte!

Script ausführbar machen:

Rechtsklick ⇒ Eigenschaften ⇒ Berechtigungen ⇒ Auf ausführbar klicken ⇒ OK

Script als root nach /etc/cron.daily kopieren! (es wird dann jeden Tag ein Update ausgeführt)

KMail für den Empfang des täglichen Berichtes einrichten:

1. KMail öffnen
2. Einen neuen Ordner erstellen in dem die Berichte abgelegt werden sollen
3. Einstellungen ⇒ KMail einrichten ... ⇒ Accounts ⇒ Hinzufügen ⇒ Lokales Postfach ⇒ OK:
Hacken bei "Posteingang regelmässig überprüfen" machen und Zeit auf 15-30 Minuten einstellen.
Unter "Zielordner" den vorher erstellten Ordner angeben
4. OK ⇒ OK

Fertig

6. Pakete sperren

Man kann Pakete sperren, damit sie nicht verändert (installieren, aktualisiert, deinstalliert, usw) werden.

Das ist in Zusammenhang mit dem automatischen Update sehr praktisch, damit er nichts ungewolltes aktualisiert.

(Anwendungsbeispiel: Wenn jemand einen Treiber für seine GraKa installiert hat empfiehlt es sich kernel und kernel-source zu sperren.)

Man kann entweder das gesamte Paket sperren, oder nur einzelne Versionen.

Auf der Konsole in der Form:

```
smart flag --set lock pkgname           (ganzes Paket sperren)
smart flag --set lock 'pkgname >= 1.0' (bestimmte Version sperren)
```

Mit der GUI einfach rechts auf das entsprechende Paket klicken und dann zwischen allen oder nur dieser Version entscheiden.

Entfernt wird die Sperrung auf der Konsole auf dem selben Weg, man muss nur das --set durch ein --remove ersetzen.

Auf der GUI muss man einfach nochmals auf den entsprechenden Eintrag im Kontextmenü klicken!

```
smart flag --show lock
```

 gibt alle gesperrten Pakete aus.

7. Erweitert

1. smart

Action commands:

2. update
3. install
4. reinstall
5. upgrade
6. remove
7. check
8. fix
9. download
10. clean

Query commands:

11. search
12. query
13. info
14. stats

Setup commands:

15. channel
16. priority
17. mirror
18. flag

Optionen:

--version	Zeige die Programmversion und beende
--config-file=DATEI	Konfigurationsdatei (Standard ist <data-dir>/config)
--data-dir=Verzeichnis	Datenverzeichnis (Standard ist /var/lib/smart/)
--log-level=LEVEL	Setze den log Level zu level (debug, info, warning, error)
--gui	Benutze die Standard GUI
--shell	Benutze die Standard Shell Oberfläche
--interface=Name	Benutze die angegebene Oberfläche
--ignore-locks	Beachte keine Sperrungen

Beispiele:

```
smart install --help
smart install pkgname
smart --gui
smart --gui install pkgname
smart --shell
```

2. smart update:

Der Befehl updated die bekannten Informationen der gegebenen Channel.
Wenn keine Channel angegeben sind, werden alle, die nicht ausgeschaltet sind
oder auf manuell stehen, geupdated.

Optionen:

-h, --help	Zeige die Hilfe und beende
--after=MIN	Nur updaten, wenn das letzte erfolgreiche update vor dem eingestellten zeitraum lag

Beispiele:

```
smart update
smart update mychannel
smart update mychannel1 mychannel2
```

3. smart install:

Dieser Befehl installiert eines oder mehrere Pakete. Wenn eine neuere Version eines bereits installierten Paketes verfügbar ist, wird es für die Installation ausgewählt.

Optionen:

-h, --help	Zeige die Hilfe und beende
--stepped	Ausführung in Einzelschritte aufteilen
--urls	Dumpe benötigte URLs und führe keine Funktion aus
--download	Herunterladen der Pakete und keine Funktion ausführen
--explain	Füge zusätzliche Informationen über Änderungen ein, wenn möglich
-y, --yes	Frage nicht nach Bestätigung

Beispiele:

```
smart install pkgname
smart install '*kgna*'
smart install pkgname-1.0
smart install pkgname-1.0-1
smart install pkgname1 pkgname2
smart install ./somepackage.file
smart install http://some.url/some/path/somepackage.file
```

4. smart reinstall

Dieser Befehl installiert ein oder mehrere Pakete, die bereits installiert sind, erneut.

Verwendung: Gleich wie smart install.

5. smart upgrade

Upgraded eines oder mehrere Pakete, die installiert sind. Ist keine Vorgabe gemacht, werden alle installierten Pakete geprüft.

Optionen:

-h, --help	Zeige die Hilfe und beende
--stepped	Ausführung in Einzelschritte aufteilen
--urls	Dumpe benötigte URLs und führe keine Funktion aus
--download	Herunterladen der Pakete und keine Funktion ausführen
--update	Update Channel Informationen vor dem Upgrade
--check	Nur prüfen, ob es Upgrades gibt

<code>--check-update</code>	Prüfe, ob Upgrades vorhanden sind und update die vorhandenen
<code>--explain</code>	Füge zusätzliche Informationen über Änderungen ein, wenn möglich
<code>-y, --yes</code>	Frage nicht nach Bestätigung

Beispiele:

```

smart upgrade
smart upgrade pkgname
smart upgrade '*kgnam*'
smart upgrade pkgname-1.0
smart upgrade pkgname-1.0-1
smart upgrade pkgname1 pkgname2

```

6. smart remove

Dieser Befehl wird ein oder mehrere Pakete, die momentan installiert sind, entfernen.

Optionen:

<code>-h, --help</code>	Zeige die Hilfe und beende
<code>--stepped</code>	Ausführung in Einzelschritte aufteilen
<code>--urls</code>	Dumpe benötigte URLs und führe keine Funktion aus
<code>--download</code>	Herunterladen der Pakete und keine Funktion ausführen
<code>--explain</code>	Füge zusätzliche Informationen über Änderungen ein, wenn möglich
<code>-y, --yes</code>	Frage nicht nach Bestätigung

Beispiele:

```

smart remove pkgname
smart remove '*kgnam*'
smart remove pkgname-1.0
smart remove pkgname-1.0-1
smart remove pkgname1 pkgname2

```

7. smart check

Dieser Befehl prüft die Beziehungen zwischen Paketen. Wenn kein Paket angegeben ist, werden alle Pakete in ausgewählten Channel geprüft.

Nutze den `fix` Befehl um defekte Beziehungen von Paketen zu reparieren.

Optionen:

<code>-h, --help</code>	Zeige die Hilfe und beende
<code>--all</code>	Prüfe Pakete aller Channel
<code>--installed</code>	Prüfe Pakete, die mindestens in einem installierten Channel sind (Standard)
<code>--available</code>	Prüfe Pakete, die mindestens in einem nicht installierten channel sind
<code>--channels=ALIASES</code>	Prüfe Pakete, die in dem eingestellten Channel

sind (Komma teilt Aliase)

Beispiele:

```
smart check
smart check pkgname
smart check '*kgna*'
smart check pkgname-1.0
smart check pkgname-1.0-1
smart check pkgname1 pkgname2
```

8. smart fix

Dieser Befehl wird versuchen alle Abhängigkeiten der installierten Pakete, die zu den angegebenen Paketen gehören aufzulösen. Beachte, dass die angegebenen Pakete evtl. Installiert sein können oder nicht. Wenn kein Paket angegeben ist, werden alle installierten Pakete geprüft.

Optionen:

-h, --help	Zeige die Hilfe und beende
--stepped	Ausführung in Einzelschritte aufteilen
--urls	Dumpe benötigte URLs und führe keine Funktion aus
--download	Herunterladen der Pakete und keine Funktion ausführen
--explain	Füge zusätzliche Informationen über Änderungen ein, wenn möglich
-y, --yes	Frage nicht nach Bestätigung

Beispiele:

```
smart fix
smart fix pkgname
smart fix '*kgna*'
smart fix pkgname-1.0
smart fix pkgname-1.0-1
smart fix pkgname1 pkgname2
```

9. smart download

Dieser Befehl lädt eines oder mehrere Pakete herunter

Optionen:

-h, --help	Zeige die Hilfe und beende
--target=DIR	Pakete werden in dem eingestellten Verzeichnis gesichert
--urls	Dump benötigt URLs und lädt nichts herunter
--from-urls	Lade Dateien von der angegebenen URL und/oder von den angegebenen Dateien mit der Liste der URLs herunter.

Beispiele:

```
smart download pkgname
smart download '*kgna*'
smart download pkgname-1.0
```

```
smart download pkgname-1.0-1
smart download pkgname1 pkgname2
smart download pkgname --urls 2> pkgname-url.txt
smart download --from-urls pkgname-url.txt
smart download --from-urls http://some.url/some/path/somefile
```

10. smart clean

Dieser Befehl löscht den Paketinformationsspeicher.

Optionen:

-h, --help Zeige die Hilfe und beende

11. smart search

Hier kann man nach einem Ausdruck suchen, und zwar in dem Namen, in der Zusammenfassung und der Beschreibung der Pakete.

Optionen:

-h, --help Zeigt die Hilfe und beendet

Beispiele:

```
smart search ldap
smart search kernel module
smart search rpm 'package manager'
smart search pkgname
smart search 'pkgn*e'
```

12. smart query

Dieser Befehl fragt die bekannten Pakete auf vielen verschiedenen Wegen ab. Siehe auch: 'search' Befehl.

Optionen:

-h, --help	Zeige die Hilfe und beende
--installed	Beachte nur installierte Pakete
--provides=DEP	Zeige nur Pakete an, die die notwendige Abhängigkeit bereitstellen
--requires=DEP	Zeige nur Pakete an, die die gegebene Abhängigkeit benötigen
--conflicts=DEP	Zeige nur Pakete an, die in Konflikt mit der gegebenen Abhängigkeiten stehen
--upgrades=DEP	Zeige nur Pakete an, die die gegebene Abhängigkeit Upgraden
--name=STR	Zeigt nur Pakete an, die auf den Namen der Eingabe passen
--summary=STR	Zeige nur Pakete an, die auf die Zusammenfassung passen
--description=STR	Zeige nur Pakete an, die auf die eingegebene

Beschreibung passen

--path=STR Zeige nur Pakete an, die den gegebenen Pfad der verfügbaren meta Infos beinhalten

--url=STR Zeige nur Pakete an, die die eingestellten Referenz-URL in den meta Informationen beinhalten

--hide-version Verstecke Paket Version

--show-summary Zeige Paket Zusammenfassungen

--show-provides Zeige Bereitstellungen für die angegebenen Pakete

--show-requires Zeige benötigte Dateien an für die angegebenen Pakete

--show-prerequires Zeige nur ausgewählte benötigte Daten an, die in Vorabhängigkeit stehen

--show-upgrades Zeige Upgrades der eingestellten Pakete an

--show-conflicts Zeige Konflikte der eingestellten Pakete an

--show-providedby Zeige Pakete an, die Abhängigkeit bereitstellen

--show-requiredby Zeige Pakete an, die benötigte Informationen bereitstellen

--show-upgradedby Zeige Pakete an, die bereitgestellte Informationen Upgraden

--show-conflictedby Zeige Pakete an, die in Konflikt mit den bereitgestellten Informationen stehen

--show-priority Zeige Paket Prioritäten an

--show-channels Zeige Channel an, die dieses Paket beinhalten

--show-all Aktiviere alle --show-* Optionen

--format=FMT Ändere Ausgabeformat

--output=FILE Leite Ausgabe auf den eingestellten Dateinamen um

Beispiele:

```
smart query pkgname
smart query '*kgnam*'
smart query pkgname-1.0
smart query pkgname --show-requires
smart query --requires libpkg.so --show-providedby
smart query --installed
smart query --summary ldap
```

13. smart info

Zeigt Informationen der selektierten Pakete.

Optionen:

```
-h, --help Zeige die Hilfe und beende
--urls Zeige URLs
--paths Zeige Pfad Liste
```

Beispiele:

```
smart info pkgname
smart info pkgname-1.0
smart info pkgname --urls --paths
```

14. smart stats

Dieser Befehl zeigt einige Statistiken.

Optionen:

-h, --help Zeige die Hilfe und beende

Beispiele:

smart stats

15. smart channel

Dieser Befehl erlaubt es einem, die Channel zu bearbeiten. Channel werden als Quellen-Information über installierte und verfügbare Pakete benutzt. Abhängig von der Channel Art sind verschiedene Verwaltungsoberflächen vorhanden, um die Beziehungen zwischen dem Betriebssystem und den Herausziehen von Informationen der eingestellten Kanälen zu behandeln.

Die folgenden Channel Arten sind verfügbar:

apt-deb	APT-DEB Repository
apt-rpm	APT-RPM Repository
deb-dir	DEB Verzeichnis
deb-sys	DPKG installierte Pakete
mirrors	Mirror Information
red-carpet	Red Carpet Channel
rpm-dir	RPM Verzeichniss
rpm-hdl	RPM Kopfdatenliste
rpm-md	RPM MetaDaten
rpm-sys	Als RPM installierte Pakete
slack-site	Slackware Repository
slack-sys	Unter Slackware installierte Pakete
up2date-mirrors	Mirror Information (up2date format)
urpmi	URPMI Repository
yast2	YaST2 Repository

Benutze --help-type <type> für mehr Informationen.

Optionen:

-h, --help	Zeige die Hilfe und beende
--add	Argument ist ein alias und ein oder mehrere Schlüssel=wert paare die einen Channel definieren, oder ein Dateiname/UR die auf eine Channel Beschreibung im gleichen Format zeigen wie von --show genutzt, oder ein Verzeichnispfad, in dem eine automatische Erkennung versucht wird.
--set	Argument ist ein alias und ein oder mehrere Schlüssel=wert paare die einen Channel modifizieren
--remove	Argumente sind Channel Aliase die entfernt werden sollen
--remove-all	Remove all existent Channels
--show	Zeige Channel mit Aliasen, oder alle Channel wenn keine Argumente übergeben sind
--edit	Bearbeite Channel mit Editor \$editor

--enable Aktiviere Channel mit eingestelltem alias
--disable Deaktiviere Channel mit eingestelltem alias
-y, --yes Ausführen ohne Nachfrage
--help-type=TYPE Zeige mehr Informationen über die eingestellten arten

Beispiele:

```
smart channel --help-type apt-rpm
smart channel --add mydb type=rpm-sys name="RPM Database"
smart channel --add mychannel type=apt-rpm name="Some repository" \
    baseurl=http://somewhere.com/pub/repos components=extra
smart channel --set mychannel priority=-100
smart channel --disable mychannel
smart channel --remove mychannel
smart channel --show
smart channel --show mychannel > mychannel.txt
smart channel --add ./mychannel.txt
smart channel --add http://some.url/mychannel.txt
smart channel --add /mnt/cdrom
```

16. smart priority

Dieser Befehl ändert die Priorität der Pakete. Pakete mit höherer Priorität werden bevorzugt. Wenn keine Priorität angegeben ist, wird die Channel Priorität verwendet. Die Channel Priorität kann mit 'channel' geändert werden. Standardeinstellung ist 0 und wird auf diesen Wert gesetzt, wenn nichts angegeben ist.

Beachte: Negative Prioritäten müssen mit vorgestelltem '--' eingegeben werden, ansonsten werden sie als Befehl interpretiert!

Optionen:

-h, --help Zeige die Hilfe und beende
--set Setze Priorität
--remove Setze Priorität zurück
--show Zeige Prioritäten
--force Ignoriere Probleme

Beispiele:

```
smart priority --set pkgname 100
smart priority --set pkgname mychannel -- -200
smart priority --remove pkgname
smart priority --remove pkgname mychannel
smart priority --show
smart priority --show pkgname
```

17. smart mirror

Dieser Befehl erlaubt die Bearbeitung der Mirror Server. Mirror Server sind URLs, die eigentlich die selben Informationen enthalten, wie die Originalquelle. Es gibt keine internen Beschränkungen, was Mirror Server enthalten. Wenn eine

Ursprungs URL angesprochen wird und es sind Mirror URLs vorhanden, werden die Mirror URLs auch beachtet. Ob der Mirror angesprochen wird, ist abhängig davon, ob der Mirror bereits angesprochen wurde.

Es wird automatisch die schnellste Verbindung versucht. Wenn ein Fehler auftritt, wird der nächste Mirror ausgewählt.

Zum Beispiel, wenn folgender Mirror eingestellt ist "http://mirror.url/path/" für die Originalquelle "ftp://origin.url/other/path/", und eine Datei in "ftp://origin.url/other/path/subpath/somefile" abgeholt wird, wird auch der Mirror angesprochen, und die Adresse "http://mirror.url/path/subpath/somefile" wird benutzt, wenn der Mirror ausgewählt ist. Beachte, dass strings verglichen und ersetzt werden ohne jegliche Vorbearbeitung, sodass es möglich ist, URLs die in Präfixen von Verzeichnisseinträgen enden zu benutzen.

Optionen:

-h, --help	Zeige die Hilfe und beende
--show	Zeige jetzige Mirror
--add	Füge zu einer Ursprungs-URL eine Mirror-URL hinzu, entweder als paar oder in einer angegebenen Datei/URL im Format von --show
--remove	Entferne in einer Ursprungs URL eine Mirror URL, entweder als paar oder in einer angegebenen Datei/URL im Format von --show
--remove-all	Entferne alle Mirror für die angegebene Ursprungs-URL
--sync=FILE	Synchronisiere Mirror von der angegebenen Datei/URL, so dass die Ursprungsdateien exakte spezifizierte Spiegel haben.
--edit	Bearbeite Mirror mit Editor \$editor
--clear-history	Lösche die Historie für die eingegebenen Ursprünge/Mirror, oder für alle Mirror
--show-penalties	Zeige laufende Fehler für die Ursprünge/Mirror aus der Historie

Beispiele:

```
smart mirror --show
smart mirror --add ftp://origin.url/some/path/ http://mirror.url/path/
smart mirror --remove ftp://origin.url/some/path/ http://mirror.url/path/
smart mirror --add http://some.url/path/to/mirrors.txt
smart mirror --sync http://some.url/path/to/mirrors.txt
smart mirror --clear-history ftp://origin.url/some/path/
smart mirror --clear-history ftp://mirror.url/path/
smart mirror --clear-history
```

18. smart flag

Dieser Befehl setzt, entfernt oder zeigt die Paket-Zeiger. Paket-Zeiger(Flags) stimmen das Verhalten einiger Algorithmen ab, wenn Pakete miteinander arbeiten

Momentan bekannte Flags sind:

lock	Pakete werden nicht deinstalliert, wenn sie momentan
------	--

installiert sind, bzw. auch nicht installiert falls sie momentan verfügbar sind.

new Pakete gelten als neue Pakete, wenn das letzte Update gemacht wurde. Dieser Flag wird automatisch gesetzt.

multi-version Es gibt mehrere Versionen dieses Paketes im System. (backend abhängig).

Optionen:

-h, --help Zeige die Hilfe und beende

--set Setze die Flags als paar Flagname/Ziel, wo das Ziel ein Paketname ist, oder Paketname, Beziehung, Version, wie:
lock 'python > 1.0'

--remove Entferne die Flags als paar Flagname/Ziel, wo das Ziel ein Paketname ist, oder Paketname, Verbindung, Version, wie:
lock 'python > 1.0'

--show Zeige Pakete mit den Flags entsprechend der Eingabe oder alle, wenn keine Eingabe erfolgt

--force Ignoriere Probleme

Beispiele:

smart flag --show
smart flag --show new
smart flag --set lock pkgname
smart flag --remove lock pkgname
smart flag --set lock 'pkgname >= 1.0'
smart flag --remove lock 'pkgname >= 1.0'

8. Links

[TIP] Tool (GUI) für smart zur Erzeugung von channel-Daten
<http://www.linux-club.de/ftopic65645.html>

[TIP] Repositories für ATI & NVIDIA Treiber
<http://www.linux-club.de/ftopic65601.html>

[TIP] Sperren von (defekten) Paketen mit smart
<http://www.linux-club.de/ftopic64551.html>

[TIP] eigenes Repo mit createrepo für SuSE DVD iso
<http://www.linux-club.de/ftopic64077.html>

[TIP] smart-cache aufräumen
<http://www.linux-club.de/viewtopic.php?p=361797>

[TIP] smart - Feintuning's
<http://www.linux-club.de/viewtopic.php?p=355405>

[TIP] Mirrors für BuildService-Pakete (z.B. KDE3 + Backports)
<http://www.linux-club.de/viewtopic.php?t=62449>

[TIP] Smart Channel SuSE-10.1 x86 und x86_64

<http://www.linux-club.de/ftopic60190.html>

[Tip] YUM-Metadaten für SuSE-Updates sind da
<http://www.linux-club.de/ftopic58975.html>

[Tip] Übersicht aller smart-Optionen.
<http://www.linux-club.de/viewtopic.php?t=53049>

aptosmart - convert apt-sources to smartpm
Ein Konverter Script um apt sources.list in smart-channels umzuwandeln
<http://www.linux-club.de/viewtopic.php?t=45241>

[Tip] Fehlermeldungen von smart
<http://www.linux-club.de/viewtopic.php?t=44904>

Wiki Seite für SuSE
<http://de.susewiki.org/index.php?title=Smart>

Die smart-Homepage:
<http://labix.org/smart>

Die smart-FAQ Seite:
<http://labix.org/smart/faq>

Die smart FAQ in deutsch
http://wiki.hackerboard.de/index.php/Smart_FAQ_Uebersetzung

smart -- Smart Package Manager mailing list:
<http://lists.labix.org/listinfo.cgi/smart-labix.org>

Pro-Linux: Suse aktualisiert Paketmanager
<http://www.pro-linux.de/news/2006/9815.html>

Pro-Linux: Smart Package Manager soll Konflikte besser lösen
<http://www.pro-linux.de/news/2005/7668.html>

Smart FAQ Uebersetzung
http://wiki.hackerboard.de/index.php/Smart_FAQ_Uebersetzung

9. Viel Spass :-)