

**Table des matières**

**Debian** ..... 3  
  *apt-get* ..... 3  
  *apt-cache* ..... 3  
  *dpkg* ..... 3  
**Red Hat** ..... 4  
  *yum* ..... 4  
  *rpm* ..... 4  
**Troubleshooting** ..... 4  
  This package requires ... ..... 4  
  unrecognized db option: "db3" ignored ..... 5



## Debian

On trouve plusieurs outils pour gérer les packages, ci-dessous les commandes utilisées avec les paramètres les plus courants :

### apt-get

- Mettre à jour la liste des packages d'après les sources définies dans `/etc/apt/sources.list` :

```
apt-get update
```

- Chercher un package contenant *kernel* sur un dépôt :

```
apt-cache search kernel
```

- Pour installer :

```
apt-get install <nom_du_package>
```

- Pour réinstaller un package :

```
apt-get install --reinstall <nom_du_package>
```

- Pour supprimer un package :

```
apt-get remove <nom_du_package>
```

- Pour supprimer un package et ses fichiers de conf associés :

```
apt-get remove --purge <nom_du_package>
```

- Pour mettre à jour tous les packages installés :

```
apt-get -u upgrade test (-s pour simuler et -u pour afficher la liste des paquets qui seront mis à jour)
```

### apt-cache

`apt-cache` gère la base de données des paquets disponibles.

- Lister les packages contenant un pattern dans le nom ou la description :

```
apt-cache search kernel
```

- Lister les packages contenant un pattern dans le nom uniquement :

```
apt-cache search --names-only kernel
```

### dpkg

`dpkg` gère les paquets installés et présents physiquement sur le disque.

- Lister les packages

```
dpkg -l
```

- Lister un package contenant *kernel* :

```
dpkg -l kernel
```

```
dpkg -l | grep kernel
```

- Lister les fichiers du package *toto* :

```
dpkg -L toto
```



Parfois le nom complet est tronqué. On peut utiliser :

```
COLUMNS=142 dpkg -l | grep kernel
```

- Reconfigurer un package en appelant le script lancé lors de l'installation :

```
dpkg-reconfigure <nom_du_package>
```

## Red Hat

### yum

yum permet d'installer, de lister les packages installés et d'interroger le(s) dépôt(s).

- **Lister les packages dispo sur le dépôt :**

```
yum list xorg*
```

- **Lister les packages installées sur la machine :**

```
yum search xorg*
```

- **Installer un package :**

```
yum install gcc.i386
```

- **Trouver le rpm quivabien :**

```
/usr/include/gnu/stubs.h:9:27: error: gnu/stubs-64.h: No such file or directory
```

On peut déterminer avec yum quel(s) package(s) fournisse(nt) le(s) fichier(s) en question :

```
root@serv9001979:~# yum provides \*/gnu/stubs-64.h
glibc-devel-2.5-49.x86_64 : Object files for development using standard C libraries.
Repo      : EL5_U5-x86_64
Matched from:
Filename  : /usr/include/gnu/stubs-64.h
```

```
glibc-devel-2.5-49.el5_5.7.x86_64 : Object files for development using standard C libraries.
Repo      : RHEL5.5-updates
Matched from:
Filename  : /usr/include/gnu/stubs-64.h
```

On détermine le bon kernel avec **uname -r** et on installe le package correspondant.

### rpm

- **Lister les packages :**

```
rpm -qa <nom_du_package>
```

- **Lister les fichiers d'un package :**

```
rpm -ql <nom_du_package>
```

- **Lister les fichiers d'un rpm :**

```
rpm -qlp <nom_du_package>
```

- **Extraire les fichiers d'un package :**

```
rpm2cpio zsh-html-4.2.0-4.EL.4.5.x86_64.rpm | cpio -extract -make-directories
```

## Troubleshooting

### This package requires ...

On peut obtenir cette erreur lors de l'installation d'un rpm :

```
root@server1110636:/tmp# rpm -Uvh EMCpower.LINUX-5.0.0-157.rhel.x86_64.rpm
Preparing... #***** [100%]
This package requires RedHat RHEL4.
error: %pre(EMCpower.LINUX-5.0.0-157.x86_64) scriptlet failed, exit status 1
```

```
error: install: %pre scriptlet failed (2), skipping EMCpower.LINUX-5.0.0-157
```

Pourtant tout est OK, on est bien en RHEL4, les pré-requis sont OK, etc. Utiliser `rpm --noscripts` n'apportera rien dans ce cas. Par contre on peut lister les scripts de pré-install qui sont effectués pour voir où la vérification est incorrecte.

```
rpm -q --scripts -p EMCpower.LINUX-5.0.0-157.rhel.x86_64.rpm > test
```

On cherche la fonction qui gêne :

```
root@server1110636:/tmp# grep check_vendor_rev -A 5 test
check_vendor_rev()
{
version=`sed -n 's/^.*release \([0-9][0-9]*\).*\/\1/p' /etc/issue`
test "$1" = "$version"
}

--
check_vendor_rev $EXPECTED_OS_REV
if [ $? != 0 ]; then
    pre_error 'This package requires $VENDOR_NAME $VENDOR_OS_NAME.'
    cleanup_error_exit
fi
```

Le fichier `/etc/issue` est parsé pour checker si on a bien "release X" dans le fichier. Or ce n'est pas forcément le cas. Il suffit de rajouter la ligne suivante :

```
Welcome to Red Hat Enterprise Linux AS release 4 (Nahant Update 5)
```

qui est aussi normalement présente dans le fichier `/etc/motd`. Il suffit de rajouter cette ligne dans le fichier `/etc/issue`. Ensuite on relance l'install :

```
root@parcl1110636:/tmp# rpm -ivh EMCpower.LINUX-5.0.0-157.rhel.x86_64.rpm
Preparing...      ##### [100%]
 1:EMCpower.LINUX  ##### [100%]
All trademarks used herein are the property of their respective owners.
NOTE:License registration is not required to manage the CLARiion AX series array.
```

#### unrecognized db option: "db3" ignored

Il faut réinstaller le package `rpm` sauf que la commande `rpm` ne marche plus. On récupère le package rpm de la version RedHat quivabien puis :

```
mkdir -p /tmp/toto
cd /tmp/toto
rpm2cpio rpm-XXX.rpm | cpio -dim
find . -type d -exec chmod 755 {} \;
tar cf - ./usr ./etc | (cd /; tar xvf -)
```

From:  
<https://unix-bck.ndlp.info/> - Where there is a shell, there is a way

Permanent link:  
[https://unix-bck.ndlp.info/doku.php/informatique:nix:linux:linux\\_gestion\\_packages](https://unix-bck.ndlp.info/doku.php/informatique:nix:linux:linux_gestion_packages)

Last update: 2021/09/13 15:08