

# Commandes utiles pour l'administration des paquets Debian

**Auteur:** Jérémie DECOCK  
**Contact:** [jd.jdhp@gmail.com](mailto:jd.jdhp@gmail.com)  
**Version:** 1  
**Date:** 22 septembre 2015  
**Licence:** [Creative Commons 4.0 \(CC BY-SA 4.0\)](https://creativecommons.org/licenses/by-sa/4.0/)

Ce document liste quelques commandes utiles pour l'administration des paquets Debian. Ces commandes s'appliquent aux système Gnu/Linux [Debian](#) et à ses dérivés tels qu'[Ubuntu](#), [Linux Mint](#) ou encore [Raspbian](#)<sup>1</sup>.

Les commandes de base sont brièvement décrites dans la première partie de l'article. La seconde partie présente des commandes plus "avancées".

## **Note**

Nous ne parlerons pas ici des outils graphiques tels que [Synaptic](#). Si vous êtes intéressés par Synaptic, vous trouverez de l'aide sur <http://doc.ubuntu-fr.org/synaptic> et <https://wiki.debian.org/fr/Synaptic>.

# Sommaire

<b>1</b>	<b>Avant-propos</b>	<b>3</b>
<b>2</b>	<b>Rappel des commandes de base</b>	<b>3</b>
2.1	Définir les dépôts de paquets à utiliser [TODO]	3
2.2	Rechercher le nom d'un paquet	3
2.3	Afficher les informations disponibles sur un paquet [TODO]	4
2.4	Installer un ou plusieurs paquets	4
2.5	Supprimer un ou plusieurs paquets	4
2.6	Mettre à jour tous les paquets du système	5
2.7	Nettoyer le cache d'apt [TODO]	5
2.8	La commande <code>aptitude</code>	5
<b>3</b>	<b>Commandes plus "avancées"</b>	<b>6</b>
3.1	Télécharger un paquet sans l'installer	6
3.2	Extraire le contenu d'un paquet téléchargé	6
3.3	Afficher la liste des fichiers contenus dans un fichier <code>.deb</code>	7
3.4	Installer un fichier <code>.deb</code> [TODO]	7
3.5	Télécharger le code source d'un paquet [TODO]	7
3.6	Découvrir à quel paquet appartient un fichier installé sur le système	7
3.7	Afficher la liste des fichiers installés par un paquet	8
3.8	La commande <code>apt-file</code>	8
3.9	Obtenir et déchiffrer le statut des paquets installés	8
3.10	Obtenir la taille effective d'un paquet [TODO]	9
3.11	Lister les paquets installés triés par taille croissante [TODO]	10
3.12	Sauvegarder la liste des paquets installés sur le système [TODO]	10
3.13	Afficher la liste des dépendances d'un paquet [TODO]	10
3.14	Supprimer le serveur X et toutes ses dépendances [TODO]	10
3.15	<code>Apt-rdepends</code> [TODO]	11
3.16	<code>Deborphan</code> [TODO]	11
3.17	TODO	11
3.18	Convertir un paquet RPM en paquet Debian [TODO]	11
<b>4</b>	<b>À lire également</b>	<b>11</b>
<b>5</b>	<b>Licence</b>	<b>12</b>

# 1 Avant-propos

Les morceaux de texte écrits en police à espacement fixe comme

```
apt-cache search <motif>
```

ou

```
sudo apt-get install <paquets>
```

doivent être tapés dans un [terminal](#); <paquet> doit être remplacé par le nom d'un paquet (par exemple `vlc`), <paquets> par le nom d'un ou plusieurs paquets (séparés par des espaces) et <motif> par un motif de recherche (par exemple un simple terme tel que `python` ou une [expression régulière](#) telle que `^lib.*`). Les commandes préfixées par le mot `sudo` sont uniquement accessibles en mode [administrateur](#).

## 2 Rappel des commandes de base

### 2.1 Définir les dépôts de paquets à utiliser [TODO]

TODO: explication dépôts

`/etc/apt/source.list`

### 2.2 Rechercher le nom d'un paquet

La plupart du temps, on ne connaît pas le nom exacte du paquet que l'on souhaite installer ou supprimer. Heureusement il est souvent facile de retrouver le nom d'un paquet à partir de mots clés grâce à l'outil de recherche `apt-cache search`.

On peut ainsi afficher la liste des paquets contenant un mot clé donné dans leur nom avec la commande:

```
apt-cache search -n <motif>
```

Par exemple, si on souhaite installer la bibliothèque logicielle `numpy` pour Python3, nous savons avec à la commande:

```
apt-cache search -n numpy
```

que le paquet à installer s'appelle `python3-numpy` (sur un système *Debian 8*).

On peut aussi supprimer l'option `-n` pour étendre la recherche du motif à la description des paquets. Par exemple, pour chercher tous les paquets contenant le mot `mp3` dans leur nom ou leur champ de description, on utilise la commande:

```
apt-cache search mp3
```

La liste retournée est probablement trop longue pour être vraiment utile. Dans ce cas, on peut affiner la recherche en utilisant plusieurs mots clés:

```
apt-cache search mp3 encoder
```

ou en utilisant des motifs plus sophistiqués tels que:

```
apt-cache search ^lib.*mp3
```

Dans le dernier exemple, `^lib.*mp3` est ce qu'on appelle une *expression régulière* (ou *expression rationnelle*). La présentation des expressions régulières dépasse le cadre de cet article. Pour en savoir plus, je vous invite à consulter [l'article correspondant sur wikipedia](#) ainsi que les livres *Expressions régulières, le guide de survie* de Bernard Desgraupes aux éditions Pearson et *Les expressions régulières par l'exemple* de Vincent Fourmond aux éditions H&K.

## 2.3 Afficher les informations disponibles sur un paquet [TODO]

On peut afficher les informations relatives à un paquet (version, taille, description, auteurs, dépendances, etc.) avec:

```
apt-cache show <paquets>
```

Par exemple:

```
apt-cache show vlc
```

TODO: Notion de dépendances

## 2.4 Installer un ou plusieurs paquets

Avant d'installer un paquet, il est recommandé de mettre à jours la liste des paquets disponibles avec la commande:

```
sudo apt-get update
```

On installe ensuite un ou plusieurs paquets avec:

```
sudo apt-get install <paquets>
```

Par exemple, pour installer [VLC](#):

```
sudo apt-get install vlc
```

Pour installer [VLC](#) et [Libre Office](#):

```
sudo apt-get install vlc libreoffice
```

## 2.5 Supprimer un ou plusieurs paquets

On peut supprimer un ou plusieurs paquets avec:

```
sudo apt-get remove <paquets>
```

La suppression d'un paquet avec `apt-get remove` laisse ses fichiers de configuration sur le système.

Pour supprimer un paquet et les fichiers de configuration qu'il a généré, tapez:

```
sudo apt-get purge <paquets>
```

ou

```
sudo apt-get remove --purge <paquets>
```

Pour supprimer les dépendances paquet devenues inutiles sur le système (i.e. utilisées par aucun autre paquet installé), utilisez la commande:

```
sudo apt-get autoremove
```

ou dans la commande `apt-get remove`:

```
sudo apt-get remove --auto-remove <paquets>
```

## 2.6 Mettre à jour tous les paquets du système

Les paquets sont régulièrement mis à jours pour corriger d'éventuels bugs ou failles de sécurité. La commande `apt-get upgrade` permet d'appliquer toutes les mises à jours disponibles pour les paquets installés sur le système:

```
sudo apt-get update
sudo apt-get upgrade
```

Comme pour `apt-get install`, il est recommandé de mettre à jours la liste des paquets disponibles au préalable avec `apt-get update`.

## 2.7 Nettoyer le cache d'apt [TODO]

Lorsque l'on utilise les commandes `apt-get install` et `apt-get upgrade`, les paquets Debian sont téléchargés et stockés dans `/var/cache/apt/archives` avant d'être installés. Pour diverses raisons, ils sont conservés dans ce répertoire, même après leur installation.

Vous pouvez donc rapidement vous retrouver avec des centaines de mégaoctets de fichiers `.deb` dans `/var/cache/apt/archives`.

Ces fichiers sont inutiles pour la plupart des utilisateurs et ils peuvent être supprimés sans problème avec la commande:

```
sudo apt-get clean
```

## 2.8 La commande `aptitude`

La commande `aptitude` est une alternative efficace <sup>2</sup> aux commandes `apt`. Elle est installée par défaut sur Debian mais pas sur Ubuntu. Vous pouvez l'installer avec la commande suivante:

```
sudo apt-get install <paquets>
```

Le tableau qui suit résume les principales équivalences entre les commandes `apt` et `aptitude`.

<code>apt-get</code>	<code>aptitude</code>
<code>apt-get update</code>	<code>aptitude update</code>
<code>apt-get upgrade</code>	<code>aptitude safe-upgrade</code>
<code>apt-get install &lt;paquets&gt;</code>	<code>aptitude install &lt;paquets&gt;</code>
<code>apt-get remove &lt;paquets&gt;</code>	<code>aptitude remove &lt;paquets&gt;</code>
<code>apt-get remove --purge &lt;paquets&gt;</code>	<code>aptitude purge &lt;paquets&gt;</code>

<b>apt-get</b>	<b>aptitude</b>
apt-get clean	aptitude clean
apt-cache search <motif>	aptitude search <motif>
apt-cache show <paquets>	aptitude show <paquets>

## 3 Commandes plus "avancées"

### 3.1 Télécharger un paquet sans l'installer

Chaque paquet Debian est contenu dans un fichier `.deb`. Ces fichiers sont stockés dans des dépôts (généralement des serveurs web affiliés à Debian, Ubuntu, etc.). C'est de là que viennent les paquets installés avec `apt-get install <paquets>`.

Il est possible de télécharger les paquets Debian provenant de ces dépôts, sans les installer, avec les commandes:

```
apt-get download <paquets>
```

ou

```
aptitude download <paquets>
```

Les paquets téléchargés (fichiers `.deb`) sont placés dans le répertoire courant.

### 3.2 Extraire le contenu d'un paquet téléchargé

Nous avons vu dans la section précédente comment télécharger des paquets Debian depuis les dépôts de votre système. Voyons maintenant leur contenu.

Les paquets Debian sont en fait des *archives* Unix portant l'extension `.deb`. Ainsi, ils sont semblables aux fichiers `.tar` très répandus sur les systèmes Unix ou aux fichiers `.zip` fréquemment utilisés sous Windows.

Tous les paquets Debian contiennent exactement 3 fichiers: `control.tar.gz`, `data.tar.xz` et `debian-binary`.

Ces trois fichiers peuvent être extraits de n'importe quel paquet Debian avec la commande suivante:

```
ar -x <fichier.deb>
```

Les fichiers sont extraits dans le répertoire courant. Deux des fichiers extraits sont eux même des archives:

- `control.tar.gz` est une archive au format `tar` compressé avec l'algorithme *Deflate* (via la commande `gzip`);
- `data.tar.xz` est une archive au format `tar` compressé avec l'algorithme *LZMA* (via la commande `xz`).

On peut extraire leur contenu respectif avec les commandes suivantes:

```
tar -xzvf control.tar.gz  
tar -xJvf data.tar.xz
```

### 3.3 Afficher la liste des fichiers contenus dans un fichier .deb

Si vous voulez obtenir la liste des fichiers et des répertoires contenus dans un fichier `.deb` sans rien extraire, tapez:

```
dpkg -c <fichier.deb>
```

On peut désactiver l'affichage des répertoires avec:

```
dpkg -c <fichier.deb> | grep -v "^d"
```

### 3.4 Installer un fichier .deb [TODO]

La commande `apt-get install` permet uniquement d'installer des paquets stockés sur les dépôts du système. Elle ne permet pas d'installer des fichiers `.deb` stockés localement, hors des dépôts.

Il arrive toutefois de devoir installer un paquet récupérés par exemple sur le web. Pour installer de tels paquets, il faut utiliser:

```
sudo dpkg -i <fichier.deb>
```

Cette commande suppose que les autres paquets requis pour le bon fonctionnement de `<fichier.deb>` soient déjà installées sur le système. Contrairement à `apt-get install`, la commande `dpkg -i` n'installera pas elle même ces *dépendances*.

### 3.5 Télécharger le code source d'un paquet [TODO]

TODO: différence entre paquet binaire et paquet source

On peut très facilement étudier le code source de n'importe quel paquet Debian à l'aide de la commande suivante:

```
apt-get source <paquet>
```

Le code source est placé dans le répertoire courant.

Il n'existe pas d'équivalent à cette commande pour aptitude.

### 3.6 Découvrir à quel paquet appartient un fichier installé sur le système

Il est souvent très utile de savoir quel paquet à installé un exécutable donné sur notre système ou de savoir quel paquet est à l'origine de tel ou tel fichier de configuration, de données, etc.

On peut facilement retrouver le nom du paquet qui a installé un fichier présent sur le système avec:

```
dpkg -S <fichier>
```

Par exemple:

```
dpkg -S /etc/init.d/networking
```

nous apprend que le fichier `/etc/init.d/networking` a été installé par le paquet `ifupdown` (sur *Debian 8* du moins).

Pour découvrir directement à quel paquet appartient une commande du système, tapez:

```
dpkg -S $( which <commande> )
```

Par exemple:

```
dpkg -S $( which vlc )
```

nous apprend que la commande `vlc` (i.e. le fichier `/usr/bin/vlc`) a été installé par le paquet `vlc-nox` (sur *Debian 8*).

Notez que `which <commande>` ne fait que retourner l'emplacement d'une commande sur le système.

### 3.7 Afficher la liste des fichiers installés par un paquet

On peut obtenir la liste des fichiers installés par un paquet avec:

```
dpkg -L <paquets>
```

### 3.8 La commande apt-file

Les commandes `dpkg -L` et `dpkg -S` présentées ci-dessus ne tiennent compte que des paquets déjà installés sur le système.

Dans certains cas il peut être utile d'effectuer ces recherches sur l'ensemble des paquets disponibles sur le dépôt et non pas seulement sur les paquets installés. C'est ce que permet la commande `apt-file`.

On peut installer `apt-file` et mettre à jours sa base de données avec:

```
sudo apt-get install apt-file  
apt-file update
```

On peut ensuite découvrir à quel paquet appartiendrait un fichier installé sur le système avec:

```
apt-file search -F <paquets>
```

et afficher la liste des fichiers qui seraient installés par un paquet avec:

```
apt-file list -F <paquets>
```

`apt-file` nécessite d'être mis à jours régulièrement avec `apt-file update` pour tenir compte des modifications opérées sur les dépôts de paquets.

### 3.9 Obtenir et déchiffrer le statut des paquets installés

On peut obtenir le statut de tous les paquets installés avec la commande:

```
dpkg -l
```

ou, si on souhaite supprimer l'entête retournée:

```
dpkg -l | tail -n +6
```

La première colonne de chaque ligne est formée de 2 ou 3 lettres. Elle traduit le statut du paquet correspondant.

La première lettre définit l'état souhaité du paquet:



- **u** ... Inconnu
- **i** ... Installer
- **r** ... Désinstaller
- **p** ... Purger (supprimer le programme et les fichiers de configuration)
- **h** ... Ignorer ce paquet (marqué *hold*)

La deuxième lettre définit l'état actuel du paquet:

- **n** ... Le paquet n'est pas installé sur le système
- **i** ... Le paquet est installé (correctement dépaqueté et configuré)
- **c** ... Seuls les fichiers de configuration sont installés
- **u** ... Le paquet est dépaqueté mais n'est pas configuré
- **f** ... Le paquet est partiellement configuré (la configuration a échoué)
- **h** ... Le paquet est partiellement installé (l'installation a échoué)
- **w** ... Le paquet attend l'exécution d'une action différée qui est à la charge d'un autre paquet (*triggers-awaited*)
- **t** ... Une action différée de ce paquet a été activée, il reste à l'exécuter (*triggers-pending*)

La troisième lettre signale une éventuelle erreur (cette lettre est généralement absente):

- **r** ... Le paquet est cassé et sa réinstallation est nécessaire

Sur un système sain (sauf cas particuliers) la plupart des paquets doivent avoir le statut **ii**. On peut afficher la liste des paquets qui n'ont pas le statut **ii** avec:

```
dpkg -l | tail -n +6 | grep -v "^ii "
```

Plutôt que d'afficher le statut de tous les paquets installés, on peut afficher uniquement le statut d'un ou plusieurs paquets donnés avec:

```
dpkg -l <paquets> | tail -n +6
```

### 3.10 Obtenir la taille effective d'un paquet [TODO]

On peut obtenir une approximation de la taille totale des fichiers installés par un paquet en regardant le champ "*Installed-Size*" dans le résultat retourné par la commande `apt-cache show <packet>`.

Mais ce n'est pas très pratique car `apt-cache show <packages>` retourne plein d'autres informations sur le paquet.

TODO Plutôt que d'ajouter `grep` à la commande précédente en écrivant:

```
``apt-cache show <packages> | grep "Installed-Size"``
```

profitons-en pour utiliser une commande spécialement faite pour ça:

```
dpkg-query -Wf '$ {Installed-Size} \t $ {Package} \n' <paquets>
```

Quelle que soit la méthode utilisée pour récupérer sa valeur, la taille décrite dans le champ "*Installed-Size*" est définie en *kibiocet*. Un kibiocet (noté Kio) correspond à 1 024 octets, c'est à dire à peu près un kilooctet (noté ko).

TODO <https://www.debian.org/doc/debian-policy/ch-controlfields.html#s-f-Installed-Size>

On peut comparer le résultat obtenu avec la valeur exacte retourné par cette commande (beaucoup moins pratique à utiliser):

```
du -ch $(for FILE in $(dpkg -L <paquet>) ; do \  
    if [ -f "${FILE}" ] ; then echo "${FILE}" ; fi ; done)
```

### 3.11 Lister les paquets installés triés par taille croissante [TODO]

```
dpkg-query -Wf '${Installed-Size}\t${Package}\n' | sort -n
```

Ou avec wajig (sudo apt-get install wajig):

```
wajig large
```

### 3.12 Sauvegarder la liste des paquets installés sur le système [TODO]

```
dpkg -l
```

```
dpkg --get-selections > LIST_FILE
```

```
dpkg --set-selections < LIST_FILE  
...
```

ne fait pas la distinction entre les paquets dont l'installation a été explicitement demandée par l'utilisateur et les dépendances automatiquement installées.

Il peut être préférable de ne lister que les paquets ...:

```
(grep "^\[INSTALLÉ\]" /var/log/aptitude & zgrep "^\[INSTALLÉ\]" /var/log/aptitude*.gz) |  
aptitude install $(tr '\n' ' ' < ${DIR_BASE}/${FILE})
```

TODO: supprimer le i386 dans cette commande... TODO: cette commande ne marche que pour les paquets installés avec aptitude...

### 3.13 Afficher la liste des dépendances d'un paquet [TODO]

```
apt-cache dotty apache2 | dot -T png | display
```

### 3.14 Supprimer le serveur X et toutes ses dépendances [TODO]

```
sudo apt-get remove --auto-remove --purge "libx11-.*"
```

Réfléchissez bien avant de taper cette commande...

### 3.15 Apt-rdepends [TODO]

...

### 3.16 Deborphan [TODO]

...

### 3.17 TODO

lister les versions disponibles pour un paquet:

```
apt-cache madison
```

afficher des statistiques sur ...:

```
apt-cache stats
```

lister les dépendances d'un ou plusieurs paquets:

```
apt-cache depends <paquets>
apt-cache depends --recursive <paquets>
apt-cache depends --installed <paquets>

apt-cache rdepends <paquets>
apt-cache rdepends --recursive <paquets>
apt-cache rdepends --installed <paquets>

apt-cache showsrc <motif>

apt-cache dotty <paquets>

apt-cache xvcg <paquets>
```

### 3.18 Convertir un paquet RPM en paquet Debian [TODO]

```
alien -d <paquet.rpm>
```

## 4 À lire également

La documentation de référence: <http://www.debian.org/doc/manuals/debian-reference/ch02.fr.html>

## 5 Licence



Ce document est distribué selon les termes de la licence [Creative Commons 4.0 \(CC BY-SA 4.0\)](#).

- 
- 1 Le système officiel du [RaspberryPi](#).
  - 2 `aptitude` est notamment réputé mieux gérer les conflits de dépendances qu'`apt`.