

apt/dpkg

- Création par : 👤 [lagrenouille](#)
- Objet : du tuto [gestion-des-paquets-.deb-avec-apt-et-dpkg](#)
- Niveau requis :
[débutant](#), [avisé](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi:
[à-placer](#)
- Testé par : 🛠️ [Fix Me!](#)

Introduction



Gestion des Paquets .deb avec apt et dpkg

APT est une interface de gestion des paquets des systèmes Linux Debian.

apt est le programme de base permettant d'installer, de mettre à jour ou de supprimer des paquets.

le fichier `/etc/apt/sources.list` vous donne la liste des sources des paquets.

- `deb` : Adresse de paquets binaires (compilés)
- `deb-src` : Adresse de paquets sources
- `main` : Les paquets principaux

`contrib` : Les paquets plus vraiment supportés

`non-free` : Les paquets de logiciels non libres (propriétaires)

-Le dépôt `non-free` a donc été créé entre autre pour pouvoir disposer de ces firmwares ne remplissant pas les conditions pour être considéré logiciel libre.

-Le dépôt `contrib` contient du logiciel libre dépendant de logiciels ou de ressources non-libres (son installation va donc généralement de paire avec `non-free`).

Nouveauté dans debian 12 “bookworm”:

non-free-firmware

La plupart des paquets de microprogrammes non libres ont été déplacés de non-free vers non-free-firmware en préparation de la publication de Debian Bookworm.

Cette séparation nette permet de construire les images d'installation officielles en utilisant les paquets des sections main et non-free-firmware,

tout en évitant ceux des sections contrib et non-free. Ces images permettent donc d'installer un système avec uniquement des paquets de main et

non-free-firmware, et sans utiliser contrib ni non-free.

Comme décrit dans Section 2.2, « Sections de l'archive », les paquets de microprogrammes non libres sont maintenant fournis à partir d'un composant de l'archive qui leur est dédié, appelé non-free-firmware.

Afin d'être sûr que les paquets de microprogrammes non libres bénéficient de mises à niveau appropriées, des modifications dans la configuration d'APT sont nécessaires. En supposant que le composant non-free a été seulement ajouté au fichier sources-list d'APT pour installer les microprogrammes l'entrée de la liste de source d'APT pourrait ressembler à cela :

```
deb https://deb.debian.org/debian bookworm main non-free-firmware
```

avis aux débutants Pour passer en root, il vous faut maintenant taper : `$ su -` ou `su -l` En quelque sorte le « su » d'avant ne respectait pas la norme posix, alors que maintenant, il la respecte.

Voir explication ici :

<https://debian-facile.org/viewtopic.php?id=24901>

Les commandes root (administrateur) sont précédées de `#`

les commandes simples utilisateurs sont précédées de `$`

```
# apt update
```

L'instruction update va rechercher les mises à jour disponibles pour votre système et vos programmes installés en se basant sur les sources définies dans `/etc/apt/sources.list`

Installation

`apt install apt dpkg`, installation automatique, par défaut sous debian

Utilisation

```
# apt upgrade
```

L'option upgrade installe les mises à jour identifiées avec apt update sans supprimer les paquets installés et

sans en installer de nouveaux. S'il y a de nouvelles dépendances à installer, les paquets présents ne sont pas mis à jour.

```
Cette commande vous propose de voir les paquets qui vont être mis à jour avec.
```

```
# apt list - -upgradable
```

Apt search peut être utilisé en root et en simple user, ceci pour rechercher un paquet.

```
# apt search nom de paquet
```

pour n'avoir que ce paquet afficher, mettre ^paquet-deb\$

```
# apt search ^nom du paquet$
```

Parfois le paquet n'existe plus en stable mais il peut exister en oldstable, un testing ou en sid.

Pour le savoir vous devez installer le paquet deb « devscripts »

```
# apt install devscripts
```

la commande sera, en root

```
# rmadison « paquet-deb
```

Exemple :

```
# rmadison libavcodec-extra
```

```
libavcodec-extra | 6:11.12-1 deb8u1 | oldoldstable | all
libavcodec-extra | 7:3.2.14-1 deb9u1 | oldstable | all
libavcodec-extra | 7:4.1.6-1 deb10u1 | stable | all
libavcodec-extra | 7:4.3.2-0+deb11u2 | testing | amd64, arm64, armel, armhf,
i386, mips64el, mipsel, ppc64el, s390x
libavcodec-extra | 7:4.3.2-0+deb11u2 | unstable | amd64, arm64, armel,
armhf, i386, mips64el, mipsel, ppc64el, s390x
libavcodec-extra | 7:4.3.2-2 | experimental | armel, mipsel
libavcodec-extra | 7:4.4-1 | experimental | amd64, arm64, armhf, i386,
mips64el, ppc64el, s390x
```

Pour chercher une dépendances

apt depends paquet-deb Si vous avez un paquet "Recommande" si vous l'avez installé et que vous décidez de le virer, tester avant avec `apt remove --simulate paquet-deb`

Pour installer un paquet # `apt install paquet-deb`

`apt install apt-listbugs` vous informera des bugs à l'installation du paquet lors des mises à jour ou des installations.

Pour supprimer un paquet # `apt remove - --purge paquet-deb`

Pour rechercher de quel paquet provient un fichier : # `dpkg -S fichier`

La commande `show` vous affiche toutes les informations détaillées sur le(s) paquet(s)

exemple :

```
$ apt show gedit
```

```
Package : gedit
Version : 3.30.2-2
Priority : optional
Section : gnome
Maintainer : Debian GNOME Maintainers
Installed-Size : 1980 kB
Depends : python3:any (>= 3.2 ), python3.7, libatk1.0-0 (>= 1.12.4), libc6 (>= 2.4), libcairo-gobject2 (>= 1.10.0)
libcairo2 (>= 1.2.4), libenchant1c2a (>= 1.6.0), libgdk-pixbuf2.0-0 (>= 2.25.2), libgirepository-1.0-1 (>= 0.9.3)
libglib2.0-0 (>= 2.44), libgspell-1-1 (>= 1.0.0), libgtk-3-0 (>= 3.21.4), libgtksourceview-3.0-1 (>= 3.21.5)
libpango-1.0-0 (>= 1.14.0), libpangocairo-1.0-0 (>= 1.14.0), libpeas-1.0-0 (>= 1.14.1), libx11-6, libxml2 (>= 2.7.4)
gir1.2-gtk-3.0 (>= 3.21.3), gir1.2-gtksource-3.0 (>= 3.21.2), gedit-common (>= 3.30), gedit-common (<= 3.0)
python3-gi-cairo (>= 3.0), gir1.2-peas-1.0, gir1.2-glib-2.0, gir1.2-pango-1.0, iso-codes
Recommends : zenity, yelp
Suggests : gedit-plugins
Homepage :
```

Description : Éditeur de texte officiel de l'environnement de bureau GNOME

gedit est un éditeur de texte qui offre la plupart des fonctions de base que vous attendez d'un éditeur de texte, avec en plus des fonctionnalités rarement trouvées dans les éditeurs simples.

C'est une application avec une interface graphique qui supporte l'édition de plusieurs fichiers texte dans une seule fenêtre (quelquefois connu comme onglets ou MDI).

gedit gère n'importe quelle langue pour les textes grâce à l'utilisation du codage Unicode UTF-8 pour les fichiers édités. Les fonctionnalités de base comprennent la coloration syntaxique du code source, l'auto-indentation et le support pour l'impression et l'aperçu avant impression.

gedit est également extensible par le biais de son système de modules, parmi lesquels on peut citer le support pour la vérification orthographique, la comparaison de fichiers, la visualisation de changements dans les versions CVS et l'ajustement des niveaux d'indentation.

Dans votre distribution, pour avoir la liste de paquets qu'il est possible de mettre à jour.

```
# apt install apt-file
```

```
# apt-file update
```

```
# apt-file search paquet-deb
```

En cas de problèmes divers : vous pouvez déjà aller voir si des paquets posent problèmes.

```
# dpkg -l | grep -v ^ii
```

```
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder  
| État=Non/Installé/fichier-Config/dépaqUeté/échec-config/H=semi-  
installé/W=attend-traitement-déclenchements  
|/ Err ?=(aucune)/besoin Réinstallation (État,Err : majuscule=mauvais)  
|/ Nom Version Architecture Description
```

+++-

dans ce cas, tout va bien, sinon il y aura des réponses :

voir explications en fin d'article (dpkg-query).

Les fichiers débutants avec l'option rc sont des paquets supprimés et mal purgés

Pour les purger

```
# dpkg - -purge paquet
```

Si il y en à beaucoup

```
# dpkg -l | grep '^rc' | awk 'print $2' | xargs dpkg -P  
ou  
# dpkg -l | awk '/^rc/ print $2
```

Si vous utilisez aptitude

```
# aptitude search ' c '
```

```
# aptitude purge ' c '
```

pour afficher la liste des paquets obsolètes (attention, certains paquets que vous voulez garder, mais dont vous vous servez très peu. peuvent être listés dans cette commande.

```
# aptitude search ' o '
```

Enlever les anciennes versions obsolètes avec :

```
# apt-get autoclean
```

Supprimer les paquets du cache avec

```
# apt-get clean
```

Si vous êtes coincés et que ça n'avance pas, essayez l'option `-fix-broken`

```
# apt install -f
```

cette option essaye de réparer un système dont les dépendances sont défectueuses

Cette option, utilisée avec `apt install` ou `apt remove`...

puis tentez : `# apt full-upgrade` ce qui installe les versions candidates des paquets installés en installant ou en supprimant d'autres paquets si nécessaire, tiens compte des dépendances.. vous obtiendrez toutes les dernières mises à jour sur votre système et du noyau.

Lors du passage à une version supérieure de votre distribution, mettez tout vos paquets à jour

Puis ré-écrivez votre fichier `source.list`

ensuite mettez tout à jour

```
# apt dist-upgrade
```

ou

```
# apt full-upgrade
```

Problèmes possibles pendant une mise à niveau

La mise à niveau échoue avec « Impossible de faire une configuration immédiate ».

Dans certains cas, l'étape `apt full-upgrade` peut échouer après le téléchargement des paquets avec :

Si cela se produit, exécuter .

```
#apt full-upgrade -o APT::Immediate-Configure=0
```

devrait permettre la mise à niveau.

Pour voir la liste des métapaquets `linux-image` disponibles, exécutez :

```
# dpkg -l "linux-image*" | grep ^ii | grep -i meta
```

```
# apt-cache policy
```

permet de consulter les priorités des différentes sources de paquets ainsi que celles des paquets qui

bénéficient d'un traitement particulier

```
# apt-cache dumpavail
```

qui affiche les en-têtes de toutes les versions disponibles de tous les paquets

```
# apt-cache pkgnames
```

affiche une liste de tous les paquets existants dans la mémoire cache.

```
# apt-get autoremove
```

pour supprimer proprement les paquets ainsi que les bibliothèques dépendantes. vérifier avant de confirmer

Pour obtenir un rapport plus détaillé pour un paquet particulier, lancez la commande :

```
# dpkg - -status packagename
```

Pour afficher des informations détaillées des versions disponibles pour un paquet et les paquets ayant des dépendances inverses sur lui :

```
# apt-cache showpkg paquet
```

Reinstaller un paquet si nécessaire

```
# apt install - -reinstall
```

```
# apt-cache gencaches
```

À installer si vraiment besoin.

apt install apt-show-source Afficher les informations des paquets source

```
apt-install apt-show-versions
```

afficher les versions de paquet disponibles dans la distribution

```
apt install apt-src
```

gestionnaire des paquets source Debian

```
apt install apt-transport-https
```

transport de téléchargement https pour APT

```
apt install apt-transport-s3
```

transport APT pour les dépôts privés sur AWS S3

```
apt install apt-transport-spacewalk
```

transport APT pour communiquer avec des serveurs Spacewalk

```
apt -install apt-transport-tor
```

transport APT pour les téléchargements de paquets anonymes via Tor

```
apt install apt-utils
```

utilitaires liés à la gestion de paquets

```
apt install apt-venv
```

environnement virtuel apt

```
apt install apt-xapian-index
```

outils de recherche et de maintenance dans un index Xapian des paquets Debian

on peut lister les paquets de transition avec :

```
aptitude search smetapackages
```

```
aptitude search soldlibs
```

```
aptitude search smetapackages soldlibs
```

pour voir les paquets de priorité « important

```
aptitude search 'pimportant'
```

Pour savoir pourquoi un paquet est installé, tu peux utiliser : `aptitude why $paquet`

Vous pouvez lister les paquets installés

```
$ dpkg -l
```

les mettre dans un fichier texte

```
$ dpkg -l > liste-paquets.txt
```

```
dpkg -l | grep vim
```

ou n'en voir qu'un

```
$ dpkg -l | grep apache2
```

```
$ man apt (manuel d'apt)
```

dpkg-query - Un outil pour interroger la base de données de dpkg il affiche les informations sur les paquets connus par la base la commande citée plus haut « dpkg -l | grep -v ^ii » en fait partie

```
# dpkg-query -l apache2
```

```
# dpkg-query -l apache2
```

```
Souhait=inconnU/Installé/suppRimé/Purgé/H=à garder
```

```
| État=Non/Installé/fichier-Config/dépaqueté/échec-config/H=semi-
installé/W=attend-traitement-déclenchements
```

```
|/ Err?=(aucune)/besoin Réinstallation (État,Err: majuscule=mauvais)
```

```
||/ Nom                Version                Architecture Description
ii  apache2              2.4.56-1~deb11u1      amd64          Apache HTTP Server
```

ii apache2 2.4.38-3+deb10u4 amd64 Apache HTTP Server Les trois premières colonnes affichées montrent dans l'ordre : l'action souhaitée, l'état du paquet et les erreurs.

u = Inconnu (Unknown)

i = Installer (Install)

h = Garder (Hold)

r = Supprimer (Remove)

p = Purger (Purge)

État du paquet :

n = Non installé (Not-installed)

c = Fichiers de configuration (Config-files)

H = Semi-installé (Half-installed)

U = Décompressé (Unpacked)

F = Semi-configuré (Half-configured)

W = Attente de déclenchements (Triggers-awaiting)

t = Déclenchements en attente (Triggers-pending)

i = Installé (Installed)

cruft propose ainsi un traitement un peu plus élaboré qu'un simple balayage du système de fichiers, couplé à des requêtes sur la base dpkg.

```
# apt search cruft
```

```
En train de trier... Fait
Recherche en texte intégral... Fait
cruft/stable 0.9.38 amd64
identifier les fichiers systèmes devenus inutiles
```

Attention, il vous faut regarder à deux fois avant de supprimer quelque fichier que ce soit ! Man cruft est fortement conseillé.

L'outil debsums

```
$ apt search debsums
```

```
En train de trier... Fait
Recherche en texte intégral... Fait
debsums/stable,stable,now 2.2.3 all [installé]
outil de vérification des fichiers de paquets installés grâce à des sommes
de contrôle MD5
```

Les options de debsums.

- a,
Vérifie aussi les fichiers de configuration (ignorés par défaut).
- e
Vérifie seulement les fichiers de configuration.
- c
Envoie la liste des fichiers modifiés sur la sortie standard (ceci implique -s).
- l
Liste les paquets (ou fichiers .deb) qui n'ont pas de somme de contrôle MD5.
- s
Affiche seulement les erreurs.
- m, -md5sums=fichier
Lit la liste des sommes de contrôle dans fichier.
- r, - -root=répertoire
Répertoire racine à vérifier (/ par défaut).
- d, - -admindir=répertoire
Répertoire administratif de dpkg (/var/lib/dpkg par défaut).
- p, - -deb-path=répertoire[:répertoire...]
Répertoires où chercher les fichiers .deb dérivés du nom du paquet (répertoire courant par défaut).

Notes: Quand on utilise apt-get autoclean ou qu'on ne vide pas du tout le cache, /var/cache/apt/archives est un paramètre intéressant pour cette option. La commande : apt-get - -reinstall -d install `debsums -l` peut être utilisée pour remplir à nouveau le cache avec les paquets qui ne sont pas déjà dans le cache.

aptitude est une interface «ncurses» en ligne de commande basée sur plusieurs bibliothèques Apt, basé sur du texte et se lance à partir d'un terminal

Qu'est ce que Ncurses ? Retour ligne automatique ncurses est une bibliothèque qui fournit des fonctions de définition de touches du clavier, de couleurs d'écran et permet l'affichage de plusieurs fenêtres sans recouvrement sur un terminal texte.

Voulez vous que vos programmes disposent d'une interface couleur sur un terminal texte ?

Ncurses est une bibliothèque qui offre ce type de fonctionnalité pour les terminaux en mode texte. Ncurses peut Retour ligne automatique utiliser tout l'écran,, créer et gérer des fenêtres, en permettant l'utilisation de la souris Retour ligne automatique Utiliser les touches de fonction du clavier.Retour ligne automatique comme midnight-commander, ou vi Mettre à jour la liste des paquets :

```
# aptitude update
```

Mettre à jour les paquets : Retour ligne automatique

```
# aptitude safe-upgrade
```

installer vim-noxRetour ligne automatique

```
# aptitude install vim-nox
```

Supprimer vim-noxRetour ligne automatique

```
# aptitude remove vim-nox
```

```
# aptitude purge vim-nox
```

Pour mettre en œuvre les mises à jour de sécurité Retour ligne automatique

```
# aptitude updateRetour ligne automatique
```

```
# aptitude full-upgrade
```

unattended-upgrade

```
# apt search unattended-upgrade
```

```
En train de trier... Fait
Recherche en texte intégral... Fait
apt-config-auto-update/stable,stable 2.1 all
configuration APT pour mettre à jour le cache automatiquement
unattended-upgrades/stable,stable 1.11.2 all
installation automatique des mises à jour de sécurité
```

le paquet apt-listchanges - Affiche les nouvelles entrées du journal des modifications des paquets de l'archive Debian, il affiche ce qui a

changé dans la nouvelle version d'un paquet Debian par rapport à la version actuellement installée sur le système.

Il réalise cela en extrayant les entrées appropriées des fichiers NEWS.Debian et changelog[.Debian], qui se trouvent habituellement

dans le répertoire /usr/share/doc/paquet, de l'archive d'un paquet Debian. (voir man)

Le fichier de configuration par défaut du package unattended-upgrades se trouve dans `/etc/apt/apt.conf.d/50unattended-upgrades`

Pour activer les mises à jour sans surveillance, vous devez vous assurer que le fichier de configuration `/etc/apt/apt.conf.d/20auto-upgrades`

contient au moins les lignes suivantes.

```
# éditeur /etc/apt/apt.conf.d/20auto-upgrades
```

```
APT::Periodic::Update-Package-Lists "1" ;  
APT::Periodic::Mise à niveau sans surveillance "1" ;
```

Le fichier `/etc/apt/apt.conf.d/20auto-upgrades` peut être créé manuellement ou en exécutant la commande suivante en tant que root :

```
# dpkg-reconfigure -plow unattended-upgrades
```

```
# apt install unattended-upgrades apt-listchanges
```

À propos du message :

« package linux-image-amd64 upgradable but fails to be marked for upgrade Packages that will be upgraded : »

en gros ça dit :

Démarrage du script de mise à niveau automatique

Les origines permises sont : `origin=Debian,codename=buster,label=Debian-Security`

Le paquet `linux-image-amd64` peut être mis à niveau mais n'est pas marqué pour la mise à niveau () `apt upgrade` n'ajoute pas de nouveaux packages. Le `linux-image-amd64` est un méta-paquet qui dépend souvent de l'installation d'un nouveau paquet qui est la version la plus récente du noyau.

Quoi qu'il en soit, si vous souhaitez passer à la dernière version du noyau, utilisez `apt dist-upgrade`, ce qui peut-être dangereux `dist-upgrade` peut être assez dangereux, car il peut supprimer des packages pour résoudre des situations de dépendance complexes.

Contrairement à vous, APT n'est pas toujours assez intelligent pour savoir si ces ajouts et suppressions pourraient faire des ravages.

Trouvé cette explication sur le net. à potasser avant toute chose Si la mise à niveau nécessite la suppression d'un autre package ou l'installation d'un nouveau package, le package sera "conservé". Les packages actuellement installés avec les nouvelles versions disponibles sont récupérés et mis à niveau ; en aucun cas les packages actuellement installés ne sont supprimés, ou les packages non déjà installés ne sont récupérés et installés. Pour surmonter cela, vous pouvez faire

```
# apt --with-new-pkgs upgrade
```

Cela permet d'installer de nouveaux packages. Il vous indiquera quels packages seront installés et vous demandera avant de procéder à l'installation.

Le rapport de base est qu'il apt upgrade ne mettra pas à niveau les packages pour lesquels le gestionnaire de packages souhaite supprimer et réinstaller le package.

Cela se produit beaucoup dans les correctifs de sécurité car, souvent, ce sont en fait des bibliothèques précurseurs de la compilation finale distribuable qui ont été corrigées.

Half-Installed »(à moitié installé) ou « Failed-Config »(Échec de la configuration), ainsi que ceux qui sont dans un état d'erreur.

```
# dpkg - -audit
```

les paquets installés

```
# dpkg - -get-selections "*" > /paquets-actuels.txt
```

Si vous désirez vérifier quels paquets étaient bloqués pour apt, il vous faudra utiliser :

```
# dpkg - -get-selections | grep 'hold$'
```

La commande apt-mark Elle peut être utilisée comme interface unique pour diverses configurations d'un paquet, par exemple pour le marquer comme installé automatiquement ou manuellement, ou pour modifier les sélections de dpkg telles que hold install deinstall et purge qui sont suivies par exemple par les commandes apt-get dselect-upgrade ou aptitude.

PAQUETS INSTALLÉS AUTOMATIQUEMENT ET MANUELLEMENT

Lorsque l'installation d'un paquet est demandée et que par voie de dépendances d'autres paquets sont installés, ces paquets sont marqués comme ayant été automatiquement installés, tandis que le paquet que vous avez installé explicitement est marqué comme manuellement installé.

Une fois que ces paquets automatiquement installés ne sont plus liés à un paquet installé manuellement, ils ne seront plus considérés comme nécessaires et apt ou aptitude, par exemple, proposeront au moins de les supprimer.

Voir le man pour toutes les options

```
uname -r
```

```
4.19.0-17-amd64
```

```
cat /etc/apt/apt.conf.d/01autoremove-kernels
```

```
// DO NOT EDIT! File autogenerated by /etc/kernel/postinst.d/apt-auto-removal
APT::LastInstalledKernel "5.10.0-21-amd64";
```

Mise à jour de mon serveur vers bullseye

Mise à jour du serveur "funambule.org" de buster à bullseye

Il me faut mettre à jour nextcloud avant.

```
apt update && apt upgrade
```

```
apt full-upgrade
```

Changement du source list

```
#DEBIAN BULLSEYE
deb http://deb.debian.org/debian/ bullseye main contrib non-free
#deb-src http://deb.debian.org/debian/ bullseye main contrib non-free
deb http://security.debian.org/debian-security bullseye-security main
contrib non-free
#deb-src http://security.debian.org/debian-security bullseye-security main
contrib non-free
deb http://deb.debian.org/debian/ bullseye-updates main contrib non-free
#deb-src http://deb.debian.org/debian/ bullseye-updates main
deb http://deb.debian.org/debian/ bullseye-backports main contrib non-free
```

```
apt update
```

```
apt upgrade --without-new-pkgs
```

Cette commande met à niveau les paquets qui peuvent l'être sans entraîner l'installation ou la suppression d'autres paquets

```
apt full-upgrade
```

cette commande installe les versions candidates des paquets installés en installant ou en supprimant d'autres paquets si nécessaire

quelques explications s'affichent parfois, vous demandent de taper « q » pour quitter

quelques messages d'avertissement sur des répertoires qui ne sont pas vides, donc laisser en l'état.

Suite à cette mise à jour :

lufi ne fonctionnait plus, il faut donc, virer le répertoire local :var/www/html/lufi/local

et faire une nouvelle install avec :

```
carton install
```

il faut aussi réinstaller php7.4-gd et php-gd pour le bon fonctionnement de nextcloud

Ensuite, j'ai cherché les paquets à purger avec la commande

```
dpkg -l | grep -v ^ii
```

J'ai trouvé une quinzaine de paquets avec rc qui veut dire : désinstallé mais pas purgé

commande:

```
dpkg -l | awk '$1~/^rc$/{print $2}' | xargs echo
```

ou encore:

```
dpkg -l | awk '/^rc/ { print $2 }'
```

Purger les paquets marqués avec rc `apt purge $(dpkg -l | awk '/^rc/ { print $2 }')`

on peut faire les paquets 1 par 1 si on le veut

```
dpkg --purge paquet
```

Si vous utilisez aptitude, vous pouvez également utiliser cette commande au lieu de la précédente :

```
aptitude search '~c'
```

```
aptitude purge '~c'
```

dans la catégorie « Paquets obsolètes ou créés localement » et ils peuvent être listés et purgés depuis la ligne de commande avec :

```
aptitude search '~o'
```

```
aptitude purge '~o'
```

pour finir

```
# apt autoremove
```

on à toutes les infos dans les notes de publications de debian

<https://www.debian.org/releases/bullseye/ppc64el/release-notes.fr.pdf#43>

entre autre ceci :

Problèmes possibles pendant une mise à niveau

Les parties suivantes décrivent les problèmes connus pouvant survenir lors d'une mise à niveau vers Bullseye.4.5.1 La mise à niveau échoue avec « Impossible de faire une configuration immédiate ».

Dans certains cas, l'étape `apt full-upgrade` peut échouer après le téléchargement des paquets avec :

E: Impossible de faire une configuration immédiate sur 'paquet'. Veuillez ←↔

consulter `man 5 apt.conf` à la section `APT::Immediate-Configure` pour plus de précisions.

Si cela se produit, exécuter `apt full-upgrade -o APT::Immediate-Configure=0` à la place devrait permettre la mise à niveau.

Une autre possibilité pouvant permettre de contourner ce problème consiste à ajouter temporairement des sources pour buster et bullseye dans vos listes de sources d'APT puis d'exécuter la commande

```
apt update
```

Mise à jour de mon serveur vers bookworm

j'ai fait exactement la même procédure que pour le passage à Bullseye, voir plus haut.

aucun problème, tout s'est bien passé, même si lors de full-upgrade, l'installateur a fermé un service:

Pour terminer, si vous voulez voir les services démarrés:

Ou les failles de sécurités :

avant tout faites un

```
apt install debsecan
```

```
systemctl list-unit-files | grep enabled
```

```
systemctl | grep running
```

```
debsecan --suite bullseye
```

```
debsecan --suite bullseye --format detail
```

voir aussi ici

<https://security-tracker.debian.org/tracker/>

Vous pouvez aussi vérifier l'état de tous les paquets du système (en root)

```
dpkg -l | pager
```

tous les paquets qui contiennent linux

```
dpkg -l | pager | grep linux
```

voir les paquets bloqués qui bloquent la mise à jour

```
aptitude search ~ahold
```

ou

```
dpkg --get-selections | grep hold$
```

Un problème avec votre display-manager (lightdm, gdm etc...)

```
systemctl status graphical.target display-manager.service
```

voir l'état en en générale

```
systemctl status  
ou  
systemctl --failed
```

Pour vérifier l'état corrompu du noyau dans les journaux du noyau :

```
journalctl -k | grep taint
```

Pour vérifier l'état corrompu du noyau lors de l'exécution :

```
cat /proc/sys/kernel/tainted
```

From:

<http://debian-facile.org/> - **Documentation - Wiki**

Permanent link:

<http://debian-facile.org/utilisateurs:lagrenouille:apt-dpkg>

Last update: **16/05/2024 23:23**

