

aptitude

- Objet : aptitude
- Niveau requis : [avisé](#)
- Commentaires : *Rechercher, installer ou mettre à jour les logiciels et autres paquets disponibles.*
- Suivi : [à-compléter](#)
 - Création par [smolski](#) 23-11-2010
 - Testé par [smolski](#) le 16-07-2014
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto^{1\)}](#)

Introduction

Aptitude est le gestionnaire de paquets recommandé pour la gestion des paquets en [console](#), c'est-à-dire leur installation, l'informations qui les concerne, leur suppression et leur mise à jour.

De façon général, il gère donc la mise à jour de votre système et s'utilise en [super-utilisateur](#) pour cela mais peut (*doit...*) être utilisé en session *simple utilisateur* pour vous informer sur les paquets.

Nota important :

N'hésitez pas à respecter l'usage de *simple utilisateur* lorsque cela vous est permis, quitte à ouvrir une seconde session console, bien des erreurs vous seront évitées ainsi, sûr ! 😊

Il fournit les mêmes fonctionnalités que `dselect` et [apt-get](#), ainsi que des fonctions additionnelles non présentes dans les autres programmes :

- il effectue les tâches d'administration de manière rapide et facile.
- il permet d'accéder à toutes les versions d'un paquet.
- il garde un journal de toutes ses actions dans **`/var/log/aptitude`**.
- il inclut un système assez puissant pour rechercher des paquets particuliers.
- il recherche les paquets qui ont été installés pour satisfaire les dépendances, et les supprimera automatiquement quand les paquets qui les utilisaient seront supprimés.
- etc.

Vous pouvez utiliser aptitude au travers d'une [interface graphique](#) (en lançant simplement aptitude) ou directement en ligne de commande.

La syntaxe utilisée en ligne de commande est très similaire à celle utilisée avec apt-get.

Mise à jour de la liste des paquets

Avec les privilèges du [super-utilisateur](#) :

```
aptitude update
```

Cette opération consiste à mettre à jour sur votre machine la liste des paquets disponibles sur les

serveurs déclarés dans le fichier [/etc/apt/sources.list](#).

- Voir: [la gestion et l'administration des dépôts Debian](#).

Si vous ne l'avez pas faites depuis plus de 24H, vous devez impérativement l'exécuter avant toute modification de votre installation !

Mise à jour des logiciels installés sur le système:

```
aptitude safe-upgrade
```

Met à jour les paquets d'après la liste des *mises à jour possibles* obtenue auparavant avec **aptitude update**. Le solveur de dépendance utilisé dans l'opération de **safe-upgrade** a pour contrainte de ne supprimer aucun paquet installé manuellement. Ainsi, les seuls paquets éventuellement désinstallés lors de cette opération sont ceux ayant été automatiquement installés dans le passé (pour résoudre des dépendances) et n'étant plus nécessaires (dépendance modifiée ou supprimée).

C'est la procédure de mise à jour standard sous Debian stable.

Il est recommandé de la faire régulièrement, afin d'installer les derniers correctifs de sécurité.

Mise à jour du système:

```
aptitude full-upgrade
```

Le solveur de dépendance utilisé par **full-upgrade** se fixe comme objectif de faire la *totalité* des mises à jour possibles, quitte à supprimer d'autres paquets s'ils empêchent le système d'être mis à jour.

Cette commande s'appellait **dist-upgrade**. Pour des raisons historiques, aptitude la reconnaît toujours. **dist-upgrade** est donc un synonyme de **full-upgrade**.

Par précaution, avec **full-upgrade**, il est conseillé d'utiliser l'option `-s` (*simulation*) pour simuler l'installation de la mise à jour, ainsi vous pourrez contrôler les éventuels *logiciels/dépendances* proposés à l'installation ou à la désinstallation AVANT de les réaliser vraiment !

Soit la commande :

```
aptitude -s full-upgrade
```

Le **full-upgrade** propose parfois de désinstaller des paquets, il faut juste être attentif à ce qu'il veut désinstaller mais en dehors de ça, ça ne craint rien. 😊

J'essaye au maximum de n'utiliser que **full-upgrade** je reste à **safe-upgrade** uniquement quand il y a des dépendances cassées dans sid.

captnfab sur irc [#debian-facile](#)

Lorsqu'aptitude propose de supprimer des paquets pour résoudre les dépendances, il n'y a souvent pas qu'une seule solution valide. Pensez à répondre « n » à la question « Cette solution vous convient-elle ? » afin de consulter les autres propositions.

Faire des recherches

Rechercher un logiciel

```
aptitude search <expression rationnelle>
```

Cette commande vous retournera la liste de tous les paquets dont le nom contient, même partiellement, le motif ([W l'expression rationnelle](#)) de la recherche.

Exemples :

```
aptitude search liferea
```

[retour de la commande](#)

```
i   liferea                - Agrégateur de flux pour
GNOME
i A liferea-data          - architecture independent data
for liferea
p   liferea-dbg           - liferea debug symbols
v   liferea-webkit        -
v   liferea-xulrunner     -
```

Les résultats sont présentés ligne par ligne.

Les premiers caractères sont des indicateurs d'état²⁾, suivi du nom du logiciel, accompagné d'une brève description.

Les indicateurs d'état

- *Le premier caractère indique l'état courant du paquet :*
 - i ⇒ le paquet est installé
 - p ⇒ aucune trace du paquet n'est présente sur le système.
 - v ⇒ le paquet est virtuel.³⁾
 - c ⇒ le paquet à été supprimé mais ses fichiers de configuration sont toujours présent sur le système.
 - B ⇒ le paquet a des dépendances cassées.

Les caractères suivants , s'ils sont présents, indiquent dans l'ordre :

1. l'action qui sera entreprise sur le paquet,
2. si le paquet a été automatiquement installé,
3. si le paquet est fiable.

- *Caractères suivants susceptibles d'être rencontrés:*
 - h ⇒ le paquet est gelé.
 - F ⇒ toute mise à jour du paquet est interdite.
 - B ⇒ le paquet est cassé.

- A ⇒ le paquet a été installé automatiquement.
- d ⇒ le paquet va être supprimé, ses fichiers de configuration resteront sur le système.

Utilisation avancée de l'option "search"

Commande	Commentaire
aptitude search mot1 mot2	liste les paquets contenant le mot1 et les paquets contenant le mot2
aptitude search 'mot1 mot2'	liste les paquets contenant le mot1 et le mot2
aptitude search ~dmot1	liste les paquets dont la description contient le mot1
aptitude search 'mot1 ~sweb'	liste les paquets contenant le mot1 dans la section ⁴⁾ web
aptitude search '~dmot1 ~sweb'	liste les paquets dont la description contient le mot1 dans la section web
aptitude search ~i	liste les paquets installés
aptitude search ~mMainteneur	liste les paquets maintenus par <i>Mainteneur</i>
aptitude search ~M	liste les paquets installés automatiquement (dépendant d'autres paquets)
aptitude search ~b	liste les paquets ayant une dépendance cassée
aptitude search ~U	liste les paquets installés et pouvant être mis à jour
aptitude search ~ahold	liste les paquets gelés
aptitude search ~o	liste les paquets obsolètes ou créés localement
aptitude search ~c	liste les paquets supprimés mais non purgés
aptitude search ~E	liste les paquets essentiels

Exemples d'utilisation avancée

```
[16:32] <captnfab> et pourquoi tu n'utilises pas un émulateur gba qui est ds
les dépôts debian ?
[16:32] <galendil> Je savais pas qu'il y en avait un en fait

[16:38] <captnfab> aussi: aptitude search ~demulator~sgame
[16:39] <galendil> pourquoi ~d?
[16:39] <captnfab> pour rechercher dans la description plutôt que dans le
nom du paquet
[16:39] <captnfab> et ~s, c'est pour rechercher dans la section
[16:40] <captnfab> donc, les paquets de la section games qui parlent
d'émulateurs...
[16:40] <galendil> Ah
[16:40] <galendil> Merci pour ces infos :D
[16:41] <captnfab> :)
[16:41] <galendil> IL FONCTIONNE \o/
[16:42] <captnfab> wé, tu vas pouvoir jouer aux pokémons maintenant :)
```

Ou encore :

```
aptitude search '~ilifer !~M'
```

[retour de la commande](#)

```
i   liferea                - Agrégateur de flux
pour GNOME
```

Retourne le paquet installé (~i) contenant le mot lifer, en excluant ceux qui ont été installés automatiquement (!~M).

Autrement dit: retourne le paquet installé manuellement contenant le mot lifer

```
aptitude search '~i ~drss !~M'
```

[retour de la commande](#)

```
i   icedove                - client de courriel et
lecteur de nouvelles avec RSS
i   liferea                - Agrégateur de flux
pour GNOME
```

Retourne les paquets installés manuellement dont la description contient le mot rss.

- [Plus d'exemples](#)

Rechercher les versions disponibles d'un logiciel

```
aptitude versions nomdulogiciel
```

Les résultats de cette commande sont en fonction des serveurs déclarés dans le fichier /etc/apt/[preferences](#) quand il existe.

Exemple:

```
aptitude versions iceweasel
```

[retour de la commande](#)

```
p   3.5.16-6                stable
900
p   3.5.16-8                proposed-
updates,stable           900
p   3.5.19-3                testing
-10
i   5.0-3~bpo60+1          squeeze-backports
200
p   5.0-6                   unstable
-10
```

Chaque version est présentée sur une ligne distincte.
Les premiers caractères sont des indicateurs d'état⁵⁾

À la droite du numéro de version, vous pouvez trouver la branche à partir de laquelle la version est disponible, et la priorité du dépôt défini par le fichier </etc/apt/preferences>.

Obtenir des informations sur un logiciel

```
aptitude show nomdulogiciel
```

Cette commande affiche les caractéristiques et la description du paquet installé ou installable "nomdulogiciel"

```
aptitude show -v nomdulogiciel
```

L'option -v retournera l'affichage de toutes les versions disponibles.

```
aptitude show -t<dépôt> nomdulogiciel
```

L'option -t <dépôt> retournera l'affichage de la version disponible dans le dépôt indiqué.⁶⁾

Rechercher les dépendances, ou les conflits entre paquets.

```
aptitude why nomdulogiciel
```

Cette commande retourne la suite de dépendances qui s'enchaînent jusqu'au paquet visé. Elle permet également d'expliquer pourquoi le paquet nomdulogiciel est, ou sera installé.

Exemple:

```
aptitude why evolution-common
```

[retour de la commande](#)

```
i  gnome-core Dépend evolution (>= 2.30)
i A evolution  Dépend evolution-common (= 2.30.3-5)
```

```
aptitude why-not nomdulogiciel
```

Cette commande retourne les raisons pour lesquels les paquets qui correspondent à nomdulogiciel ne peuvent pas être installés.

Exemple:

```
aptitude why-not apache2-mpm
```

[retour de la commande](#)

```
i gitweb Dépend apache2 | httpd
i A apache2 Dépend apache2-mpm-worker (=
2.2.16-6+squeeze1) | apache2-mpm-prefork (= 2.2.16-6+squeeze1) |
                                                                    apache2-mpm-
event (= 2.2.16-6+squeeze1) | apache2--itk (=2.2.16-6+squeeze1)
i A apache2-mpm-worker Est en conflit avec apache2-mpm
```

Installation d'un Logiciel

```
aptitude install LeNomDuLogiciel
```

Cette commande installera la version candidate du paquet *LeNomDuLogiciel* ainsi que ses dépendances.

Pour plus de précautions, vous pouvez ajouter l'option `-s` (comme simulation) pour simuler l'installation du paquet logiciel, ainsi vous pourrez contrôler les éventuels logiciels/dépendances installés ou désinstallés.

```
aptitude -s install lenomdulogiciel
```

Les dépendances du logiciel installé manuellement, seront marquées comme étant installées automatiquement.

Utilisation avancée de l'option "install"

Commande	Commentaire
<code>aptitude install -t<branche> LeNomDuLogiciel</code>	Installer un logiciel situé dans une autre branche du sources.list ^{7) 8)}
<code>aptitude install -without-recommends LeNomDuLogiciel</code>	Installer que les paquets essentiels. ⁹⁾
<code>aptitude install -d LeNomDuLogiciel</code>	Télécharger un paquet deb sans l'installer. ¹⁰⁾

Utilisation avancée avec les attributs de surcharge

Commande	Commentaire
<code>aptitude install LeNomDuLogiciel+M</code>	Installer un logiciel et le marquer comme installé automatiquement.
<code>aptitude install LeNomDuLogiciel-</code>	Désinstalle le logiciel (équivalent à: <code>aptitude remove</code>).
<code>aptitude install LeNomDuLogiciel_</code>	Désinstallation complète (avec les fichiers de configurations) du logiciel (équivalent à: <code>aptitude purge</code>).
<code>aptitude install LeNomDuLogiciel&M</code>	Marque le logiciel comme ayant été installé automatiquement (équivalent à: <code>aptitude markauto</code>).
<code>aptitude install LeNomDuLogiciel&m</code>	Marque le logiciel comme ayant été installé manuellement (équivalent à: <code>aptitude unmarkauto</code>).
<code>aptitude install LeNomDuLogiciel=</code>	Gèle le logiciel, ¹¹⁾ (équivalent à: <code>aptitude hold</code>).

Commande	Commentaire
<code>aptitude install LeNomDuLogiciel:</code>	Dégèle le logiciel (équivalent à: <code>aptitude unhold</code>).

Ainsi, sur la même ligne de commande, il est possible par exemple, d'installer un "LogicielA" et de supprimer le "LogicielB":

```
aptitude install LogicielA LogicielB-
```

Utilisation avancée: Installer un paquet source

```
aptitude source le-paquet-source
```

Cette commande vous permettra de télécharger les sources d'un paquet debian.

Il faut vous assurer au préalable que votre fichier `/etc/apt/source.list` est correctement rempli avec l'adresse des dépôts contenant les paquets sources (classiquement les noms de ces dépôts commencent par `deb-src`).

Par défaut, le paquet est décompressé une fois téléchargé. Si vous souhaitez le garder en `*.tar.gz`, passez le paramètre `-download-only` à la commande :

```
aptitude source le-paquet-source --download-only
```

Si vous voulez compiler directement et automatiquement le paquet après son téléchargement, passez l'option `-compile` à la commande :

```
aptitude source le-paquet-source --compile
```

Assurez-vous au préalable d'avoir installé les bibliothèques nécessaires à la compilation du paquet. Pour ce faire, utilisez la commande

```
aptitude build-dep le-paquet
```

Elle aura pour effet d'installer toutes les dépendances nécessaires à la compilation du paquet.

Téléchargement d'un logiciel

Pour télécharger un paquet en `.deb` avec ses dépendances sans l'installer :

```
aptitude clean
```

```
aptitude -d install paquet
```

place tout dans `/var/cache/apt/archives/`

Réinstallation d'un Logiciel


```
aptitude reinstall LeNomDuLogiciel
```

Il arrive que le système soit endommagé suite à la suppression ou à la modification de fichiers appartenant à un paquet.

Le moyen le plus simple de récupérer ces fichiers est alors de réinstaller le paquet concerné.

Désinstallation d'un Logiciel

Désinstallation simple

```
aptitude remove LeNomDuLogiciel
```

Cette commande désinstallera le paquet LeNomDuLogiciel, ainsi que ses paquets dépendants installés automatiquement, mais leurs fichiers de configuration sont toujours présent sur le système.¹²⁾

Utilisation avancée de l'option "remove"

```
aptitude remove ~mot
```

Cette commande supprimera tous les paquets dont le nom contient: **mot**

A utiliser avec précaution, n'hésitez pas à utiliser l'option -s (simulation)

```
aptitude -s remove ~mot
```

```
aptitude remove LeNomDuLogiciel+
```

Cette commande utilisant l'attribut de surcharge + installera le logiciel au lieu de le supprimer.

Les attributs de surcharges présentés dans l'utilisation avancée de l'option "install" sont utilisables avec l'option "remove".

Désinstallation complète

```
aptitude purge LeNomDuLogiciel
```

Dans l'exemple précédant, l'option remove désinstalle le paquet mais les fichiers de configurations globaux (situés normalement sous /etc) sont conservés.

Cette commande désinstalle et supprime ces fichiers de configuration en même temps que le paquet, les paquets dépendant sont bien supprimés cependant leurs fichiers de configuration sont toujours présents.¹³⁾

Utilisation avancée de l'option "purge"

ATTENTION ! Cette commande purge également toutes les dépendances **automatiquement** installées dont certaines dont vous auriez peut-être voulu en garder la configuration ! Vérifiez bien la liste de la purge proposée. Si vous n'êtes pas sûrs de comprendre cet avertissement, cette commande n'est pas celle que vous voulez utiliser !

```
aptitude purge --purge-unused LeNomDuLogiciel
```

Cette commande désinstalle et supprime les fichiers de configuration du paquet et de ses dépendances. ¹⁴⁾

Par contre, les fichiers de configuration que l'application crée dans votre /home/utilisateur ne seront pas enlevés.

La solution :rechercher les répertoires *".lenomdu logiciel"* et les supprimer.

Pour rechercher de quel paquet provient un fichier qu'on désire supprimer :

- [dpkg option s](#)

C'est tout du **captnfab** ça avec la collaboration de **matux** sur le forum ici :

- [Le forum df - Supprimer les fichier de conf après désinstallation d'un programme ?](#)

Gérer les paquets automatiquement installés

Lors de l'installation d'un paquet, aptitude va automatiquement installer tout autre paquet dont il dépend.

Ces paquets sont marqués comme ayant été automatiquement installés (indicateur A).

Aptitude va les surveiller et les supprimer quand plus aucun paquet manuellement installé n'en dépendra.

Ainsi :

```
aptitude markauto LeNomDuLogiciel
```

marque le paquet LeNomDuLogiciel comme ayant été installés automatiquement¹⁵⁾

et :

```
aptitude unmarkauto LeNomDuLogiciel
```

marque le paquet LeNomDuLogiciel comme ayant été installés manuellement.

Exemple d'utilisation

Gérer un méta-paquet sans supprimer ses dépendances

Pour le méta-paquet gnome :

```
aptitude unmarkauto ~i~Rgnome
```

Cela marque toutes les dépendances de gnome comme installées manuellement.

1. Les paquets installés manuellement ne sont pas supprimés par aptitude lorsqu'aucun autre paquet ne dépend d'eux.
2. Cette manipulation permet donc de supprimer le méta-paquet gnome tout en conservant les applications qu'il avait installé.

Seul **aptitude** parvient à faire ce genre d'opération, il est donc recommandé à l'utilisation pour ce genre de manipulation à la place de [apt-get](#) ou de tout autre exercice sur le système des paquets.

Suppression du paquet **gnome-core**

Tout d'abord, il est important de spécifier que j'avais seulement le paquet **gnome-core** d'installé, pas de présence des autres méta-paquets.

Comme je croyais à tort que **gnome-core** n'était pas un méta-paquet, je me voyais pris avec des applications inutiles comme **evolution** que je ne pouvais enlever isolément des autres applications du même méta-paquet.

Or, ce n'est pas le cas, on peut virer **gnome-core** et tous les paquets qu'il contient de la même manière que les autres méta-paquet.

D'abord, je tape la commande suivante :

```
aptitude -s purge gnome-core
```

Cela me donne [la liste des paquets installés automatiquement](#) que **aptitude** veut virer avec **gnome-core**.

Dans mon cas :

[retour de la commande](#)

```
eog evolution evolution-common evolution-webcal gedit gedit-common
gnome-core gnome-disk-utility libavahi-ui0 libgdu-gtk0
libgtksourceview2.0-0 libgtksourceview2.0-common python-gtksourceview2
```

1. On peut copier/coller cette liste dans un fichier temporaire,
2. ou bien la reprendre depuis le même terminal en copier/coller comme proposé ici.

Je retire **gnome-core** de cette liste, puis je tape :

```
aptitude unmarkauto la_liste
```

Je peux maintenant virer **gnome-core** sans problème :

```
aptitude purge gnome-core
```

Suppression au choix d'autres paquets

Après cela, plus de problème non plus pour enlever **evolution** (et certains paquets qui y sont rattachés).

Remerciements

Recette validée donc. 😊

Merci à **Berillions Haricophile** et **Martin**. Voir ce post sur le forum :

- <http://debian-facile.org/forum/viewtopic.php?pid=25843>

Empêcher la mise à jour d'un paquet

```
aptitude hold nomdulogiciel
```

Cette commande gèle le paquet nomdulogiciel, annule toute action de safe-upgrade ou full-upgrade (installation, mise à jour ou suppression) sur les paquets marqués (indicateur d'état h).

Elle permet de maintenir un paquet à une version inférieure à la version de votre branche debian installée.

```
aptitude unhold nomdulogiciel
```

Pour dégeler le paquet nomdulogiciel

Nettoyer le cache

Les paquets téléchargés avant installation sont stockés dans `/var/cache/apt/archives`. Ils y restent ad vitam.

Ce dossier peut donc rapidement devenir encombrant et faire plusieurs Giga-octets. Dans un terminal user faites donc un :

```
du -h /var/cache/apt/archives
```

vous serez surpris 😊

Faire un petit nettoyage

```
aptitude autoclean
```

La commande ci-dessus va supprimer uniquement les paquets dont le numéro de version est obsolète par rapport à ceux des miroirs (les vieux trucs quoi)

Faire le ménage de printemps

```
aptitude clean
```

Cette commande videra entièrement le contenu du cache.

Je connais même certain matelot qui l'utilise systématiquement après chaque installation de paquet, c'est vous dire ! 😊

aptitude-create

Aptitude fournit deux utilitaires dédiés à la sauvegarde de l'état des paquets sur un système :

1. aptitude-create-state-bundle et
2. aptitude-run-state-bundle.

La sauvegarde de la liste des paquets se fait avec la commande :

```
aptitude-create-state-bundle liste_paquet.bz2
```

Qui préservera en plus les informations concernant les paquets listés.

Voir le man :

```
man aptitude-create-state-bundle
```

La restauration se fait avec la commande :

```
aptitude-run-state-bundle liste_paquet.bz2
```

Merci **cthuluh** 😊

Source :

- http://www.guim.info/dokuwiki/debian/system_backup

Choisir une image kernel

“ATTENTION : vous ne devez pas effectuer la mise à niveau en utilisant telnet, rlogin, rsh, ou depuis une session X gérée par gdm, kdm, etc. sur la machine que vous mettez à niveau. ”

Pour savoir les **noyaux les plus récents** qui existent :

```
apt-cache search linux-image
```

Tu copies ton choix dans la liste du (des) noyau(x) qui te convient(viennent).
Puis, dans une [Les gestionnaires de session](#))).

Par exemple sous **gnome** :

```
invoke-rc.d gdm stop
```

Puis tu installes :

```
aptitude install linux-image-que-t-a-choisi
```

Et relances la bécane par *reboot* afin que le nouveau noyau soit pris en compte.

Debian a cela de bien qu'il va gentiment mettre à jour le fichier menu.lst de grub et, **au reboot suivant**, tu auras le choix entre l'ancien noyau et le(s) nouveau(x).

hop la 😊 Et c'est freddec qui le dit !

Exemple

Pour rechercher une image linux pour une version 2.6.32... tapez dans un terminal user :

```
apt-cache search linux-image | grep 32
```

[retour de la commande](#)

```
linux-headers-2.6.32-5-amd64 - Header files for Linux 2.6.32-5-amd64
linux-headers-2.6.32-5-openvz-amd64 - Header files for Linux 2.6.32-5-
openvz-amd64
linux-headers-2.6.32-5-vserver-amd64 - Header files for Linux 2.6.32-5-
vserver-amd64
linux-headers-2.6.32-5-xen-amd64 - Header files for Linux 2.6.32-5-xen-
amd64
linux-image-2.6.32-5-amd64-dbg - Debugging infos for Linux 2.6.32-5-
amd64
linux-image-2.6.32-5-amd64 - Linux 2.6.32 for 64-bit PCs
linux-image-2.6.32-5-openvz-amd64-dbg - Debugging infos for Linux
2.6.32-5-openvz-amd64
linux-image-2.6.32-5-openvz-amd64 - Linux 2.6.32 for 64-bit PCs, OpenVZ
support
linux-image-2.6.32-5-vserver-amd64-dbg - Debugging infos for Linux
```

```
2.6.32-5-vserver-amd64
linux-image-2.6.32-5-vserver-amd64 - Linux 2.6.32 for 64-bit PCs,
Linux-VServer support
linux-image-2.6.32-5-xen-amd64-dbg - Debugging infos for Linux
2.6.32-5-xen-amd64
linux-image-2.6.32-5-xen-amd64 - Linux 2.6.32 for 64-bit PCs, Xen dom0
support
nvidia-kernel-2.6.32-5-amd64 - NVIDIA binary kernel module for Linux
2.6.32-5-amd64
nvidia-kernel-2.6.32-5-vserver-amd64 - NVIDIA binary kernel module for
Linux 2.6.32-5-vserver-amd64
```

Ici, j'ai utilisé la commande linux [GREP](#) pour sélectionner la version **32**.
Et nous installerons la version :

```
linux-image-2.6.32-5-amd64
```

Merci à **bestel74** d'avoir aidé à la création de cet exemple sur le forum df ici :

- <http://debian-facile.org/forum/viewtopic.php?pid=24029#p24029>

Et à **Darien** 😊 pour son intervention là :

- <http://debian-facile.org/forum/viewtopic.php?pid=38072#p38072>

Afficher la progression de l'installation

Pour aptitude version $\geq 6.8.2$ (Jessie ou supérieur) :

Il suffit de rajouter la ligne suivante dans les préférences apt :

```
/etc/apt/apt.conf.d/99showprogress
```

```
Dpkg::Progress-Fancy "true";
```

Pour plus d'informations :

- lisez la page de manuel aptitude

```
man aptitude
```

- installez le paquet de documentation aptitude-doc.

```
aptitude install aptitude-doc-fr
```

- [Référence du fichier de configuration d'aptitude](#)

et tapez cet url dans votre navigateur :

```
/usr/share/doc/aptitude/html/fr/index.html
```

Tasksel

Aptitude est également gérée par le tasksel.

Voir ici : [Le tasksel et la gestion des paquets via ncurses](#)

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

[pour en savoir plus sur les indicateurs](#)

3)

[Qu'est-ce qu'un paquet virtuel ?](#)

4)

liste des sections: admin, cli-mono, comm, database, devel, debug, doc, editors, electronics, embedded, fonts, games, gnome, graphics, gnu-r, gnustep, hamradio, haskell, httpd, interpreters, java, kde, kernel, libs, libdevel, lisp, localization, mail, math, misc, net, news, ocaml, oldlibs, othersofs, perl, php, python, ruby, science, shells, sound, tex, text, utils, vcs, video, web, x11, xfce, zope.<http://packages.debian.org/fr/squeeze/>

5)

voir: les indicateurs d'états dans rechercher un logiciel

6)

Exemple:

```
aptitude show -t squeeze-backports iceweasel
```

[retour de la commande](#)

```
Paquet : iceweasel
État: installé
Automatiquement installé: non
Version : 5.0-3~bpo60+1
Priorité : optionnel
Section : web
Responsable : Maintainers of Mozilla-related packages <pkg-mozilla-maintainers@lists.aliases.debian.org>
Taille décompressée : 5 685 k
Dépend: libc6 (>= 2.2.5), libglib2.0-0 (>= 2.16.0), libgtk2.0-0 (>= 2.10),
        libnspr4-0d (>= 1.8.0.10), libstdc++6 (>= 4.1.1), fontconfig,
        procps,
        debianutils (>= 1.16), xulrunner-5.0 (>= 5.0-3~bpo60+1)
Suggère: ttf-lyx | latex-xft-fonts, xfonts-mathml, ttf-mathematica4.1, xprint,
        mozplugger, libgssapi-krb5-2 | libkrb53
Fournit: gnome-www-browser, www-browser
Description : Navigateur web basé sur Firefox
Firefox est une refonte du navigateur de Mozilla. Il est similaire à
```


Galeon,
K-Meleon et Camino, mais est écrit en utilisant le langage
d'interface
utilisateur XUL, et pensé pour être léger et multi-plates-formes.

Ce navigateur est basé sur le code source de Firefox avec des
modifications
mineures. Historiquement, il était précédemment connu sous les noms
de Firebird
et Phoenix.

7)

[illustration de l'option -t dans manuel: fichier preferences](#)

8)

[pinning](#)

9)

[pour gérer tous les paquets "recommends" en empêchant l'installation systématique, vous pouvez créer le fichier apt.conf](#)

10)

le paquet et toutes ses dépendances seront téléchargés dans le répertoire /var/cache/apt/archives/
vous trouverez le tout au format deb. [voir: Installer un paquet .deb](#)

11)

annule toute action de safe-upgrade ou full-upgrade (installation, mise a jour ou suppression)

12)

Exemple:

```
aptitude search kompozer
```

[retour de la commande](#)

```
i kompozer - complete Web Authoring System
i A kompozer-data - kompozer data files
p kompozer-dev - KompoZer development files
# aptitude remove kompozer
Les paquets suivants seront ENLEVÉS :
kompozer kompozer-data{u}
**Résultat de la commande:**
aptitude search kompozer
c kompozer - complete Web Authoring System
c kompozer-data - kompozer data files
p kompozer-dev - KompoZer development files
```

Le paquet kompozer et son paquet dépendant kompozer-data ont été désinstallés, mais leurs fichiers de configuration sont toujours présent sur le système.

13)

Exemple:

```
aptitude search kompozer
```

[retour de la commande](#)

```

i   kompozer          - complete Web Authoring System
i A kompozer-data    - kompozer data files
p   kompozer-dev     - KompoZer development files
# aptitude purge kompozer
Les paquets suivants seront ENLEVÉS :
kompozer{p} kompozer-data{u}

```

Résultat de la commande:

```
aptitude search kompozer
```

[retour de la commande](#)

```

p   kompozer          - complete Web Authoring System
c   kompozer-data    - kompozer data files
p   kompozer-dev     - KompoZer development files

```

Aucune trace du paquet kompozer n'est présente sur le système, son paquet dépendant kompozer-data à été supprimé mais ses fichiers de configuration sont toujours présent.

¹⁴⁾**Exemple:**

```
aptitude search kompozer
```

[retour de la commande](#)

```

i   kompozer          - complete Web Authoring System
i A kompozer-data    - kompozer data files
p   kompozer-dev     - KompoZer development files

```

```
aptitude purge --purge-unused kompozer
```

[retour de la commande](#)

```

Les paquets suivants seront ENLEVÉS :
kompozer{p} kompozer-data{pu}

```

Résultat de la commande:

```
aptitude search kompozer
```

[retour de la commande](#)

```

p   kompozer          - complete Web Authoring System
p   kompozer-data    - kompozer data files

```

```
p kompozer-dev - KompoZer development files
```

Plus aucunes traces du paquet kompozer et de ses dépendances sur le système.

¹⁵⁾

un paquet marqué comme ayant été installés automatiquement, sera supprimé automatiquement dès que plus aucun paquet installé manuellement ne dépend de lui.Exemple:

```
aptitude search kompozer
```

[retour de la commande](#)

```
i kompozer - complete Web Authoring System
i A kompozer-data - kompozer data files
p kompozer-dev - KompoZer development files
```

```
aptitude markauto kompozer
```

[retour de la commande](#)

```
Les paquets suivants seront ENLEVÉS :
kompozer{u} kompozer-data{u}
```

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

Permanent link:
<http://debian-facile.org/doc/systeme:apt:aptitude>

Last update: **25/10/2016 06:59**

