

# Gestion des logiciels / Paquets sur CentOS

Paquet = archive (ensemble de fichiers)

## Pour installer sur CentOS :

1. Soit par les paquets en base : `yum install firefox`
  2. Soit en téléchargeant directement le paquet \*.rpm sur internet > installation locale
- 

Aller chercher un paquet rpm sur le net pour l'installer manuellement (**zsh**) :

### **Dans le bash :**

```
su root
```

```
rpm -Vp /home/centos/Download/zsh-4.3.10.5.el6.i686.rpm
```

```
rpm -U zsh (mettre à jour un paquet déjà installé)
```

```
rpm -qi zsh (obtenir des infos sur le paquet)
```

```
rpm -qip monpaquetzsh.rpm (obtenir des infos sur le paquet rpm)
```

```
rpm -qf paquet (obtenir des infos sur un fichier)
```

```
rpm -e paquet (desinstaller un paquet)
```

### **Exercice :**

#### rpm

Pour les liens de téléchargement, voir : <https://noelmace.framapad.org/1>

#### A] avec zsh

1) télécharger, vérifier et installer zsh 4.3

```
$ rpm -Vp zsh-4.3...rpm
```

```
# rpm -ihv zsh-4.3...rpm
```

2) afficher les informations

a) du fichier rpm zsh-4.3....rpm

```
$ rpm -qip zsh-4.3...rpm
```

b) du paquet zsh actuellement installé

```
$ rpm -qi zsh
```

et les comparer

3) télécharger zsh-4.2 (voir framapad 1)

4) lister les fichiers du zsh actuellement installé (tous, puis les fichiers de configuration puis fichiers de documentation), avec leur état

```
$ rpm -qils zsh
```

```
$ rpm -qics zsh
```

```
$ rpm -qids zsh
```

5) lister ceux du zsh-4.2...rpm, et les comparer

```
$ rpm -qilp zsh-4.2...rpm
```

6) désinstaller zsh

```
# rpm -ev zsh
```

- 7) télécharger libtermcap (voir framapad 1)
- 8) lancer un test d'installation de zsh-4.2

```
# rpm -iv --test zsh-4.2...rpm
```
- 9) installer la dépendance

```
# rpm -ivh compat-libtermcap-2.0.8-49.el6.i686.rpm
```
- 10) installer zsh-4.2

```
# rpm -ivh zsh-4.2...rpm
```
- 11) comparer les informations du zsh actuellement installé et de zsh-4.3

```
# rpm -qi zsh
# rpm -qip zsh-4.3...rpm
```
- 12) tenter d'installer zsh-4.3

```
# rpm -ivh zsh-4.3...rpm
file /usr/share/man/man1/zshzle.1.gz from install of zsh-4.3.10-5.el6.i686 conflicts with file from package zsh-4.2.6-8.el5.i386
```
- 13) mettre à jour vers zsh-4.3

```
# rpm -Uvh zsh-4.3...rpm
```
- 14) désinstaller zsh

```
# rpm -ev zsh
```
- 15) tenter d'afficher les informations sur le "mot clé" (pas d'option -p) zsh

```
# rpm -qi zsh
package zsh is not installed
```

## **Utilisation de YUM**

[http://doc.fedora-fr.org/wiki/YUM:\\_Configuration\\_du\\_gestionnaire\\_de\\_paquets#Configuration\\_de\\_YUM](http://doc.fedora-fr.org/wiki/YUM:_Configuration_du_gestionnaire_de_paquets#Configuration_de_YUM)

Une fois **YUM** configuré, je ne peux que vous conseiller de mettre à jour tout votre système via la commande (toujours en root) :

```
# yum update
```

Si c'est la première fois que vous mettez votre système à jour sur une installation toute fraîche de Fedora, lancez la commande

```
# yum update yum
```

Cette commande permet à **YUM** de se mettre à jour indépendamment des autres paquets.

Attention toutefois, le haut débit est indispensable pour réaliser les mises à jour du système car la quantité de paquets à mettre à jour est conséquente !

**Pour rechercher un paquet, faites la commande :**

```
# yum list <nom du paquet>
```

Bien sûr, ceci n'est pas très pratique si l'on ne connaît pas très bien le nom du paquet. Heureusement pour vous, **YUM** accepte le caractère joker \* (étoile). Ce caractère joker fonctionne également avec les autres commandes. Remarque : il est nécessaire d'« échapper » le joker en plaçant le caractère \ devant celui-ci ou bien en sécurisant toute la chaîne de caractère avec les guillemets.

**Exemple :**

```
# yum list kernel\*
```

ou

```
# yum list "kernel*"
```

**YUM** vous listera les paquets correspondants à votre recherche installés et ceux que vous pouvez installer.

Il est aussi possible de faire une recherche plus large, plus uniquement sur le nom du paquet mais aussi sur la description:

```
# yum search <mot clef>
```

**Pour installer un paquet :**

```
# yum install <nom du paquet>
```

**Pour supprimer un paquet :**

```
# yum remove <nom du paquet>
```

**Pour une mise à jour sélective :**

```
# yum --exclude=<nom du paquet> update
```

Ceci est surtout pratique pour mettre à jour votre système sans mettre à jour votre noyau si avez de modules que vous ne voulez pas réinstaller à chaque fois ou pour une autre raison :

```
# yum --exclude=kernel\* update
```

**Pour exclure un dépôt temporairement :**

```
# yum --disablerepo=<nom du dépôt> update
```

**Pour inclure un dépôt temporairement :**

```
# yum --enablerepo=<nom du dépôt> update
```

**Yum** peut de plus gérer des groupes de RPMs, pour faciliter l'installation d'ensembles complexes, comme par exemple les environnements graphiques.

**Pour connaître la liste des groupes disponible:**

```
# yum grouplist
```

**Pour avoir des informations sur un groupe, comme les paquets qui le composent:**

```
# yum groupinfo "Nom du Groupe"
```

**Pour installer un groupe:**

```
# yum groupinstall "Nom du Groupe"
```

**Pour supprimer un groupe:**

```
# yum groupremove "Nom du Groupe"
```

**Pour connaître la liste des dépôts configurés:**

```
# yum repolist all
```

Ce sont les principales commandes disponibles avec **YUM** mais il en existe de nombreuses autres que vous pouvez découvrir en tapant simplement

```
# yum help_
```

*Sauf exeptions explicitement mentionnées, toutes les opérations de se chapitre devront se faire avec yum.*

1) vérifier les mises à jours nécessaires pour votre système

```
# yum check-update
```

2) mettre à jour le paquet de votre choix

```
# yum update firefox
```

3) répéter l'opération 10 (install zsh-4.2) du TP précédent (toujours via la commande rpm)

pour le faire avec yum : 

```
# yum localinstall zsh-4.2...rpm
```

4) mettre à jour zsh à partir du fichier local (paquet) zsh-4.3...rpm téléchargé à l'étape 1 du TP précédent

```
# yum localupdate zsh-4.3...rpm
```

5) vérifiez si zsh peu encore être mis à jour et si oui, le faire

```
# yum update zsh
```

6) rechercher un outil polyvalent pour l'affichage de statistiques (penser à la traduction anglaise des mots clés précédents)

*Rmq : un paquet bien précis correspond à cette recherche, mais tout autre outil pouvant répondre à cette fonctionnalité conviendra du moment qu'il est disponible*

```
# yum search statistic
```

7) installer le paquet correspondant

```
# yum install dstat
```

8) tester le programme installé

```
$ dstat
```

9) afficher le paquet correspondant au fichier /etc/passwd.

```
# yum provides /etc/passwd
```

...

```
setup-2.8.14-20.el6.noarch ...
```

Est-il installé ? A jour ?

```
# yum list setup-2.8...
```

10) quel(s) paquet(s) peut / peuvent vous offrir la "fonctionnalité" vim ?

```
# yum provides vim
```

11) quel(s) paquet(s) peut / peuvent résoudre la dépendance à la librairie libc.so.6 ?

```
# yum resolvedep libc.so.6
```

13) quelles sont les dépendances de bash ? Par quels paquets sont-elles satisfaites ?

```
# yum deplist bash
```

14) désinstaller zsh

```
# yum remove zsh
```

15) nettoyer le cash

```
# yum clean all
```

A] configuration de base

*Conseil : copiez le fichier de configuration vers une autre destination afin de pouvoir récupérer la configuration d'origine après ce TP ou en cas d'erreur*

```
sauvegarde : # cp /etc/yum.conf /etc/yum.conf.bak
```

```
restauration : # cat /etc/yum.conf.bak > /etc/yum.conf
```

```
# vim /etc/yum.conf
```

1) augmentez le niveau de verbosité de yum en plusieurs étapes, et observez le résultat en effectuant quelques tests

```
debuglevel=10
```

```
# yum .....
```

2) activer la vérification des signatures

```
gpgcheck=1
```

3) configurer pour que yum fasse 2 essais avant de retourner une erreur

```
retries=2
```

4) exclure le package vim-enhanced de la mise à jour

```
exclude=vim-enhanced
```

5) limiter le nombre de versions simultanées installées d'un paquet à 3

```
installonly_limit=3
```

6) faire en sorte que yum accepte de mettre à jour un paquet même si la nouvelle version proposée n'est pas la même que celle du paquet déjà installé

```
exactarch=0
```

```
:wq!
```

**Pour configurer Yum** > /etc/yum.conf et pour configurer les dépôts : /etc/yum.repos.d/

Gestion des priorités

**Pour installer les priorités** >

```
# yum install yum-priorities
```

```
/etc/yum/pluginconf.d/priorities.conf
```

**Ajout de dépôt via rpm :**

Télécharger le paquet :

```
$ wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.3-1.el6.rf.i386.rpm
```

Importer la clé GPG

```
$ rpm --import http://dag.wieers.com/rpm/packages/RPM-GPG-KEY.dag.txt
```

Vérifier le checksum du paquet

```
$ rpm -K rpmforge-release-0.5.1-1.el5.rf.i386.rpm
```

Installer la paquet et tout mettre à jour

```
# rpm -i rpmforge-release-0.5.1-1.el5.rf.i386.rpm
```

```
# yum check-update
```

**B] Gestion des dépôts**

1) essayer d'installer vlc

```
# yum install vlc
```

No package vlc available.

2) en vous renseignant sur les différents dépôts additionnels pour CentOS grâce à ce document (<http://wiki.centos.org/AdditionalResources/Repositories> et ceux liés, ainsi que [http://doc.fedora-fr.org/wiki/YUM : Configuration du gestionnaire de paquets#Configuration de YUM](http://doc.fedora-fr.org/wiki/YUM:_Configuration_du_gestionnaire_de_paquets#Configuration_de_YUM) tâchez de voir quel est le dépôt le plus susceptible de vous offrir ce logiciel  
*(dans le cas où cet exercice vous semble réellement prendre un temps très excessif, vous pouvez parfaitement effectuer une recherche google concernant vlc et centos6, et reporter l'exploration de ce document à plus tard)*

3) ajouter ce dépôt grâce à son rpm et installer vlc

importer la clé :

```
# rpm --import http://apt.sw.be/RPM-GPG-KEY.dag.txt
```

téléchargement du paquet rpm :

```
$ wget http://packages.sw.be/rpmforge-release/rpmforge-release-0.5.2-2.el6.rf.i686.rpm
```

vérification du paquet

```
$ rpm -K rpmforge...rpm
```

**installation**

```
# yum localinstall rpmforge...rpm
```

**ou**

```
# rpm -i rpmforge...rpm
```

installation de vlc

```
$ yum search vlc
```

```
# yum install vlc
```

4) activer le dépôt Vault base, puis effectuer quelques recherches pour en voir les conséquences. Faites de même avec CentOSPlus (le normal, pas celui de Vault)

```
# vim /etc/yum.repos.d/CentOS-Vault.repo
```

```
[c6.0-Base]
```

```
...
```

```
enabled=1
```

```
# vim /etc/yum.repos.d/CentOS-Base.repo
```

```
[centosplus]
```

```
...
```

```
enabled=1
```

5) faites en sorte que les dépôts soient considérés dans l'ordre de priorité suivant :

**Base > Extra > CentOSPlus > Vault > autres ...**

```
priority=1 (ou 2, 5, 10, 20 ...) pour chacun
```

**Bonus 1 ] installer google chrome ou tout autre programme de votre choix sortant totalement du cadre des dépôts "habituels" de CentOS**

<http://www.tecmint.com/install-google-chrome-on-redhat-centos-fedora-linux/>

## La famille Debian : equivalent de rpm = dpkg

Pour avoir des infos sur des paquets : <http://packages.debian.org>

### Installer un paquet deb

```
$ dpkg {-i|--install} monpaquet.deb
```

### Options supplémentaires :

-R ou recursive : mode récursif (indiquer un répertoire au lieu d'un paquet)

--ignore-depends=package : ignorer les informations de dépendances du paquet

-G ou refuse-downgrade : ne pas installer si une version plus récente du paquet est déjà installée

-E ou skip-same-version : ne pas installer si la même version du paquet est déjà installée

--no-act : simple test

---

### dpkg

note:

pour autoriser l'installation de paquets d'une autre architecture que celle de votre distribution:

```
#dpkg --add-architecture <archi>
```

exemple: pour installer un .deb 32bits sur un systeme x86\_64

```
#dpkg --add-architecture i386
```

<https://wiki.debian.org/Multiarch/HOWTO>

1) Répéter les questions du TPs RPM avec 2 versions de zsh parmi celles indiquées sur le framapad  
1. Pour savoir quels versions utiliser, affichez la version actuelle de votre debian par un `$ cat debian_version` et regardez à quel nom de code correspond la version affichée sur ce site (<http://distrowatch.com/table.php?distribution=debian>). Utilisez alors le paquet correspondant à votre version en lieu et place du zsh-4.2...rpm et celui correspondant à la version ultérieure en lieu et place du zsh-4.3.rpm

1) télécharger, vérifier et installer zsh 4.3

```
# dpkg -i zsh-5....deb
```

2) afficher les informations

a) du fichier deb zsh-4.3....rpm

```
$ dpkg -I zsh...deb
```

b) du paquet zsh actuellement installé

```
$ dpkg -p zsh
```

et les comparer

3) télécharger zsh-4.2 (voir framapad 1)

4) lister les fichiers du zsh actuellement installé (tous, puis les fichiers de configuration puis fichiers de documentation), avec leur état

```
$ dpkg -L zsh
```

5) lister ceux du zsh-4.2...rpm, et les comparer

```
$ dpkg -c zsh...deb
```

6) désinstaller zsh

```
$ dpkg -r zsh
```

8) lancer un test d'installation de zsh-4.2

```
# dpkg -i --no-act zsh...deb
```

10) installer zsh-4.2

```
# dpkg -i zsh...deb
```

11) comparer les informations du zsh actuellement installé et de zsh-4.3

```
$ dpkg -I zsh...deb
```

```
$ dpkg -P zsh
```

12) tenter d'installer zsh-4.3

```
# dpkg -i zsh-5...deb
```

13) mettre à jour vers zsh-4.3

la 12 fonctionne

14) désinstaller zsh

```
# dpkg -r zsh
```

15) tenter d'afficher les informations sur le "mot clé" (pas d'option -p) zsh

```
# dpkg -s zsh
```

2) Remarquez la différence dans le retour à la dernière question (15). Qu'est ce que cela signifie ?

```
# dpkg -s zsh
```

comment y remédier ?

```
# dpkg --purge zsh
```

3) Aller sur le site <http://packages.debian.org/fr/> pour rechercher les paquets sl, funny-manpages et dstat. Afficher leurs dépendances, et télécharger celles nécessaires. Placer tout ces paquets (.deb) dans un même dossier, et lancer l'installation de tout ce dossier, de manière récursive. Lancer enfin les commandes suivantes :

```
$ sl
```

```
$ man rtfm
```

```
$ dstat
```

a) télécharger les .deb et tous les mettre dans un même dossier (par exemple debs/)

b) afficher les infos

```
$ dpkg -I sl...deb
```

```
$ dpkg -s libc6
```

```
etc ...
```

c) installer

```
# dpkg -iR debs/
```



## apt

**apt-cache** > Manipulation du cache de paquets

aucune manipulation du système

recherche dans les métadonnées

```
$ apt-cache [options] action
```

**Informations sur un paquet :**

```
$ apt-cache showpkg paquet
```

(afficher versions, dépendances, dépendances inverses, paquets correspondant si applicable et méta-paquets le proposant)

```
$ apt-cache show paquet (afficher, section, taille, description complète, version architecture, suggestions etc ...)
```

```
$ apt-cache stats
```

Trouver les dépendances insatisfaites : 

```
$ apt-cache unmet
```

Afficher les dépendances d'un paquet : 

```
$ apt-cache depends paquet
```

**Rechercher des paquets :**

Par noms > 

```
$ apt-cache pkgnames [expression]
```

Par expression rationnelle > 

```
$ apt-cache search [expression]
```

**Exercice :**

Pour le paquet de votre choix (vim par exemple), récupérez les informations suivantes :

- version 

```
$ apt-cache showpkg vim
```

 OU 

```
$ apt-cache show vim
```
- dépendances (de 3 manières différentes)  

```
$ apt-cache depends vim
```

```
$ apt-cache showpkg vim
```

```
$ apt-cache show vim
```
- liste des paquets dépendant de celui-ci 

```
$ apt-cache showpkg vim
```
- description complète 

```
$ apt-cache show vim
```
- taille 

```
$ apt-cache showpkg vim
```

 OU 

```
$ apt-cache show vim | grep -i installed-size
```

**Répondez également aux questions suivantes :**

1) quelle est l'url du site officiel de zsh ? 

```
$ apt-cache show zsh |grep -i homepage
```

2) quel est le nombre total de paquets disponibles sur votre système

```
$ apt-cache stats |grep "Total package name"
```

3) y a-t-il des dépendances insatisfaites ? Si oui, lesquelles ?

```
$ apt-cache unmet vim
```

 ou 

```
$ apt-cache unmet (global)
```

4) quels sont les paquets dont le nom contient le mot clé "vim"

```
$ apt-cache pkgnames vim
```

...

```
vim-nox
```

```
vim-gui-common
```

```
vim-scripts
```

```
vim-common
```

```
vim-athena
```

```
vim-conque
```

...

5) quelles sont les différentes solutions permettant d'utiliser gvim (vim en interface graphique) ?

```
$ apt-cache showpkg gvim
```

...

```
vim-gtk 2:7.3.547-7
```

```
vim-gnome 2:7.3.547-7
```

```
vim-athena 2:7.3.547-7
```

par une recherche (un tri des informations est bien entendu nécessaire) :

```
$ apt-cache search gvim
```

6) afficher les problèmes importants de dépendances sur votre système

```
$ apt-cache unmet -i
```

7) nous avons besoin d'installer gcc-doc. Retracer quel est exactement le problème, et proposer une piste pour le résoudre.

```
$ apt-cache depend
```

```
s gcc-doc
```

```
$ apt-cache unmet gcc-doc
```

Dans ce cas, nous sommes face à l'absence du paquet nécessaire sur nos dépôts. On va donc chercher des informations sur le paquet (soit via une recherche dans un moteur de recherche, afin de retrouver le site officiel des dev, soit sur <http://packages.debian.org>) afin soit de le télécharger et installer manuellement soit d'ajouter les dépôts nécessaires.

Bonus 1 ] Afficher les informations sur le paquet source vim

```
$ apt-cache showsrc vim
```

## **apt-get**

Mettre à jour les informations sur les dépôts (resynchronisation du fichier d'index)

```
$ apt-get update
```

Recherche de dépendances défectueuses (met à jour le cache des paquets)

```
$ apt-get check
```

Nettoyer le référentiel local des paquets récupérés

```
$ apt-get clean
```

Installer

```
$ apt-get install package
```

Desinstaller

```
$ apt-get remove package (garde les fichiers conf)
```

```
$ apt-get purge package
```

Desinstaller les paquets qui ne sont pas utilisés

```
$ apt-get autoremove
```

Récupérer un paquet source

```
$ apt-get source package
```

Mettre à jour tout les packages installés

```
$ apt-get upgrade
```

Avec système intelligent de résolution de conflits (prend en compte les changements de dépendances gère un système de priorités)

```
$ apt-get dist-upgrade
```

**Les options :**

Option	Action(s)	Description
-d	upgrade, dselect-upgrade, install, source	téléchargement sans installation
-f	install, remove	répare les dépendances insatisfaites
-m	upgrade, dselect-upgrade, install, remove, source	ignorer les packages inaccessibles
-q / -qq	toutes	mode silencieux
-s	toutes	simulation
-y	toutes	répondre "yes" à tout prompt
-b	source	compiler après récupération
--no-upgrade	install	ne pas mettre à jour

### **Exercice apt-get**

- 1) mettre à jour le fichier d'index des paquets `$apt-get update`
- 2) rechercher les dépendances défectueuses `$apt-get check`
- 3) libérez de l'espace disque en nettoyant le référentiel local des paquets récupérés  
`$apt-get clean`
- 4) retrouver et installer un paquet de votre choix (zsh)  
`$apt-cache search zsh`  
`$apt-get install zsh`
- 5) récupérez le paquet source de vim `$apt-get source vim`
- 6) désinstaller zsh en supprimant ses fichiers de configuration `$apt-get purge zsh`
- 7) télécharger zsh sans l'installer, de deux manières différentes  
`$apt-get -d install zsh`  
`$apt-get -s install zsh (simulation)`
- 8) mettre à jour le paquet de votre choix `$apt-get upgrade vim`

Bonus 1 ] installer vim à partir du paquet source déjà récupéré

```
$apt-get install dpkg-dev
$apt-get -b source vim
$dpkg -i ./vim...deb
```

Bonus 2 ] Explorez un peu aptitude. (Remarque au passage : tentez de lancer :

```
$ aptitude moo
$ aptitude -v moo
$ aptitude -vv moo
$ apt-get moo
```

### **Exercice Configuration de apt**

- 1) quels sont les dépôts qui sont actuellement activés sur votre système ?

```
#less /etc/apt/source.list
```

2) installer smbclient `#apt-get install smbclient`

3) quel serait l'intérêt de mettre en place un dépôt backport ? (pour vous donner une idée, vous pouvez aller explorer le contenu des dépôts Debian sur <http://packages.debian.org/> particulièrement à la section wheezy-backport, et faire des comparaisons avec les autres versions, ainsi que lire cette page <https://wiki.debian.org/fr/Backports> et <http://backports.debian.org/>)

[Les dépôts backports chez Debian](#) permettent d'intégrer à votre distribution Debian en version stable des paquets **en provenance de la prochaine version "en test"** (actuellement [Wheezy](#)) mais adaptés pour s'intégrer sur votre système stable. En effet dès que la version stable est publiée, **elle n'est plus mise à jour que pour des bugs sérieux trouvés dans ses paquets ou des mises à jour de sécurité**. Si vous souhaitez mettre à jour d'autres paquets indispensables à la bonne marche de votre infrastructure, le **dépôt backports** devra être exploré avant de penser à dégainer votre compilateur.

4) passez toute votre configuration vers des dépôts backport, mettre à jour le fichier d'index, et vérifier les mises à jour disponibles

5) mettre à jour smbclient `#apt-get update smbclient`

6) explorez les différences entre debian oldstable, stable, testing et unstable, toujours grâce aux liens précédents. Dans quels cadre utiliser chacune ?

7) essayer d'installer chrome (pas chromium)

```
#apt-cache search google
```

```
#apt-get install google-chrome-stable
```

```
pb libxssl > reactiver les depots stable > apt-get update > pb clé > entrer ces commandes :
```

```
gpg --keyserver pgpkeys.mit.edu --recv-key [MA CLÉ]
```

```
gpg -a --export [MA CLÉ] | sudo apt-key add -
```

```
réponse serveur : OK
```

8) ajoutez un dépôt tiers vous permettant de l'installer, et installez le (vous pouvez retrouver une bonne liste de dépôts tiers ici <https://sites.google.com/site/mydebiansourceslist/>)

```
# echo "deb http://dl.google.com/linux/chrome/deb/ stable main" >> /etc/apt/sources.list
```

```
# apt-get update
```

```
# apt-get install google-chrome-stable
```

9) faites le ménage : on supprime ce dépôt, on désinstalle ça, et à la rigueur, on installe chromium à la place

```
# apt-get purge google-chrome-stable
```

```
# vim /etc/apt/sources.list
```

```
rajouter un # devant la ligne précédemment ajoutée
```

```
# apt-get install chromium-browser
```

Bonus 1 ] ajouter les dépôts contrib et non-free

```
# vim /etc/apt/sources.list
```

```
deb http://ftp.fr.debian.org/debian wheezy main contrib non-free
```

```
deb-src http://ftp.fr.debian.org/debian wheezy main contrib non-free
```

```
deb http://ftp.debian.org/debian wheezy-backports main contrib non-free
```

Remarque : ces dépôts contrib et non-free sont (du moins dans la plupart des cas) spécifiques aux dépôts officiels. Inutile donc de répéter cette opération pour un dépôt annexe, comme par exemple celui de google-chrome.