

Les paquets Debian

- Objet : Un paquet logiciel, c'est quoi
- Niveau requis :
[débutant, avisé](#)
- Commentaires : *Indispensable à connaître pour comprendre l'administration d'une Debian.*
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là ! 😊](#)
- Suivi :
[à-compléter](#)
 - Création par [captnefab](#) 23/05/2013
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto^{1\)}](#)

Définition d'un paquet logiciel

Un **paquet** est un ensemble de fichiers :

- des fichiers exécutables,
- des fichiers de configuration,
- des scripts d'installation / mise à jour / désinstallation
- des pages de man/info,
- la licence d'utilisation,
- etc.

Un **programme** est constitué d'un ou plusieurs paquets qui sont liés entre eux, c'est-à-dire qu'un paquet peut nécessiter l'installation d'un autre paquet pour pouvoir fonctionner : **c'est le système des dépendances**.

Les .deb

Tous ces fichiers sont regroupés au sein d'un paquet donc (ou une archive), dans un format propre à Debian sous l'extension : **.deb**

Dans les faits, chacun des “.deb” fait appel à la lib **apt**, qui, utilisant les listes des paquets (stockée dans **/var/lib/apt/lists/**) des différents dépôts (renseignés dans **/etc/apt/sources.list{,.d/*}**) va récupérer les paquets demandés et leurs dépendances depuis ces dépôts pour les recopier dans le cache apt (**/var/cache/apt/archives**), d'où ils seront installés grâce à **dpkg**.

- Voir : [dpkg - Extraire un .deb](#)

Un .deb, c'est une archive au format “ar”, qui contient 3 fichiers: un fichier **debian-binary** (indiquant le format du paquet), un fichier **control.tar.gz** (contenant les infos de dépendances, version, description, pré-installation, etc. du paquet), et un fichier **data.tar.gz** (ou **data.tar.bz2**) contenant les fichiers à installer.

On peut voir ce que contient un .deb avec la série de commandes suivante :

```
mkdir /tmp/prout
```

```
ar xv /mon/paquet.deb /tmp/prout
```

```
ls /tmp/prout
```

Merci au captnfab pour ce fol instant partagé sur irc dans le chan toujours vivant de [#debian-facile](#)²⁾

Dépendances

Un paquet Y est une **dépendance** du paquet X s'il faut absolument que Y soit installé pour pouvoir faire fonctionner X.

Illustrons cela par une petite histoire :

Il était une fois... le mariage d'un logiciel (libre) avec une bibliothèque (libre aussi) dont il va dépendre pour fonctionner :

- Le développeur de feh (développeur = créateur) veut que son logiciel (un visionneur d'image léger) puisse lire aussi le PNG.
- Un autre développeur a déjà écrit du code sous licence libre permettant de faire cela, le développeur de feh peut donc l'utiliser avec feh.
- Il indique simplement que feh a besoin de la libpng12-0 pour fonctionner correctement et le tour est joué.
- La bibliothèque libpng12-0 est ainsi devenue une dépendance de feh.

Il existe plusieurs niveaux de dépendance entre deux paquets Debian X et Y :

- **dépend** : X ne peut être installé que si Y est installé
- **recommande** : l'installation de X recommande l'installation de Y
- **suggère** : l'installation de X suggère l'installation de Y
- **conflit** : X ne peut pas être installé si Y est installé et vice versa

Ces informations peuvent être récupérées grâce [aux outils apt](#) :

```
apt-cache show feh
```

[retour de la commande](#)



```
Package: feh
Version: 2.12-1
Installed-Size: 343
Maintainer: Debian PhotoTools Maintainers <pkg-phototools-
devel@lists.alioth.debian.org>
Architecture: amd64
Depends: libc6 (>= 2.15), libcurl3 (>= 7.16.2), libexif12,
libimlib2 (>= 1.4.5), libpng12-0 (>= 1.2.13-4), libx11-6,
libxinerama1
```



Recommends: libjpeg-progs

...

Par défaut, `apt-get` installe les paquets recommandés mais pas les paquets suggérés. Donc par défaut l'installation d'un environnement de bureau installera le serveur graphique X.org.

Lorsque dans l'historiette le développeur crée son paquet Debian de `feh`, il indique que ce dernier a besoin de la `libpng12-0` pour fonctionner correctement. La `libpng12-0` est donc, de fait, une dépendance de `feh`

Il faut enfin distinguer deux cas :

1. les dépendances à la construction et
2. les dépendances à l'exécution.

Par exemple, pour un programme écrit en langage C, la construction dépend d'un compilateur C. Mais pas l'exécution.

Bibliothèques

Quoi, y'a des bouquins dans mon ordi et on ne pas pas prévenu ? 🤪

Non, mais presque 😊

Une bibliothèque logicielle est une collection de fonctions, qui peut être déjà compilée et prête à être utilisée par des programmes³⁾. Les bibliothèques, qu'on trouve sous forme de paquets elles aussi, ne sont pas utilisées seules, mais sont des dépendances d'autres paquets.



En Français, on voit souvent utilisé le mot *librairie* à la place de *bibliothèque*. Cela provient d'une erreur de traduction, car en anglais, *bibliothèque* se dit *library*.

Ces bibliothèques vont être liées au programme via l'éditeur de lien. On va distinguer deux types de liaison :

1. Liaison **statique** : les fonctions des bibliothèques utilisées par le programme sont incluses dans ce dernier lors de la construction. Il n'a pas besoin des bibliothèques lors de l'exécution.
Exemple : `busybox-static`.
Avantages : programme indépendant, évite les problèmes d'incompatibilités entre versions d'une même bibliothèque, parfois gain de place quand un programme est le seul à dépendre d'une bibliothèque dont il n'utilise que quelques fonctions (seules ces fonctions sont incluses dans le programme au lieu d'installer la bibliothèque complète).
2. Liaison **dynamique** : les fonctions des bibliothèques utilisées par le programme ne sont pas incluses dans ce dernier lors de la construction. Il a besoin de la bibliothèque lors de l'exécution.
Avantages : gain de place (les fonctions des bibliothèques sont partagées au lieu d'être dupliquées), simplicité de mise à jour (la correction d'un bug d'une bibliothèque ne nécessite pas de mettre à jour tous les programmes qui l'utilisent).

Lien vers le forum

Donc, les dépendances... Nous disions : dépendances. Ah ! Je sèche !

¹⁾

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

²⁾

Voir : [Le salon irc de df](#)

³⁾

Dixit [Wikipedia](#)

From:

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

Permanent link:

<http://debian-facile.org/doc:systeme:apt:paquets>



Last update: **08/10/2017 08:47**