

# Votre premier paquet Debian

- Objet : Créer un mini paquet Debian, et faire connaissance avec les différents points importants
- Niveau requis : [avisé](#)
- Commentaires : *Vous voulez vous lancer dans la construction de paquet Debian, empaqueter un nouveau logiciel ou reprendre la maintenance d'un paquet existant.*
- À savoir : [Contribuer à Debian](#) 😊
- Suivi : [à-tester](#)
  - Création par [captfnfab](#) 01/03/2014
  - Mis à jour par [greenmerlin](#) 28/10/2016
  - Testé par <...> le <...>
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto](#) <sup>1)</sup>

## Introduction

Bien que le nombre de paquets proposé par Debian soit démentiel (48608 environs)

<https://packages.debian.org/stable/allpackages?format=txt.gz>

Il est possible que vous trouviez un logiciel Open-source qui ne soit pas encore disponible par votre **apt install**

Là 2 choix s'offrent à vous :

<b>1ER choix</b>
Faire une demande officiel via le RPF ( Request for Package ) <a href="https://www.debian.org/devel/wnpp/requested">https://www.debian.org/devel/wnpp/requested</a>
<b>2EME choix</b>
Faire le paquet vous meme

Alors pourquoi faire un paquet me direz vous?

> Bah ouaii green t'oublies que je suis un gros Nerds de Debian&co et le ./configure && make && make install je maîtrise

» Ok mais si tu dois déployer ton programme sur 100 ordinateurs et qu'il met 4 ans à se compiler tu seras mort avant d'avoir pu faire le déploiement et du coup ton patron ira pourrir la vie de tes petits enfants

> HA d'accord j'ai compris!!!

» deuxièmement si tu regardes attentivement la page de **RPF** tu t'apercevras que certains attendent depuis très longtemps.....



Donc c'est parti pour le tuto

## Installation des paquets nécessaire à la création de paquet (fixme le titre de ce chapitre est null)

```
apt install build-essential automake autoconf libtool pkg-config libcurl4-  
openssl-dev intltool libxml2-dev libgtk2.0-dev libnotify-dev libglib2.0-dev  
libevent-dev checkinstall
```

## Récupération du code source du programme que vous voulez transformer en paquet Debian

Là c'est facile, il suffit de récupérer le code source du programme désiré. La plupart du temps il est proposé soit en \*.tar.gz ou en \*.tar.bz2 voir \*.zip pour github

Donc je me rend sur le site du programme/projet en question et je télécharge les sources.

```
cd /home/green/src
```

```
wget http://downloads.sourceforge.net/project/le_projet_en_question.tar.gz
```

ensuite je fais comme dans le super tutorial de Paul **cf : lien vers le tuto ./configure make et truc & co**

```
tar xzvf le_projet_en_question.tar.gz
```

```
cd le_projet_en_question
```

```
./configure --les-options-qui-vont-bien && make
```

```
checkinstall
```



C'est cette commande qui va créer votre paquet (.deb) mais avant cela elle va vérifier l'existence d'un dossier doc-pak à la racine qui est en fait un dossier qui contiendra la documentation de votre paquet.

Si elle ne trouve pas le dossier en question elle vous proposera donc d'en créer un

```
The package documentation directory ./doc-pak does not exist.  
Should I create a default set of package docs? [y]: y  
Preparing package documentation...OK
```

```
Please write a description for the package.  
End your description with an empty line or EOF.  
>>
```

La vous écrivez la description de votre paquet

```
Gros rootkit dissimuler dans un faux plugins d'evolution
```

Ensuite vous arriverez devant un menu qui vous permettra d'entrer tout les informations "vitales" pour un paquet Debian

```
This package will be built according to these values:
```

```
0 - Maintainer: [ root@debian ] //Qui harceler quand le paquet fonctionne plus  
1 - Summary: [ Gros rootkit dissimuler dans un faux plugins d'evolution ]  
//La description rentrer plus haut  
2 - Name: [ fish ] //Le nom du programme  
3 - Version: [ 20161027 ] // la version  
4 - Release: [ 1 ] // ...  
5 - License: [ GPL ] //etc ...  
6 - Group: [ checkinstall ] /etc...  
7 - Architecture: [ i386 ]  
8 - Source location: [ Fish ]  
9 - Alternate source location: [ ]  
10 - Requires: [ ]  
11 - Provides: [ fish ]  
12 - Conflicts: [ ]
```

```
13 - Replaces: [ ]
```

```
Enter a number to change any of them or press ENTER to continue
```

Une fois toutes les infos renseignées, valider en appuyant sur

Si tout c'est bien déroulé **Checkinstall** devrait vous donner une sortie comme ça

```
*****
Done. The new package has been installed and saved to
/usr/src/Fish/fish_20161027-1_i386.deb
You can remove it from your system anytime using:
    dpkg -r fish
*****
```

## Test du paquet avant mise en production

Là très simple on installe via un

```
dpkg -i nom_du_paquet.deb
```

puis on teste le programme si tout est ok on désinstalle via un

```
dpkg -r nom_du_paquet
```

voilà ensuite il ne vous reste plus qu'à implémenter votre nouveau paquet dans un miroir apt et le tour est joué

1)

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

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

Permanent link:  
<http://debian-facile.org/atelier:chantier:mentors-creer-un-paquet-simple>

Last update: **21/04/2019 14:00**

