



Packaging Debian



Introduction au packaging Debian

Y. Collette



Objectifs

L'objectif du packaging :

Préserver le code source du paquet original + patches correctifs pour assurer le build

Masquer les différentes étapes du build du paquet et des dépendances

Référencer les différentes modifications

Faire des builds pour plusieurs architectures

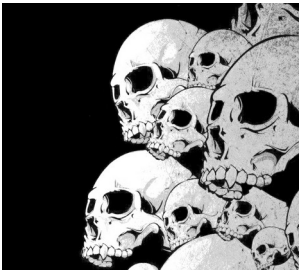
Tout cela, encapsulé dans un simple fichier SPEC

Automatisation, répétabilité, documentation du build

Définition :

Un container installable OS-spécifique ou distro-spécifique pour un software





Objectifs

Standardiser le déploiement	→	Savoir ce qu'on a installé
Simplifier l'environnement	→	Une seule source de logiciels
Conformité aux normes	→	Uniformisation de l'installation
Gestion des risques	→	Des mises à jour centralisées
Reproductible	→	Fiabilité des builds



APT / DPKG

Les quelques commandes pour gérer les paquets sous Debian :

apt	Gestion des paquets (pour un utilisateur final)
apt-get, apt-cache, apt-file	Gestion des paquets (pour un script)
aptitude	Gestion des paquets - interface semi-graphique
dpkg	Gestion des paquets hors dépôts
deborphan	Trouver les paquets non utilisés, appelés orphelins
dselect (param DPKG)	Gestion des paquets - interface à dpkg
wajig	Outil d'administration simplifié pour apt, dpkg et dselect



APT / DPKG

apt update

apt upgrade

apt install <package_name>

apt install <package_name>=<version_number>

apt remove <package_name>

apt purge <package_name>

apt remove just removes the binaries of a package. It leaves residue configuration files.

apt purge removes everything related to a package including the configuration files.

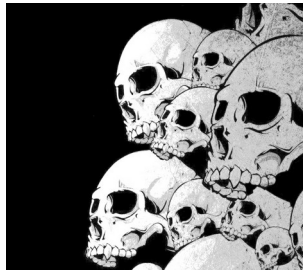
apt search <search term>

apt show <package_name>

apt list --upgradable

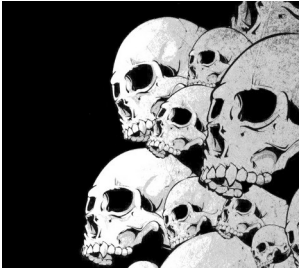
apt list --installed

apt autoremove



APT / DPKG

<code>dpkg -i <package_name></code>	install install
<code>dpkg -l</code>	list installed packages
<code>dpkg -r <package_name></code>	remove
<code>dpkg -p <package_name></code>	purge
<code>dpkg -c <package_name></code>	show content
<code>dpkg -s <package_name></code>	verify if installed
<code>dpkg -L <package_name></code>	check the location of installed packages
<code>dpkg --help</code>	show version of dkpg

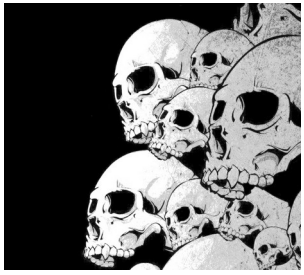


Configuration initiale

Avant de se lancer dans le packaging, il est nécessaire de configurer certaines variables d'environnement :

Ajouter dans ~/.bashrc

```
DEBEMAIL="ycollette.nospam@free.fr"  
DEBFULLNAME="Yann Collette"  
export DEBEMAIL DEBFULLNAME
```



La structure d'un paquet Debian

debian

changelog	the changelog file of the package
compat	contains a version number. Don't change this
control	a file which describes all the informations related to the package
copyright	the license of the package
install	a file which indicated which files are installed and where (optional)
rules	a 'makefile' which rules to execute various steps of the packaging
source	
format	a version number. Don't change ths



Le fichier control

Source: greetings

Section: unknown

Priority: optional

Maintainer: Yann Collette <ycollette.nospam@free.fr>

Build-Depends: debhelper (>= 11)

Standards-Version: 4.1.3

Homepage: <insert the upstream URL, if relevant>

#Vcs-Browser: <https://salsa.debian.org/debian/greetings>

#Vcs-Git: <https://salsa.debian.org/debian/greetings.git>

Package: greetings

Architecture: any

Depends: \${shlibs:Depends}, \${misc:Depends}

Description: <insert up to 60 chars description>

<insert long description, indented with spaces>



Générer un paquet

Before that, we need to install some required packages :

```
apt-get install debhelper dh-make devscripts  
apt-get install autoconf automake autotools-dev  
apt-get install fakeroot lintian  
apt-get install patch patchutils pbuilder  
apt-get install g++  
apt-get install quilt
```

Then :

```
mkdir -p package/greetings-0.1  
cd package/greetings-0.1/  
vi hi.sh  
chmod a+x hi.sh  
dh_make --indep --createorig  
ls debian/  
rm debian/*.ex debian/*.EX  
vi debian/install  
debuild -us -uc
```

generate the debian repo with initial files

cleanup

add 'hi.sh /usr/bin' to tell debian to install hi.sh

start package building





Générer un paquet

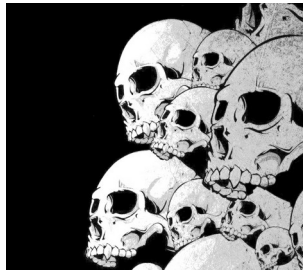
The files generated by debuild :

```
greetings-0.1/*  
greetings_0.1-1_all.deb  
greetings_0.1-1_amd64.build  
greetings_0.1-1_amd64.buildinfo  
greetings_0.1-1_amd64.changes  
greetings_0.1-1.debian.tar.xz  
greetings_0.1-1.dsc  
greetings_0.1.orig.tar.xz
```



Patcher un paquet Quilt

Ajouter les patches au paquet :



pbuilder

Packager dans un chroot

```
/sbin/pbuilder --create  
/sbin/pbuilder --build --basetgz  
/var/cache/pbuilder/base.tgz  
/home/artelys/package/greetings_0.1-1.dsc
```



Créer un dépôt local

Installer le paquet createrepo puis générer la structure du dépôt :

```
$ dnf install createrepo  
$ mkdir /opt/local-repo  
$ createrepo /opt/local-repo
```

Créer un fichier de configuration du dépôt /etc/yum.repos.d/local.repo contenant :

```
[local-repo]  
name=Local Repo  
baseurl=file:///opt/local-repo  
enabled=1  
gpgcheck=0
```

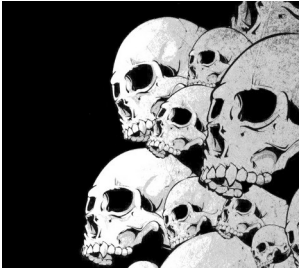
Copier les nouveaux paquets dans le dépôt local :

```
$ sudo cp ~/rpmbuild/RPMS/x86_64/mypackage.rpm /opt/local-repo/  
$ ...
```

Mettre à jour le dépôt :

```
$ createrepo --update /opt/local-repo  
$ dnf makecache
```





Signer un paquet

First, install dpkg-sig:

```
$ apt-get install dpkg-sig
```

Then sign your package(s):

```
dpkg-sig -k keyid --sign builder your_packages_${VERSION}_${ARCHITECTURE}.deb
```



Ressources

Guide du packageur :

<https://www.debian.org/doc/manuals/maint-guide/dreq.fr.html>

Utilisation de Quilt :

<https://wiki.debian.org/UsingQuilt>

Utilisation de pbuilder :

<https://doc.ubuntu-fr.org/pbuilder>

Configurer un dépôt local :

<https://wiki.debian.org/DebianRepository/Setup>

<https://wiki.debian.org/DebianRepository/SetupWithReprepro>

Configurer les dépôts :

<https://linuxhint.com/how-to-add-a-package-repository-to-debian/>