

La création d'un paquet Debian

Aurélien Jarno

Journées du Logiciel Libre

15 octobre 2005



Plan

La création
d'un paquet
Debian

Aurélien
Jarno



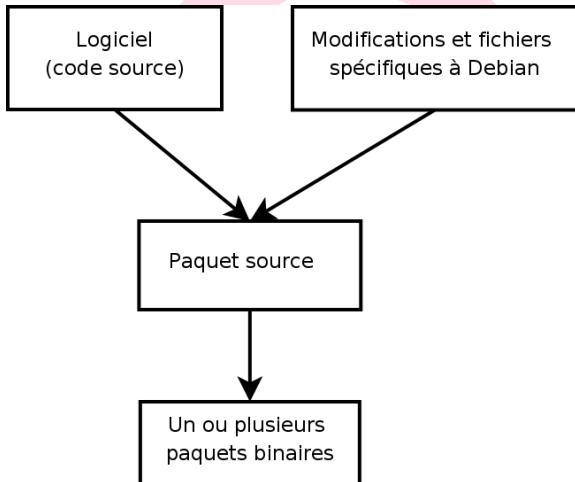
Les paquets Debian

Du code source aux binaires

Qu'est ce que créer un paquet Debian ?

La création
d'un paquet
Debian

Aurélien
Jarno



Les paquets sources

La création
d'un paquet
Debian

Aurélien
Jarno

Composés de trois fichiers :

- `.orig.tar.gz`
Le code source original, non modifié.
- `.diff.gz`
Les modifications apportés par Debian au format diff.
- `.dsc`
Informations sur le paquet source.
Somme MD5 des deux fichiers précédents.

NB : Il existe aussi des paquets sources dit « natifs » composés uniquement de deux fichiers, un fichier `.tar.gz` et un fichier `.dsc`, lorsque le paquet est spécifique à Debian.

Les paquets binaires (.deb)

La création
d'un paquet
Debian

Aurélien
Jarno

- Contiennent des logiciels, des bibliothèques ou des données dans leur version **compilée**
- C'est ce que l'utilisateur installe sur son système
- Deux types de paquets binaires :
 - Paquets communs à toutes les architectures (.all.deb)
exemple : fichiers textes, images, sons, etc.
 - Paquets spécifiques à une architecture (.i386.deb, .powerpc.deb)
exemple : exécutables, bibliothèques, headers, etc.

Les paquets binaires (.deb)

Un paquet binaire Debian est une archive ar contenant :

- `debian-binary`
Indique la version du format `.deb`
- `control.tar.gz`
Contient les informations liées au paquet (nom, version, description, dépendances, etc.) et éventuellement des scripts à exécuter lors de l'installation ou la désinstallation du paquet.
- `data.tar.gz`
Contient les données du paquet, i.e. tous les fichiers qui seront installés.

La phase de compilation du paquet source

Le répertoire `debian/` contient un certain nombre de fichiers nécessaires à la compilation du paquet. Il doit contenir au minimum :

- `rules`
Script principal de compilation du paquet.
- `control`
Description du paquet source, du ou des paquets binaires et relations avec les autres paquets.
- `changelog`
Historique des modifications du paquet.

C'est un fichier de type Makefile qui doit être exécutable et qui doit contenir les cibles suivantes :

- **build**
Cette cible doit exécuter les différentes étapes de la compilation du logiciel.
- **binary, binary-arch, binary-indep** Ces cibles doivent créer un ou plusieurs paquets binaires (.deb) à partir des fichiers générés à l'étape précédente.
- **clean** Cette cible doit effacer tout les fichiers créés lors de la compilation ou de la création des paquets binaires.

L'ensemble de ces opérations doit répondre à un certain nombre de règles qui se trouve dans la Charte Debian (Debian Policy Manual), ou dans les chartes annexes (python, perl, java, menu, etc...)

Les outils d'aide pour debian/rules

Écrire toutes les opérations nécessaires pour arriver jusqu'au `.deb` dans le fichier `debian/rules` est fastidieux. Heureusement des outils existent pour nous simplifier la vie :

- `debhelper`
Le plus utilisé, simplifie grandement le fichier `debian/rules`
- `yada`
Le plus ancien, aujourd'hui dépassé.
- `δbs` (surcouche à `debhelper`)
Aujourd'hui dépassé.
- `cdbs` (surcouche à `debhelper`)
Utile pour les paquets « classiques » (`./configure ; make ; make install`)

Exemple de Galette

Gestionnaire d'Adhérents en Ligne Extrêmement Tarabiscoté
mais Tellement Efficace...



Se faire une idée du logiciel

La création
d'un paquet
Debian

Aurélien
Jarno

- Télécharger le logiciel
- L'installer
- L'utiliser
- Regarder rapidement comme il fonctionne
- Éventuellement contacter l'auteur

Installation des outils nécessaires

- `apt-get install build-essential`
Le paquet `build-essential` contient le minimum nécessaire à la compilation d'un paquet (`make`, `gcc`, `libc-dev`, `dpkg-dev`).
- `apt-get install fakeroot`
Nécessaire pour compiler un paquet sans être `root`.
- `apt-get install devscripts`
Un ensemble d'outils facilitant la vie du mainteneur Debian.
- `apt-get install dh-make`
Utilitaire permettant de créer plus ou moins automatiquement un paquet source à partir des sources du logiciel.
- `apt-get install son-éditeur-favori`

Création du paquet

La création
d'un paquet
Debian

Aurélien
Jarno

- Lire les différentes chartes qui s'appliquent à se paquet
- Lancer `dh_make`
 - Décompacter les sources
 - Se placer dans la racine des sources
 - `dh_make -f ../nomdelarchive.tar.gz`
 - répondre aux questions
- Renommer/supprimer/modifier les fichiers du répertoire `debian/` en fonction du paquet
- Première compilation du paquet : lancer `dpkg-buildpackage` ou `debuild`
- Corrections éventuelles et retour à l'étape précédente
- Essais du paquet, et éventuellement retour à l'étape précédente

Vérification du paquet

Quand le paquet semble prêt, il est temps de faire vérifier le paquet par un vérificateur automatique. Il en existe deux dans Debian :

- `lintian`
Le plus utilisé, le plus performant
- `linda`
Une alternative possible, ou plutôt un complément

Tous deux comportent un certain nombre de règles, qui sont principalement dérivées de la Charte Debian.

Attention ! Un paquet pour lequel aucune erreur n'est détectée, n'est pas un paquet parfait. Inversement, une erreur détectée par un de ces outils n'est pas forcément une erreur. Avoir un sens critique.

Pour en savoir plus

La création
d'un paquet
Debian

Aurélien
Jarno

- <http://www.debian.org/devel/>
 - La charte Debian (Debian Policy Manual)
À lire absolument !
 - Le guide des nouveaux responsables de paquets
 - Les chartes spécifiques à certains paquets (perl, python, java, dictionnaires, etc.)
- `apt-get source`
Ne pas réinventer la roue, ne pas hésiter à aller voir comment les autres font.