

# Les scripts SlackBuild

Un script SlackBuild est un 'shell script' (généralement compatible avec le shell bash: Bourne-Again SHell.) dont le rôle est d'automatiser la configuration, la compilation et la création de paquets Slackware (des fichiers \*.tgz, \*.txz, \*.tbz) à partir d'archives du code source.

## Sommaire

La méthode générale pour créer un paquet Slackware à partir des sources peut être résumée comme suit, même si certains détails peuvent varier d'un système de construction à l'autre.

1. Décompresser l'archive source (habituellement un fichier .tar.gz ou .tar.bz2) dans un "répertoire de travail."
2. Créer un fichier slack-desc, qui décrit le logiciel suivant un format normalisé.
3. Configurer le fichier source, et appliquer les correctifs, si nécessaire.
4. Lancer la compilation/construction sur les fichiers-source afin de produire les fichiers binaires.
5. Installer le programme dans un répertoire provisoire. Cette étape peut s'avérer délicate, suivant quelle procédure dans `make install` (ou dans un outil semblable) est employée par le système d'agencement des paquets. Dans les Makefiles standards GNU, la variable `DESTDIR` passée à `make install` peut suffire. Dans le cas de Makefiles non-standards et de méthodes de construction de paquets où cette option n'est pas prévue, on devra envisager d'écrire des correctifs au Makefile (ou au fichier équivalent) on devra aussi adapter les réglages de la configuration du système de construction (build system).
6. Copier la documentation du programme et les autres fichiers nécessaires (y compris les scripts de post-installation) dans les sous-répertoires appropriés du répertoire temporaire d'installation. Le choix de ces emplacements doit respecter la structure adoptée par Slackware pour l'installation des programmes par le gestionnaire de paquets, à savoir dans le dossier `/usr/`.
7. Employer l'utilitaire `makepkg` depuis le répertoire temporaire (où se fait l'installation, suivant le présent exemple) pour créer un paquet Slackware qu'on peut ensuite installer en appelant la commande `installpkg`.

Bien qu'on puisse faire à la main, point par point, ce qui vient d'être décrit, la démarche est ennuyeuse, parfois compliquée, et elle nécessite la saisie au clavier de beaucoup de commandes. L'automatisation de ce travail par un script shell permet à l'utilisateur de partager son script, donc sa démarche, avec l'ensemble de la communauté. De plus il est logique d'utiliser et de modifier des scripts SlackBuilds existants pour la mise à jour d'un programme donné vers une version plus récente.

## Création de scripts SlackBuild

Un bon tutoriel qui montre comment créer un script SlackBuild est disponible [ici](#), et plus loin vous trouverez quelques exemples ou [modèles](#) qui permettent de voir précisément ce qu'on attend d'un script.

Cependant, les logiciels peuvent être écrits dans différents langages de programmation, utiliser différents systèmes de construction, ou peuvent même être écrits tout simplement dans un langage

interprété de sorte qu'on ne peut pas vraiment procéder à leur "installation". Dans ces cas là, un grand nombre d'étapes requises pour l'écriture d'un script SlackBuild différeront de la norme. Le copier-coller de commandes provenant d'autres scripts SlackBuild peut fonctionner, ou pas!

Un bon conseil à donner à quiconque envisage d'écrire un script SlackBuild est d'étudier et comprendre la structure d'un projet avant d'entreprendre l'écriture du script SlackBuild pour ce logiciel. Une bonne compétence dans l'écriture de shell scripts est également un bon atout, puisque en pratique l'écriture de SlackBuilds est faite avec des scripts compatibles avec le shell script 'bash' - Bourne-again shell. La façon idéale de commencer à écrire des scripts SlackBuild est de s'exercer avec des projets simples en C employant le système standard GNU Make.



Une bonne idée pour les novices en écriture de scripts est de suivre les étapes indiquées ci-dessus à la main, pour créer un paquet à partir des sources, avant de se lancer dans l'écriture d'un script SlackBuild. Il y aura ainsi moins de risques de bugs et de comportements inattendus provoqués par le script.

La présentation de scripts SlackBuild sur un dépôt public comme <http://www.slackbuilds.org> peut en outre exiger le respect de certaines conventions et normes de codage. Par exemple, ce sera au script de déterminer l'architecture-cible à l'exécution, ce qui est mieux que d'avoir codé "en dur" une architecture de machine dans ce script. Ces standards ont pour but de rendre les scripts SlackBuild aussi portables que possible.

## Aides automatisées à la création de SlackBuilds

Il existe quelques outils qui peuvent vous aider à la création de scripts SlackBuild. En voici deux exemples.

- [Alien's SlackBuild Toolkit \(AST\)](#)  
Un outil "magicien" fondé sur le web, de [Eric Hameleers](#)
- [mkslack](#)  
Un script créateur de SlackBuilds écrit par David Woodfall

## Liens Externes

- Dépôt de SlackBuilds - <http://www.slackbuilds.org/>
- Modèles de SlackBuilds - <http://slackbuilds.org/templates/>

## Sources

- Originally written by [V.Harishankar](#) for the SlackDocs Wiki Project, traduction française de [P-M. Averseng](#)

[slackware](#), [slackbuild](#), [pierreaverseng](#)

From:  
<https://docs.slackware.com/> - **SlackDocs**

Permanent link:  
[https://docs.slackware.com/fr:slackware:slackbuild\\_scripts](https://docs.slackware.com/fr:slackware:slackbuild_scripts)

Last update: **2017/05/08 08:16 (UTC)**

