

# Utilisation de cowbuilder

- Objet : Présentation, installation et configuration de cowbuilder
- Niveau requis : [avisé](#)
- Commentaires : *Pour compiler proprement vous-même des paquets Debian à partir d'un paquet source Debian.*
- À savoir : [Contribuer à Debian](#) 😊
- Suivi :
  - Création par [captnfab](#) 28/02/2014
  - Testé par [captnfab](#) le 28/02/2014
- Commentaires sur le forum : [ici](#) <sup>1)</sup>

## Introduction

Cowbuilder est un outil qui va permettre d'effectuer efficacement les étapes suivantes :

- Création d'un chroot contenant une Debian minimale
- Installation des dépendances de construction du paquet à construire
- Construction du paquet
- Sauvegarde du paquet généré
- Suppression du chroot

Pour faire cela efficacement, **cowbuilder** va générer la debian minimale une première fois, puis s'en servir de base à chaque utilisation. La préparation de l'environnement de construction du paquet se fait alors grâce à l'outil **pbuilder**, dépendance de cowbuilder.



Nous vous conseillons fortement d'associer à cowbuilder un proxy pour les paquets Debian.

- Voir: [apt-cacher-ng](#)

## Installation

Tout simplement :

```
apt-get install cowbuilder
```

## Configuration

Dans le dossier personnel de *user*, créer le fichier `.pbuilderrc` contenant:

~/.pbuilderrc

```

# this file extends or overrides /usr/share/pbuilder/pbuilderrc

# ----- get and set distribution and arch vars -----

UNSTABLE_CODENAME="sid"
TESTING_CODENAME="stretch"
STABLE_CODENAME="jessie"
STABLE_BACKPORTS_SUITE="$STABLE_CODENAME-backports"
DEBIAN_SUITES=($UNSTABLE_CODENAME $TESTING_CODENAME $STABLE_CODENAME
"unstable" "testing" "stable")
DEBIAN_MIRROR="httpredir.debian.org"

# Si la distribution n'est pas précisée mais qu'il y a un fichier
d/changelog, l'en déduire
if [ -z "${DIST}" ] && [ -r "debian/changelog" ]; then
    DIST=$(dpkg-parsechangelog | awk '/^Distribution: / {print $2}')
    # Use the unstable suite for certain suite values.
    if $(echo "experimental UNRELEASED" | grep -q $DIST); then
        DIST="$UNSTABLE_CODENAME"
    fi
fi

# Sinon, prendre la distribution du système
: ${DIST:="$(lsb_release --short --codename)"}

case "$DIST" in
    unstable)
        DIST="$UNSTABLE_CODENAME"
        ;;
    testing)
        DIST="$TESTING_CODENAME"
        ;;
    stable)
        DIST="$STABLE_CODENAME"
        ;;
esac

# Si l'architecture à utiliser n'est pas définie, prendre celle du
système.
: ${ARCH:="$(dpkg --print-architecture)"}

NAME="$DIST"
if [ -n "${ARCH}" ]; then
    NAME="$NAME-$ARCH"
    DEBOOTSTRAPOPTS=(" --arch" "$ARCH" "${DEBOOTSTRAPOPTS[@]}")
fi

if $(echo ${DEBIAN_SUITES[@]} | grep -q $DIST); then
    MIRRORSITE="http://$DEBIAN_MIRROR/debian/"

```

```

    COMPONENTS="main contrib non-free"
else
    echo "Unknown distribution: $DIST"
    exit 1
fi

# ----- overrides for pbuilder -----

BASETGZ="/var/cache/pbuilder/base-$NAME.tgz"
BASEPATH="/var/cache/pbuilder/base-$NAME.cow/"
BUILDPLACE="/var/cache/pbuilder/build/"
ALLOWUNTRUSTED=yes
BUILDRESULT="/var/cache/pbuilder/result/$NAME/"
DISTRIBUTION="$DIST"
APTCACHE="/var/cache/pbuilder/aptcache/$NAME/"
BUILDUSERID=65534
BUILDUSERNAME=nobody
EXTRAPACKAGES="apt-utils"
REMOVEPACKAGES="lilo"
HOOKDIR=/var/cache/pbuilder/hooks
unset DEBOOTSTRAPOPTS
unset CCACHEDIR
BINDMOUNTS="$BUILDRESULT"

# -----

if [ ! -d $BUILDRESULT ]; then
    mkdir -p $BUILDRESULT
fi
if [ ! -e $BUILDRESULT/Packages ]; then
    touch $BUILDRESULT/Packages
fi
if [ ! -e $BUILDRESULT/Release ]; then
    cat << EOF > $BUILDRESULT/Release
Archive: $DIST
Component: main
Origin: pbuilder
Label: pbuilder
Architecture: $ARCH
EOF
fi

# ---- END ----

```

Dans le dossier /var/cache/pbuilder, créer les dossiers hooks et repo:

```
mkdir -p /var/cache/pbuilder/{hooks,repo}
```

Créer les deux fichiers suivants:

[/var/cache/pbuilder/hooks/D05deps](#)

```
#!/bin/bash

# Before the chroot is started by pbuilder and friends, you can copy
# any .deb files into /var/cache/pbuilder/repo. The following commands
# creates an Packages.gz file, which is readable by apt. You usually do
# that when the .deb you want to build BuildDepends on other gems that
# are not yet in official/private/PPA repos.
# (cd /var/cache/pbuilder/repo; apt-ftparchive packages . > Packages &&
# gzip -c Packages > Packages.gz)

echo ""
echo "XXXXXXXXXXXXXXXXXXXX"
echo "Calling $0"
echo "XXXXXXXXXXXXXXXXXXXX"
echo ""

NAME="$DIST-$ARCH"
BUILDRESULT="/var/cache/pbuilder/result/$NAME/"

# create apt archive of previously built packages
#( cd $BUILDRESULT ; apt-ftparchive packages . > $BUILDRESULT/Packages
# )
cd $BUILDRESULT || exit 1
/usr/bin/dpkg-scanpackages . /dev/null > Packages

echo "===== ORIG sources.list"
cat /etc/apt/sources.list
echo "===== "

OTHERMIRROR="deb file://$BUILDRESULT /"
mv /etc/apt/sources.list /etc/apt/sources.list2
touch /etc/apt/sources.list
echo $OTHERMIRROR > /etc/apt/sources.list
cat /etc/apt/sources.list2 >> /etc/apt/sources.list
rm /etc/apt/sources.list2

echo "===== NEW sources.list"
cat /etc/apt/sources.list
echo "===== "

# apt-get will prefer the OTHERMIRROR repository
# (/var/cache/pbuilder/repo) over the MIRRORSITE repository, since it
# will appear first in /etc/apt/sources.list.
apt-get update
```

et

[/var/cache/pbuilder/hooks/B10lintian](#)

```
#!/bin/sh

# lintian_pdeb version 0.4
# a hook script to run lintian (from a pdebuild environment)
#
# Copyright (c) 2009-2012 Francesco Poli
#
# Permission is hereby granted, free of charge, to any person obtaining
# a copy of this software and associated documentation files (the
# "Software"), to deal in the Software without restriction, including
# without limitation the rights to use, copy, modify, merge, publish,
# distribute, sublicense, and/or sell copies of the Software, and to
# permit persons to whom the Software is furnished to do so, subject to
# the following conditions:
#
# The above copyright notice and this permission notice shall be
# included
# in all copies or substantial portions of the Software.
#
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND,
# EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF
# MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND
# NONINFRINGEMENT.
# IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY
# CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT,
# TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE
# SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

# Changed by Michael Franzl:
# 0.3->0.4 lintian user changed to nobody

LC_ALL=C
export LC_ALL

TMP_LOG=$(mktemp --tmpdir aptitude-log.XXXXXX)
APTITUDE_TLOG="aptitude -o Aptitude::Log=$TMP_LOG"
AOPTS="-q -y"
$APTITUDE_TLOG $AOPTS install awk dpkg-dev lintian
TO_BE_PURGED=$(awk -F] '/INSTALL/ { print $2 ; }' ${TMP_LOG})
cat $TMP_LOG >> /var/log/aptitude
rm -f $TMP_LOG

PKG_DIR=$(readlink -f ${HOME}/*/debian/..)
if test $? -ne 0
then
    echo " => ERROR! found the following debianized source
directories:"
    for DSD in ${HOME}/*/debian/..
    do
        readlink -f $DSD
    done
done
```

```

    echo "(there should be exactly one, instead)"
    exit 22
fi

cd $PKG_DIR
PKG=$(dpkg-parsechangelog |
    awk '/^Source: / { src = $2 ; }
        /^Version: / { vers = $2 ; print src "_" vers ; }')
ARCH=$(dpkg --print-architecture)
CHANGES_FILE=${PKG}_${ARCH}.changes

echo " => performing ftp-master-rejects lintian checks on
$CHANGES_FILE"
su -p nobody -c "lintian -viF ${HOME}/${CHANGES_FILE}"

echo " => performing general and experimental lintian checks on
$CHANGES_FILE"
su -p nobody -c "lintian -EviIL +pedantic ${HOME}/${CHANGES_FILE}"

echo " => cleaning up"
aptitude $AOPTS --purge-unused purge $TO_BE_PURGED

```

Rendre les deux fichiers exécutables :

```
chmod a+x /var/cache/pbuilder/hooks/{D05deps,B10lintian}
```

## Configuration de sudo

pbuilder et cowbuilder nécessitent d'être lancés avec les droits super-utilisateurs, avec certaines variables d'environnement. Cela peut être fait via sudo. Nous devons nous assurer que cela se passe bien, et pouvons même demander à ce que le mot de passe ne soit pas demandé à chaque fois (dangereux).

Rajouter ce texte dans un nouveau fichier `/etc/sudoers.d/01_pbuilders` permet de s'assurer cela.

[/etc/sudoers.d/01\\_pbuilders](#)

```

Cmd_Alias PBUILDERS = /usr/sbin/pbuilder, /usr/sbin/cowbuilder
Defaults! PBUILDERS setenv
#user ALL=NOPASSWD: PBUILDERS
user ALL=PBUILDERS

```



- Pour que le mot de passe ne soit pas demandé, commentez la ligne `user ALL=PBUILDERS` et décommentez la ligne avec `NOPASSWD`.
- Si vous préférez autoriser un groupe plutôt qu'un utilisateur, vous pouvez créer

le groupe *builders* et remplacer *user* par *%builders* dans le fichier ci-dessus.



Assurez-vous alors que votre utilisateur est bien membre de ce groupe et que vous avez relancé sa session.

## Préparation

Génération du chroot:

```
DIST=unstable; ARCH=amd64;  
sudo -E cowbuilder --create --architecture ${ARCH} --distribution ${DIST}
```

ou si vous utilisez un proxy pour les paquets :

```
DIST=unstable; ARCH=amd64;  
sudo -E cowbuilder --create --architecture ${ARCH} --distribution ${DIST} --  
http-proxy http://localhost:3142
```

```
chown -R user: /var/cache/pbuilder
```

## Entretien

De temps en temps, pensez à faire des mises à jour :

```
DIST=unstable; ARCH=amd64;  
sudo -E cowbuilder --update --architecture ${ARCH} --distribution ${DIST}
```

Vous pouvez par exemple définir une tâche *cron* hebdomadaire effectuant cette opération.

## Utilisation

Si vous êtes dans un dossier contenant les sources d'un paquet debian, la commande suivante le construit :

```
export DIST=unstable; export ARCH=amd64;  
pdebuild --pbuilder cowbuilder --debbuildopts '-d'
```



Pensez à remplacer *user* par votre *login* et assurez-vous de bien avoir configuré **sudo** comme indiqué ci-dessus.



Pour forcer la signature automatique des paquets par votre clé GPG<sup>2)</sup> :



```
export DIST=unstable; export ARCH=amd64; export  
AUTO_DEBSIGN=yes;  
pdebuild --pbuilder cowbuilder --debbuildopts '-d'
```

Les paquets créés ainsi que les fichiers `.changes`, `.dsc`, etc. sont placés automatiquement dans le répertoire parent.

## Références

- [How to build Debian packages correctly](#)
- [HOWTO: sudo + cowbuilder \(+git-buildpackage\)](#)

<sup>1)</sup>

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

<sup>2)</sup>

Voir comment [signer-un-paquet](#)

From:

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

Permanent link:

<http://debian-facile.org/doc:mentors:cowbuilder>



Last update: **04/08/2015 17:46**