

## RENDEZ-VOUS SUR ARRAKIS

*C'est lorsque nous croyons savoir quelque chose qu'il faut justement réfléchir un peu plus profondément. F. Herbert*

. .. Astuces diverses Cloner une debian  
Comment aider debian **Créer sa propre distribution  
avec live-build** Installer debian sans CD ni cle usb  
Installer debian sans ecran automatiquement Miroirs  
plus rapides Que faire en cas de bug et comment le  
resoudre Une debian dans une debian Utiliser  
reportbug asus netbook X102B

## CRÉER SA PROPRE DISTRIBUTION AVEC LIVE-BUILD

Live-build est un petit outil permettant de construire des images pour cdrom ou usb de debian. Vous pourrez pré-configurer cette «distribution perso» comme si c'était la vôtre, l'utiliser comme système "live", et aussi l'installer telle quelle.

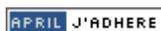
Cette page tentera d'apporter quelques éclaircissements sur l'utilisation de live-build, parfois obscure, en particulier sur la version 3.

Sachez toutefois que la meilleure documentation se trouve sur votre ordinateur. Une fois live-build d'installé, allez jeter un oeil à la documentation présente dans `/usr/share/doc/live-manual/html/`.

### Installation

Je vous conseille d'installer la version disponible dans les dépôts. Pour une version plus récente si vous êtes en "oldstable", utilisez les dépôts suivants :

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut



```
deb http://live.debian.net/ wheezy-  
snapshots main contrib non-free
```

Puis installez *live-build*, *live-manual* et *live-tools*.

```
apt-get update && apt-get install  
live-build live-manual live-tools
```

## Préparons le travail

On va créer un répertoire de travail, s'y déplacer, puis créer l'arbre de live build pour la suite :

```
mkdir masuperdebian  
cd masuperdebian  
lb config
```

Désormais sont présent dans ce dossier de nouveaux répertoires, tels que config.

## Configuration

La configuration se réalise par l'édition de plusieurs fichiers présents dans le répertoire config : binary, bootstrap, chroot, common. Cependant, comme les options de live-build peuvent changer lors du changement de version, nous allons plutôt utiliser la ligne de commande pour faire ça. Autrement dit, lorsque lb config sera lancé, il appliquera automatiquement certains réglages.

Tout d'abord, copiez les scripts clean, config et build présents dans /usr/share/live/build/examples/auto/ dans le répertoire auto du dossier masuperdebian :

```
$ cp /usr/share/doc/live-build/examples  
/auto/* auto/
```

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut



Avec une ancienne version, c'était `cp /usr/share/live/build/examples/auto/* auto/`.

Désormais, ce seront ces scripts qui seront utilisés lors des `lb build`, `live config`, etc. Nous les modifierons directement, afin d'éviter des lignes de commande à rallonge. Autrement dit, lorsque la commande `lb config` sera de nouveau exécutée, ce sera le script présent dans `/auto/config` qui sera lancé.

Voyons donc comment configurer le tout, en modifiant le fichier `auto/config`. Tout d'abord, voici à quoi il ressemble par défaut :

```
#!/bin/sh
lb config noauto \
"${@}"
```

Nous allons rajouter des options selon les besoin à la suite. Le caractère `\` permet de passer à la ligne sans souci. Les options seront donc rajoutées entre ces deux lignes :

```
#!/bin/sh
lb config noauto \
# AJOUTER DES OPTIONS ICI \
# ICI AUSSI \
# ET ENCORE ICI SI VOUS VOULEZ \
"${@}"
```

## Installer une liste de paquets personnalisée

Créez un fichier contenant la liste des paquets à installer, dans le dossier `config/package-lists`. Attention : le fichier doit avoir l'extension `.list.chroot`. Par exemple : `maliste.list.chroot`. Remplissez ce fichier avec les paquets vous intéressant.

## Préparer les fichiers de configuration de l'utilisateur (le home)

Vous souhaitez avoir vos marques pages de navigateur déjà prêt? Utiliser un thème graphique précis? Des raccourcis déjà tout prêts? Il s'agit de la configuration de l'utilisateur, généralement présente sous forme de fichier caché dans le /home/.

Pour avoir tout ça de déjà prêt, on va utiliser le dossier `config/includes.chroot` . Dans ce dernier, créez un dossier `etc/skel` . Il s'agit du dossier contenant tout ce qu'un utilisateur a dans son dossier personnel lorsqu'il est créé. Copiez dans le `config/includes.chroot/etc/skel` les fichiers de configuration, comme un `.bashrc`, ou un `.config/openbox ...` Tout ce que vous voulez!

L'ensemble du contenu de ce dossier sera rajouté à votre système personnalisé. Bien sûr, cette fonctionnalité fonctionne pour tout, pas seulement pour /etc/skel.

*note en passant* : L'utilisateur par défaut s'appelle **user**

## Un système en français

Dans `auto/config`, ajoutez ces options :

```
--bootappend-live "locales=fr_FR.UTF-8
keyboard-layouts=fr" \
--bootappend-install "locales=fr_FR.UTF-8"
\
```

## Préciser une autre distribution

Vous pouvez construire `sid`, ou `oldstable`, avec l'option `-distribution` :

## Utiliser autre chose que main

Vous pouvez ajouter les sections contrib et non-free (bouh!):

```
--archive-areas "main contrib non-free" \
```

## Installer un clone du système préconfiguré

Pour installer le système tel qu'il est sur votre clé sur un ordinateur, voici l'option à utiliser :

```
--debian-install "live" \
```

Puis assurez vous d'installer le paquet debian-install-launcher dans la liste des paquets à installer.

## Pour une clé usb, pas une iso cdrom

```
--binary-images "hdd" \
```

## Configurer l'utilisateur (nom d'utilisateur, groupes)

Pour configurer différents aspects de la session utilisateur, cela se passe en modifiant les paramètres de démarrage, c'est à dire en rajoutant des choses dans la partie après "boot" de cette ligne

```
--bootappend-live "locales=fr_FR.UTF-8  
keyboard-layouts=fr boot=live" \
```

Ainsi, pour changer les groupes et par exemple ajouter l'utilisateur au groupe fuse, rajoutez :

```
live-config.user-default-  
groups=audio,cdrom,dip,floppy,video,plugdev  
,netdev,powerdev,scanner,bluetooth,fuse`
```

```
--bootappend-live "locales=fr_FR.UTF-8
keyboard-layouts=fr boot=live user-default-
groups=audio,cdrom,dip,floppy,video,plugdev
,netdev,powerdev,scanner,bluetooth,fuse" \
```

Pour modifier le nom d'utilisateur, c'est avec `username=nom_d_utilisateur`

## Lancer des scripts pour configurer le système

Vous pouvez avoir besoin de lancer des scripts supplémentaire pour personnaliser un peu plus le système. Ces scripts doivent être lancés dans le chroot, avant que l'image soit construite.

Heureusement, tout est déjà prévu. Il suffit de placer ces scripts dans le dossier `config/hooks`. **Attention**, ces scripts doivent avoir l'extension `.chroot`.

Par exemple, pour changer l'alternative par défaut du terminal, on peut créer un script `alternatives.chroot` :

```
#!/bin/sh
set -e
update-alternatives --install /usr/bin
/x-terminal-emulator x-terminal-emulator
/usr/bin/urxvtcd 90
```

Dans le système, ce sera `urxvtcd` qui sera le terminal par défaut grâce à ce script.

## Persistence des données

Si vous souhaitez retrouver vos données entre chaque démarrage, vous pouvez créer ce qu'on appelle un live persistant. Pour cela, assurez-vous d'ajouter cette option :

```
--bootappend-live "persistence"
```

Ce qui peut donner avec d'autres paramètres :

```
--bootappend-live "persistence  
locales=fr_FR.UTF-8 keyboard-layouts=fr  
boot=live" \
```

Je passe rapidement sur la suite des détails puisqu'ils sont détaillés ensuite :

- Construction de l'image : `lb config && lb build`
- Copie de l'image : `# dd if=binary.hybrid.iso of=/dev/sdb`

On crée ensuite une nouvelle partition sur la clé avec `fdisk` ou `gparted`. Ensuite, on formate cette partition, par exemple en `ext4`. Le point important ici est de donner le label "persistence" à cette partition :

```
# mkfs.ext4 -L persistence /dev/sdb2
```

Pour finir, on va créer un fichier `persistence.conf` (anciennement `live-persistence.conf`) dans cette partition, et y préciser quel dossier on veut garder pour les démarrages suivants. Ici, on choisit de conserver le `/home`, qui contient toutes les données des utilisateurs :

- Montage de la partition : `# mount -t ext4 /dev/sdb2 /mnt`
- Création du fichier `persistence.conf` avec l'option voulue : `# echo "/home" >> /mnt/persistence.conf`
- Démontage de la clé : `#umount /mnt`

Et voilà, lors du premier démarrage sur la

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut

/dev/sdb2, autrement dit la deuxième partition de la clé. À chaque redémarrage suivant, vous retrouverez les changements réalisés au dernier lancement, sans autre manipulation.

Un autre exemple de fichier `persistence.conf` pour garder aussi les éventuels programmes installés à posteriori sur la clé, ainsi que la configuration de l'utilisateur (merci LeDub) :

```
/usr union
/home
/var/cache/apt
```

## Construction de l'image

Il suffit de lancer, toujours dans le répertoire `masuperdebian` :

```
lb config
lb build
```

`lb config` prépare l'image selon les options définies dans le fichier `auto/config`, et `lb build` fabrique l'image.

## Gravure/ copie sur usb

Une image iso est maintenant disponible dans le répertoire `masuperdebian`.

Si vous avez choisi le format `usb`, vous pouvez copier le tout sur une clé de cette façon :

```
dd if=binary.img of=/dev/sdb``
```

où **binary.img** est l'image de votre debian personnalisée, et **/dev/sdb** est le chemin vers

/dev/sdc2, mais seulement /dev/sdb. De plus, tout sera effacé sur votre clé. Assurez-vous de copier sur le bon périphérique!

Il se peut aussi que vous ayez plutôt **binary.hybrid.iso**, ce n'est pas un problème, il s'agit juste d'un format pouvant servir à la fois pour les clé usb que pour les cdroms.

## On recommence

---

La dernière image ne correspondait pas à vos attentes? On recommence alors. Tout d'abord, on nettoie le tout :

```
lb clean
```

Puis vous appliquez vos changements de configuration, et ensuite :

```
lb config
```

on recommence au début.

## Rendre les constructions futures plus rapides

---

Si vous planifiez de construire des ISO régulièrement, une bonne idée serait de mettre en cache les paquets localement. Installez simplement apt-cacher-ng et configurez la variable d'environnement http\_proxy avant la construction:

```
apt-get install apt-cacher-ng
/etc/init.d/apt-cacher-ng start
export http_proxy=http://localhost:3142/
.... # setup and configure your live build
lb config --apt-http-proxy
http://127.0.0.1:3142/
```

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut



D'après la documentation de Kali linux

## Exemple

```
#!/bin/sh

lb config noauto \
  --architectures "i386" \
  --linux-flavours "686" \
  --bootappend-live "locales=fr_FR.UTF-8
keyboard-layouts=fr" \
  --bootappend-install "locales=fr_FR.UTF-8"
\
  --binary-images "hdd" \
  --distribution "wheezy" \
  --archive-areas "main contrib non-free" \
  --apt-indices "false" \
  --apt-recommends "false" \
  --includes "none" \
  --memtest "none" \
  --win32-loader "false" \
  --source "false" \
  --debug \
  "${@}"
```

## Liste des options disponibles à lb config

Pour connaître toutes les options pouvant être ajoutées au fichier auto/config, tapez la commande `man lb_config`, et vous obtiendrez quelque chose du genre :

```
[--apt apt|aptitude]
[--apt-ftp-proxy URL]
[--apt-http-proxy URL]
[--apt-indices true|false|none]
[--apt-options OPTION|"OPTIONS"]
[--aptitude-options OPTION|"OPTIONS"]
[--apt-pipeline DEPTH]
[--apt-recommends true|false]
```

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut

```
[--apt-source-archives true|false]
[-a|--architectures ARCHITECTURE]
[-b|--binary-images iso|iso-
hybrid|net|tar|hdd|virtual-hdd]
[--binary-filesystem
fat16|fat32|ext2|ext3|ext4]
[--bootappend-install
PARAMETER|"PARAMETERS"]
[--bootappend-live PARAMETER|"PARAMETERS"]
[--bootloader grub|syslinux|yaboot]
[--bootstrap cdebootstrap|cdebootstrap-
static|debootstrap|copy]
[-f|--bootstrap-flavour minimal|standard]
[--bootstrap-keyring PACKAGE]
[--cache true|false]
[--cache-indices true|false]
[--cache-packages true|false]
[--cache-stages STAGE|"STAGES"]
[--checksums md5|sha1|sha256|none]
[--compression bzip2|gzip|lzip|none]
[--build-with-chroot true|false]
[--chroot-filesystem
ext2|ext3|ext4|squashfs|jffs2|none]
[--clean]
[-c|--conffile FILE]
[--debconf-frontend
dialog|editor|noninteractive|readline]
[--debconf-nowarnings true|false]
[--debconf-priority
low|medium|high|critical]
[--debian-installer
true|cdrom|netinst|netboot|businesscard|liv
e|false]
[--debian-installer-distribution
daily|CODENAME]
[--debian-installer-preseedfile FILE|URL]
[--debian-installer-gui true|false]
[--debug]
[-d|--distribution CODENAME]
[--parent-distribution CODENAME]
[--parent-debian-installer-distribution
CODENAME]
```

Le 10/06/2017 à 14:38 avec swx - Contact - Plan - OpenBSD - ↑ En haut

## Références

---

- <http://www.esdebian.org/wiki/live-helper>
- <http://live.debian.net/>
- <http://live.debian.net/devel/live-build/>