



Live Debian-Facile : DIY

- Objet : Construction du [Live-DF](#)
- Commentaires : Documentation complémentaire au [manuel simplifié live-build](#)
- Suivi :
 - Création par  [arpinux](#) le 21/10/2016
- Membres :
 -  [arpinux](#)
- Dépôt git :
 - URL Gitweb live-df/doc : <https://git.framasoft.org/Live-DF/live-df/blob/master/doc>
- Commentaires sur le forum : [FIXME^{1\)}](#)

Cette page relate le processus complet de construction du live-df Ces sources permettent de construire les versions i386 (586/686-pae) et amd64.

Conseils :

- suivez le tuto jusqu'au clonage des sources, puis affichez les sources à coté du tuto pour visualiser rapidement la structure du build et le rôle de chaque fichier.
- utilisez de préférence un navigateur de fichiers en CLI comme midnight-commander ou ranger qui permettent de visualiser rapidement le contenu (aperçu du texte), les permissions, cibles des liens etc
- amusez-vous 😊

need help ? arpinux@member.fsf.org

Introduction

Le live-df est construit sur 2 ressources :

- les paquets Debian présents dans les dépôts du [sources.list](#),
- les sources du build à utiliser avec le programme live-build.

Les paquets sont appelés pas le fichier `/config/package-lists/livedf.list.chroot` et peuvent être mis à jour après installation.

Les sources modifient 'en dur' la construction de l'image ISO, soit en intégrant des fichiers de configuration, soit en modifiant la construction grâce à certains scripts.

Principe du live-cd

Live-DF est distribué sous forme d'image ISO destinée à être gravée ou transférée sur une clé USB afin de créer un 'live-cd'.

un live-cd est constitué d'une image système (toute l'architecture embarquée d'une installation) compressée dans une archive de type "squashfs" et d'un programme de boot permettant de lancer ce système compressé.

ce programme de boot peut être accompagné d'un installeur, permettant ainsi de reporter le système

compressé sur un disque dur. dans ce cas, c'est un live-cd-installable. Live-DF utilise live-build, le programme de construction des images disque Debian.

Principe du live-build

live-build est constitué d'une série de scripts destinés à construire une image live et/ou installable d'un système Debian.

ces scripts acceptent des arguments permettant de personnaliser l'image finale.

les arguments peuvent être réunis dans des scripts qui sont placés dans le dossier 'auto' des sources.

Processus de construction

- lorsqu'on lance la commande de construction, live-build lit les instructions que vous avez laissé.
- live-build détecte l'architecture, le noyau à utiliser etc, puis il utilise debootstrap pour créer un chroot (un système dans le système) de base.
- ensuite il complète l'installation dans le chroot avec les paquets listés dans le /config/package-lists/blah.list.chroot .
- tous les paquets sont mis en cache dans un dossier qui se crée au moment de la construction directement à la racine des sources (/cache).
- une fois le chroot installé, live-build va reporter le contenu de /config/includes.chroot/ (votre personnalisation du système) dans le chroot.
- viennent ensuite les 'hooks', le(s) script(s) qui vont modifier le chroot juste avant de le démonter.
- puis live-build démonte le chroot et commence à le compresser en squashfs (la partie la plus gourmande et la plus longue).
- enfin, il va chercher les paquets nécessaires pour l'installeur et la création de l'iso (isolinux, xorriso, etc) et génère l'image ISO.

... en très gros :D

Mise en place

Vous aurez besoin d'environ 8G d'espace libre sur votre système pour construire les images ISOs live-df

Installation des dépendances

Live-df se sert programme de construction live-build.

Pour ses sources, c'est le protocole git qui est utilisé :

```
apt-get update
```

```
apt-get install live-build live-manual live-tools git make
```

Clonage des sources

Pour utiliser et modifier les sources localement, vous devez les cloner sur votre système.

```
git clone https://git.framasoft.org/DFLinux/live-df.git
```

Vous obtenez alors un dossier live-df qui contient l'intégralité des sources du live-df.

Contenu des sources du build

liste des dossiers et fichiers des sources Live-DF
aka koya dedans... kesafula ??

- /Makefile
script de lancement de la construction.
permet de construire les deux versions i386 et amd64 en précisant le type de construction en argument. lancer 'make' pour la liste des options disponibles.
- /README.md
fichier de présentation générale Live-DF.
- /TODO.md
les trucs à faire.
- /VERSION.md
fichier indiquant la version courante des sources et la date de sortie.
- /latotale
script de construction de toutes les ISOs (à lancer avant d'aller se coucher :P)

/addons/ : dossiers et fichiers spécifiques selon le type d'ISO construite. ces données sont copiées dans les sources du build par le Makefile.

/auto/ : contient les scripts de construction et de nettoyage pour les commandes 'lb config', 'lb build' et 'lb clean'. ces scripts permettent de passer en arguments de façon plus claire, les options pour live-build.

- build : script de construction qui va lire le fichier 'config' et porter le retour du build dans un log.
- clean : script de nettoyage du dossier de construction.
- config : script principal de construction contenant la grande majorité des options passées à live-build. il y a plusieurs 'config' pour pouvoir construire les différentes variantes du live-df.

/config/hooks : dossier contenant les scripts de modification en cours de build.

- dflinux.chroot
script exécuté en cours de build, alors que le squashfs n'est pas encore créé. ce script met à jour la base de donnée de "command-not-found"

/config/includes.binary/isolinux/ : fichiers intégrés dans le dossier 'isolinux' du livecd.
les fichiers présents servent à la personnalisation du menu isolinux, le menu d'ouverture du liveDVD.
Ils seront complétés lors du build par les fichiers tirés du dossier 'addons' et copiés par le Makefile.

/config/package-lists/ : dossier contenant les listes des paquets à installer.

si vous ajoutez une liste maliste.live, les paquets contenus seront présents dans la session live mais seront supprimés lors de l'installation.

- **livedf.list.chroot**
liste des paquets par défaut pour DFLinux les lignes '#' ne sont pas prises en compte. la liste est copiée depuis les addons en début de build en fonction de la variante spécifiée et n'est donc pas présente à son emplacement dans les sources.
- **live.list.chroot**
liste créée automatiquement par live-build afin d'intégrer les paquets live-build dans le liveDVD.

/config/includes.chroot : tout ce qui sera placé dans ce dossier sera intégré dans le squashfs en respectant l'architecture et les permissions du système GNU/Linux de base.
ex, un fichier placé dans /config/includes.chroot/etc/issue appartiendra à root dans le squashfs et sera situé dans /etc/issue

! attention ! rien de ce qui sera placé dans ce dossier en 'dur' ne pourra être mis à jour depuis le gestionnaire de paquets : préférez la modification depuis les paquets.

/config/includes.chroot/etc/skel/

dossier de configuration de l'utilisateur par défaut (skel=skeleton). ce dossier sera détaillé plus tard.

/config/includes.chroot/usr/share/live-df/


ce dossier contient les cahiers du débutant et les petits outils ajoutés pour faciliter la prise en main.

/config/includes.chroot/etc/skel : la configuration de l'utilisateur par défaut.

- dossiers et fichiers cachés (certains sont copiés depuis les addons)
 - /.config/autostart/ : lanceurs d'applications à démarrer automatiquement. le lanceur "welcome.desktop" sera supprimé lors de la première connexion.
 - /.config/gtk-3.0 /gtk-2.0 : préférences GTKpar défaut pour l'utilisateur. les thèmes définis sont installés par handylinuxlook.
 - /.config/mimeapps.list : définition des applications par défaut selon le type de fichier.
 - /.config/Trolltech.conf : fichiers de configuration de l'apparence des logiciels 'Qt' type VLC.
 - /.config/user-dirs.dirs : définition des dossiers XDG par défaut.
 - /.config/user-dirs.locale : définition de la langue des dossiers XDG.
 - /.fluxbox/ : configuration pour fluxbox.
 - /.gtkrc-2.0 : préférences de thème pour fluxbox issues de lxappearance.
- dossiers et fichiers visibles
 - /Documents/ : liens vers /usr/share/live-df/les_cahiers_du_debutant.pdf et fluxbox_initiation.pdf.
 - /Images/wallpapers : lien vers le dossier système des fonds d'écran.
 - /Modèles/ : différents modèles pour le menu contextuel de thunar "créer un document...".
 - /Musique/Jamendo.desktop : lanceur pour le site Jamendo.
 - /guide.desktop : lanceur pour les cahiers du débutant.
 - /welcome.desktop : lanceur pour afficher de nouveau le message d'accueil.

les dossiers vides ne sont pas pris en compte par git, pour les afficher dans les sources, j'ai placé des fichiers cachés ".file" pour résoudre le soucis. Si vous avez une autre méthode, je suis preneur :)

Modification des sources

Cf le PDF HOWTO-livebuild.pdf intégré aux sources de live-df pour une vulgarisation de live-build et de la personnalisation d'un live-cd.  : intégrer le howto livebuild sur DF

Construction de L'ISO

La construction se lance depuis le Makefile intégré à lancer en root.

`nom.sh`

```
# live-df Makefile <https://debian-facile.org/projets:live-df>

all:
    @echo "Usage: as root"
    @echo "make 586          : build live-df 586"
    @echo "make 686          : build live-df 686-pae"
    @echo "make 64           : build live-df amd64"
    @echo "make clean        : clean up build directories"
    @echo "make cleanfull    : clean up cache directories"

586:
    @echo "building live-df 586"
    # copie du script principal du build
    cp auto/config586 auto/config
    # copie du menu de boot du live
    cp addons/live586.cfg config/includes.binary/isolinux/live.cfg
    # construction du live
    lb build
    # renommer les fichiers
    mkdir -p live-df-586
    mv live-image-i386.hybrid.iso live-df-586/live-df-586.iso
    mv chroot.packages.install live-df-586/live-df-586.pkgs
    mv *.log live-df-586/live-df-586.log
    md5sum live-df-586/live-df-586.iso > live-df-586/live-df-586.md5

686:
    @echo "building live-df 686-pae"
    # copie du script principal du build
    cp auto/config686 auto/config
    # copie du menu de boot du live
    cp addons/live686.cfg config/includes.binary/isolinux/live.cfg
    # construction du live
    lb build
    # renommer les fichiers
    mkdir -p live-df-686
    mv live-image-i386.hybrid.iso live-df-686/live-df-686-pae.iso
    mv chroot.packages.install live-df-686/live-df-686-pae.pkgs
```

```
mv *.log live-df-686/live-df-686-pae.log
md5sum live-df-686/live-df-686-pae.iso > live-df-686/live-df-686-
pae.md5

64:
@echo "building live-df amd64"
# copie du script principal du build
cp auto/config64 auto/config
# copie du menu de boot du live
cp addons/live64.cfg config/includes.binary/isolinux/live.cfg
# construction du live
lb build
# renommer les fichiers
mkdir -p live-df-64
mv live-image-amd64.hybrid.iso live-df-64/live-df-amd64.iso
mv chroot.packages.install live-df-64/live-df-amd64.pkgs
mv *.log live-df-64/live-df-amd64.log
md5sum live-df-64/live-df-amd64.iso > live-df-64/live-df-amd64.md5

clean:
@echo "cleaning build directories"
# nettoyage du script principal du build
rm -f auto/config
# nettoyage du menu de boot du live
rm -f config/includes.binary/isolinux/live.cfg
# nettoyage du live
lb clean

cleanfull:
@echo "cleaning cache directories"
rm -R -f cache
```

Vous obtenez un dossier nommé selon la version construite avec l'ISOs, le log, la liste des paquets et la somme md5.

... et voilà 😊

des questions ? arpinux@member.fsf.org

1)

N'hésitez pas à * intervenir !

From:

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

Permanent link:

<http://debian-facile.org/projets:live-df:diy>

Last update: **16/02/2019 07:58**

