

[Pour créer une ISO de base](#)

[Pour faire une ISO particulière :](#)

[Mettre son logo en fond d'écran :](#)

[Les répertoires :](#)

[Blocages :](#)

[Astuces avancées :](#)

Pour créer une ISO de base :

Il faut s'installer une Débian 7.1 ou plutôt une Ubuntu 14.04 avec une partition home de 50Go, puis

Se créer un compte utilisateur avec un nom générique (parce que celui-ci sera utilisé dans l'ISO) et dans ce mode utilisateur. J'insiste:créer un compte spécifique, parce que beaucoup de fichiers seront inscrit sur le compte.

Rajouter le programme Git:
sudo apt-get install git

Installer ce qu'il faut :
cd \$HOME

(Sur Voyager-Ubuntu 14.04, le programme install est remplacé par autre chose via un Alias, donc, il faut le dé-aliaser.)

```
install -d src
cd src
git clone https://bitbucket.org/yledoare/dibab.git
cd dibab
cd 2.3
sudo sh configure
```

(Installe ce qu'il manque : au moins 350Mo sont installés, mais il se peut que lors de la compilation, vous vous aperceviez qu'il en manque encore quelques uns.)

Si on fait une modification sur les scripts, mettre à jour par :

```
cd $HOME/src/dibab/2.3
git pull
```

Et relancer le make ou le make spécifique

par défaut, la compilation se lancera pour x86 (x86_64, si c'est l'architecture de votre ordinateur)

Un fichier config.ini est généré, il suffit de modifier le numéro de noyau Linux dans ce fichier si vous voulez une autre version que celle proposée par défaut. **Important** : Pour une ISO dérivée, remplacer "linuxconsole" par le nom de votre distribution dérivée.

Pour faire une ISO particulière :

Rajouter les dépendances nécessaires (voir le fichier de script)

Exemple Linuxfr.org (et il doit en manquer):

```
apt-get install libalien-sdl-perl libtinymce-dev libxi-dev libxrandr-dev libfreetype6-dev
libglew-dev libjpeg62-dev libopenal-dev libsndfile1-dev libghc-sdl-image-dev
libghc-sdl-mixer-dev liblzma-dev libvlc-dev
```

Si on a besoin du compilateur langage D (jeux Japonais de Kenta Cho ou Jumpei Isshiki):

```
apt-get install gdc
```

Dans config.ini, remplacer "linuxconsole" par le nom de votre distribution dérivée.

Et lancer :

```
cd $HOME/src/dibab/2.3  
git pull
```

```
make lanpower
```

(Pour l'ISO LanPower)

Si tout se passe bien, vous aurez (au bout de 40 heures, environ...sur un Pentium 2Ghz) un fichier ISO (300Mio,pour la branche principale compatible unetbootin), avec un environnement LXDE et la gestion du réseau via NetworkManager.

Mettre son logo en fond d'écran :

Convertir l'image en png et l'appeler du nom de sa dérivée par exemple lanpower.png et la placer dans \$HOME_DIBAB/logos où \$HOME_DIBAB est votre répertoire /home suivi du nom de login (utilisateur) puis de src/dibab/2.3 pour la version 2.3 de dibab ce qui chez moi donne :

```
/home/lg/src/dibab/2.3/logos
```

Les répertoires :

L'ISO se trouve dans le répertoire utilisateur (\$HOME) sous répertoire dibab/iso

Les fichiers temporaires dans \$HOME/dibab/tmp

Les fichiers téléchargés dans \$HOME/dibab/tarballs . Les .deb (paquages Debian) sont décompressés dans ce même répertoire.

Les autres sont décompressés dans \$HOME/dibab/src

Les fichiers source à installer sont donc dans \$HOME/dibab/src

Dans \$HOME/dibab/build-\$ARCH (remplacer \$ARCH par l'architecture que vous avez choisie : exemple x86), ce sont les packages, installés en général avec "make install DESTDIR=\$HOME/dibab/build-\$ARCH" . Intérêt : on pourrait faire un .opk avec chaque package (comme les autres distribs), et par ex, si je modifie un fichier dans \$HOME/\$ARCH/bin, je peux ensuite "restaurer" l'original depuis \$HOME/dibab/build-\$ARCH

Une fois compilés, ils sont mis dans \$HOME/dibab/packages-\$ARCH

Les packages de votre distribution particulière se mettent dans \$HOME/nomdeladistrib par exemple \$HOME/lanpower

Dans \$HOME/dibab/opkg on a les programmes supplémentaires accessibles par internet.

Dans \$HOME/src/dibab/ on a l'ensemble des scripts Dibab pour faire les ISO (avec un N° de version comme sous répertoire).

Dans \$HOME/\$ARCH on a les packages compilés pour le module "core" (\$ARCH.squashfs) : on fait ./configure --prefix=\$HOME/\$ARCH . Ca pourrait être ailleurs ...

Les logs de l'extraction des tarballs sont dans \$HOME/dibab/tmp, ainsi que l'initramfs.

Blocages :

De nombreux blocages peuvent se produire dus à des dépendances manquantes, des archives absentes (sur l'ordinateur de compilation ou sur la compilation en train de se construire), un changement de version d'archive ou des archives non standard, qui nécessitent des adaptations des scripts : contacter Patrice et Yann.

Astuces avancées :

On peut lancer un make et un make spécifique en même temps pour gagner du temps, mais le make spécifique peut avoir besoin d'une librairie compilée dans le make principal. De plus, si l'un des 2 make s'arrête (break), l'autre s'arrête aussi. Donc, il faut relancer

1 - je veux supprimer complètement le paquet bzip2-1.0.6

```
sh scripts/uninstall-package bzip2-1.0.6  
make # ça recompilera et installera
```

2 - je veux compiler "à la main"

```
BUILDME=OK make  
# on se retrouve dans le dossier de compilation du paquet, on peut  
faire des ./configure et des make pour tester des options
```