

---

# Documentation constructeur

*Version 0.1.3-4b-jessie*

**Gregory DAVID <[gregory.david@ac-nantes.fr](mailto:gregory.david@ac-nantes.fr)>**

10/05/2016



<b>1</b>	<b>Prologue</b>	<b>1</b>
1.1	Nomenclature . . . . .	1
1.2	Numérotation des versions . . . . .	1
<b>2</b>	<b>Mécanisme général d’une construction</b>	<b>3</b>
<b>3</b>	<b>Mise en œuvre de la construction</b>	<b>5</b>
3.1	Pré-requis . . . . .	5
3.2	Récupération de la configuration initiale et des outils accessoires . . . . .	6
3.3	Choix de la configuration spécifique . . . . .	6
3.4	Construction de l’image système dans sa version v0.1.3 . . . . .	6
<b>A</b>	<b>Nomenclature</b>	<b>7</b>
<b>B</b>	<b>Pré-requis</b>	<b>9</b>
<b>C</b>	<b>Contributions</b>	<b>11</b>



---

# Prologue

---

Cette documentation est une simplification de la documentation officielle de [live-build](http://live-build.1024gr.org/)<sup>1</sup> et est adaptée à la configuration fournie par ce projet.

## 1.1 Nomenclature

Voir [Nomenclature](#) (page 6) pour des explications sur la nomenclature concernant les lignes de commande.

## 1.2 Numérotation des versions

Nous nous basons sur le principe de [numérotation sémantique des versions](#)<sup>2</sup> proposé par [Tom Preston Werner](http://tom.preston-werner.com/)<sup>3</sup>.

---

1. <http://debian-live.alioth.debian.org/live-manual/unstable/manual/html/live-manual.en.html>  
2. <http://www.semver.org>  
3. <http://tom.preston-werner.com/>



---

## Mécanisme général d'une construction

---

La construction d'une clef USB *bootable* requiert d'assurer les étapes suivantes :

- **Installation** des *Pré-requis* (page 9) sur votre système de construction
- *Récupération de la configuration initiale et des outils accessoires* (page 6)
- *Choix de la configuration spécifique* (page 6) (RC ou entière)
- construction

---

**Note : Les 3 stades de la construction**

- la construction d'une **version distribuable** (mise en production)
- la construction d'une **RC** qui mènera à une distribuable (test et validation)
- la construction d'une **intermédiaire** qui mènera à une RC (développement)

Quel que soit le stade de la construction, le mécanisme est toujours le même. La variance d'un stade à l'autre correspond à la position dans l'arbre des configurations figées (sur quel `tag` nous plaçons-nous pour réaliser la construction).

---





---

## Mise en œuvre de la construction

---

Voir la vidéo de démonstration<sup>4</sup> dont les étapes sont détaillées ci-après.

### 3.1 Pré-requis

Les exigences pour la création ou la distribution de l'images système **liveUSB** sont faibles :

- Accès super-utilisateur (`root`) sur le système
- Une version à jour de `git`
- Une version mise à jour de `live-build` (voir `nondeb主an`)
- Une version à jour de `debootstrap`
- Un shell `bash`
- Un noyau Linux 2.6.x ou supérieur
- Une version à jour de `whiptail` ou `dialog`
- Une version à jour de GNU `make`
- Une version à jour de `syslinux` fournissant l'EFI (`syslinux-efi` pour Debian)
- Une version à jour de `sfdisk`, version 2.26 minimum

#### 3.1.1 Super-utilisateur ?

La commande suivante permet de connaître le nom de l'utilisateur que nous sommes :

```
# whoami
root
```

si l'utilisateur n'est pas `root`, alors il faut s'assurer que nous sommes un utilisateur faisant partie du groupe `sudo` :

```
$ groups
lambda cdrom floppy sudo audio video plugdev ...
```

Si dans la sortie de la commande `groups` vous n'apercevez pas le mot clef `sudo`, c'est que vous ne faites pas partie du groupe `sudo`. Dans ce cas, vérifiez avec l'administrateur de votre système afin d'obtenir un compte vous permettant de réaliser en tant que super-utilisateur les commandes suivantes : `lb` et `apt-get`.

#### 3.1.2 Installer les logiciels nécessaires

```
# apt-get update && apt-get install git live-build debootstrap make
```

ou avec `sudo`

---

4. <https://www.youtube.com/watch?v=mSmJwO6gATY>

```
$ sudo apt-get update && sudo apt-get install git live-build debootstrap make
```

## 3.2 Récupération de la configuration initiale et des outils accessoires

```
$ cd ~  
$ git clone https://framagit.org/groolot/liveUSB.git  
$ cd liveUSB  
$ ls  
auto burn config documentation Makefile README.md
```

## 3.3 Choix de la configuration spécifique

### 3.3.1 Lister les configurations disponibles

```
$ git tag  
v0.1  
v0.1.1  
v0.1.2  
v0.1.3
```

## 3.4 Construction de l'image système dans sa version v0.1.3

```
# make v0.1.3-build
```

ou bien avec sudo

```
$ sudo make v0.1.3-build
```

À l'issue de la construction, l'image système se trouve disponible dans le répertoire courant portant le nom `liveUSB-v0.1.3-amd64.img` (ou tout autre numéro de version en fonction du choix de la configuration fait au préalable).

---

## Nomenclature

---

Les lignes de commande commençant par \$ doivent être saisies par l'utilisateur lambda, exemple :

```
$ cd ~  
$ ip link show
```

Les lignes de commande commençant par # doivent être saisies par le super utilisateur, ou en tant qu'utilisateur avec élévation éphémère de pouvoir (sudo), exemple :

```
# touch /etc/shadow  
# rm -rf /tmp/*
```

ou l'équivalent avec sudo :

```
$ sudo touch /etc/shadow  
$ sudo rm -rf /tmp/*
```

Les lignes ne commençant pas par \$ ou # **ne doivent pas être saisies** car elles sont les affichages de sortie des commandes précédentes, exemple :

```
$ cat /etc/hostname  
ISNconstructeur  
# ls /var  
backups  cache  games  lib  local  lock  log  mail  opt  run  spool  tmp
```



---

## Pré-requis

---

**Les exigences pour la création ou la distribution de l'images système `liveUSB` sont faibles :**

- Accès super-utilisateur (`root`) sur le système
- Une version à jour de `git`
- Une version mise à jour de `live-build` (voir `nondeb主an`)
- Une version à jour de `debootstrap`
- Un shell `bash`
- Un noyau Linux 2.6.x ou supérieur
- Une version à jour de `whiptail` ou `dialog`
- Une version à jour de GNU `make`
- Une version à jour de `syslinux` fournissant l'EFI (`syslinux-efi` pour Debian)
- Une version à jour de `sfdisk`, version 2.26 minimum



---

## Contributions

---

### Documentations

- Vincent LABBÉ <[vincent.labbe@ac-nantes.fr](mailto:vincent.labbe@ac-nantes.fr)>

### Tests et validations

- Vincent LABBÉ <[vincent.labbe@ac-nantes.fr](mailto:vincent.labbe@ac-nantes.fr)>
- Dimitri TRICARD <<https://git.framasoft.org/u/D.Tricard>>