

# Dump et backup d'un appareil Android avec Debian et ses dérivés

- Objet : créer un dump de son smartphone ou tablette sous Android
- Niveau requis :  
[débutant](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Commentaires : *Contexte d'utilisation du sujet du tuto.* 🛠️ **Fix Me!**
- Suivi :  
[à-tester](#)
- Création par [👤 slyfox](#) 05.06.2016

## Modèles testés

Vous pouvez inscrire les appareils dans le tableau ci-dessous qui ont testé avec ce wiki.

Marque	Modèle	Type	Remarque
Trekstor	M-Budget	Tablette	RAS

## Introduction

Le dump ou nandroid est une copie brut d'un système. Il peut s'avérer fort utile en cas de plantage de d'un smartphone ou d'une tablette sous Android comme par exemple suite à une fausse manipulation ou dans le cas où l'appareil reste bloquée sur le logo de démarrage. Réinstaller ou flasher le firmware reste alors la seule manière de refaire fonctionner l'appareil.

Le problème c'est que pour de nombreux appareils sous Android il n'existe pas forcément de firmware disponible en particulier pour les tablette bas prix (low-cost). Le dump ou un backup de l'appareil et la réinstallation via le recovery s'avère dans ce cas une manière de solutionner ce problème.

Il est tout a fait possible de suivre ce wiki si vous avez un recovery custom comme par exemple ClockWorkMod (CWM) ou TeamWin Recovery Project (TWRP) qui en passant permettent de faire un backup de l'appareil plus facilement de ce qui est expliqué dans ce wiki, mais ce wiki peut fonctionner si on a un recovery custom seul la réinstallation sera légèrement différente. La réinstallation avec un recovery custom ne sera pas expliqué dans ce wiki, internet est votre ami.

Il est peut-être judicieux de faire un dump de l'appareil (avec toutes les précautions qui s'imposent) à la sortie de sa boîte ou dans le cas où l'on veut retrouver son appareil dans l'état qu'on l'avait laissé avant qu'il plante ou pour toutes autres raisons.

Dans ce wiki nous allons voir comment:

- Installer les outils nécessaires.
- Préparer notre appareil.
- Visualiser les diverses partitions.
- Extraire les partitions.
- Copier les partitions sur un ordinateur.

- Créer l'archive zip.
- Signer l'archive zip.
- Réinstaller le système via le recovery d'origine ou du fabricant.

## Vocabulaire

Voici quelques définitions des termes techniques que nous allons rencontrer dans ce wiki:

### **ADB**

L'Android Debug Bridge (ADB) est un outil qui se compose d'un programme client et d'un programme serveur qui communiquent entre eux. En d'autre terme c'est un outil qui permet de communiquer entre l'appareil Android et l'ordinateur.

---

### **Backup**

En informatique, c'est un emprunt de l'anglais pour signifier une sauvegarde de données.

---

### **Dump ou nandroid**

En informatique, désigne généralement une copie brute (sans transformation) de données d'un périphérique à un autre, par exemple pour effectuer une sauvegarde de données.

---

### **Linux**

Dans le langage courant on trouve souvent l'emploi du terme « Linux » seul pour désigner une distribution du système d'exploitation GNU/Linux (Debian, Ubuntu, Linux Mint, Fedora...) ce qui peut porter à confusion étant donné qu'il existe quelques systèmes fondés sur Linux mais pas sur GNU comme par exemple Android. En général Linux désigne le noyau ou le kernel (cœur) du système portant le même nom.

 *Tux, la mascotte du noyau Linux.*

---

### **Recovery**

Le recovery ou récupération en français est un menu ou un mode de démarrage du mobile qui permet d'effectuer différentes opérations dont entre autres:

- Installer une archive spéciale (thème, mod, patch ou autre)
- Effacer les données utilisateur de la mémoire interne
- Effacer le cache du mobile
- ...

## **Recovery custom**

Le recovery custom est un recovery qui a été développé pour faciliter certaines opérations de maintenances. Les plus connus sont ClockWorkMod (CWM) ou TeamWin Recovery Project (TWRP).

---

## **ROM, Firmware, ROM custom**

Comme Windows, MacOS ou les distributions GNU/Linux, Android est le système d'exploitation de notre appareil. Dans le cas d'une ROM personnalisée on parle de ROM custom.

---

## **Root, roter**

C'est le mode super utilisateur ou administrateur qui permet de modifier le système d'exploitation et permet d'obtenir le contrôle total de son appareil, donc prudence!

---

## **Partition**

En informatique, le partitionnement d'un disque est l'opération qui consiste à le diviser en plusieurs parties appelées partitions.

---

## **Console ou terminal**

La console ou terminal est une interface qui permet d'exécuter des lignes de commandes. La différence entre la console et le terminal réside du fait que le terminal est une console émulée dans une interface graphique qui est très apprécié des débutants. 😊

# **Avertissements et conseils**

**Il vraiment conseiller de faire une sauvegarde de toutes vos données importantes avant de poursuivre ce wiki, comme par exemple un backup. Beaucoup de tutoriels et d'application existent sur internet à ce sujet, vous pouvez aussi suivre le chapitre “Sauvegarder ses données personnelles avec adb” dans ce wiki. 😊**

Certaines manipulations risquent de rendre votre appareil **inutilisable** si elles sont exécutées de manières incorrectes. Si vous avez un doute posez la question sur ce forum ou un autre forum spécialisé. Debian Facile ne pourra pas être tenu responsable. Vous voilà averti.

Le fait de rooter son appareil peut selon les marques **annuler la garantie**.

Ce wiki est un principe, certaines manipulations peut être différentes selon les appareils.

Ce wiki est spécialement dédié aux utilisateurs d'une distribution GNU/Linux ayant comme base Debian (Debian, Ubuntu, Linux Mint...).

Pour les utilisateurs d'autre distributions ou de Windows ou Mac il est conseillé de faire se wiki via une machine virtuelle ou via une live installation ou même avec une installation en dualboot. 😊

Lien: [Installer Debian](#)

## Prérequis

- Avoir un câble USB / mini USB.
- Avoir un appareil fonctionnel sous Android... 😊
- Avoir un ordinateur "si possible" avec une distribution GNU/Linux basée sous Debian.
- Activer le mode débogage USB de notre appareil sous Android, si ce n'est pas le cas voir la suite du wiki.
- Avoir installer les outils ADB, si ce n'est pas le cas voir la suite du wiki.
- Rooter son appareil Android (de nombreux tuto existent sur internet).

---

## Activer le mode débogage USB

Depuis notre appareil Android allons sur **Paramètres**, si nous voyons pas le menu "**Options pour les développeurs**" nous allons devoir l'activer.

Pour ce faire allons dans **Paramètres** -> **A propos du téléphone** ou **A propos de la tablette** Appuyons une dizaine de fois sur la partie **Numéro de build**.

Un message devrait nous indiquer que nous sommes maintenant un développeur !

A présent nous pouvons activer le Débogage USB via **Paramètres** -> **Options pour les développeurs** -> et cochons **Débogage USB**

---

## Installation adb

Voici la commande pour installer les paquets adb et fastboot:

```
su
apt-get install android-tools-adb android-tools-fastboot
exit
```

[https://wiki.cyanogenmod.org/w/Doc:\\_adb\\_intro](https://wiki.cyanogenmod.org/w/Doc:_adb_intro)

Pour vérifier que adb est bien installer et qu'il reconnaît notre appareil sous Android, ouvrons un terminal.

```
adb devices
```

Et le résultat devrai "ressembler" à ça:

```
List of devices attached
HT05BPLXX5058    device
```

Le cas échéant il faudra chercher d'où vient le problème comme par exemple un problème de driver ou pilote.

## Sauvegarder ses données personnelles avec adb

Pour sauvegarder ses données personnelles (backup), on peut utiliser des apk ou applications comme Titanium Backup, Super Backup, App Backup & Restore... Mais nous pouvons aussi le faire avec adb en ligne de commandes.

**AVERTISSEMENT Debian Facile décline toutes responsabilités en cas de pertes de données et de mauvaises manipulations, si vous n'êtes pas sûr de ce que vous faites utilisez une autre méthode !**

```
adb backup -apk -shared -all -f /home/$USER/backup_android.ab
```

Ne pas oublier de confirmer l'action sur notre appareil en confirmant par Sauvegarder mes données.  
et pour restaurer

```
adb restore /home/$USER/backup_android.ab
```

Ne pas oublier de confirmer l'action sur notre appareil par Restaurer mes données.

Simple et efficace ! 😊

## Rooter son appareil

Le root d'un appareil sous Android est nécessaire pour pouvoir effectuer ce wiki cependant le fait de rooter un appareil peut annuler la garantie de votre appareil, il n'est pas prévu de donner des détails à ce sujet dans ce wiki. Merci de faire une recherche sur internet à ce sujet. 😊

Lien 1: <http://www.androidpit.fr/root-android-comment-rooter-smartphone>

## Points de montage

Dans cette partie nous allons visualiser toutes les partitions de notre appareil sous Android via un Terminal.

Relions notre appareil Android à notre ordinateur via un câble USB/mini USB (en principe c'est le même câble qui nous sert à recharger celui-ci 😊).



```
adb shell
ls -al /dev/block/by-name/
exit
```

ou

```
adb shell
ls -al /dev/block/platform/dw_mmc/by-name
exit
```

Cette commande n'est pas universel et nous aurons peut-être besoin de rechercher le bon chemin avec les termes /dev /block /platform et by-name. Si vous avez du mal à trouver le bon chemin internet est votre ami.

Une astuce pour trouver le bon chemin, consiste à utiliser la touche de tabulation pour prévisualiser la suite du chemin correct, pour plus d'info à ce sujet, nous pouvons consulter le super wiki de l'ami smolski 😊

Lien: <https://debian-facile.org/doc:programmation:bash:autocompletion>.

Voici un exemple des points de montages avec leurs chemins (le résultat peut être différent selon les appareils).

```
lrwxrwxrwx root    root    2016-05-31 23:56 UDISK ->
/dev/block/nandl
lrwxrwxrwx root    root    2016-05-31 23:56 alog ->
/dev/block/nandk
lrwxrwxrwx root    root    2016-05-31 23:56 boot ->
/dev/block/nandc
lrwxrwxrwx root    root    2016-05-31 23:56 bootloader ->
/dev/block/nanda
lrwxrwxrwx root    root    2016-05-31 23:56 cache ->
/dev/block/nandg
lrwxrwxrwx root    root    2016-05-31 23:56 env ->
/dev/block/nandb
lrwxrwxrwx root    root    2016-05-31 23:56 klog ->
/dev/block/nandj
lrwxrwxrwx root    root    2016-05-31 23:56 metadata ->
/dev/block/nandh
lrwxrwxrwx root    root    2016-05-31 23:56 misc ->
/dev/block/nande
lrwxrwxrwx root    root    2016-05-31 23:56 private ->
/dev/block/nandi
lrwxrwxrwx root    root    2016-05-31 23:56 recovery ->
/dev/block/nandf
lrwxrwxrwx root    root    2016-05-31 23:56 system ->
/dev/block/nandd
```

Analysons ce retour de commande, par exemple la ligne:

```
lrwxrwxrwx root    root    2016-05-31 23:56 system ->
/dev/block/nandd
```

System est le nom de l'une des partitions que nous allons extraire dans un moment.

/dev/block/nandd est le chemin complet de la partition system.

**Remarquons que chaque chemin est similaire hormis la fin qui est identifié par une lettre différente.**

Il est vivement conseillé de faire un copie/coller du résultat de la commande ci-dessus dans un fichier texte surtout si celle-ci n'est pas identique à celle obtenue dans l'exemple ci-dessus.

## Préparation carte SD

Pour extraire nos partitions et les copier sur notre ordinateur allons devoir les stockés temporairement sur un support.

Pour ce faire il existe plusieurs supports:

- La carte SD interne
- Une carte SD externe

Avant d'aller plus loin, il est judicieux de vérifier l'espace nécessaire qu'il faut avoir ou libéré sur la carte SD de notre appareil sous Android.

Voilà comment faire.

```
adb shell
cd /
du -hc
exit
```

La dernière ligne nous indique l'espace total nécessaire.

Pour connaître l'espace utilisé sur la SD carte interne:

```
adb shell
cd sdcard/
du -hc
exit
```

## Extraction des partitions

Maintenant que tout est prêt nous allons pouvoir extraire les partition de notre appareil et par la même occasions les copiées sur notre ordinateur afin de créer une archive zip mais ça c'est pour maintenant.

Pour cela nous allons utiliser la commande dd qui permet de copier une image disque à l'identique.

```
dd if=/chemin/de/la/partition
of=/chemin_du_dossier_de_destination/nom_de_la_partition.img
```

et de la commande adb pull qui permet de copier un fichier ou un dossier de notre carte SD vers notre ordinateur.

```
adb pull /sdcard/nom_du_dossier /home/$USER/nom_du_dossier
```

Vérifions d'abord que notre appareil est reconnu si le terminal est déjà ouvert, fermons le et

réouvrons le.

```
adb devices
```

Le résultat devrait ressembler à ça (le code HT05BPLXX5058 peut-être différent chez vous, c'est juste un exemple)

```
List of devices attached
HT05BPLXX5058    device
```

Dans le cas contraire il faut s'arrêter et chercher le problème...

Si tout est OK, nous allons pouvoir extraire les partitions sur notre carte SD interne

Voici les partitions qui nous intéressent:

- alog
- boot
- bootloader
- cache
- env
- klog
- metadata
- misc
- private
- recovery
- system

Voici un exemple pour extraire les partitions avec leurs chemins de montage (/dev/block/nand\*) avec l'exemple vu au chapitre "Points de montage"

**Dans le cas échéant il faut adapter les commandes ci-dessous avec vos propres chemins de point de montage. 😊**

Il se peut que suite à la saisie de la commande su qu'il nous soit demandé de confirmer sur l'appareil l'accès en super-utilisateur qui faudra confirmer par "allow".

```
adb shell
su

mkdir /sdcard/dump_android

dd if=/dev/block/nandk of=/sdcard/dump_android/alog.img

dd if=/dev/block/nandc of=/sdcard/dump_android/boot.img

dd if=/dev/block/nanda of=/sdcard/dump_android/bootloader.img

dd if=/dev/block/nandg of=/sdcard/dump_android/cache.img

dd if=/dev/block/nandb of=/sdcard/dump_android/env.img

dd if=/dev/block/nandj of=/sdcard/dump_android/klog.img
```



```
dd if=/dev/block/nandh of=/sdcard/dump_android/metadata.img
dd if=/dev/block/nande of=/sdcard/dump_android/misc.img
dd if=/dev/block/nandi of=/sdcard/dump_android/private.img
dd if=/dev/block/nandf of=/sdcard/dump_android/recovery.img
dd if=/dev/block/nandd of=/sdcard/dump_android/system.img

exit
exit
```

... ne pas oublier les 2 "exit".

Après environ 15 minutes d'attente (en particulier par la partition system) les partitions doivent toutes être extraites.

En suite nous pouvons copier le tout sur notre ordinateur.

```
adb pull /sdcard/dump_android /home/$USER/dump_android
```

Maintenant si nous allons dans notre /home nous devrions avoir un dossier du nom de "dump\_android" avec tous les fichiers .img que nous venons d'extraire.

Si nous voulons supprimer le dossier "dump\_android" histoire de libérer de l'espace voici la commande:

```
adb shell
rm -rf /sdcard/dump_android/
exit
```

## Création de la structure de l'archive

La suite de ce wiki ce passe sur l'ordinateur. Nous allons créer l'archive au format zip qui contiendra notre dump. Dans le dossier "dump\_android" qui contient (normalement) toutes les partitions que nous venons d'extraire de notre appareil Android, nous allons devoir rajouter un dossier "spécial" qui doit s'appeler obligatoirement "META-INF". Pour faire bref ce dossier contiendra toutes des informations importantes comme par exemple le script d'installation (updater-script) de notre futur archive.

Via la commande "mkdir" et "touch" nous allons créer cette structure de divers dossiers:

```
mkdir /home/$USER/dump_android/META-INF
mkdir /home/$USER/dump_android/META-INF/com
mkdir /home/$USER/dump_android/META-INF/com/google
mkdir /home/$USER/dump_android/META-INF/com/google/android
touch /home/$USER/dump_android/META-INF/com/google/android/updater-script
```

Et téléchargeons le fichier “[update-binary](#)” qui viendra placé dans le dossier  
“/home/\$USER/dump\_android/META-INF/com/google/android/”

```
wget https://debian-facile.org/images/file-R25ac9925cdd9fe07763413ca6210a1c2-0 /home/$USER/dump_android/META-INF/com/google/android/update-binary
```



## Script d'installation ou updater-script

Maintenant nous allons créer le script d'installation ou updater-script qui doit être actuellement un fichier vide.

Pour faire bref, c'est via ce script que lors de la réinstallation notre appareil “comprendra” ce qu'on attends de lui. 😊

Ouvrons le fichier “updater-script” qui ce trouve dans le dossier  
/home/\$USER/Bureau/dump\_android/META-INF/com/google/android et copions ce texte:

ATTENTION ! de bien remplacer **/dev/block/nand\*** par les bons chemins de points de montage qui correspondent à **votre** appareil sous Android (voir chapitre “Point de montage”)

```
package_extract_file("bootloader.img", "/dev/block/nanda");
package_extract_file("env.img", "/dev/block/nandb");
package_extract_file("boot.img", "/dev/block/nandc");
package_extract_file("system.img", "/dev/block/nandd");
package_extract_file("misc.img", "/dev/block/nande");
package_extract_file("recovery.img", "/dev/block/nandf");
package_extract_file("cache.img", "/dev/block/nandg");
package_extract_file("metadata.img", "/dev/block/nandh");
package_extract_file("private.img", "/dev/block/nandi");
package_extract_file("metadata.img", "/dev/block/nandj");
package_extract_file("klog.img", "/dev/block/nandk");
package_extract_file("alog.img", "/dev/block/nandl");

delete_recursive("/data/dalvik-cache/");
delete_recursive("/cache/dalvik-cache/");
delete_recursive("/cache");

ui_print("Done!");
```

Et enregistrons, rien de compliquer normalement. 😊

## Création de l'archive zip

Maintenant nous allons créer l'archive au format zip.

Si le paquet “zip” n'est pas installé voilà comme faire:

```
su
apt-get install zip
exit
```

Et créons notre archive.zip via cette commande:

```
cd /home/$USER/dump_android
zip -r9 /home/$USER/dump_android.zip *
```

## Signature de l'archive zip

Signer une archive zip c'est lui apposer une signature numérique qui permet de valider son contenu et sa provenance, si on ne le faisait pas l'archive risquerait d'être refusé par le Recovery lors de l'installation ou flash.

Pour pouvoir signer l'archive, nous aurons besoin d'un mini "soft" qui s'appelle "SignApk".

Lien 1: <https://debian-facile.org/images/file-Rb3866f5849cc53ea3c0468a27a5dd51f>

Une fois "SignApk" télécharger il faut l'extraire dans notre /home.

Pour info, nous trouverons les fichiers "testkey.pk8" et "testkey.x509.pem". Ces deux fichiers sont le certificat et la clé privée fournie par Android. Nous trouverons aussi "signapk.jar" qui est script qui nous permet d'exécuter le travail de signature.

Maintenant, copions dump\_android.zip dans le dossier signapk afin de le signer.

```
cp dump_android.zip /home/$USER/SignApk/dump_android.zip
java -Xmx10240m -jar signapk.jar -w testkey.x509.pem testkey.pk8
dump_android.zip dump_android_signed.zip
```

Et voilà notre archive est prête à être réinstaller! 😊

**Voici les commandes qui permettent de faire tous ce qu'il a été vu dans ce chapitre (pour autant que le lien de "wget" soit toujours valable!)**

```
cd /home/$USER
wget https://debian-facile.org/images/file-Rb3866f5849cc53ea3c0468a27a5dd51f
-O SignApk.zip
unzip SignApk.zip
cp dump_android.zip /home/$USER/SignApk/dump_android.zip
cd /home/$USER/SignApk
java -Xmx10240m -jar signapk.jar -w testkey.x509.pem testkey.pk8
dump_android.zip dump_android_signed.zip
```

## Réinstallation

Nous sommes arrivés à la dernière partie de ce wiki. Nous allons réinstaller notre archive ! 😊

Éteignons notre appareil.

Nous allons maintenant le rallumer en mode Recovery, pour ce faire il faut généralement maintenir la touche **Power** et **Volume bas** pendant 5 à 10 secondes (si ce n'est pas le cas chez vous et que vous ne savez pas comment faire une recherche sur internet devra vous donner la réponse).

Nous voici sur le mode Recovery, pour information il existe une multitude de Recovery mais en principe ils tous très ressemblant. Voici un exemple de Recovery:



Pour réinstaller notre archive on a plusieurs possibilités, nous allons en voir 2 dans ce wiki: "apply update from ADB" et "apply update from extsd storage".

## Réinstallation via apply update from ADB

Si ce n'est pas déjà fait, commençons par connecter notre appareil sous Android à notre ordinateur via un câble USB / mini USB.

Via les **Touches de volume** naviguons jusqu'à l'option "apply update from ADB" et appuyons sur la touche **Power** pour confirmer notre choix. Un message nous confirme que nous pouvons maintenant faire appel à ADB.



Sur notre ordinateur ouvrons un Terminal

```
adb kill-server
su
adb start-server
adb devices
```

Nous pouvons continuer, si nous avons un résultat "ressemblant" à ça:

```
List of devices attached
03384658c80500000000    sideload
```

C'est OK? Alors continuons...

```
adb sideload /home/$USER/SignApk/dump_android_signed.zip
```

Dans la commande ci-dessus, nous avons saisi le chemin de l'archive (signée) où on l'avait laissé, le cas échéant il faudra modifier le chemin par le bon. 😊

Normalement et après quelques instants d'attente (environ 10 à 15 minutes) l'archive devrait être installée. Il suffit d'éteindre et de rallumer notre appareil Android qui devrait être à nouveau fonctionnelle !!!

## Réinstallation via une clé USB ou une carte micro SD

Dans un premier temps il faudra insérer notre support (clé USB ou carte mini SD) dans notre ordinateur.

Pour utiliser la méthode avec une clé USB il faudra disposer dans câble USB/mini USB permettant le branchement de la clé USB sur notre appareil Android.



Pour utiliser la méthode avec une carte micro SD il faudra certainement disposer d'un adaptateur micro SD/USB ou micro SD/SD permettant le branchement sur l'ordinateur.



Il suffit de copier notre archive signée (dump\_android\_signed.zip) sur la clé USB ou la carte micro SD puis de la retirer et de l'insérer sur notre appareil Android.

Via les Touches de volume naviguons jusqu'à l'option:

- "apply update from usbhost storage" si on utilise une clé USB.
- "apply update from extsd" si on utilise une carte micro SD .

Appuyons sur la touche Power pour confirmer notre choix.

Un message indiquant la progression devrait s'afficher.

Normalement et après quelques instants d'attente (environ 10 à 15 minutes) l'archive devrait être installée.

Il suffit d'éteindre et de rallumer notre appareil Android qui devrait être à nouveau fonctionnelle !!!

## Réinstallation ses données personnelles

Maintenant que tout est à nouveau fonctionnelle (normalement) il nous reste plus qu'à réinstaller notre backup selon le chapitre "[Sauvegarder ses données personnelles avec adb](#)"

From:

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

Permanent link:

<http://debian-facile.org/utilisateurs:slyfox:tutos:dump-android>

Last update: **12/06/2016 21:15**

