

Android - dump et backup avec Debian

- Objet : créer un dump de son smartphone ou tablette sous Android
- Niveau requis :
[avisé](#)
- Débutant, à savoir : [Utiliser GNU/Linux en ligne de commande, tout commence là !](#) 😊
- Suivi :
[à-tester](#)
- Création par [slyfox](#) 05.06.2016
- Commentaires sur le forum : [Lien vers le forum concernant ce tuto^{1\)}](#)]]

Modèles

Ajoutez votre appareil testé dans le tableau ci-dessous :

Marque	Modèle	Type	Remarque
Trekstor	M-Budget	Tablette	RAS

Introduction

Le dump²⁾ est la copie brute d'un système.

Il est utile en cas de plantage d'un smartphone ou d'une tablette³⁾ suite par exemple à :

- une fausse manipulation ou
- si l'appareil reste bloqué sur le logo de démarrage.

Réinstaller ou flasher le firmware est alors la seule façon pour refaire fonctionner l'appareil.

Pour de nombreux appareils sous Android il n'existe pas forcément de firmware disponible, en particulier pour les tablettes bas prix (low-cost). Le simple dump ou un backup complet de l'appareil puis la réinstallation via le recovery sont alors une solution possible.

Vous pouvez suivre ce wiki si vous avez un recovery custom comme par exemple CWM⁴⁾ ou TWRP⁵⁾ qui permettent également de faire un backup de l'appareil plus facilement que celui expliqué ici.



Avec un *recovery custom* ; la réinstallation sera légèrement différente de ce wiki.
Voir alors sur internet pour la réinstallation avec le recovery custom de votre appareil.



Il peut être judicieux de faire *préalablement* un dump de l'appareil (avec toutes les précautions qui s'imposent) dès la sortie de sa boîte pour retrouver l'appareil dans l'état où il était avant un plantage ou toutes autres raisons à inventer soi-même...

Dans ce wiki nous allons voir comment :

1. Installer les outils nécessaires.

2. Préparer l'appareil.
3. Visualiser les partitions.
4. Extraire les partitions.
5. Copier les partitions sur un ordinateur.
6. Créer l'archive zip.
7. Signer l'archive zip.
8. Réinstaller le système via le recovery d'origine ou celui du fabricant.

Vocabulaire

Voici quelques définitions des termes techniques utilisés dans ce wiki :

ADB

L'Android Debug Bridge (ADB) est un outil avec un programme client et un programme serveur pour permettre la communication entre l'appareil Android et l'ordinateur.

Backup

En informatique, c'est la sauvegarde des 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 la sauvegarde de données.

Linux

Linux désigne le noyau employé par un [Système d'exploitation](#).

Dans le langage courant on trouve souvent l'emploi du terme « *Linux* » seul pour désigner plus particulièrement une distribution du système d'exploitation [GNU/Linux](#) comme Debian, ce qui porte à confusion car il existe d'autres systèmes non GNU fondés aussi sur l'utilisation de Linux, comme Android donc.

 *Tux, la mascotte du noyau Linux.*

Recovery

Le recovery ou récupération est un menu ou un mode de démarrage du mobile qui permet d'effectuer différentes opérations comme :

- 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 maintenance.

Les plus connus sont ClockWorkMod (CWM) et 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, rooter

C'est le mode [superutilisateur](#) (nommé aussi *administrateur*) qui permet de contrôler l'ensemble de son appareil et en modifier le Système d'Exploitation, donc prudence !

Partition

En informatique, [le partitionnement d'un disque](#) consiste à le diviser en plusieurs parties⁶.

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 dans le fait que le terminal est le plus souvent une console émulée dans une interface graphique, plus apprécié ainsi par les débutants. 😊

Avertissements et conseils



Faites une sauvegarde (faire un backup) de toutes vos données importantes avant d'appliquer ce tuto.

Pour le backup, de nombreux tutoriels et applications existent sur internet, vous pouvez aussi suivre le chapitre [Sauvegarder ses données personnelles avec adb](#). 😊



Certaines manipulations de ce tuto exécutées de manières incorrectes peuvent de rendre votre appareil **inutilisable**.

En cas de doute et obtenir des précisions, n'hésitez pas à en faire part sur le forum df [via ce lien concernant le tuto](#) ou à consulter un forum spécialisé.



En aucun cas Debian Facile ne pourra pas être tenu responsable de vos manipulations sur votre appareil.



Le fait de rooter son appareil peut en **annuler la garantie**.



Ce wiki est un principe, certaines manipulations peuvent différées selon les appareils.



Ce wiki est dédié aux distributions GNU/Linux utilisant Debian⁷⁾.

Pour les utilisateurs d'autres distributions, il est conseillé de suivre ce wiki via une [machine virtuelle](#) ou une installation live ou même avec une installation en [dualboot](#). 😊

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.
- Rooter son appareil Android (de nombreux tuto existent sur internet).
- Activer le mode débogage USB de notre appareil sous Android, si ce n'est pas le cas voir la suite du wiki.
- Avoir installé les outils ADB, si ce n'est pas le cas voir la suite du wiki.

Rooter son appareil

Pour effectuer ce wiki il est nécessaire de rooter son appareil, cependant :



Le fait de **rooter** un appareil sous Android peut annuler la garantie de celui-ci !

C'est pourquoi il n'est pas prévu de donner plus de détails à ce sujet ici et pour avoir plus de détails à ce sujet, nous vous invitons à faire une recherche sur internet. 😊

Exemple de lien possible : <http://www.androidpit.fr/root-android-comment-rooter-smartphone>

Activer le mode débogage USB

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

Pour ce faire allons dans :

1. **Paramètres** -> **À propos du téléphone** (ou **À propos de la tablette**)
2. Appuyons une *dizaine* de fois sur la partie **Numéro de build** et un message nous indique alors que nous sommes maintenant un développeur !
3. Activons le Débogage USB via **Paramètres** -> **Options pour les développeurs**.
4. Cochons⁸⁾ maintenant : **Débogage USB** 😊

Installation adb

Installation des paquets adb et fastboot :

```
apt-get update && apt-get install android-tools-adb android-tools-fastboot
```

Voir https://wiki.cyanogenmod.org/w/Doc:_adb_intro

Pour vérifier que adb est bien installé et qu'il reconnaît notre appareil sous Android, ouvrons un terminal et validons :

```
adb devices
```

Le résultat devrait ressembler à ça :

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



Le cas échéant il faudra chercher d'où vient le problème comme par exemple celui posé par un 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 sur votre appareil, si vous n'êtes pas sûr de ce que vous faites posez la question sur le forum df ou utilisez une autre méthode !

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

Ne pas oublier de confirmer l'action sur votre appareil par : Sauvegarder mes données.

et pour restaurer

```
adb restore $HOME/backup_android.ab
```

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

Simple et efficace ! 😎

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/
```

ou

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



Cette commande n'est pas universelle 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 stocker 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 du -hc /
```

La dernière ligne nous indique l'espace total nécessaire.

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

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

Extraction des partitions

Maintenant que tout est prêt nous allons pouvoir extraire les partitions de notre appareil et par la même occasion les copier 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_répertoire_de_destination/nom_de_la_partition.img
```

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

```
adb pull /sdcard/nom_du_répertoire $HOME/nom_du_répertoire
```

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 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.

Ensuite nous pouvons copier le tout sur notre ordinateur.

```
adb pull /sdcard/dump_android $HOME/dump_android
```

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

Si nous voulons supprimer le répertoire "dump_android" histoire de libérer de l'espace voici la commande :

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

Création de la structure de l'archive

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

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

```
mkdir -p $HOME/dump_android/META-INF/com/google/android  
touch $HOME/dump_android/META-INF/com/google/android/updater-script
```

Et téléchargeons le fichier "update-binary" qui sera placé dans le répertoire "\$HOME/dump_android/META-INF/com/google/android/"

```
wget https://debian-facile.org/images/file-R25ac9925cdd9fe07763413ca6210a1c2  
-O $HOME/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 se trouve dans le répertoire \$HOME/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 compliqué 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 :

```
apt-get install zip
```

Et créons notre archive.zip via cette commande :

```
cd $HOME/dump_android
zip -r9 $HOME/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 refusée 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” téléchargé 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 un script qui nous permet d’exécuter le travail de signature.

Maintenant, copions dump_android.zip dans le répertoire signapk afin de le signer.

```
cp dump_android.zip $HOME/SignApk/dump_android.zip
cd $HOME/SignApk
java -Xmx1024m -jar signapk.jar -w testkey.x509.pem testkey.pk8
dump_android.zip dump_android_signed.zip
```

Et voilà notre archive est prête à être réinstallée ! 😊

Voici les commandes qui permettent de faire tous ce qui a été vu dans ce chapitre (pour autant que le lien de “wget” soit toujours valable!)

```
cd $HOME
wget https://debian-facile.org/images/file-Rb3866f5849cc53ea3c0468a27a5dd51f
-O SignApk.zip
unzip SignApk.zip
cp dump_android.zip $HOME/SignApk/dump_android.zip
cd $HOME/SignApk
java -Xmx1024m -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 appuyées les touches 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 devrait vous donner la réponse).

Nous voici sur le mode Recovery, pour information il existe une multitude de Recovery mais en principe ils sont tous très ressemblants. 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/SignApk/dump_android_signed.zip
```

Dans la commande ci-dessus, nous avons saisi le chemin de l'archive (signée) où on l'avait laissée, 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 fonctionnel !!!

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 d'un 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 fonctionnel !!!

Réinstallation ses données personnelles

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

1)

N'hésitez pas à y faire part de vos remarques, succès, améliorations ou échecs !

2)

ou **nandroid**

3)

sous Android

4)

ClockWorkMod

5)

TeamWin Recovery Project

6)

nommées : partitions

7)

Debian, Ubuntu, Linux Mint...

8)

qui s'en dédit...

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

Permanent link:
<http://debian-facile.org/atelier:chantier:dump-et-backup-d-un-appareil-android-avec-debian-et-ses-derives2>

Last update: 03/11/2016 20:16

