

URL: <https://www.plop.at/en/lineageos.html>

LineageOS - Backup, Restore, Transfer, Migrate

I prefer to use my own scripts to create backups. If you also want to manage your backups by yourself and don't want to use the backup feature of [LineageOS](#) or an app, then you can easily use ADB to create backups of your device and restore them later. It is also no big deal to migrate/transfer your data to another device.

[LineageOS](#) is a free and open-source operating system for various devices, based on the Android mobile platform.

Useful scripts: [adb-scripts-20231101.zip](#)

[Requirements](#)

[Setup](#)

[Scripts / Downloads](#)

[Backup](#)

[Restore / Transfer / Migrate](#)

[Examples: Backup and restore/transfer apps](#)

[Use adb exec-out & tar](#)

[Access via MTP](#)

[Troubleshooting](#)

Requirements

On your device:

- When LineageOS is running: Developer options enabled, USB debugging enabled, rooted debugging enabled.
- When Lineage Recovery is running: ADB enabled.

On your computer:

- ADB installed.

Setup

LineageOS: How to enable developer options

Open the settings app. Go to 'About phone'. Tap at 'Build number' ten times.

LineageOS: How to enable ADB access via USB on the device

Open the settings app. Go to 'System' / 'Developer options'. Enable 'USB debugging', enable 'Rooted debugging'.

Lineage Recovery: How to enable ADB access via USB on the device

In the menu 'Advanced' touch 'Enable ADB'.

On your computer: ADB root

To have full access for the backup, you have to set ADB to root mode. Connect your device with USB to your computer. On your computer run 'adb root', on your device touch 'Allow' at the 'Allow USB debugging' popup.

Scripts / Downloads

Here are some backup scripts that gives you a starting point. Download: [adb-scripts-20231101.zip](#)

Backup

You can make a backup of the whole device with disk images or backup individual files and directories. This can be done when LineageOS or Lineage Recovery is running.

Backup when LineageOS or Lineage Recovery is booted? Disk image or file based?

When you make the backup via ADB when LineageOS is booted, then your device is up and running as usual. You stay online, you are reachable and you can use your device as expected. But there is the risk, that some data became invalid when LineageOS is writing to the file system during the backup. If you want to create clean backups, then you have to reboot the device to Lineage Recovery, but this is more uncomfortable.

If you use as backup strategy the file and directory download instead of the disk image, then the risk of invalid data is reduced. The file and directory download is faster (more specific) than creating a disk image. But it also increases the risk to forget to backup important data. When you create the disk image, then you backup everything as it is at the moment of the backup. Restoring gives you back what you had. But creating the disk image is slower, because it saves the whole storage, regardless if there is data or not.

Start ADB

Connect your device with USB to the computer. Run on your computer 'adb root' and on your device touch 'Allow' at the 'Allow USB debugging' popup.

Create disk image

Usually, the device node of the internal drive is '/dev/block/mmcblk0' or '/dev/block/sda'. To figure out what device node is used by LineageOS, connect to your device with 'adb shell'. You find all device nodes in '/dev/block/'. When you run the command 'mount' you get a list of all devices mounted by LineageOS.

Here are some examples, running 'mount' on LineageOS.

Example output:

```
=== CUT ===
tmpfs on /mnt type tmpfs (rw,seclabel,nosuid,nodev,noexec,relatime,size=86...
/dev/block/mmcblk0p23 on / type ext4 (ro,seclabel,nodev,relatime,discard)
tmpfs on /apex type tmpfs (rw,seclabel,nosuid,nodev,noexec,relatime,size=8...
tmpfs on /linkerconfig type tmpfs (rw,seclabel,nosuid,nodev,noexec,relatim...
/dev/block/mmcblk0p26 on /data type ext4 (rw,seclabel,nosuid,nodev,noatime...
/dev/block/mmcblk0p24 on /cache type ext4 (rw,seclabel,nosuid,nodev,noatim...
/dev/block/mmcblk0p12 on /efs type ext4 (rw,seclabel,nosuid,nodev,noatime,...
/dev/block/mmcblk0p26 on /data/user/0 type ext4 (rw,seclabel,nosuid,nodev,...
=== CUT ===
```

Here you see '/dev/block/mmcblk0p23' is mounted to '/'. '/dev/block/mmcblk0p23' stands for device mmcblk0 partition 23. The device node for the internal drive is '/dev/block/mmcblk0'. Use 'exit' to quit.

To make a backup of the internal drive run 'adb pull /dev/block/mmcblk0 mmcblk0.img'.

Another device example:

```
=== CUT ===
tmpfs on /mnt/androidwritable type tmpfs (rw,seclabel,nosuid,nodev,noexec,...
/dev/block/sda29 on /metadata type ext4 (rw,seclabel,nosuid,nodev,noatime,...
/dev/block/sda25 on / type ext4 (ro,seclabel,nodev,relatime)
/dev/block/sda26 on /vendor type ext4 (ro,seclabel,relatime)
/sys/kernel/debug on /sys/kernel/debug type debugfs (rw,seclabel,relatime)
none on /config type configfs (rw,nosuid,nodev,noexec,relatime)
none on /sys/fs/fuse/connections type fusectl (rw,relatime)
=== CUT ===
```

Here you see '/dev/block/sda25' is mounted to '/'. '/dev/block/sda25' stands for device sda partition 25. The device node for the internal drive is '/dev/block/sda'. Use 'exit' to quit.

To make a backup of the internal drive run 'adb pull /dev/block/sda sda.img'.

Backup files and directories

LineageOS stores the data of the apps under '/data/data/'. Media files like photos and videos are stored on the internal drive under '/data/media/0/'.

Example: Backup data from all apps without media files

```
adb pull -a /data/data
```

Note: When you want to do the backup with Lineage Recovery, then you have to mount the data partition manually before you can use 'adb pull'.

The second example is to make a backup from the SD Card. The SD Card is mounted to a directory under '/storage/'. To figure out which directory is the SD Card on your device, connect to LineageOS with 'adb shell'. Then run 'ls /storage/'. Here is an example output

```
emulated self 1865-F8CF
```

On this device, the SD Card is mounted to /storage/1865-F8CF. The directory name will be different on your device. When you list the content of the directory, then you should see your files and directories. On the example device, the command would look like 'ls /storage/1865-F8CF'.

Example: Backup the whole SD Card

```
adb pull -a /storage/1865-F8CF
```

Example: Backup a directory called DCIM from the SD Card

```
adb pull -a /storage/1865-F8CF/DCIM
```

Restore / Transfer / Migrate

You can restore the whole device with disk images or individual files and directories. You can also transfer app data when you migrate to another device. Below you find some examples how to restore the data of the apps Phone, Contacts, SMS/MMS, Notepad, WhatsApp.

Connect your device with USB to the computer. Run on your computer 'adb root' and on your device touch 'Allow' at the 'Allow USB debugging' popup.

Restore a disk image

Restoring from a disk image has to be done from Lineage Recovery. You have to enable ADB in the 'Advanced' menu. Select 'Enable ADB'.

Example: Restore LineageOS on the internal drive. You have to use the same device node that you used for the backup. The device node for this example device is '/dev/block/sda'. Your device may use a different device node. Maybe '/dev/block/mmcblk0'.

Warning: All data on the destination drive will be overwritten by the backup!

Restore:

```
adb push sda.img /dev/block/sda
```

Restore a file or a directory and its files and sub directories

'adb push' copies files and directories to the device. Use '--sync' to overwrite only older files.

Example: Backup the directory Pictures from the SD Card:

```
adb pull -a /storage/1865-F8FC/Pictures
```

Example: Restore/transfer the directory Pictures to the SD Card:

```
adb push Pictures /storage/1865-F8FC
```

Examples: Backup and restore/transfer apps

Every app has its own data directory. LineageOS is storing the data of the apps in '/data/data/[package name]'. You find the package name in the settings app, choose 'Apps' and 'See all apps'. Select the app you are interested in. At the bottom you see the package name. For example the browser has the package name org.lineageos.jelly. The data is stored in '/data/data/org.lineageos.jelly/'.

Note: Close the app before starting with restoring/transferring the data.

[Telephone & Calling list](#)

[Contacts](#)

[SMS & MMS messages](#)

[Notepad](#)

[WhatsApp](#)

Telephone & Calling list

Backup:

```
adb pull -a /data/data/com.android.dialer
```

Restore:

```
adb push com.android.dialer /data/data
```

Contacts & Call logs

Backup:

```
adb pull -a /data/data/com.android.providers.contacts
```

Restore:

```
adb push com.android.providers.contacts /data/data
```

SMS & MMS messages

Backup:

```
adb pull -a /data/data/com.android.messaging  
adb pull -a /data/data/com.android.providers.telephony
```

Restore:

```
adb push com.android.messaging /data/data  
adb push com.android.providers.telephony /data/data
```

Notepad

Backup:

```
adb pull -a /data/data/com.farmerbb.notepad
```

Restore:

```
adb push com.farmerbb.notepad /data/data
```

WhatsApp

Backup messages:

```
adb pull -a /data/data/com.whatsapp
```

Backup media files (photos, videos, ...):

```
adb pull -a /data/media/0/Android/media/com.whatsapp/WhatsApp/Media
```

Install WhatsApp on the new device before restoring/transferring the files. The WhatsApp installation ensures that all required directories are created on the device. Then you can transfer your data to the device with ADB.

Restore messages:

```
adb push com.whatsapp /data/data
```

Restore media files (photos, videos, ...):

```
adb push Media /data/media/0/Android/media/com.whatsapp/WhatsApp/
```

Use adb exec-out & tar

When you make a backup with 'adb pull' then adb creates internal at first a file list and then adb copies all files from that list to your computer. If you do a backup of the apps data directory, then it is possible that apps have temporary files while adb is creating the file list for copying. When such files are removed while adb is copying then adb stops with the error 'adb: warning: stat failed for path: No such file or directory'. adb has no option to skip copying missing files. To solve this issue use 'adb exec-out' and 'tar' instead of 'adb pull'. This creates a tar archive and missing files are ignored.

Command:

```
adb exec-out "tar -c /data/data 2> /dev/null" > data.tar
```

Enhanced with date in file name:

```
adb exec-out "tar -c /data/data 2> /dev/null" > data-$(date +%Y%m%d).tar
```

See also [ADB pull stops after first error](#)

Access via MTP

MTP is not adb related, but it is another way to access the internal data storage and the SD Card of a smartphone/tablet. You can copy files and directories through a mount point.

Connect the smartphone with USB to your computer.

Now some messages will popup on the smartphone. Tap on

Charging this device via USB

Tap for more options

Now the setting screen opens. In the section 'Use USB for' select 'File transfer'.

Create a mount point. For example 'smartphone':

```
mkdir smartphone
```

Mount the smartphone with:

```
jmtvfs smartphone
```

Now you can change to the directory 'smartphone' and copy files. Note, if you have a lot of files in a directory, then opening the directory is a bit slow and takes some time.

Unmount the smartphone with:

```
sudo umount smartphone
```

jmtarfs has a lot of options. See 'jmtarfs -h'.

Troubleshooting

Error message: Permission denied

You have to switch ADB to root mode. Run on your computer 'adb root' and on your device touch 'Allow' at the 'Allow USB debugging' popup.

Error message: adb: unable to connect for root: device unauthorized.

```
* daemon not running; starting now at tcp:5037
* daemon started successfully
adb: unable to connect for root: device unauthorized.
This adb server's $ADB_VENDOR_KEYS is not set
Try 'adb kill-server' if that seems wrong.
Otherwise check for a confirmation dialog on your device.
ADB on device enabled?
```

On your computer run again 'adb root', on your device touch 'Allow' at the 'Allow USB debugging' popup.

Error message: adb: warning: stat failed for path: No such file or directory

See section 'Use adb exec-out & tar'.
