

# How to Fix Corrupted USB in Linux

Usually, users use USB drives to store data safely. However, in some cases, USB drives may be corrupted or simply not working. If you are using Linux operating system and encounter the same error on your USB drive, then you can use the commands to restore the USB drive.

Everyone has a flash drive. They're wonderful little things that make moving data around a breeze. But sometimes flash drives can get corrupted or stop working altogether. Luckily, if you're using Linux, you have access to a variety of tools that can help you troubleshoot the problem. This guide uses Ubuntu, but everything should apply to most modern Linux distributions. Here's how to fix a corrupted USB drive in Linux.

This article prioritizes data safety. You'll first learn how to back up the contents of your flash drive, then perform various fixes to the problem at hand. That way, if your troubleshooting efforts make things worse, you can still restore a snapshot of the current state of your flash drive.

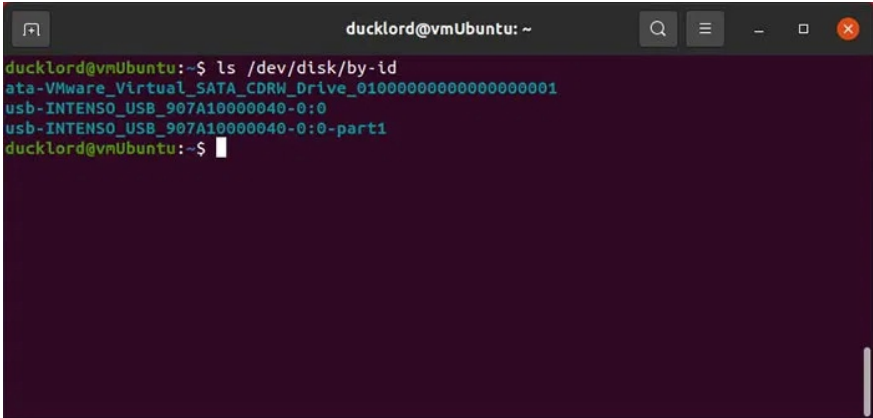
## Create a compressed full backup image

In Linux, there are many tools to backup any storage device. However, the tried and tested method is based on `dd` and `gzip`.

Before you jump into backup mode, you should check to see if the drive is actually broken. A simple first step is to plug the USB into a different USB port or even a different computer. Sometimes the problem isn't the drive itself — it could be something as simple as a loose connection or an unstable port.

Now, to back up your flash drive, first connect your USB to your computer. Launch your favorite terminal (or press **CTRL + Alt + T**). Then, locate your flash drive using the following command:

```
ls /dev/disk/by-id
```

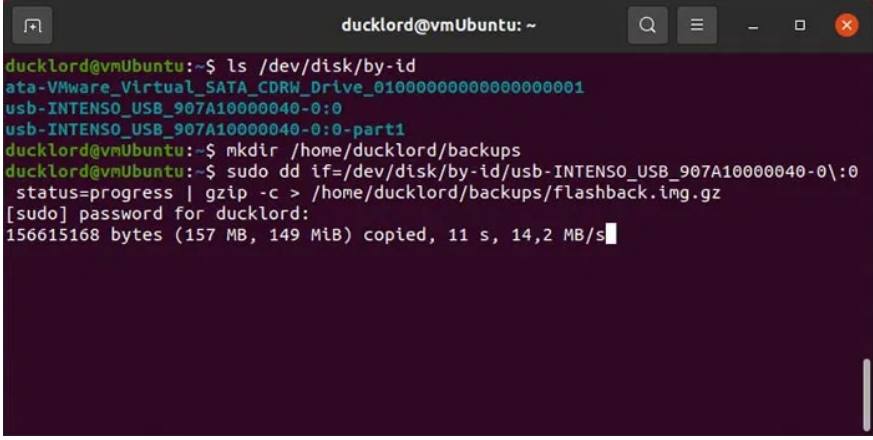
A terminal window titled 'ducklord@vmUbuntu: ~' with search, menu, and window control icons. The terminal shows the command 'ls /dev/disk/by-id' being executed, resulting in the following output:

```
ducklord@vmUbuntu:~$ ls /dev/disk/by-id
ata-VMware_Virtual_SATA_CDRW_Drive_01000000000000000001
usb-INTENSO_USB_907A10000040-0:0
usb-INTENSO_USB_907A10000040-0:0-part1
ducklord@vmUbuntu:~$
```

Alternatively, you can identify your flash drive by running **lsblk** or **sudo fdisk -l** to find the device name (e.g. **/dev/sdb** , not the partition like **/dev/sdb1** ).

Next, back up the flash drive to a GZIP compressed image file in a single command, using:

```
sudo dd if=/dev/disk/by-id/YOUR_FLASH_DRIVE status=progress | gzip -c > /home/USERNAME/backups/flashback.img.gz
```

A terminal window titled 'ducklord@vmUbuntu: ~' showing the execution of commands to identify a USB drive and create a backup. The commands and their outputs are: 'ls /dev/disk/by-id' listing 'ata-VMware\_Virtual\_SATA\_CDRW\_Drive\_01000000000000000001', 'usb-INTENSO\_USB\_907A10000040-0:0', and 'usb-INTENSO\_USB\_907A10000040-0:0-part1'; 'mkdir /home/ducklord/backups'; and 'sudo dd if=/dev/disk/by-id/usb-INTENSO\_USB\_907A10000040-0:0 status=progress | gzip -c > /home/ducklord/backups/flashback.img.gz' with a password prompt and a progress report: '156615168 bytes (157 MB, 149 MiB) copied, 11 s, 14,2 MB/s'.

```
ducklord@vmUbuntu:~$ ls /dev/disk/by-id
ata-VMware_Virtual_SATA_CDRW_Drive_01000000000000000001
usb-INTENSO_USB_907A10000040-0:0
usb-INTENSO_USB_907A10000040-0:0-part1
ducklord@vmUbuntu:~$ mkdir /home/ducklord/backups
ducklord@vmUbuntu:~$ sudo dd if=/dev/disk/by-id/usb-INTENSO_USB_907A10000040-0:0
status=progress | gzip -c > /home/ducklord/backups/flashback.img.gz
[sudo] password for ducklord:
156615168 bytes (157 MB, 149 MiB) copied, 11 s, 14,2 MB/s
```

Also make sure that **'/home/USERNAME/backups/'** exists (create it with **mkdir -p /home/USERNAME/backups** ). Also, the **if=** parameter specifies the flash drive and **gzip** compresses the output.

To restore the backup, you will need to reverse the order of the two commands and specify your flash drive as the output device. The full command will look like this:

```
sudo gzip -cd /home/USERNAME/backups/BACKUP_NAME.img.gz | sudo dd of=/dev/disk/by-id/YOUR_FLASH_DRIVE
```

**Note** : This will overwrite the entire flash drive, erasing all data. Double check the device name to avoid data loss.

## Repair corrupted file systems with FSCK

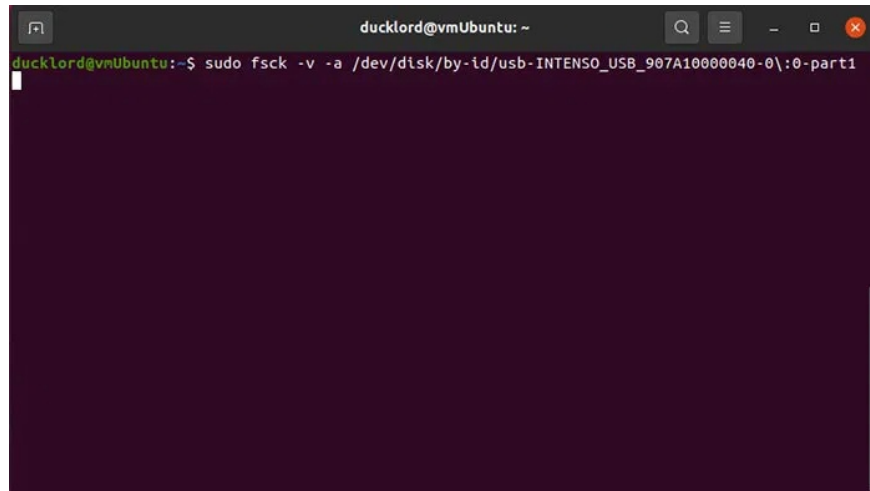
Once you've backed up the contents of your flash drive, it's time to try to repair it. To do that, you can turn to **fsck**. This is great for removing corrupted file blocks, as most (if not all) corruption and unreadability issues stem from issues like this.

For this command, you will need to specify a partition instead of a full drive. You will find the partition with the same name as your device by executing:

```
ls /dev/disk/by-id/usb*
```

Then run **fsck** on that drive with the command:

```
sudo fsck -v -y /dev/disk/by-id/YOUR_FLASH_DRIVE-PARTITION-TO-CHECK
```



```
ducklord@vmUbuntu: ~  
ducklord@vmUbuntu:~$ sudo fsck -v -a /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0\:0-part1
```

In this command:

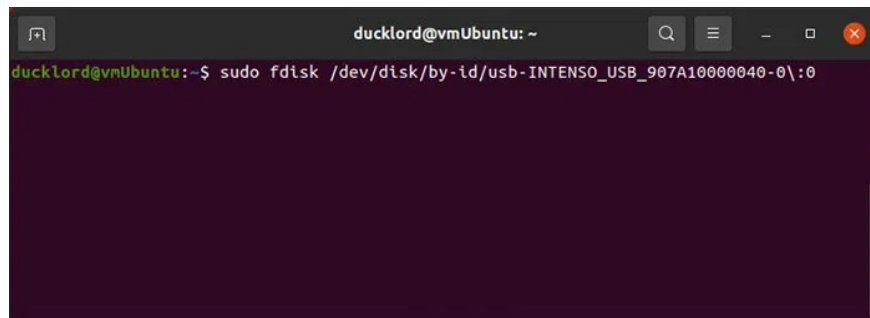
1. **sudo fsck** runs the repair tool with admin rights
2. **-v** tells the tool to display detailed information about the process.
3. **-y** tells the tool to automatically try to fix any errors it finds.
4. **/dev/disk...** is the partition that will be checked for errors.

## Format USB drive using Fdisk/MKFS from Terminal

If fsck fails to repair the device's file system, you can try formatting it to use it as if it were new.

The first step is to delete any existing filesystem structures and recreate them from scratch. You can use fdisk for this purpose. Run it with sudo on your device with:

```
sudo fdisk /dev/disk/by-id/YOUR_FLASH_DRIVE
```



```
ducklord@vmUbuntu: ~  
ducklord@vmUbuntu:~$ sudo fdisk /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0\:0
```

Press **o** and then **Enter** to create a new DOS partition table on it so your USB drive can read it everywhere. If you just want to use it on your modern computer and operating system, you can replace **o** with **g** to create a new GPT partition table.

Press **n** then **Enter** to create a new partition, then press **p** to make it a primary partition. If you use **e** instead of **p**, it will be created as an extended partition, but there is no point in doing so unless you plan on having more than 3 partitions on the drive. Then just press **Enter** when asked for the partition number, first sector, and last sector, to accept the defaults and let the partition span the entire USB drive.

```
ducklord@vmUbuntu: ~
ducklord@vmUbuntu:~$ sudo fdisk /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0\.:0

Welcome to fdisk (util-linux 2.34).
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Command (m for help): n
Partition type
  p   primary (0 primary, 0 extended, 4 free)
  e   extended (container for logical partitions)
Select (default p): p
Partition number (1-4, default 1): 1
First sector (2048-8060927, default 2048):
Last sector, +/-sectors or +/-size{K,M,G,T,P} (2048-8060927, default 8060927):

Created a new partition 1 of type 'Linux' and of size 3,9 GiB.

Command (m for help):
```

Press **p** and then **Enter** to test the new storage structure on the USB drive. Then, press **w** and then **Enter** to write the changes to the USB drive and exit fdisk.

```
ducklord@vmUbuntu: ~
Command (m for help): p
Disk /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0:0: 3,86 GiB, 4127195136 bytes, 8060928 sectors
Disk model: USB
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0x6892a409

Device                               Boot Start    End Sectors  Size
e Id Type
/dev/disk/by-id/usb-INTENSO_USB_907A10000040-0:0-part1 * 2048 8060927 8058880 3,9
G 83 Linu

Command (m for help): w
```

## Format the new partition with a file system

Your partition will not be formatted and, since it does not have a file system, it will not be usable. You can create a file system using one of the mkfs tools that come with every modern Linux distribution. To format the partition to FAT32, which is usable on most devices, use:

```
sudo mkfs.fat -F 32 /dev/disk/by-id/YOUR_FLASH_DRIVE-PARTITION
```

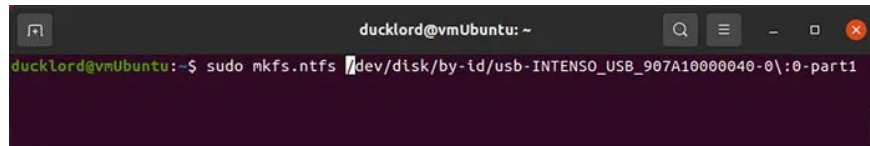
```
ducklord@vmUbuntu: ~
ducklord@vmUbuntu:~$ sudo mkfs.fat -F 32 /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0\.:0-part1
```

To format a partition with NTFS, for use with modern versions of Windows, or with EXT4, for use with Linux only, use:

```
sudo mkfs.ntfs /dev/disk/by-id/YOUR_FLASH_DRIVE-PARTITION
```

or

```
sudo mkfs.ext4 /dev/disk/by-id/YOUR_FLASH_DRIVE-PARTITION
```



```
ducklord@vmUbuntu: ~  
ducklord@vmUbuntu:~$ sudo mkfs.ntfs /dev/disk/by-id/usb-INTENSO_USB_907A10000040-0\0-part1
```

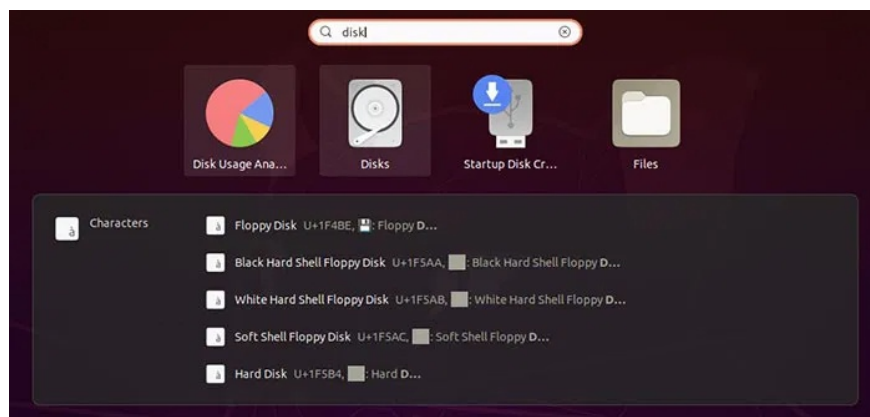
EXT4 is optimized for Linux but is not natively supported by Windows or macOS without additional software.

After formatting, you can plug in the USB and confirm that the drive is working. Just use `lsblk` again to check that the partition has a file system and is mountable.

## Check and repair USB using Disks via GUI

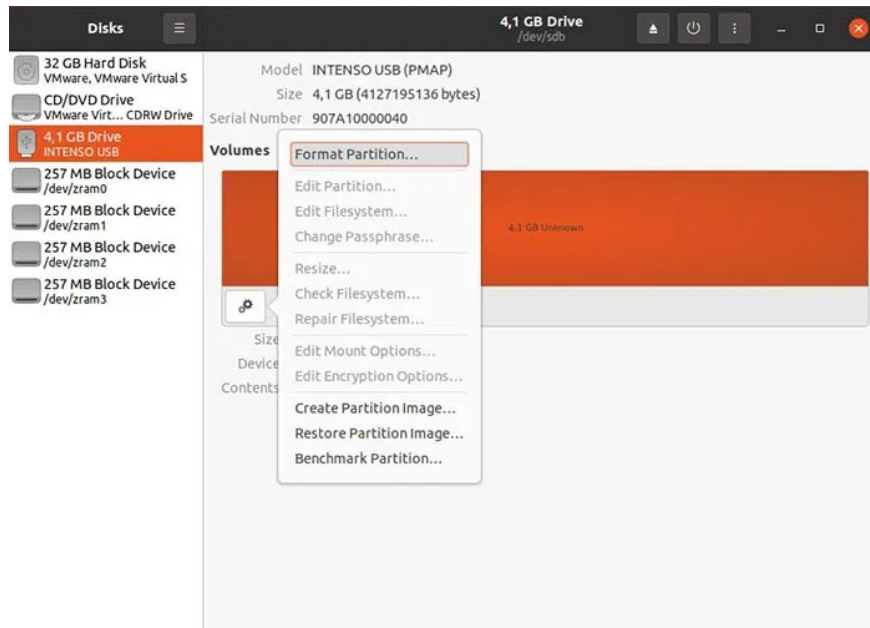
If you don't like typing commands, you can switch to the Disks tool to check and format your USB. Disks comes pre-installed on Ubuntu.

Go to the applications menu and search for **Disks** . Launch the application when you find it.



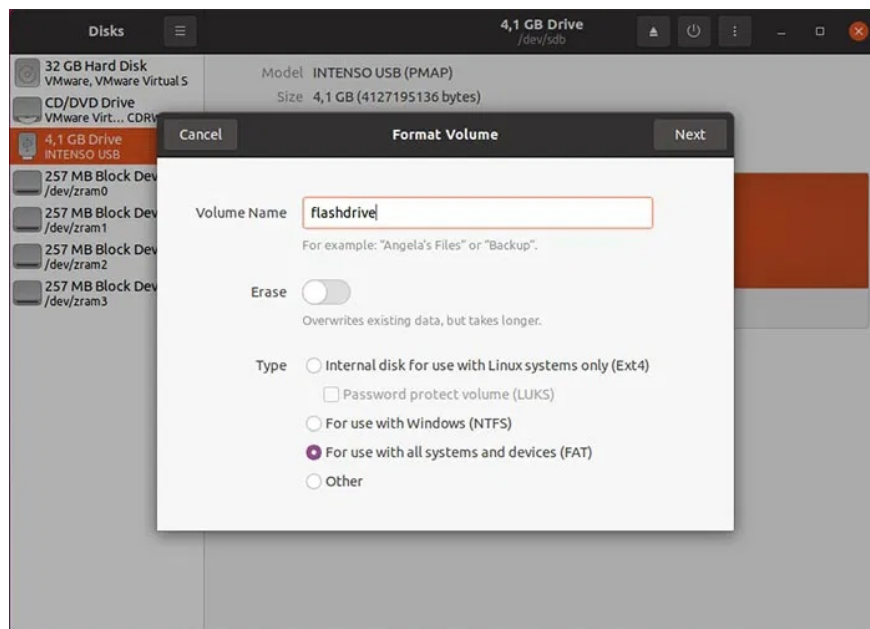
Select your USB drive from the list on the left and click the gear icon. Then, select **Repair Filesystem** and follow the wizard's steps to repair the file system.

In this case, for example, it's not a hardware problem, but a corrupted file system. While the problem can't be repaired, you can reformat the USB drive and continue using it. To do that with Disks and the USB selected, click the gear icon again and select **Format Partition** .

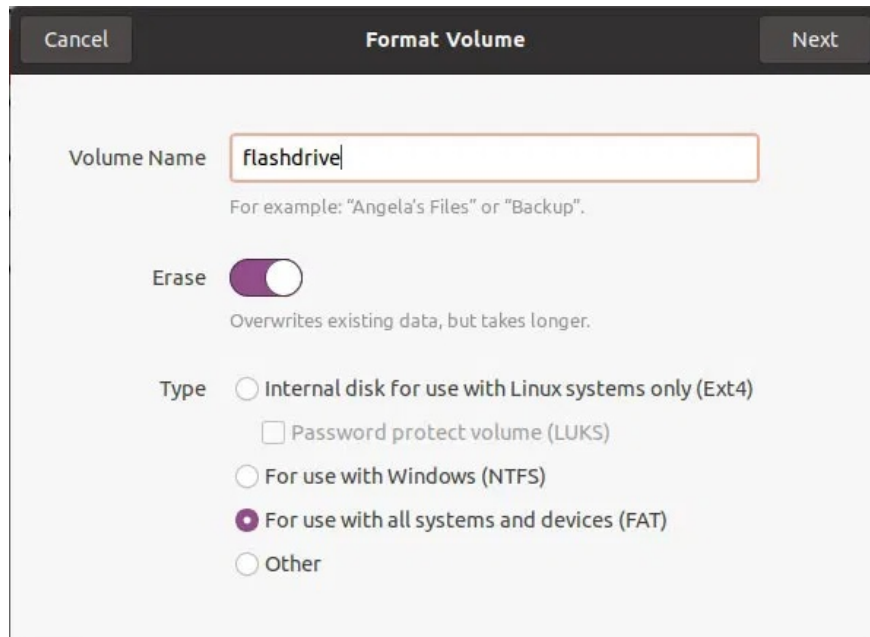


Enter a name for the USB drive in **the Volume Name** field and choose from the three most common file systems for that drive:

1. Ext4 for use with Linux
2. NTFS for use with modern versions of Windows
3. FAT for use with both, as well as other types of devices (from smartphones to game consoles)



**Pay attention to the Erase** option . Leave this option disabled for a quick format. Swipe it to **On** for a full format, completely erasing the contents of the device. When dealing with damaged devices that may also have some bad blocks, it is better to use the full erase option.



**Good luck!**

**See more articles below:**

1. Protect your Google account with a USB "security key"
2. Summary of some ways to fix USB Device Not Recognized error on Windows 7, 8 and 10
3. Want to know if your computer supports USB 3.0, read this article

You finished reading the article "**How to Fix Corrupted USB in Linux**" edited by the [TipsMake](#) team. We hope this article has provided you with many useful tech tips and tricks. You can search for similar articles on tips and guides. Thank you for reading and for following us regularly.