**Raspberry\_Pi\_Setup**

**2024\_04\_11 – EeePC**

**FAIL ON Eee-PC UBUNTU 20.04**

**Start Clean on HP W 10**

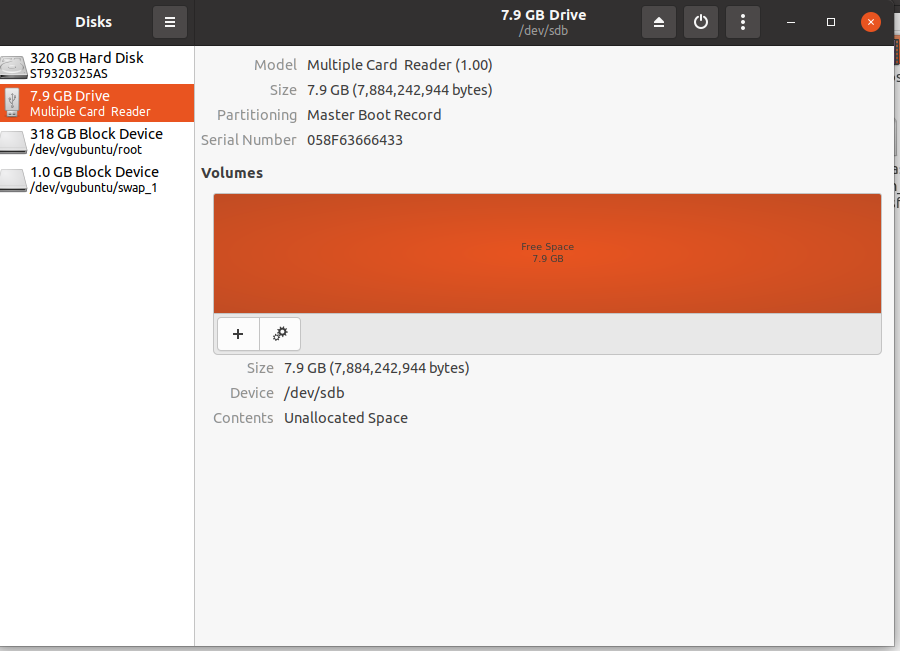
I broke the Raspberry Pi Imager → Use dd

**sudo dd if=digipi-1.8-2.img of=/dev/sdb bs=4M**

2024-03-15-raspios-bookworm-armhf-lite.img.xz

sdb looks good

sudo dd if=2024-03-15-raspios-bookworm-armhf-lite.img.xz of=/dev/sdb bs=4M



steve@EeePC:~$ sudo fdisk /dev/sdb

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): d

Partition number (1,2, default 2):

Partition 2 has been deleted.

Command (m for help): d

Selected partition 1

Partition 1 has been deleted.

Command (m for help): d

No partition is defined yet!

Command (m for help): a

No partition is defined yet!

Command (m for help): m

Help:

DOS (MBR)

a toggle a bootable flag

b edit nested BSD disklabel

c toggle the dos compatibility flag

Generic

d delete a partition

F list free unpartitioned space

l list known partition types

n add a new partition

p print the partition table

t change a partition type

v verify the partition table

i print information about a partition

Misc

m print this menu

u change display/entry units

x extra functionality (experts only)

Script

I load disk layout from sfdisk script file

O dump disk layout to sfdisk script file

Save & Exit

w write table to disk and exit

q quit without saving changes

Create a new label

g create a new empty GPT partition table

G create a new empty SGI (IRIX) partition table

o create a new empty DOS partition table

s create a new empty Sun partition table

Command (m for help): o

Created a new DOS disklabel with disk identifier 0x9ac52722.

Command (m for help): F

Unpartitioned space /dev/sdb: 7.35 GiB, 7883194368 bytes, 15396864 sectors

Units: sectors of 1 \* 512 = 512 bytes

Sector size (logical/physical): 512 bytes / 512 bytes

Start End Sectors Size

2048 15398911 15396864 7.4G

Command (m for help): w

The partition table has been altered.

Failed to remove partition 1 from system: Device or resource busy

Failed to remove partition 2 from system: Device or resource busy

The kernel still uses the old partitions. The new table will be used at the next reboot.

Syncing disks.

**steve@EeePC**:**~/Downloads**$ ls

**2024-03-15-raspios-bookworm-armhf-lite.img.xz** invite.ics

2024-03-15-raspios-bookworm-armhf-lite.img.xz.torrent **tor-browser-linux64-10.5.4\_en-US.tar.xz**

**expressvpn\_3.10.0.9-1\_amd64.deb** tor-browser-linux64-10.5.4\_en-US.tar.xz.sig

**imager\_1.6.2\_amd64.deb** **veracrypt-1.24-Update7-Ubuntu-20.04-amd64.deb**

**imager\_1.8.5\_amd64.deb** veracrypt-1.24-Update7-Ubuntu-20.04-amd64.deb.sig

**steve@EeePC**:**~/Downloads**$ sudo dpkg -i imager\_1.8.5\_amd64.deb

[sudo] password for steve:

(Reading database ... 282195 files and directories currently installed.)

Preparing to unpack imager\_1.8.5\_amd64.deb ...

Unpacking rpi-imager (1.8.5) over (1.6.2) ...

**dpkg:** dependency problems prevent configuration of rpi-imager:

rpi-imager depends on libc6 (>= 2.34); however:

Version of libc6:amd64 on system is 2.31-0ubuntu9.14.

rpi-imager depends on libgnutls30 (>= 3.7.0); however:

Version of libgnutls30:amd64 on system is 3.6.13-2ubuntu1.10.

rpi-imager depends on libqt5core5a (>= 5.15.1); however:

Version of libqt5core5a:amd64 on system is 5.12.8+dfsg-0ubuntu2.1.

rpi-imager depends on libqt5dbus5 (>= 5.14.1); however:

Version of libqt5dbus5:amd64 on system is 5.12.8+dfsg-0ubuntu2.1.

rpi-imager depends on libstdc++6 (>= 12); however:

Version of libstdc++6:amd64 on system is 10.5.0-1ubuntu1~20.04.

**dpkg:** error processing package rpi-imager (--install):

dependency problems - leaving unconfigured

Processing triggers for gnome-menus (3.36.0-1ubuntu1) ...

Processing triggers for desktop-file-utils (0.24-1ubuntu3) ...

Processing triggers for mime-support (3.64ubuntu1) ...

Processing triggers for hicolor-icon-theme (0.17-2) ...

Processing triggers for man-db (2.9.1-1) ...

Errors were encountered while processing:

rpi-imager

**So, I broke the imager!**

**Try dd**

**REFERENCE: Raspberry\_Pi\_Headless\_Setup**

<https://peppe8o.com/install-raspberry-pi-os-lite-in-your-raspberry-pi/>

**How to Create an Image of a Raspberry Pi SD Card? (Win/Linux/Mac)**

Saturday, August 7, 2021

12:17 PM

<https://raspberrytips.com/create-image-sd-card/>

**How to Create an Image of a Raspberry Pi SD Card? (Win/Linux/Mac)**

Written by [Patrick Fromaget](https://raspberrytips.com/author/admin/)in [How-To Tutorials](https://raspberrytips.com/category/how-to-tutorials/)

Creating an entire image of your SD card can be really useful on Raspberry Pi. The system and your data are on this little piece of plastic, which isn’t the safest :).

In this tutorial, I’ll show you how to make a full backup copy of your SD card (system, configuration and data).

**For a Raspberry Pi with many data or even critical data, it’s a good practice to create an image of the entire storage.**

**On Windows, Win32 Disk Imager is the best tool to do this.**

**On Linux, the dd command can do this. And on macOS, ApplePi Baker is the best choice for a graphical solution.**

Here it is for the short answer, but these tools are not easy to use for the first time. So, in this guide I’ll show you step-by-step how to do in each case.

But before that, we’ll start by a short explanation on why it’s so important to do this from time-to-time, if you aren’t sure yet.

If you are looking for quick progress on Raspberry Pi, [you can check out my e-book here](https://raspberrytips.com/book-intro). It’s a 30-day challenge, where you learn one new thing every day until you become a Raspberry Pi expert. The first third of the book teaches you the basics, but the following chapters include projects you can try on your own.

**Why do you need to create a Raspberry Pi image?**

If you are lost in all these new words and abbreviations, request my free Raspberry Pi glossary here (PDF format)!

If you are on this page, there is a good chance you already know why you want to create an image of your Raspberry Pi.

But you may not have considered all benefits of this procedure. Below is a quick list of reasons or cases to help you.

**Your SD card will stop working**

**The first thing to consider is the storage you are using on Raspberry Pi.**

As your system and critical data are on a micro SD card, don’t expect to keep years for life in the same state.

**The lifespan of a SD card is expected to be 10 years or more**. But, this mostly depends on the model, the usage and how you handle it daily. Even if you use [the best SD card with your Raspberry Pi](https://raspberrytips.com/best-sd-card-raspberry-pi/), it may still happen.

If you use your SD card with caution, always in the same device, with a low disk usage on your Raspberry Pi, it could have a long lifespan. But if you change devices regularly, travel a lot or let your Raspberry Pi run on heavy load all day, it probably won’t.

Also, **your system may be corrupted one day, due to updates, security breaches or mishandling**.

In short, don’t expect your SD card to work forever, and anticipate a malfunction at anytime.

[Raspberry Pi Bootcamp](https://raspberrytips.com/post-bootcamp)

**Sale: 10% off today.**

Take it to the next level.

I'm here to help you get started on Raspberry Pi.

Learn all the skills you need in the correct order.

[Watch now!](https://raspberrytips.com/post-bootcamp)

**A solid backup is mandatory**

Basically, **I never expect a storage method to be 100% safe**.

That’s why I recommend doing regular backups of them if you keep critical data on your card.

I already have an [entire guide on how to back up your Raspberry Pi](https://raspberrytips.com/backup-raspberry-pi/).

And today, we’ll see how to clone the entire SD card on your computer.

**This is the best way if you have a lot of critical or important data.**

By the way, even if you have a retro gaming solution like Retropie on it and think you don’t need to be concerned, think again. If there is a huge catalog of games on your Raspberry Pi, I really recommend backing up your SD card :).

A copy on your computer or on an external drive ([I recommend this one](https://geni.us/ciOWl6)) will be worth it if you spend a lot of time on your Pi.

**Use this technique to save time**

Small anecdote here, when I started on Raspberry Pi, I tested 5 to 10 new projects each week ([while writing my experiences on RaspberryTips](https://raspberrytips.com/2years-raspberrytips/)).

**SD card preparation, update and configurations (like Wi-Fi and keyboard layout) was a waste of time for me.**

My solution was to create a basic Raspbian installation on a small SD card, and create an image on my computer.

This way, I could flash this image instead of the one from the Raspberry Pi Foundation and everything would be ready to use.

I have learned a lot and generally use the configuration files to do this. I have the files on my computer, and I copy them to each SD card I flash ([you can learn how to do this in this article](https://raspberrytips.com/pi-zero-setup-without-keyboard/)). In the beginning this was very useful.

Even so, there are probably many cases where it makes sense to do this (cluster? Multiple web servers? Many Raspberry Pi deployment?).

**Create an image on Windows**

As most of you are generally on Windows, let’s start with this system.

**Win32 Disk Imager**

Win 32 Disk Imager is a well-known tool on Windows for Raspberry Pi And Linux users.

**It’s often used to create SD cards (or USB disk) from an image of an operating system downloaded on the Internet.**

You can download it [here on SourceForge](https://sourceforge.net/projects/win32diskimager/).

It looks like this:

**But the goal today is the opposite: creating an image from an SD card.**

**Win 32 Disk Imager can also do this, let’s see how!**

**Step-by-step image creation with Win32 Disk Imager**

**Here are the steps to follow to create an image of any SD card on Windows, with Win 32 Disk Imager:**

* Insert your SD card in your computer.  
  If you don’t have an SD card reader on your computer, you’ll need a USB adapter for this ([I explain everything here](https://raspberrytips.com/usb-card-reader-raspberry-pi/)).
* **Find the partition letter corresponding to your SD card**: Open the File explorer and go to “This PC”.

Take note of the drive letter, you’ll need it later.

* Open Win 32 Disk Imager.
* **Start by choosing an image location and name** for your image:

Make sure to have enough free space on your disk where you want to store the image (a 64 GB SD card can quickly fill a SSD disk ^^).  
You can use a local storage, or an external USB drive.

* Then **select the device you want to back up**:

You’ll typically only see the “boot” partition, but don’t worry, Win32DiskImager will create an entire image of all partitions on the device.

* You can now **click on “Read” to start the copy**:
* The process will start and it can take some time depending on your SD card size (between 15min and 1h in general for a standard size):

Once done, your image is safe, and we’ll now see how to flash it on another card.

[Master your Raspberry Pi in 30 days](https://raspberrytips.com/master-raspberry-pi)

**Sale: 10% off today.**

Download the eBook.

Uncover the secrets of the Raspberry Pi in a 30 days challenge.

Learn useful Linux skills and practice multiples projects.

[Get it Now!](https://raspberrytips.com/master-raspberry-pi)

**Restoring the image to any SD card**

**I recommend doing at least one test to recover the SD card from this image before considering you are safe.**

I have had too many backups in my life that didn’t work when needed, so we can never be sure if a backup will work in real conditions :).

What you can do, is **insert another SD card in your computer and flash it with the image.**

To do this, you can use Win32 Disk Imager almost the same way:

* **Choose the image file**.
* **Pick the device letter** in the list.
* And **click on “Write”** to start the copy:

If you prefer, there are other tools to do the same thing.

The one I recommend everywhere on this website is [Etcher](https://www.balena.io/etcher/). You can also use [Raspberry Pi Imager](https://www.raspberrypi.org/blog/raspberry-pi-imager-imaging-utility/) if you want (official software from the Raspberry Pi Foundation).

**You may also like:**

* [25 awesome Raspberry Pi project ideas at home](https://raspberrytips.com/raspberry-pi-projects-for-home/?related)
* [15 best operating systems for Raspberry Pi (with pictures)](https://raspberrytips.com/best-os-for-raspberry-pi/?related)
* [My book: Master your Raspberry Pi in 30 days](https://raspberrytips.com/book-link)

**Create an image on Linux**

If you are a Linux user, let’s see how to do this on your favorite system!

I’ll show you on Ubuntu, but the tool is the same on any distribution.

**The dd command**

**“dd” is a base command on Unix. The goal is to offer a tool to manage files.**

You can use it to erase a partition (filling it with zeros), generate a random file, but also to manage disk images!

I’m also using it for benchmarks (like in [this post about SD cards](https://raspberrytips.com/best-sd-card-raspberry-pi/)).

As dd can do a complete backup of any disk, it’s really useful for this.

Let’s see how to use it!

A bit lost in the Linux command line? [Check this article first](https://raspberrytips.com/raspberry-pi-commands/), which will give you the most important ones to remember with a free cheat sheet you can download to have all of them at your fingertips.

**How to use “dd” to back up the SD card**

**Find the device name**

Looking for the drive letter on Windows is pretty easy, but on Linux it’s a bit more hidden.

**A device name on Linux is something like /dev/sdX (if you use an USB adapter), or /dev/mmcblkX (if your computer has an SD card reader).**

On Ubuntu, you can use the Disk Utility to find this information:

I’m using a 16 GB SD card for this test, so this is this one (/dev/sde).

If you aren’t on Ubuntu and can’t find a similar tool, **you can also jump to the terminal and use the following command:**

sudo fdisk -l

It will show you a list of drives on your computer. You just need to **find the one corresponding to your SD card.**

In my case, it looks like this:

So, we have a disk named /dev/sde, with two partitions (/dev/sde1 and /dev/sde2).

**Create the image with dd**

Once we know the device name, we need the correct command to create the image of this device:

* **Open a terminal**.
* **Type the following command:**sudo dd bs=4M if=/dev/sde of=/home/username/MyImage.img
* **Don’t forget to replace the device name (if for input file) and the file destination (of for output file).**
* You’ll get something like this:

**Expect at least 15 minutes to create the image** (depending on the SD card size).

**Image restoration to the SD card**

Copying back the image to another SD card is almost the same thing.

**I recommend trying this at least one time, just to be sure that your image is working** (don’t try on the same SD card!).

To copy an image to a new SD card, there are two ways you can use:

* **The first one is to use dd again, in the reverse order:**
  + The command is something like:  
    sudo dd bs=4M if=/home/username/MyImage.img of=/dev/sde
  + For the first time, you need to edit this command with the correct path, image name and device name.
* **The second way, that I always recommend is to use Etcher**:
  + Etcher is a free tool you can [download here](https://www.balena.io/etcher/).  
    The good news is that it’s a graphical tool and very intuitive.  
    The dd command seems simple now because you just used it to create the image, but in 6 months, you probably won’t remember the correct options.
  + The tool looks like this:

Just select your backup image, your drive (automatic in theory), and click on “Flash!” to start the copy.

Whatever the method you use, it should create an exact replica of the original SD card.

Once done, insert it in your Raspberry Pi and check that everything is working correctly.

**Create an image on macOS**

The last operating system is macOS.

I will be quick on this tutorial, as I didn’t test because I don’t have a Mac :).

But I know they are working methods to create an image from your Raspberry Pi.

[Master Python on Raspberry Pi](https://raspberrytips.com/masterpython-ip)

**Sale: 10% off today.**

Get the eBook.

Do more with your Raspberry Pi, learn the useful concepts and take the shortcuts.

You miss half of the fun of using a Raspberry Pi if you don’t know anything about Python.

[Download here](https://raspberrytips.com/masterpython-ip)

**First method: use dd**

**The first method is to use “dd”, like in the previous part for Linux.**

As macOS is based on Unix, dd is also available on it.

Just run a terminal and follow the Linux part :).

**Second method: try ApplePi Baker**

**If you prefer a graphical tool, I found ApplePi Baker that seems to do exactly what we need.**

You can download it here on [Tweaking4All](https://www.tweaking4all.com/software/macosx-software/applepi-baker-v2/).

Scroll down to the ApplePi-Baker V2 download link, get it and install it:

The tool looks like this:

It’s intuitive, with big symbols for each step.

**Start by choosing the disk you want to back up on the left, then click on “Backup” to set the image name and location.**

Once done, the process starts, and your image is created as with dd.

**Restoration on macOS**

As on any operating system, **it’s mandatory to try your backup at least once before considering the job done**.

What you can do, is to take a new SD card and flash the image to it, then test it on your Raspberry Pi

**I already explained in the Linux part two ways you can use: dd and Etcher.** They will work perfectly on macOS.

If you want to use these tools, go to the previous part about Linux and find the “Image restoration” section.

**The other way if you installed ApplePi-Baker, is to use the “Restore” option on the right.**

Pick the disk you want to flash (“1” on my picture), and click on “Restore” to select the image to copy.

The process will starts immediately. After a few minutes, try to boot this SD card on your Raspberry Pi, and see how it goes.

If everything is OK, you can consider your backup safe.

Redo the same process regularly to keep an updated version of the image (if needed).

**Video**

<https://youtu.be/xSxNJSkSgpk>

**Conclusion**

If you are lost in all these new words and abbreviations, request my free Raspberry Pi glossary here (PDF format)!

That’s it, you now know why you should create an image of a Raspberry Pi, and how to do so on any operating system.

I hope this guide was useful for you, if it’s the case, please share it on your favorite social network!

Don’t forget this technique, even if you don’t need it yet. It will save you a lot of time in the future if you remember to create images for your most critical systems.

By the way, try to keep the image at a safe location too (or keep two copies), your computer disk can also break or be formatted :).

A good external hard drive ([my favorite is this one on Amazon](https://geni.us/ciOWl6)) is probably a good option to consider. Personally, I’m using a NAS at home for all these backups ([this one from Synology](https://geni.us/kMRF2e)), but that’s a bit more expensive.

**Related links:**

* [Complete guide to back up a Raspberry Pi](https://raspberrytips.com/backup-raspberry-pi/)
* [Benchmark of the best SD cards for Raspberry Pi](https://raspberrytips.com/best-sd-card-raspberry-pi/)
* [25 projects ideas to try at home on Raspberry Pi](https://raspberrytips.com/raspberry-pi-projects-for-home/)

From <[*https://raspberrytips.com/create-image-sd-card/*](https://raspberrytips.com/create-image-sd-card/)>

Headless SSH -- **SSH (Secure Shell)**

Saturday, August 7, 2021

12:13 PM

<https://www.raspberrypi.org/documentation/remote-access/ssh/README.md>

* [Hardware](https://www.raspberrypi.org/products/)
* [Software](https://www.raspberrypi.org/software/)
* [Books & magazines](https://www.raspberrypi.org/books-magazines/)
* [Learn](https://www.raspberrypi.org/learn/)
* [Teach](https://www.raspberrypi.org/teach/)
* [About us](https://www.raspberrypi.org/about/)

[documentation](https://www.raspberrypi.org/documentation) > [remote-access](https://www.raspberrypi.org/documentation/remote-access) > [ssh](https://www.raspberrypi.org/documentation/remote-access/ssh)

**SSH (Secure Shell)**

You can access the command line of a Raspberry Pi remotely from another computer or device on the same network using SSH.

The Raspberry Pi will act as a remote device: you can connect to it using a client on another machine.

You only have access to the command line, not the full desktop environment. For a full remote desktop, see [VNC](https://www.raspberrypi.org/documentation/remote-access/vnc/README.md).

**1. Set up your local network and wireless connectivity**

Make sure your Raspberry Pi is properly set up and connected. If you are using wireless networking, this can be enabled via the desktop's [user interface](https://www.raspberrypi.org/documentation/configuration/wireless/README.md), or using the [command line](https://www.raspberrypi.org/documentation/configuration/wireless/wireless-cli.md).

If you are not using wireless connectivity, plug your Raspberry Pi directly into the router.

You will need to note down the IP address of your Pi in order to connect to it later. Using the ifconfig command will display information about the current network status, including the IP address, or you can use hostname -I to display the IP addresses associated with the device.

**2. Enable SSH**

As of the November 2016 release, Raspberry Pi OS has the SSH server disabled by default. It can be enabled manually from the desktop:

1. Launch Raspberry Pi Configuration from the Preferences menu
2. Navigate to the Interfaces tab
3. Select Enabled next to SSH
4. Click OK

Alternatively, [raspi-config](https://www.raspberrypi.org/documentation/configuration/raspi-config.md) can be used in the terminal:

1. Enter sudo raspi-config in a terminal window
2. Select Interfacing Options
3. Navigate to and select SSH
4. Choose Yes
5. Select Ok
6. Choose Finish

Alternatively, use systemctl to start the service

sudosystemctl enablesshsudosystemctl start ssh

When enabling SSH on a Pi that may be connected to the internet, you should change its default password to ensure that it remains secure. See the [Security page](https://www.raspberrypi.org/documentation/configuration/security.md) for more details.

**3. Enable SSH on a headless Raspberry Pi (add file to SD card on another machine)**

For headless setup, SSH can be enabled by placing a file named ssh, without any extension, onto the boot partition of the SD card from another computer. When the Pi boots, it looks for the ssh file. If it is found, SSH is enabled and the file is deleted. The content of the file does not matter; it could contain text, or nothing at all.

If you have loaded Raspberry Pi OS onto a blank SD card, you will have two partitions. The first one, which is the smaller one, is the boot partition. Place the file into this one.

**4. Set up your client**

SSH is built into Linux distributions and Mac OS, and is an optional feature in Windows 10. For mobile devices third-party SSH clients are available. See the following guides for using SSH with the OS on your computer or device:

* [Linux & Mac OS](https://www.raspberrypi.org/documentation/remote-access/ssh/unix.md)
* [Windows 10](https://www.raspberrypi.org/documentation/remote-access/ssh/windows10.md)
* [iOS](https://www.raspberrypi.org/documentation/remote-access/ssh/ios.md)
* [Android](https://www.raspberrypi.org/documentation/remote-access/ssh/android.md)

[A cartoon of a cat

Description automatically generated](https://github.com/raspberrypi/documentation/blob/master/remote-access/ssh/README.md)

[View/Edit this page on GitHub](https://github.com/raspberrypi/documentation/blob/master/remote-access/ssh/README.md)

[Read our usage and contributions policy](https://www.raspberrypi.org/creative-commons/)

[Creative Commons Licence](https://www.raspberrypi.org/creative-commons/)

About Us

From <[*https://www.raspberrypi.org/documentation/remote-access/ssh/README.md*](https://www.raspberrypi.org/documentation/remote-access/ssh/README.md)>