

Backup and Restore Entire Rpi System

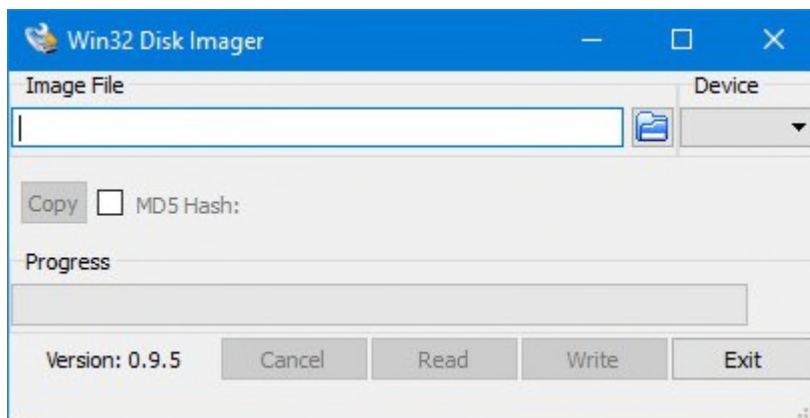
Saturday, February 24, 2018 2:29 PM

Note: This method will create an image file exactly the size of the SD card's total capacity. For example, if you have a 16 GB SD card, the resulting image file will also be 16 GB, no matter how much space your installation is actually using. **There is a method to shrink the image size, but it works only on Linux.** It is covered later in these instructions.

Windows Instructions

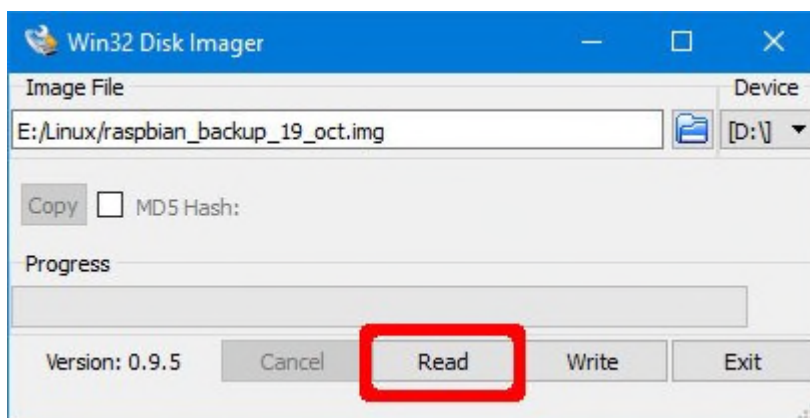
Backup Raspberry Pi SD Card

1. Insert the micro SD card that you want to clone in your PC using a USB or built-in card reader.
2. [Download](#) and install Win32DiskImager, and run it. You will see a screen like this:

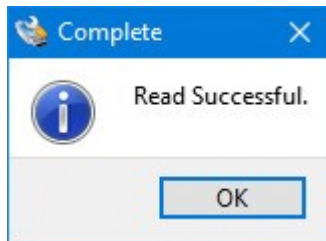


3. Under the **Device** section in the top right corner, **select the drive of your SD card**. It is D: in my case. Now click on the folder icon to its left, and **choose a location and filename** where the image file will be saved. I chose the file name raspbian_backup_19_oct.img. You can choose whichever filename you like, but it is preferable to have an .img extension. Once done, click on the **Read** button.

Important: When Windows recognizes the SD card, it will probably "freak out" and you could see multiple screens pop up. Just keep closing them until you see the one that has the "real" image you want to back up (probably says "recovery" on it). Note the device letter associated with it and **make sure the correct device is selected in Win32 Disk Imager**.



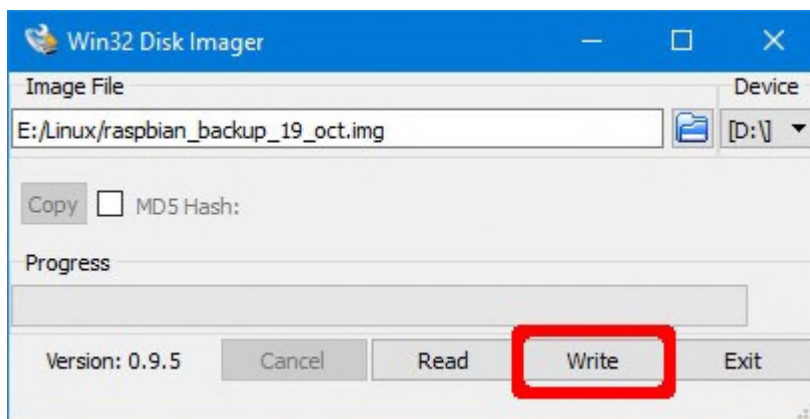
4. This will make a clone of the SD card in the location you specified. The copying **will take a while** to complete, so do not turn off or put your PC to sleep during this period. Once it is done, you will get a “Read Successful” message.



You are now free to insert the card back in your Raspberry Pi and break things or install a new distro. Once you are done playing and want to restore the backed up image, follow the steps in the next section.

Restore Raspberry Pi SD Card

Insert the micro SD card in your PC, and open Win32DiskImager. **Select the image file** you created previously, and the appropriate **drive** in the **Device** section. Now, click on the **Write** button. The image will be saved to the SD card, exactly how it was, at the time you copied it.



Again, this process will take a while depending on the size of the SD card. Once the restore is complete, you can remove the card from your PC, and insert it back in the Raspberry Pi.

Linux Instructions

Backup Raspberry Pi SD Card

1. Insert the SD card in your PC using a USB or built-in card reader. Now open a **Terminal** window, and enter the command `sudo fdisk -l`. This will list all the filesystems present on your system.
2. Try to find out the *device name* of your SD card. I have a 16GB SD card, so it is easily identified as the device `/dev/sdb` which has a size of 14.9GB. This is because the actual storage on a device is always slightly lower than advertised. **Note down this device name.**

```
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 4096 bytes
I/O size (minimum/optimal): 4096 bytes / 4096 bytes
Disklabel type: dos
Disk identifier: 0xa11cac6d

Device      Boot      Start          End      Sectors  Size Id Type
/dev/sda1   *           2048         206847     204800   100M 7 HPFS/NTFS/exFAT
/dev/sda2             206848     184322047   184115200  87.8G 7 HPFS/NTFS/exFAT
/dev/sda3       184322048   434464767   250142720  119.3G 83 Linux
/dev/sda4       434466814   976771071   542304258  258.6G f W95 Ext'd (LBA)
/dev/sda5       440324096   696324095   256000000  122.1G 7 HPFS/NTFS/exFAT
/dev/sda6       696326144   976771071   280444928  133.7G 7 HPFS/NTFS/exFAT
/dev/sda7       434466816   440322047     5855232    2.8G 82 Linux swap / Solaris

Partition 4 does not start on physical sector boundary.
Partition table entries are not in disk order.

Disk /dev/sdb: 14.9 GiB, 15931539456 bytes, 31116288 sectors
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: 0x04ac9d6c

Device      Boot      Start          End      Sectors  Size Id Type
/dev/sdb1             8192         137215     129024    63M c W95 FAT32 (LBA)
/dev/sdb2       137216   31116287   30979072  14.8G 83 Linux
shivam@beebom-HP-15-Notebook-PC ~ $
```

3. Use the dd command to write the image to your hard disk. For example:

```
sudo dd if=/dev/sdb of=~/.raspbian_backup.img
```

Here, the *if* parameter (input file) specifies the file to clone. In my case, it is /dev/sdb, which is my SD card's device name. Replace it with the device name of yours. The *of* parameter (output file) specifies the file name to write to. I chose raspbian_backup.img in my home directory.

Note: Be careful, and double check the parameters before executing the dd command, as entering the wrong parameters here can potentially destroy the data on your drives.

You will not see any output from the command until after the cloning is complete, and that might take a while, depending on the size of your SD card. Once it is complete, you will see an output like the following.

```
shivam@beebom-HP-15-Notebook-PC ~ $ sudo dd if=/dev/sdb of=~/.raspbian_backup.img
31116288+0 records in
31116288+0 records out
15931539456 bytes (16 GB, 15 GiB) copied, 1140.69 s, 14.0 MB/s
shivam@beebom-HP-15-Notebook-PC ~ $
```

You can now remove the SD card and use it in your Pi. Once you are ready to restore the backed up image, follow the instructions below:

Restore Raspberry Pi SD Card

1. Insert the SD card in your PC. Before we restore the image, it is important to make sure that the SD card's partitions are *unmounted*. To verify this, open the **Terminal**, and execute the command `sudo mount | grep sdb`. Here, replace sdb with your SD card's device name.

If you see a blank output, you do not need to do anything. If you do see some mounted partitions, *umount* the listed ones. For example:

```
sudo umount /dev/sdb1 /dev/sdb2 /dev/sdb3 /dev/sdb4
```

2. Use the dd command to write the image file to the SD card:

```
sudo dd if=~/.raspbian_backup.img of=/dev/sdb
```

This is like the command we used to make a clone, but **reversed**. This time, the *input file* is the backup image, while the *output file* is the SD card device.

Again, verify, and **double-verify the parameters here**, as entering the wrong command here will cause permanent data loss.

Once the write is complete, you will see a confirmation from dd. You can then remove the card from your PC, and insert it back in the Raspberry Pi.

Shrink the Cloned Raspberry Pi Image (Linux-only)

As mentioned at the starting of the article, these methods create an image file that is equal to the total capacity of the SD card. For example, cloning an SD card with a capacity of 32GB will create an image file of 32 GB, even if only 5 GB is actually in use on the card. This is fine if you only have one or two such images, but any more than that (especially if you use an SSD) will cause you to run out of space.

To work around this limitation, we will use [PiShrink](#), a script that **automatically shrinks a Pi image** that will then automatically resize to the max size of the SD card on boot. This also makes copying the image back onto the SD card much faster.

Unfortunately, this tool is only available on Linux. If you do not have Linux installed, you can install the latest version of Ubuntu or Linux Mint in a virtual machine, and run this script there. Here is how to use it:

1. Download the **PiShrink script**, and make it executable. In a **Terminal**, enter the following two commands:

```
wget https://raw.githubusercontent.com/Drewsif/PiShrink/master/pishrink.sh
```

```
chmod +x ./pishrink.sh
```

2. **Run the script**, followed by the **name of the image** that you want to shrink.

```
sudo ./pishrink.sh ./raspbian_backup.img
```

```
shivam@beebom-HP-15-Notebook-PC ~ $ sudo ./pishrink.sh ./raspbian_backup.img
Creating new /etc/rc.local
e2fsck 1.42.13 (17-May-2015)
Pass 1: Checking inodes, blocks, and sizes
Pass 2: Checking directory structure
Pass 3: Checking directory connectivity
Pass 4: Checking reference counts
Pass 5: Checking group summary information
/dev/loop0: 133190/971040 files (0.1% non-contiguous), 1020663/3872384 blocks
resize2fs 1.42.13 (17-May-2015)
resize2fs 1.42.13 (17-May-2015)
Resizing the filesystem on /dev/loop0 to 1372187 (4k) blocks.
Begin pass 3 (max = 119)
Scanning inode table      XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
The filesystem on /dev/loop0 is now 1372187 (4k) blocks long.
Shrunk ./raspbian_backup.img from 15G to 5.3G
```

3. The image will then be shrunk. Once done, you can write the shrunk image file to the SD card as mentioned in the methods listed above. For example, on Linux:

```
sudo dd if=~/.raspbian_backup.img of=/dev/sdb
```