Guide : Portable Debian-Based Distro Install Usable With BIOS & UEFI

Disclaimer :

If BSD is just as good for your use-case (which it really can be !) , you can skip this and just use NomadBSD which (*by-design*) functions portably. It will have significant differences from a Linux system, though.

NOTES :

- *This guide is apt for beginners , on nearly any distro, and the end result will most likely be an OS that can work on any modern x86-based PC, and which can easily be moved from one system to another.*

- *Whether or not your install will be compatible with old x86_32 processors will be determined by your choice of 32/64 bit version of ISO. A 32-bit version is unavailable for new releases of Ubuntu-family of distros. A 32-bit OS is necessary for using old 32-bit processors and 16-bit software, both of which are very rare and mostly obsolete. 64-bit allows you to address more than 4GB , such as using 4GB+ RAM, files, etc and all 32-bit software can run on a 64-bit OS, while the reverse is not possible. Hence, a 64-bit version is **recommended**.*

- *This guide was tested on Lubuntu 20.04 . It requires : Internet connection, computer(s) with BIOS and UEFI boot options , 2 Drives, Live install image of distro.*

Stepwise Instructions :

1. Go to your chosen distro's website, and download the live-installer image (generally a .iso file)
2. Following the instructions on the distro's website , flash the ISO to a drive. Keeping it physically plugged in, shut down.
3. This USB is now your installation media. You must now boot from it **through BIOS/Legacy mode** . How exactly you do this depends on how the manufacturer has set-up your machine. Searching the internet for "`your_computer/motherboard_model + boot options menu`" will help.
4. Once at the desktop of your live system, plug in the drive you wish to use as the target (which Linux will be installed onto).
5. Open a terminal and run `sudo apt install gparted`. There are default tools similar to and based on Gparted, and they might work, but this guide will use Gparted as it is a very standard utility with all necessary features.
6. Launch Gparted. Change the dropdown menu in the top right-hand corner to select your target USB device.
7. Right click any existing mounted partitions on said target USB and unmount/swapoff them.
8. Select "Device" on the top bar, and select "Create partition table". Change the dropdown to "GPT" (note: this will erase all data on the disk, ensure data is backed up to your satisfaction first!).
9. Create a new partition. The size should be **1MB** and the file system should be **unformatted**.
10. Create another partition. The size should be **100MB** and the file system should be **FAT32**.
11. Create another partition. The size should be around **~1GB** and the file system should be **swap**.
12. Create one last partition, which will be used as the root filesystem of your installation. The size can be all the space left on your disk. File system should be **ext4**.
13. Apply the changes by clicking the tick button in Gparted.
14. Right click the 1MB partition, select "Manage flags", and set the flags to `bios_grub`.
15. Also set the flags of the 100MB partition to `boot` and `esp`.
16. Close Gparted, launch the distro's install tool and proceed with the installation as you normally would, until you reach the partitioning step.
17. Select "Manual partitioning" or "Advanced" at the partitioning step. You'll be asked to specify the partitions to use for installation.
18. Set the ext4 partition as the root partition (mount point = `/`). If it is shown, check the box to specify to format this partition.
19. Don't forget to change the bootloader dropdown to your USB drive!

20. Confirm that the partitions are correct and then continue.
21. Wait for installation to finish, then power off your machine.
22. Boot from your installation media again, this time in UEFI mode. Follow the same steps to get to the live system desktop.
23. Plug in your target USB again. Open a terminal.
24. Run `sudo fdisk -l`. Take note of the `/dev/sdx` label of your USB drive, it will probably be `/dev/sdb` or `/dev/sdc`. In these next commands, `/dev/sdx` will refer to your disk. Make sure you substitute the `x` for the actual letter shown for your one!

```
sudo umount /dev/sdx4
```

```
sudo swapoff /dev/sdx3
```

`sudo umount /dev/sdx2` (this one may not be mounted but it's best to make sure they're all unmounted anyway)

(`sdx1` is the BIOS boot partition so it will never be mounted).

```
sudo mount /dev/sdx4 /mnt
```

```
sudo mkdir -p /mnt/boot/efi
```

```
sudo mount /dev/sdx2 /mnt/boot/efi
```

```
sudo swapon /dev/sdx3
```

```
sudo mount --bind /dev /mnt/dev
```

```
sudo mount --bind /dev/pts /mnt/dev/pts
```

```
sudo mount -t proc proc /mnt/proc
```

`sudo mount -t sysfs sysfs /mnt/sys`

`sudo mount -t tmpfs tmpfs /mnt/run`

`sudo chroot /mnt`.

`apt install grub-efi-amd64` (For 32-bit instead of 64-bit, change `amd64` to `i386`).

`grub-install --efi-directory=/boot/efi --target=x86_64-efi --removable` (Replace `x86_64-efi` with `i386-efi` if on a 32-bit system. This will install the UEFI bootloader(GRUB) . If you experience problems at this stage, `mount /dev/sdx2 /boot/efi` and retry. It may also be possible to copy the EFI bootloader files from the live system instead of installing, but I felt it was more robust to do a clean install of the bootloader)

`update-grub`.

`blkid | grep /dev/sdx2` (`x` is *still a placeholder*). From the output of this command, take note of the `UUID` number, which is in the format `XXXX-XXXX`.

`echo "UUID=XXXX-XXXX /boot/efi vfat umask=0077 0 1" >> /etc/fstab` (`XXXX-XXXX` is a *placeholder*, replace it with your partition's UUID).

Now `exit` your chroot.

25. Power off your system (all currently mounted partitions will be unmounted automatically).
26. Boot from your new installation in either Legacy or UEFI mode; both will work!

Proprietary Drivers For Portability:

After booting, the following steps can be optionally followed to increase portability, by installing *proprietary* drivers. Without these steps, you should still be able to boot up, but functionality may be limited in some ways.

- Wifi support for mac hardware : This AskUbuntu forum post.
- Nvidia Drivers : `sudo apt install nvidia-driver-VERSION_NUMBER_HERE`

  Go to Nvidia's driver download page and input details & click search. From the result, the non-decimal part of the version number can be entered in the above terminal command, replacing 'VERSION_NUMBER_HERE' with the number.

  For example, the website may say Version : 440.100, you must enter `sudo apt install nvidia-driver-440` .

  Nvidia driver installation may not be necessary on Linux Mint since recent versions come with them built-in. Similarly, Pop!_OS Comes with the option of an iso image with nvidia drivers baked-in.

Credits: "*Stepwise Instructions*" 4 to 26 : This answer answer by Daniel Masey

---