

# Installation of Debian 12 UEFI on USB stick using 'debootstrap'

[ivanb.neocities.org/blogs/y2024/debootstrap](https://ivanb.neocities.org/blogs/y2024/debootstrap)

## Requirements

- Computer with 64 bit processor (Intel or AMD) and installed either *Debian 10*, *Debian 11*, or *Debian 12* on HDD/SSD in **UEFI** mode.
- For better results, 'Legacy mode' should be disabled in UEFI settings (so-called BIOS settings).
- USB stick with minimum size of 4 GiB (32 GiB recommended).

Warning: all data on the USB stick will be destroyed during the installation!

In this guide the installation is performed on **32 GiB** USB stick. Change the size of your partitions accordingly to your USB stick's maximum capacity.

## Preparation

Install 'debootstrap'.

```
sudo apt update
sudo apt upgrade -y
sudo apt install debootstrap
```

Connect an USB stick to a computer.

Find the device name.

```
sudo dmesg
```

Read through the last entries of 'dmesg' output. In my case, the lines 'usb 2-5: New USB device found', 'usb 2-5: Product: Flash Drive' and '[sdb] 62656641 512-byte logical blocks: (32.1 GB/29.9 GiB)' indicate that the USB stick is identified as **/dev/sdb**.

Warning: if you select the wrong device during installation, all the data on that device will be destroyed.

Run 'lsblk' to list all available devices.

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPPOINT
sda	8:0	0	953.9G	0	disk	
├─sda1	8:1	0	560M	0	part	/boot/efi
├─sda2	8:2	0	4G	0	part	[SWAP]
├─sda3	8:3	0	120G	0	part	/
└─sda4	8:4	0	829.3G	0	part	/home
sdb	8:16	1	29.9G	0	disk	
└─sdb1	8:17	1	29.9G	0	part	

My USB stick has single partition formatted as FAT32 by manufacturer. The USB stick will be repartitioned and all data on the partition along with partition table will be lost.

Warning: if you select the wrong device during the installation, all the data on that device will be destroyed.

## Partitioning the USB stick

---

Make sure you are working with the correct device! Use 'su -l' command to switch user to root.

```
su -l  
cd /
```

Check your \$PATH variable.

```
echo "$PATH" | tr ':' '\n'
```

If it lacks '/sbin' and '/usr/sbin', add them (bash syntax).

```
declare -x PATH="$PATH:/sbin:/usr/sbin"
```

Use 'gdisk' to partition the device.

```
gdisk /dev/sdb
```

On 'gdisk' prompt type 'o' and press 'Enter' to create a new empty GUID partition table (GPT).

Type 'n' to create a **new partition**, press 'Enter' on prompts to select partition number and the first sector. For last sector type '+560M'. If the boot partition will be smaller than 560 mebibytes, some buggy UEFI firmware may fail to read the contents of boot partition.

Type 'ef00' in the 'Hex code or GUID' prompt to change the type of partition to the 'EFI system partition' type. You can always press 't' to change the partition's type code, if you've accidentally chosen a wrong partition type.

Type 'n' to create a **new partition**, press 'Enter' on prompts to select partition number and the first sector. For last sector type '+1500M'. Type '8200' in the 'Hex code or GUID' prompt to change the type of partition to the 'Linux swap' type. It is important to have a swap partition on Linux system. Without swap, if the kernel runs out of memory, it will kill random process to free the memory. Depending on the process, the system may crash resulting in data loss.

Type 'n' to create a **new partition**, press 'Enter' on prompts to select partition number and the first sector. For last sector type '+18G'. Type '8300' in the 'Hex code or GUID' prompt to change the type of partition to the 'Linux filesystem' type.

Type 'n' to create a **new partition**, press 'Enter' on prompts to select partition number, the first sector and the last sector. Type '0700' in the 'Hex code or GUID' prompt to change the type of partition to the 'Microsoft basic data' type. This partition will have size around

9.9 GiB and will be formatted as FAT32 filesystem; any foreign operating system will be able to mount this partition allowing you to use this USB stick as a regular USB flash drive on operating systems such as 'Microsoft Windows' and 'MacOS'.

Type 'p' to view the full partition table, type 'i' to view detailed information on each individual partition. It is advisable to save that information into a text file, for you to be able to restore partitions manually if partition table gets damaged in the future.

Type 'w' and press 'Enter' to save partition table and exit 'gdisk'.

An automatic backup of the partition table can be performed with this command:

```
sgdisk -b usbBackup.gpt /dev/sdb
```

To restore the partition table from 'usbBackup.gpt' file, run 'gdisk /dev/sdb', press 'r' to enter 'recovery and transformation' menu, then press 'i' to load partition data from 'usbBackup.gpt' file. Type 'w' to save partition table.

Run 'lsblk' to list all available devices.

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	953.9G	0	disk	
└─sda1	8:1	0	560M	0	part	/boot/efi
└─sda2	8:2	0	4G	0	part	[SWAP]
└─sda3	8:3	0	120G	0	part	/
└─sda4	8:4	0	829.3G	0	part	/home
sdb	8:16	1	29.9G	0	disk	
└─sdb1	8:17	1	560M	0	part	
└─sdb2	8:18	1	1.5G	0	part	
└─sdb3	8:19	1	18G	0	part	
└─sdb4	8:20	1	9.9G	0	part	

## Formatting the USB stick

---

EFI partition must have FAT32 filesystem. Make sure to specify '-F 32' parameter to 'mkfs.fat' or it may automatically choose FAT16 filesystem. Some UEFI systems are unable to boot from EFI partition formatted as FAT16. Use '-n' to set the volume name (label) to be able to easily identify it later. For FAT32 partition labels must be no longer than 11 characters and contain only uppercase letters.

```
mkfs.fat -F 32 -n 'USBB00T' /dev/sdb1
```

Format /dev/sdb2 as swap. '-L' parameter sets the label of the partition.

```
mkswap -L 'USBSWAP' /dev/sdb2
```

Format /dev/sdb3 as ext4 filesystem. '-L' parameter sets the label of the partition.

```
mkfs.ext4 -L 'USBR00T' /dev/sdb3
```

Format /dev/sdb4 as FAT32 filesystem. '-n' parameter sets the label of the partition. For FAT32 partition labels must be no longer than 11 characters and contain only uppercase letters.

```
mkfs.fat -F 32 -n 'USBSTORE' /dev/sdb4
```

Run 'blkid' with 'grep' to check newly created and formatted partitions.

```
blkid | grep 'sdb'
```

## Mounting the USB stick's partitions

---

Mount the /dev/sdb3 partition on /mnt using 'mount' command.

```
mount /dev/sdb3 /mnt
```

Create the /mnt/boot/efi directories using 'mkdir -p' command.

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

Mount the /dev/sdb1 partition on /mnt/boot/efi using 'mount' command.

```
mount /dev/sdb1 /mnt/boot/efi
```

Create the /mnt/store directory using 'mkdir' command.

```
mkdir /mnt/store
```

Mount the /dev/sdb4 partition on /mnt/store using 'mount' command.

```
mount /dev/sdb4 /mnt/store
```

## Installation of Debian 12 on the USB stick

---

Temporarily move **swap space** from the device on which you are currently booted to the USB stick to not interfere with the installation of the new system.

Change the **swap space** accordingly to your devices, in my case they are /dev/sda2 and /dev/sdb2.

```
swapon /dev/sdb2
```

```
swapoff /dev/sda2
```

Run 'lsblk' ensure everything is done correctly.

```
lsblk
```

NAME	MAJ:MIN	RM	SIZE	RO	TYPE	MOUNTPOINT
sda	8:0	0	953.9G	0	disk	
├─sda1	8:1	0	560M	0	part	/boot/efi
├─sda2	8:2	0	4G	0	part	
├─sda3	8:3	0	120G	0	part	/
└─sda4	8:4	0	829.3G	0	part	/home
sdb	8:16	1	29.9G	0	disk	
├─sdb1	8:17	1	560M	0	part	/mnt/boot/efi
├─sdb2	8:18	1	1.5G	0	part	[SWAP]
├─sdb3	8:19	1	18G	0	part	/mnt
└─sdb4	8:20	1	9.9G	0	part	/mnt/store

Install Debian 12 system on /mnt using 'debootstrap'. 'bookworm' is the codename of Debian 12. The process will take some time, so be patient.

```
debootstrap bookworm /mnt
```

If your current installation was performed without Internet connection from a 'Live DVD', you will need to add online sources to your `/etc/apt/sources.list` to be able to update.

Example of Debian 12/Bookworm (stable) `/etc/apt/sources.list` containing only free software:

```
deb http://deb.debian.org/debian bookworm main
deb http://deb.debian.org/debian-security/ bookworm-security main
deb http://deb.debian.org/debian bookworm-updates main
```

Example of Debian 12/Bookworm (stable) `/etc/apt/sources.list` containing free software and proprietary firmware:

```
deb http://deb.debian.org/debian bookworm main non-free-firmware
deb http://deb.debian.org/debian-security/ bookworm-security main non-free-firmware
deb http://deb.debian.org/debian bookworm-updates main non-free-firmware
```

Example of Debian 12/Bookworm (stable) `/etc/apt/sources.list` containing free software, proprietary firmware and proprietary software:

```
deb http://deb.debian.org/debian bookworm main non-free-firmware contrib non-free
deb http://deb.debian.org/debian-security/ bookworm-security main non-free-firmware contrib non-free
deb http://deb.debian.org/debian bookworm-updates main non-free-firmware contrib non-free
```

If you are currently running *Debian 12*, copy your `/etc/apt/sources.list` to `/mnt/etc/apt/` directory using 'cp'.

```
cp /etc/apt/sources.list /mnt/etc/apt/
```

If you are currently running *Debian 10*, or *Debian 11*, create `/mnt/etc/apt/sources.list` file with the contents from the appropriate listing above.

If you are planning to use desktop environment based on the GTK toolkit, you will need to install and use 'qt5ct' program (after you install the desktop environment) to be able to change appearance of programs based on the Qt toolkit such as 'VLC media player', 'KchmViewer' and 'ghostwriter'. For the 'qt5ct' program to be able to work, this line must be present in `/mnt/etc/environment` file: 'QT\_QPA\_PLATFORMTHEME=qt5ct'. This line will only affect desktop environments **not** based on Qt toolkit and will have no effect on those based on Qt toolkit such as 'KDE' and 'LXQt'.

Edit `/mnt/etc/environment` file by adding 'QT\_QPA\_PLATFORMTHEME=qt5ct' line to the list of existing variables (if any).

```
nano /mnt/etc/environment
```

Example of `/mnt/etc/environment`:

```
QT_QPA_PLATFORMTHEME=qt5ct
```

Copy `/etc/fstab` to `/mnt/etc/` and edit it changing UUIDs to those listed by your 'blkid' command. Add 'noatime' to increase the lifespan of the USB stick.

```
cp /etc/fstab /mnt/etc/  
blkid | grep 'sdb'  
nano /mnt/etc/fstab
```

If you are using 'vim' text editor, there is a convenient way to import the output of 'blkid' into the /mnt/etc/fstab file to be able to copy UUIDs from that output and then edit /mnt/etc/fstab: open /mnt/etc/fstab, create an empty line at the bottom of the file, move to that line, while in normal mode type ':' to enter command mode, then type:

```
!.blkid | grep 'sdb'
```

Example of /mnt/etc/fstab:

```
UUID=8b2e7df4-9c48-d452-f5ef-6aa664a8a853 / ext4 noatime,errors=remount-ro  
0 1  
UUID=6621-56E3 /boot/efi vfat noatime,umask=0077 0 1  
UUID=edbd6535-6e5c-4c23-7611-88fc64a8a7e3 none swap sw 0 0  
UUID=4959-2E62 /store vfat nosuid,nodev,noexec,noatime,fmask=111,dmask=000  
0 2  
/dev/sr0 /media/cdrom0 udf,iso9660 user,noauto 0 0
```

Set the hostname by editing /mnt/etc/hostname. It must contain one line with one word in it. Default is 'debian'. Then add your hostname to /mnt/etc/hosts.

```
nano /mnt/etc/hostname  
nano /mnt/etc/hosts
```

Example of /mnt/etc/hostname:

```
yourhostname
```

Example of /mnt/etc/hosts:

```
127.0.0.1      localhost  
127.0.1.1      yourhostname  
::1           localhost ip6-localhost ip6-loopback  
ff02::1       ip6-allnodes  
ff02::2       ip6-allrouters
```

Create /mnt/etc/papersize file to set the default printer paper size for your region. It must contain one line with either 'letter', or 'a4' word in it. Default is 'letter'.

```
nano /mnt/etc/papersize
```

Example of /mnt/etc/papersize:

```
a4
```

Copy /etc/resolv.conf to /mnt/etc/ to have Internet connection while being in **chrooted environment**.

```
cp /etc/resolv.conf /mnt/etc/
```

Edit /mnt/etc/security/limits.conf to disable core dumps. Unless you are software developer and/or willing to debug the core files, there is no reason for the system to generate those files when programs crash.

```
nano /mnt/etc/security/limits.conf
```

Scroll down below the line '#<domain> <type> <item> <value>' and add the following:

```
* soft core 0
* hard core 0
root soft core 0
root hard core 0
```

Mount virtual filesystems on /mnt.

```
mount -v --bind /dev /mnt/dev
mount -vt devpts /dev/pts /mnt/dev/pts
mount -vt proc /proc /mnt/proc
mount -vt sysfs /sys /mnt/sys
mount -vt tmpfs /run /mnt/run
```

## Changing root into USB stick

---

Use 'chroot' command to change root into /mnt, which is the / filesystem of USB stick.

```
chroot /mnt
```

Check your \$PATH variable.

```
echo "$PATH" | tr ':' '\n'
```

If it lacks '/sbin' and '/usr/sbin', add them (bash syntax).

```
declare -x PATH="$PATH:/sbin:/usr/sbin"
```

Set password for user 'root' using 'passwd' command.

```
passwd
```

Update and upgrade the system using 'apt'.

```
apt update
```

```
apt upgrade -y
```

Install locales.

```
apt install locales
```

Select your locale settings by running 'dpkg-reconfigure locales' command.

```
dpkg-reconfigure locales
```

Select your time zone settings by running 'dpkg-reconfigure tzdata' command.

```
dpkg-reconfigure tzdata
```

Install Linux kernel, sudo, network-manager.

```
apt install linux-image-amd64 sudo network-manager
```

You can obtain the UUID of your swap partition by using 'grep' and 'cut' commands:

```
grep 'sw[[:blank:]]' /etc/fstab | cut -c '6-41'
```

If it looks correct, create /etc/initramfs-tools/conf.d/resume file with sting

'RESUME=UUID=' and UUID of your swap partition by redirecting the output of 'printf' command into /etc/initramfs-tools/conf.d/resume file:

```
> /etc/initramfs-tools/conf.d/resume printf 'RESUME=%s\n' "$(\grep
'sw[[:blank:]]' /etc/fstab | cut -c '-41')"
```

Example of /etc/initramfs-tools/conf.d/resume:

```
RESUME=UUID=edbd6535-6e5c-4c23-7611-88fc64a8a7e3
```

Install 'grub2' bootloader.

```
apt install grub-efi-amd64
```

Open /etc/default/grub file and remove 'quiet' from 'GRUB\_CMDLINE\_LINUX\_DEFAULT' line to be able to see kernel messages during system boot. Then uncomment 'GRUB\_DISABLE\_OS\_PROBER=false' line by removing '#' symbol from the beginning of the line to allow 'grub2' to detect other operating systems installed on the computer. Read the commentary above that line explaining why it is disabled by default in *Debian 12*.

```
nano /etc/default/grub
```

Example of modified 'GRUB\_CMDLINE\_LINUX\_DEFAULT' line:

```
GRUB_CMDLINE_LINUX_DEFAULT=""
```

Example of modified 'GRUB\_DISABLE\_OS\_PROBER=false' line:

```
GRUB_DISABLE_OS_PROBER=false
```

Update bootloader's configuration.

```
grub-install --target=x86_64-efi --force-extra-removable /dev/sdb  
update-initramfs -u  
update-grub
```

The '--force-extra-removable' parameter is required to make removable USB drive bootable. It will create /boot/efi/EFI/BOOT/ directory with files bootx64.efi, fbx64.efi, grub.cfg, grubx64.efi in it. The UEFI firmware will look for /boot/efi/EFI/BOOT/bootx64.efi file on the removable media during the boot process of the computer.

You will need to reinstall 'grub2' with '--force-extra-removable' parameter after every kernel upgrade to update the /boot/efi/EFI/BOOT/ directory.

```
grub-install --target=x86_64-efi --force-extra-removable /dev/sdb  
update-initramfs -u  
update-grub
```

On some cheap motherboards it is possible that UEFI firmware will only boot from /boot/efi/EFI/MICROSOFT/BOOT/bootmgfw.efi, which is a location of Windows boot manager. In that case you need to copy contents of /boot/efi/EFI/BOOT/ to /boot/efi/EFI/MICROSOFT/BOOT/ and rename bootx64.efi to bootmgfw.efi. The FAT32 filesystem is case-insensitive, in your case it may be /boot/efi/efi/boot/, BOOTX64.EFI, or any combination of the above. If your motherboard boots only from /boot/efi/EFI/MICROSOFT/BOOT/bootmgfw.efi file, you will have to repeat this procedure every time the kernel gets upgraded.

```
mkdir -p /boot/efi/EFI/MICROSOFT/BOOT/  
cd /boot/efi/EFI/MICROSOFT/BOOT/  
cp /boot/efi/EFI/BOOT/* ./  
cp bootx64.efi bootmgfw.efi  
cd /
```



Install the 'X11' graphical server and client.

```
apt install xorg
```

Install login manager and desktop environment.

```
apt install lightdm xfce4 xfce4-goodies xfce4-power-manager gvfs-backends  
gvfs-fuse qt5ct
```

Install other packages:

```
apt install cups-bsd printer-driver-cups-pdf dc bc groff gawk vim-gtk3 htop  
mupdf firefox-esr libreoffice gimp
```

...

If you are using *Debian 12* and got a prompt to enter root's password while installing 'printer-driver-cups-pdf', press 'ENTER' instead of typing root's password. You will have to reinstall some packages later, when you'll boot from the USB stick. Refer to 'Troubleshooting' section at the bottom of this guide on how to reinstall packages.

Add user with administrative privileges using 'useradd' command (user must be a member of the 'sudo' group to be able to use 'sudo' command), set user's password using 'passwd' command.

```
useradd -mG  
cdrom,floppy,sudo,audio,dip,video,plugdev,netdev,lpadmin,scanner -s  
/usr/bin/bash -c 'Name' username  
passwd username
```

Exit **chrooted environment** using 'exit' command.

```
exit
```

## Unmount the /mnt

---

Change **swap space** back to your drive currently running the operating system. In my case it's /dev/sda2.

```
swapon /dev/sda2  
swapoff /dev/sdb2
```

Unmount the virtual filesystems.

```
umount /mnt/run  
umount /mnt/sys  
umount /mnt/proc  
umount /mnt/dev/pts  
umount /mnt/dev
```

Unmount physical filesystems.

```
umount /mnt/boot/efi/  
umount /mnt/store  
umount /mnt
```

Exit from root's shell.

```
exit
```

## Booting from USB stick

---

Read your motherboard's manual to find which key will launch the 'Boot menu' during system boot. Usually it's 'F8' or 'F12'.

Reboot your computer, during boot sequence in the beginning rapidly press the boot key ('F8' in my case) to launch the 'Boot menu'. Using arrow keys on the keyboard, navigate to your USB device menu entry. If you have 'Legacy mode' enabled, select the entry with 'UEFI:' text in it. If everything was done correctly, you'll be booting into *Debian 12* that is located on your USB stick.

## Troubleshooting

---

- If the graphical login manager did not launch, press 'Ctrl+Alt+F1', or 'Ctrl+Alt+F2' keys to launch text console, login as 'root', enable and start 'lightdm' using 'systemctl' command.

```
systemctl enable lightdm
systemctl start lightdm
```

Reboot using 'reboot' command.

```
reboot
```

- If you are using *Debian 12* you may experience some errors while installing packages in **chrooted environment**. In *Debian 12* some activity is forbidden in **chrooted environment** due to security reasons. You will have to reinstall problematic packages using 'sudo apt reinstall' command after you have been booted from the USB stick.

```
sudo apt reinstall linux-image-amd64 pkexec policykit-1 polkitd cups-bsd
printer-driver-cups-pdf
```

- On a freshly installed *Debian 12* system look at '/var/log/boot.log' right after the boot process is complete by using 'sed' and 'date' commands.

```
sudo sed -n "$(date '+%a %b %d %H')/,,$p" /var/log/boot.log | less -r
```

If there is no output, change the '%H' to numeric value of previous hour (double-digit 24-hour format).

Example of the same command with explicitly set hour value:

```
sudo sed -n "$(date '+%a %b %d 03')/,,$p" /var/log/boot.log | less -r
```

If this error is present:

```
[FAILED] Failed to start logrotate.service - Rotate log files.
See 'systemctl status logrotate.service' for details.
```

Enable 'logrotate.service' by using 'systemctl' command.

```
sudo systemctl daemon-reload
sudo systemctl start logrotate
```

Check the status of 'logrotate.service'.

```
systemctl status logrotate.service
```