

OPENCORE: LINUX EFI LABELING ALTERNATIVE GUIDE

Warning: This is an alternative method which recommends any Linux Distro is already installed. Exact EFI partition will be renamed. Use this method with precaution. Manual config.plist editing via [ProperTree](#) is recommended. This guide is not responsible if any issues occur. Storage is case sensitive, use the [official](#) support method if you are not confident through this process.

Reminder: This method doesn't require and depends on [OpenLinuxBoot](#). Please read appropriately.

Method 1: Linux - Using [Gparted](#) or [KDE Partition Manager](#)

Step 1

At first, boot to Linux using possible BIOS key without boot to OpenCore. Use list below as references. If not listed, please [Google](#).

No.	MANUFACTURER	BIOS KEY	No.	MANUFACTURER	BIOS KEY
1.	ASUS	ESC, F2	5.	Intel	F10
2.	Gigabyte	F12	6.	Asrock	F11
3.	MSI	F11	7.	EVGA	F7
4.	IBM	F11	8.	Gateway	F11 or R

Remark: BIOS Key from table above is not really accurate. The key is depending on how the manufacturer designs the board. Please pay attention. This is basic knowledge to understand how your motherboard works. Better to refer to any source, especially your motherboard manufacturer, for better understanding.

Step 2

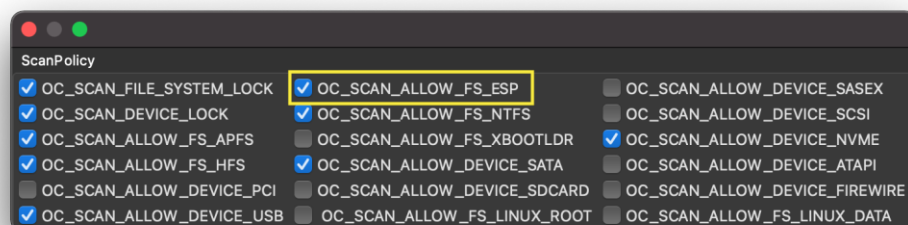
Edit config.plist by using [Xplist](#) or [ProperTree](#) for manual editing.

Step 3

Find Misc > Security > ScanPolicy on config.plist.

Step 4

Change ScanPolicy value to 2690819 or Enable OC_SCAN_ALLOW_FS_ESP. Save config.plist and reboot.

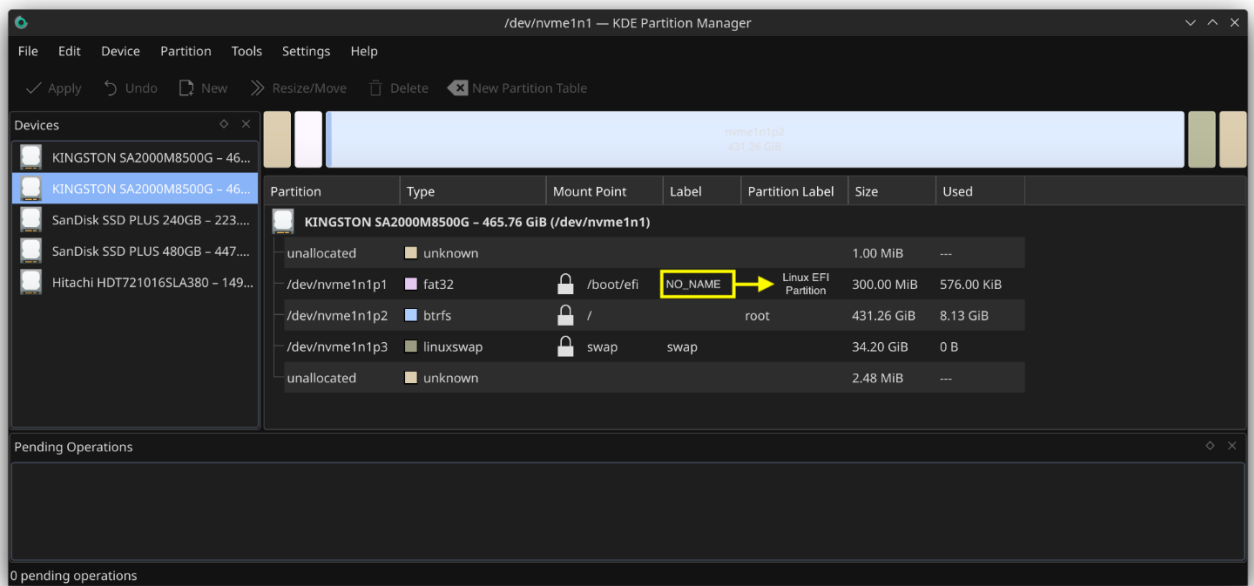


Step 5

Use any Linux built-in partition manager such as **GParted** or **KDE Partition Manager** to find Linux EFI. As an example, `"/dev/nvme1np1"` will be chosen. Rename Linux EFI partition which labelled as NO NAME to any desired name i.e. *Arch*

Step 6

Reboot PC.



Method 2: Linux Terminal - Using **fdisk**

Step 1

Same as Method 1, use `sudo fdisk -l` command on terminal to find EFI partition labeled as NO NAME. As example, `/dev/nvme1n1`.

Output:

```
Disk /dev/nvme1n1: 465.76 GiB, 500107862016 bytes, 976773168 sectors
Disk model: KINGSTON SA2000M8500G
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: gpt
Disk identifier: 945B20C6-EB46-234C-A4DC-A70A2985D995
```

Device	Start	End	Sectors	Size	Type
/dev/nvme1n1p1	4096	618495	614400	300M	EFI System <----- Linux EFI Partition
/dev/nvme1n1p2	618496	905040661	904422166	431.3G	Linux filesystem
/dev/nvme1n1p3	905040662	976768064	71727403	34.2G	Linux swap

```
Disk /dev/nvme0n1: 465.76 GiB, 500107862016 bytes, 976773168 sectors
Disk model: KINGSTON SA2000M8500G
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: gpt
Disk identifier: 4C938ED0-0EBB-41B1-A33A-6645E6CAAD62
```

Device	Start	End	Sectors	Size	Type
/dev/nvme0n1p1	40	409639	409600	200M	EFI System <----- MacOS EFI Partition
/dev/nvme0n1p2	409640	976773127	976363488	465.6G	Apple APFS

```
Disk /dev/sdb: 447.14 GiB, 480113590272 bytes, 937721856 sectors
Disk model: SanDisk SSD PLUS
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: 0x0778dc02
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdb1		2048	937721855	937719808	447.1G	af	HFS / HFS+

```
Disk /dev/sda: 223.57 GiB, 240057409536 bytes, 468862128 sectors
Disk model: SanDisk SSD PLUS
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: 0x0778dc00
```

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sda1		2048	468860927	468858880	223.6G	af	HFS / HFS+

Disk /dev/sdc: 149.01 GiB, 159998918144 bytes, 312497887 sectors
Disk model: Hitachi HDT72101
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: 0x0778dc0c

Device	Boot	Start	End	Sectors	Size	Id	Type
/dev/sdc1	2048	312496127	312494080	149G	af	HFS	HFS+

Step 2

Type **sudo** (superuser do) to get root access. Use **fatlabel** command to label **NO NAME** EFI partition.
I.e.: ***sudo fatlabel /dev/nvme1np1 Arch***

Step 3

Press Enter/Return. Then, reboot and boot back to OpenCore. *Arch* EFI Partition now visible using OpenCore boot menu.

Method 3: Using **DiskGenius** - Windows Only

Step 1

Boot to Windows

Step 2

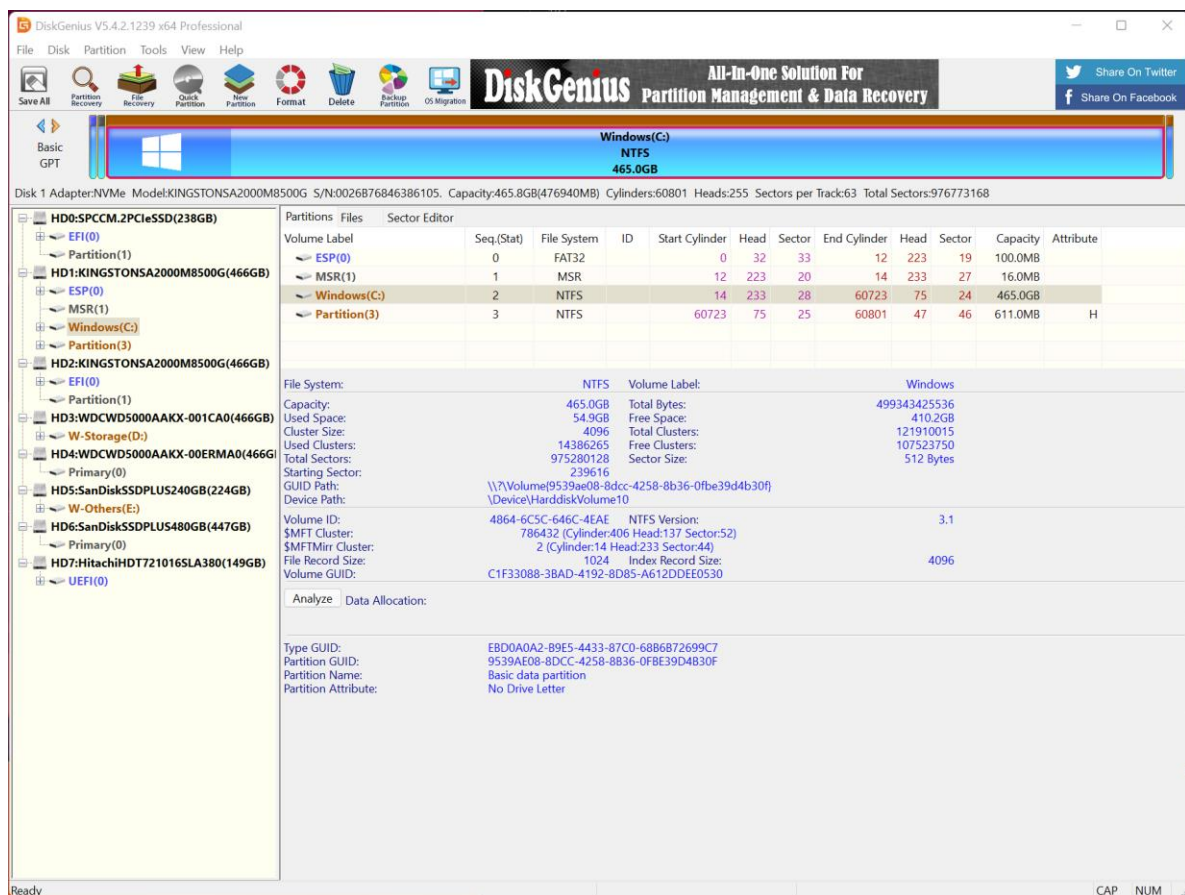
Download **DiskGenius**

Step 3

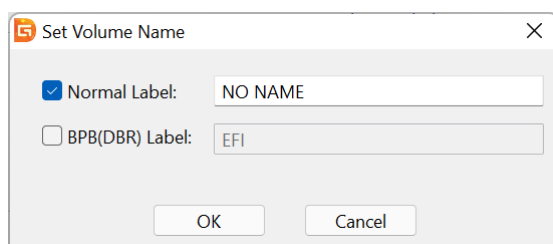
Edit "config.plist" in by using **Xplist** or **ProperTree** for manual editing

Step 4

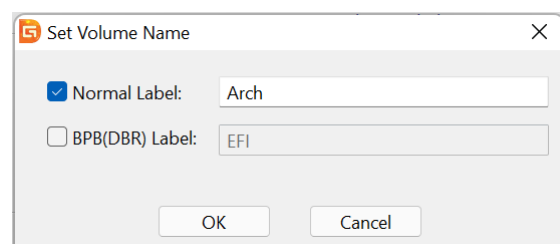
Change **ScanPolicy** value to **2690819** or **Enable OC_SCAN_ALLOW_FS_ESP**. Save config.plist and reboot.



4.1 Before



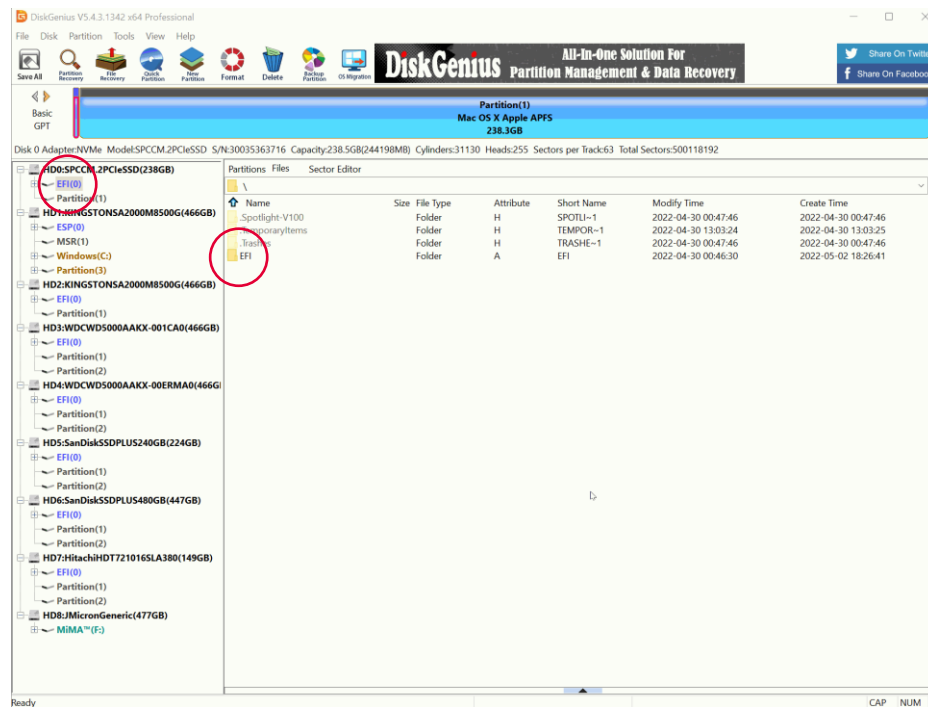
4.2 After



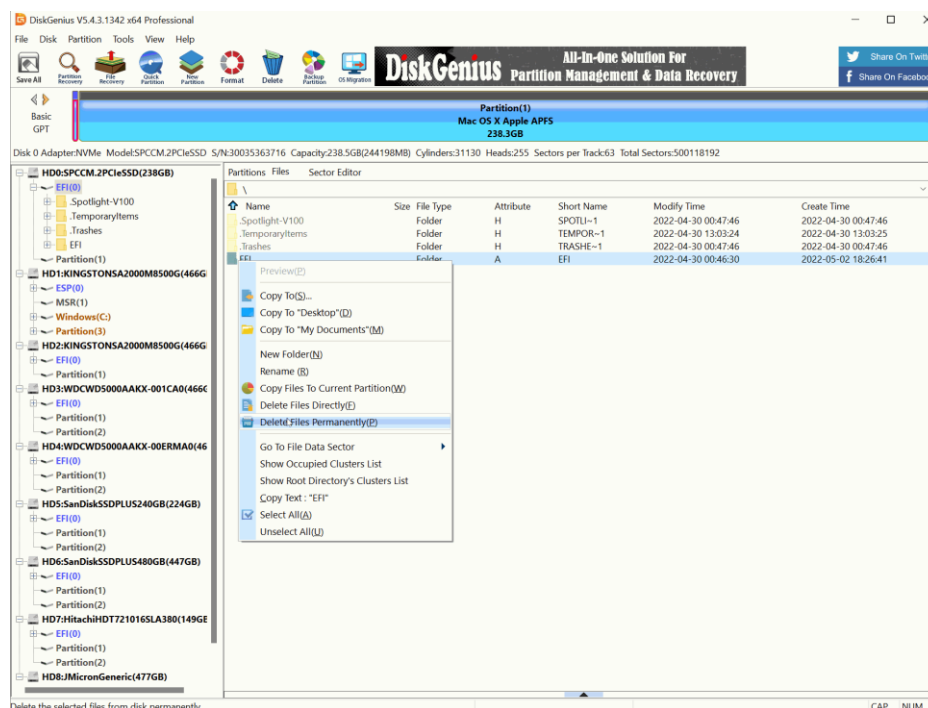
Step 5

Launch DiskGenius, and select any Linux EFI, right click and Select Volume Name. Rename Normal Label. I.e.: NO NAME to any desired name. As an example, Arch.

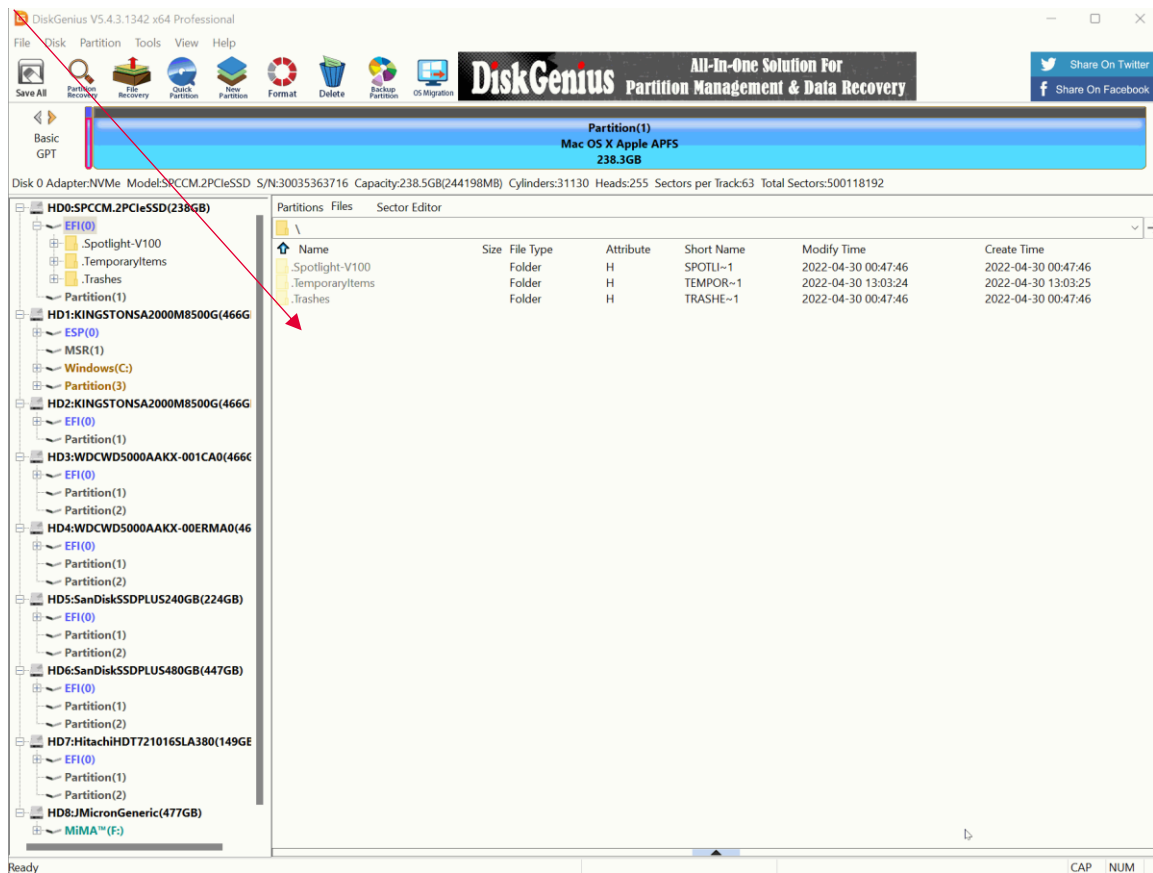
5.1 Copy to Desktop (Right Click on folder to reveal option)



5.2 Delete current EFI from partition and edit the copy on Desktop using [ProperTree](#) or [Xplist](#)



5.3 Move/Drag edited copy from Desktop to EFI partition



Step 6

Close the application, and reboot. Arch EFI partition now visible on OpenCore boot menu.
Arch EFI Partition now visible using OpenCore boot menu.

ACKNOWLEDGEMENT

I would like to thanks all folks in Hackintosh Community especially:

- [Dortania](#) 🥰 a great guide
- [corpNewt](#) 🧑🏻💻 developing [ProperTree](#)
- [Hackintosh Malaysia](#) 😊 an official [Facebook](#) community for Hackintosh
- [r/Hackintosh](#) 🗨️ my favourite [reddit](#) Hackintosh discussion platform
- [ic005k](#) 🧑🏻💻 develop [Xplist](#)

FOLLOW ME

- [reddit](#) ☆ [u/mohdismailmatasin](#)

