

**NAME**

efibootmgr – manipulate the EFI Boot Manager

**SYNOPSIS**

```
efibootmgr [ -a ] [ -A ] [ -b XXXX ] [ -B XXXX ] [ -c ] [ -d DISK ] [ -e I/3/-I ] [ -E NUM ] [ -g ] [ -H XXXX ]
[ -i NAME ] [ -I NAME ] [ -L LABEL ] [ -n XXXX ] [ -N ] [ -o XXXX,YYYY,ZZZZ ... ] [ -O ] [ -p PART ] [
-q ] [ -t seconds ] [ -T ] [ -u ] [ -U XXXX ] [ -v ] [ -V ] [ -w ] [ -@ file ]
```

**DESCRIPTION**

**efibootmgr** is a userspace application used to modify the Intel Extensible Firmware Interface (EFI) Boot Manager. This application can create and destroy boot entries, change the boot order, change the next running boot option, and more.

Details on the EFI Boot Manager are available from the EFI Specification, v1.02 or later, available from:  
<URL:<http://developer.intel.com>>

**Note:** efibootmgr requires that the kernel support access to EFI non-volatile variables (through `/proc/efi/vars` on 2.4 kernels, `/sys/firmware/efi/vars` on 2.6 kernels). **modprobe efivars** should do the trick.

**OPTIONS**

The following is a list of options accepted by efibootmgr:

- a | --active**  
Sets bootnum active
- A | --inactive**  
Sets bootnum inactive
- b | --bootnum XXXX**  
Modify BootXXXX (hex)
- B | --delete-bootnum**  
Delete bootnum (hex)
- c | --create**  
Create new variable bootnum and add to bootorder
- d | --disk DISK**  
The disk containing the loader (defaults to `/dev/sda`)
- e | --edd I/3/-I**  
Force EDD 1.0 or 3.0 creation variables, or guess.
- E | --device NUM**  
EDD 1.0 device number (defaults to 0x80)
- g | --gpt**  
Force disk with invalid PMBR to be treated as GPT
- H | --acpi\_hid XXXX**  
set the ACPI HID (used with **-i**)
- i | --iface NAME**  
create a netboot entry for the named interface
- l | --loader NAME**  
Specify a loader (defaults to `\\elilo.efi`)
- L | --label LABEL**  
Boot manager display label (defaults to "Linux")

- n | --bootnext XXXX**  
Set BootNext to XXXX (hex)
- N | --delete-bootnext**  
Delete BootNext
- o | --bootorder XXXX,YYYY,ZZZZ**  
Explicitly set BootOrder (hex)
- O | --delete-bootorder**  
Delete BootOrder
- p | --part PART**  
Partition number containing the bootloader (defaults to 1)
- q | --quiet**  
Quiet mode - supresses output.
- test filename**  
Don't write to NVRAM, write to *filename*.
- t | --timeout seconds**  
Boot Manager timeout, in *seconds*.
- T | --delete-timeout**  
Delete Timeout variable.
- u | --unicode | --UCS-2**  
pass extra command line arguments as UCS-2 (default is ASCII)
- U | --acpi\_uid XXXX**  
set the ACPI UID (used with **-i**)
- v | --verbose**  
Verbose mode - prints additional information
- V | --version**  
Just print version string and exit.
- w | --write-signature**  
write unique signature to the MBR if needed
- @ | --append-binary-args**  
append extra variable args from file (use - to read from stdin). Data in file is appended as command line arguments to the boot loader command, with no modification to the data, so you can pass any binary or text data necessary.

## EXAMPLES

1.

### DISPLAYING THE CURRENT SETTINGS (MUST BE ROOT).

```
[root@localhost ~]# efibootmgr BootCurrent: 0004 BootNext: 0003 BootOrder: 0004,0000,0001,0002,0003
Timeout: 30 seconds Boot0000* Diskette Drive(device:0) Boot0001* CD-ROM Drive(device:FF)
Boot0002* Hard Drive(Device:80)/HD(Part1,Sig00112233) Boot0003* PXE Boot:
MAC(00D0B7C15D91) Boot0004* Linux
```

This shows:

- BootCurrent - the boot entry used to start the currently running system
- BootOrder - the boot order as would appear in the boot manager. The boot manager tries to boot the first active entry in this list. If unsuccessful, it tries the next entry, and so on.
- BootNext - the boot entry which is scheduled to be run on next boot. This supercedes BootOrder for one boot only, and is deleted by the boot manager after first use. This allows you to change the next boot behavior without changing BootOrder.

- Timeout - the time in seconds between when the boot manager appears on the screen until when it automatically chooses the startup value from BootNext or BootOrder.
- Five boot entries (0000 - 0004), along with the active/inactive flag (\* means active) and the name displayed on the screen.

2.

### CREATING A NEW BOOT OPTION

An OS installer would call **efibootmgr -c**. This assumes that */boot/efi* is your EFI System Partition, and is mounted at */dev/sda1*. This creates a new boot option, called "Linux", and puts it at the top of the boot order list. Options may be passed to modify the default behavior. The default OS Loader is *elilo.efi*.

3.

### CHANGING THE BOOT ORDER

Assuming the configuration in Example #1, **efibootmgr -o 3,4** could be called to specify PXE boot first, then Linux boot.

4.

### CHANGING THE BOOT ORDER FOR THE NEXT BOOT ONLY

Assuming the configuration in Example #1, **efibootmgr -n 4** could be called to specify that the Linux entry be taken on next boot.

5.

### DELETING A BOOT OPTION

Assuming the configuration in Example #1, **efibootmgr -b 4 -B** could be called to delete entry 4 and remove it from the BootOrder.

6.

### CREATING NETWORK BOOT ENTRIES

A system administrator wants to create a boot option to network boot (PXE). Unfortunately, this requires knowing a little more information about your system than can be easily found by efibootmgr, so you've got to pass additional information - the ACPI HID and UID values. These can generally be found by using the EFI Boot Manager (in the EFI environment) to create a network boot entry, then using efibootmgr to print it verbosely. Here's one example: Boot003\* Acpi(PNP0A03,0)/PCI(5|0)/Mac(00D0B7F9F510) \ ACPI(a0341d0,0)PCI(0,5)MAC(00d0b7f9f510,0) In this case, the ACPI HID is "0A0341d0" and the UID is "0". For the zx2000 gigE, the HID is "222F" and the UID is "500". For the rx2000 gigE, the HID is "0002" and the UID is "100". You create the boot entry with: **efibootmgr -c -i eth0 -H 222F -U 500 -L netboot**

### BUGS

Please direct any bugs, features, patches, etc. to Matt Domsch <Matt\_Domsch@ dell.com>.

### AUTHOR

This man page was generated by dann frazier <dannf@debian.org> for the Debian GNU/Linux operating system, but may be used by others.

### SEE ALSO

elilo(1)