Full credit is given to the above companies including the OS that this PDF file was generated!

Linux Ubuntu 22.4.5 Manual Pages on command 'efibootmgr.8'

$ man efibootmgr.8

EFIBOOTMGR(8) EFIBOOTMGR(8)

NAME

efibootmgr - manipulate the UEFI Boot Manager

SYNOPSIS

efibootmgr [-a] [-A] [-b XXXX] [-r | -y] [-B] [-c] [-d DISK] [-D] [-e 1|3|-1] [-E NUM] [-g] [-i NAME] [-l NAME] [-L LABEL] [-m t|f] [-M X] [-n XXXX] [-N] [-o XXXX,YYYY,ZZZZ ...] [-O] [-p PART] [-q] [-t seconds] [-T] [-u] [-v] [-V] [-w] [-@ file]

DESCRIPTION

efibootmgr is a userspace application used to modify the UEFI 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 UEFI Boot Manager are available from the UEFI Specification, v1.02 or later, available from: http://www.uefi.org

Note: efibootmgr requires that the kernel support access to EFI non-volatile variables through /sys/firmware/efi/vars or /sys/firmware/efi/efivars/.

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

-c | --create

Create new variable bootnum and add to bootorder

-d | --disk DISK

The disk containing the loader (defaults to /dev/sda)

-D | --remove-dups

Remove duplicated entries from BootOrder

-e | --edd30 1|3|-1

Force EDD 1.0 or 3.0 creation variables, or guess.

-E | --edd-device NUM

EDD 1.0 device number (defaults to 0x80)

-g | --gpt

Force disk with invalid PMBR to be treated as GPT

-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")

-m | --mirror-below-4G t|f

Set t if you want to mirror memory below 4GB

-M | --mirror-above-4G X

X percentage memory to mirror above 4GB. Floating-point value with up to 2

decimal places is accepted.

-n | --bootnext XXXX

Set BootNext to XXXX (hex)

-N | --delete-bootnext

Delete BootNext

-o | --bootorder XXXX,YYYY,ZZZZ

Explicitly set BootOrder (hex). Any value from 0 to FFFF is accepted so long

as it corresponds to an existing Boot#### variable, and zero padding is not required.

-O | --delete-bootorder

Delete BootOrder

-p | --part PART

Partition number containing the bootloader (defaults to 1)

-q | --quiet

Quiet mode - supresses output.

-r | --driver

Operate on Driver#### variables instead of Boot#### variables.

-t | --timeout seconds

Boot Manager timeout, in seconds.

-T | --delete-timeout

Delete Timeout variable.

-u | --unicode | --UCS-2

Handle extra command line arguments as UCS-2 (default is ASCII)

-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

-y | --sysprep

Operate on SysPrep#### variables instead of Boot#### variables.

-@ | --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 neces? sary.

EXAMPLES

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

Each of the above are boot variables, which are defined as follows:

? 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 unsuccess? ful, 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.

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.

Changing the boot order

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

Changing the boot order for the next boot only

Assuming the configuration in the first example, efibootmgr -n 4 could be called to specify that the Linux entry be taken on next boot.

Deleting a boot option

Assuming the configuration in the first example, efibootmgr -b 4 -B could be called to delete entry 4 and remove it from the BootOrder.

Creating network boot entries

A system administrator wants to create a boot option to network boot. You create the boot entry with: efibootmgr -c -i eth0 -L netboot [-l '\filename.efi']

BUGS

Please direct any bugs, features, patches, etc. to the Red Hat bootloader team at https://github.com/rhboot/efibootmgr .

AUTHOR

This man page was generated by dann frazier <dannf@debian.org> for the Debian GNU/Linux operating system and updated by Robert Bisewski <contact@ibiscybernet?ics.com>, but may be used by others.

SEE ALSO

elilo(1)