GRUB/EFI examples

< GRUB

It is well known that different motherboard manufactures implement UEFI differently. The purpose of this page is to show hardware-specific methods known to work when installing/restoring GRUB in efi mode.

Related articles

Grub

Contents

Apple Macs

Use bless command from within macOS to set `grubx64.efi` as the default boot option. You can also boot from the macOS install disc and launch a Terminal there if you only have Linux installed. In the Terminal, create a directory and mount the EFI System Partition:

```
# cd /Volumes
# mkdir efi
# mount -t msdos /dev/disk0s1 /Volumes/efi
```

Then run bless on `grub.efi` and on the EFI partition to set them as the default boot options.

```
# bless --folder=/Volumes/efi --file=/Volumes/efi/efi/arch_grub/grubx64.efi --setBoot
# bless --mount=/Volumes/efi --file=/Volumes/efi/efi/arch_grub/grubx64.efi --setBoot
```

More info at **https://help.ubuntu.com/community/UEFIBooting#Apple_Mac_EFI_systems_.28both_EFI_architecture.29**.

Note: TODO: GRUB upstream Bazaar mactel branch **http://bzr.savannah.gnu.org/lh/grub/branches/mactel/changes**. No further update from grub developers.

Note: TODO: Experimental "bless" utility for Linux by Fedora developers - `mactel-boot (https://aur.archlinux.org/packages/mactel-boot/)`AUR. Requires more testing.

Asus

Z68 Family and U47 Family

```
# cp /boot/efi/EFI/arch_grub/grubx64.efi /boot/efi/shellx64.efi
```

After this launch the UEFI Shell from the UEFI setup/menu (in ASUS UEFI BIOS, switch to advanced mode, press Exit in the top right corner and choose "Launch EFI shell from filesystem device"). The GRUB2 menu will show up and you can boot into your system. Afterwards you can use efibootmgr to setup a menu entry, for example if you have the uefi partition in /dev/sda1: (read **Unified Extensible Firmware Interface**)

```
efibootmgr -c -g -d /dev/sda -p 1 -w -L "Arch Linux (GRUB)" -l /EFI/arch_grub/grubx64.efi
```

If your motherboard has no such option (or even if it does), you can use UEFI shell (**Unified Extensible Firmware Interface#UEFI Shell**) to create a UEFI boot option for the Arch partition temporarily.

Once you boot into the EFI shell, add a UEFI boot menu entry:

```
Shell> bcfg boot add 0 fs1:\EFI\arch_grub\grubx64.efi "Arch Linux (GRUB2)"
```

where `fs1` is the mapping corresponding to the UEFI System Partition and `\EFI\arch_grub\grubx64.efi` is the the from the `--bootloader-id` from the `grub-install` command above.

Note: The above path may need to be something like "fs1:\EFI\grub\grubx64.efi", Ex: Shell> bcfg boot add 0 fs***n***:\EFI\grub\grubx64.efi

This will temporarily add a UEFI boot option for the next boot to get into Arch. Once in Arch, modprobe `efivars` and confirm that `efibootmgr` creates no errors (no errors meaning you successfully booted in UEFI mode). Then **GRUB#UEFI systems** can be performed again and should successfully permanently add a boot entry in the UEFI menu.

ux32vd

N.B.: The BIOS does not allow computer to boot from GPT disk if there is no properly set-up EFI boot entry. The disk even may not be seen in BIOS in this case. The fix is to make a proper efi boot entry.

There is a caveat. If the machine was booted from MBR then grub-install (or efibootmgr) will fail to create the efi boot entry with the following error:

```
EFI variables are not supported on this system
```

You first need to boot the machine with EFI and then create the boot entry. This can be done the way described for Z68 Family: by copying /boot/efi/EFI/arch_grub/grubx64.efi into /boot/efi/shellx64.efi and selecting "Launch EFI shell from filesystem device". After successful boot it is possible to create a boot entry using grub-install or efibootmgr.

P8Z77 Family

- Boot to live media and chroot into the target system.
- Make sure that a 100 MB fat32 partition is marked as "EFI System" (gdisk terminology uses hex code ef00).

Note: If you get the message WARNING: Not enough clusters for a 32 bit FAT!, reduce cluster size with mkfs.vfat -s2 -F32 ... otherwise the partition may be unreadable by UEFI.

FROM WITHIN THE CHROOT

```
# mount -t vfat /dev/sdXY /boot/efi
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=arch --recheck
# grub-mkconfig -o /boot/grub/grub.cfg
# wget https://edk2.svn.sourceforge.net/svnroot/edk2/trunk/edk2/ShellBinPkg/UefiShell/X64/Shell.efi
# umount /boot/efi
```

The EFI partition should be contain just two files:

```
/Shell.efi
/EFI/arch/grubx64.efi
```

- Reboot and enter the BIOS (the Delete key will do this).
- Using the arrow keys, move to the 'exit' menu and drop down to the EFI shell.
- Add an entry for Arch to the menu. Below is an example, see the **UEFI#Launching UEFI Shell** article for more.

FROM WITHIN THE EFI SHELL

```
Shell> bcfg boot dump -v
Shell> bcfg boot add 1 fs0:\EFI\arch\grubx64.efi "Arch Linux (grub manually added)"
Shell> exit
```

- Reboot the machine and enter the BIOS.
- Navigate to the 'Boot' section and adjust the boot order to with the "Arch Linux (grub manually added)" being the one on the SSD.
- Boot to this entry and enjoy.

Note: This procedure is most likely no longer necessary and you can just create the entry via efibootmgr -d.

M5A97

Finish the standard Arch install procedures, making sure that you install `grub (https://www.archlinux.org/packages/?name=grub)` and partition your boot hard disk as GPT.

From **GRUB#UEFI systems**:

The UEFI system partition will need to be mounted at `/boot/efi/` for the GRUB install script to detect it:

```
# mkdir -p /boot/efi
# mount -t vfat /dev/sdXY /boot/efi
```

Where X is your boot hard disk and Y is the efi partition you created earlier.

Install GRUB UEFI application to `/boot/efi/EFI/arch_grub` and its modules to `/boot/grub/x86_64-efi` using:

```
# modprobe dm-mod
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=arch_grub --recheck --debug
# mkdir -p /boot/grub/locale
# cp /usr/share/locale/en\@quot/LC_MESSAGES/grub.mo /boot/grub/locale/en.mo
```

Generate a configuration for GRUB

```
# grub-mkconfig -o /boot/grub/grub.cfg
```

Then copy the **modified UEFI Shell v2 binary (http://dl.dropbox.com/u/17629062/Shell2.zip)** UefiShellX64.efi into your ESP root.

```
# cp ~/Shell2/UefiShellX64.efi /mnt/boot/efi/shellx64.efi
```

The reason that we need this shell application is that the efibootmgr command will fail silently during grub-install.

After this launch the UEFI Shell from the UEFI setup/menu (in ASUS UEFI BIOS, switch to advanced mode, press Exit in the top right corner and choose "Launch EFI shell from filesystem device"). The UEFI shell will show up. From here we need to add our GRUB UEFI app to the bootloader.

```
Shell> bcfg boot add 3 fs0:\EFI\Arch_Grub\grubx64.efi "Arch_Grub"
```

where `fs0` is the mapping corresponding to the UEFI System Partition and `3` is the zero based boot entry index.

Note: UEFI Shell commands usually support `-b` option which makes output pause after each page. `map` lists recognized filesystems (`fs0` , ...) and data storage devices (`blk0` , ...). Run `help -b` to list available commands. **Unified Extensible Firmware Interface#Important UEFI Shell commands**

To list the current boot entries you can run:

```
Shell> bcfg boot dump -v
```

Asrock

Z97M Pro4

This is a similar procedure to Asus Z68 Family. This was tested on a Z97M Pro4 BIOS P1.90.

```
# cp /boot/efi/EFI/grub/grubx64.efi /boot/efi/shellx64.efi
```

After this launch the UEFI Shell from the UEFI setup/menu (in ASROCK UEFI BIOS, goto Exit tab and choose "Launch EFI Shell From Filesystem Device"). The GRUB2 menu will show up and you can boot into your system. Afterwards you can use efibootmgr to setup a menu entry, for example if you have the uefi partition in /dev/sda1: (read **Unified Extensible Firmware Interface**)

```
# efibootmgr -c -g -d /dev/sda -p 1 -w -L "Arch Linux (GRUB)" -l /EFI/grub/grubx64.efi
```

Dell

PowerEdge T30

The Dell UEFI implementation needs the **UEFI firmware workaround** to load grub. Otherwise it will drop into a "no OS found" screen.

MSI

B250M PRO-VH

This MSI motherboard seems to want the EFI program to exist in a different location from where GRUB installs it. Do the following after following the instructions for installing **GRUB**:

```
# mkdir /boot/EFI/BOOT
# cp /boot/EFI/grub/grubx64.efi /boot/EFI/BOOT/shellx64.efi
```

Note: The procedure above probably also works for other MSI motherboards.

HP

EliteBook 840 G1

See **HP EliteBook 840 G1#UEFI Setup** for details.

> **Note:** The procedures in the link above probably also work for a range of other HP models.

Intel

S5400 Family

This board can run in BIOS or in EFI mode. BIOS mode requires an MBR-partitioned hard drive, EFI a GPT hard drive. Please note that this board operates on the Intel EFI v1.10 specification, and is i386 only. The normal procedure for UEFI installation can be followed, with the exception of the following changes.

- Instead of using the `grub-efi-x86_64` package, `grub-efi-i386` has to be used
- The `bcfg` command is not available for pre-UEFI (v2.0) firmware. A `startup.nsh` file can be used on the root of the EFI partition containing the path to the bootloader. For example:

`fs0:\EFI\arch_grub\boot.efi` has to be placed in the `startup.nsh` file on the root of the EFI partition.

- The `grub.cfg` file has to be placed in the same directory as the grub EFI file, otherwise grub will not find it and enter the interactive shell

Lenovo

K450 IdeaCentre

The "EFI System" partition requires the file `EFI\Boot\bootx64.efi` to be present in order to boot, otherwise you will receive "Error 1962: No operating system found. Boot sequence will automatically repeat." Assuming the "EFI System" partition is mounted on `/boot/efi`:

```
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id=grub --recheck --debug
# mkdir /boot/efi/EFI/Boot
# touch /boot/efi/EFI/Boot/bootx64.efi
```

This is a workaround for what is likely a bug in the UEFI implementation.

M92p ThinkCentre

This system whitelists efi labels. It will only boot from a label called "Red Hat Enterprise Linux". So specify the bootloader id appropriately:

```
# grub-install --target=x86_64-efi --efi-directory=/boot/efi --bootloader-id="Red Hat Enterprise Linux" --recheck --debug
```

VirtualBox

See **VirtualBox#Installation in EFI mode**.

Retrieved from "https://wiki.archlinux.org/index.php?title=GRUB/EFI_examples&oldid=510083"

- This page was last edited on 7 February 2018, at 22:31.
-