NAME

UEFI - Unified Extensible Firmware Interface bootstrapping procedures

DESCRIPTION

The **UEFI** Unified Extensible Firmware Interface provides boot- and run-time services to operating systems. **UEFI** is a replacement for the legacy BIOS on the i386 and amd64 CPU architectures, and is also used on arm, arm64 and riscv architectures.

The UEFI specification is the successor to the Extensible Firmware Interface (EFI) specification. The terms UEFI and EFI are often used interchangeably.

The **UEFI** boot process loads system bootstrap code located in an EFI System Partition (ESP). The ESP is a GPT or MBR partition with a specific identifier that contains an msdosfs(5) FAT file system with a specified file hierarchy.

Partition Scheme	ESP Identifier
GPT	C12A7328-F81F-11D2-BA4B-00A0C93EC93B
MBR	0xEF

The **UEFI** boot process proceeds as follows:

1. **UEFI** firmware runs at power up and searches for an OS loader in the EFI system partition. The path to the loader may be set by an EFI environment variable managed by efibootmgr(8). If not set, an architecture-specific default is used.

Architecture	Default Path
amd64	*/EFI/BOOT/BOOTX64.EFI*
arm	*/EFI/BOOT/BOOTARM.EFI*
arm64	*/EFI/BOOT/BOOTAA64.EFI*
i386	*/EFI/BOOT/BOOTIA32.EFI*
riscv	*/EFI/BOOT/BOOTRISCV64.EFI*

   The default **UEFI** boot configuration for FreeBSD installs *loader.efi* in the default path.
2. *loader.efi* reads boot configuration from */boot.config* or */boot/config*.
3. *loader.efi* loads and boots the kernel, as described in loader.efi(8).

The vt(4) system console is automatically selected when booting via **UEFI**.

FILES

UEFI bootstrap
/boot/loader.efi

Final stage bootstrap

/boot/kernel/kernel

Default kernel

/boot/kernel.old/kernel

Typical non-default kernel (optional)

SEE ALSO

vt(4), boot.config(5), msdosfs(5), boot(8), efibootmgr(8), efidp(8), efivar(8), gpart(8), loader.efi(8), uefisign(8)

HISTORY

EFI boot support for the ia64 architecture first appeared in FreeBSD 5.0. **UEFI** boot support for amd64 first appeared in FreeBSD 10.1; for arm64 in FreeBSD 11.0; for armv6 and armv7 in FreeBSD 12.0; and for riscv in FreeBSD 13.0.

BUGS

There is no support for 32-bit i386 booting via UEFI.