IPXE

The wiki is being retired!

Documentation is now handled by the same processes we use for code: Add something to the Documentation/ directory in the coreboot repo, and it will be rendered to https://doc.coreboot.org/. Contributions welcome!

iPXE is a tool for loading an operating system over a network. It is a fork of GPXE.

Like GPXE, iPXE uses legacy BIOS callbacks, and it works well with SeaBIOS. See GPXE for more information.

Contents

Building and Running in SeaBIOS howto

Here are the goals of this howto:

- fits in 128k for easier building process(if you find an easy way that makes it possible to have bigger iPXE, update that howto Â…)
- boot over an ath9k WiFi over an open/unencrypted wifi to an image that resides on the internet Â…
- Use the following boot procedure: coreboot?>SeaBIOS?iPXE?image on the internet.

Building

Identify your wireless network card:

```
# lspci
[Â…]
03:06.0 Network controller: Atheros Communications Inc. AR922X Wireless Network Adapter (rev 01)
```

Identify its PCI IDs:

```
# lspci -s 03:06.0 -nnn
03:06.0 Network controller [0280]: Atheros Communications Inc. AR922X Wireless Network Adapter [168c:0029] (rev 01)
```

Get iPXE:

```
git clone git://git.ipxe.org/ipxe.git
cd ipxe/src/
```

Copy the following into config/local/general.h:

```
#undef PXE_STACK               /* PXE stack in iPXE - you want this! */
#undef PXE_MENU                /* PXE menu booting */
#undef DOWNLOAD_PROTO_TFTP     /* Trivial File Transfer Protocol */
#undef SANBOOT_PROTO_ISCSI     /* iSCSI protocol */
#undef SANBOOT_PROTO_AOE       /* AoE protocol */
#undef SANBOOT_PROTO_IB_SRP    /* Infiniband SCSI RDMA protocol */
#undef SANBOOT_PROTO_FCP       /* Fibre Channel protocol */
#undef CRYPTO_80211_WEP        /* WEP encryption (deprecated and insecure!) */
#undef CRYPTO_80211_WPA        /* WPA Personal, authenticating with passphrase */
#undef CRYPTO_80211_WPA2       /* Add support for stronger WPA cryptography */
#undef IMAGE_NBI               /* NBI image support */
#undef IMAGE_ELF               /* ELF image support */
#undef IMAGE_MULTIBOOT         /* MultiBoot image support */
#undef IMAGE_PXE               /* PXE image support */
#define        IMAGE_SCRIPT            /* iPXE script image support */
```

```
#define	IMAGE_BZIMAGE		/* Linux bzImage image support */
#undef IMAGE_COMBOOT		/* SYSLINUX COMBOOT image support */
#undef IMAGE_EFI		/* EFI image support */
#undef IMAGE_SDI		/* SDI image support */
#undef NVO_CMD			/* Non-volatile option storage commands */
#define CONFIG_CMD		 /* Option configuration console */
#undef FCMGMT_CMD		/* Fibre Channel management commands */
#undef ROUTE_CMD		/* Routing table management commands */
#define IMAGE_CMD		 /* Image management commands */
#undef SANBOOT_CMD		/* SAN boot commands */
#undef MENU_CMD			/* Menu commands */
#undef LOGIN_CMD		/* Login command */
#undef SYNC_CMD			/* Sync command */
#undef NSLOOKUP_CMD		/* DNS resolving command */
#undef TIME_CMD			/* Time commands */
#undef DIGEST_CMD		/* Image crypto digest commands */
#undef LOTEST_CMD		/* Loopback testing commands */
#undef VLAN_CMD			/* VLAN commands */
#undef PXE_CMD		/* PXE commands */
#undef REBOOT_CMD		/* Reboot command */
#undef IMAGE_TRUST_CMD /* Image trust management commands */
```

Copy the following in the shell.ipxe file:

```
#!ipxe
shell
```

shell.ipxe is an ipxe script that ipxe will run when starting, here it will instruct ipxe to drop directly to a shell. The syntax is documented on ipxe website.

Use the previously gathered PCI ID to include only the ath9k driver:

```
make clean; make -j3 bin/168c0029.rom EMBED=./shell.ipxe
```

Go into your coreboot directory:

```
cd ../../coreboot/
```

Add the iPXE option rom:

```
./build/cbfstool ./build/coreboot.rom print
./build/cbfstool ./build/coreboot.rom add -f ../ipxe/src/bin/168c0029.rom -n pci168c,0029.rom -t raw
./build/cbfstool ./build/coreboot.rom print
```

Booting

- boot on ipxe, it will give you a shell:

```
iPXE>
```

You will then need to type some commands to make it boot over the WiFi from the Internet.

- Identify your ESSID:

```
iwlist
```

- Set the ESSID:

```
config
```

- Get an IP address:

```
dhcp
```

- Test the official demo (requires a PS/2 keyboard)

```
chain http://boot.ipxe.org/demo/boot.php
```

Running from grub as a payload

Introduction

Ipxe depend on the BIOS or EFI interfaces. Grub doesn't provide a BIOS interface. When grub is loaded from a BIOS (like SeaBIOS for instance), the interfaces are provided by the BIOS. but when grub is the payload(coreboot loads grub directly), then this BIOS interfaces are absent.

The way arround this is to make grub load SeaBIOS which then loads ipxe.

SeaBIOS

grub can load payloads.

Refer to the corresponding SeaBIOS section for making grub load SeaBIOS.