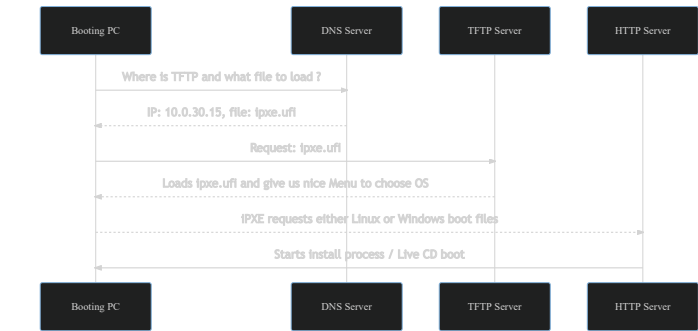


# ipXE

## Intro

ipXE is open-source boot firmware. It's the "software" binary that is loaded by your network card after DHCP tells it where to look for it and what the name of the files is. I know, you want to just get on with it, but it's vital to know, at least in basics, what the steps are in net boot, so you can debug it when something does not work.

Here is a fancy high-level diagram of how the process is going. I have excluded SAMBA service for now, we will have another graphs for Linux and Windows separately.



## Getting ipXE

I dont want to freak you out right now, but we need to compile it from source. "Panic noises...". Do not worry, it will be simple.

### Install packages

Install the following packages on your PXE server.

```
apt install -y make gcc binutils perl mttools mkisofs syslinux liblzma-dev isolinux
```

### Clone the ipXE git

Duplicate the latest ipXE repository

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

### Embedding file into ipXE

**This is an important part**, that pestered me for a day. When the ipXE loads, it kind of expect a specially formatted DNS reply which tells it what "menu" to load. Under "menu" we can imagine a simple file that tells ipXE what OS to offer you, where are the booting files for that OS and so on. Since I'm using UniFi there is no option to easily tell the build in DHCP server to serve that information which ipXE expect by default.

To go around this, and make life simpler for us, we are going to embed the first "menu" file directly into the binary. All it will tell ipXE is to look locally for another file, that will be our actual menu. We can edit it at will later on, without the need of recompiling the ROM again.

Create file: embed.ipxe

```
cd ~/ipxe/src
nano embed.ipxe
```

Add this into it, including the #ipxe at start !

```
#!ipxe
dhcp && goto netboot || goto dhcperror

:dhcperror
prompt --key s --timeout 10000 DHCP failed, hit 's' for the ipXE shell; reboot in 10 seconds && shell || reboot

:netboot
chain tftp://${next-server}/main.ipxe ||
prompt --key s --timeout 10000 Chainloading failed, hit 's' for the ipXE shell; reboot in 10 seconds && shell || reboot
```

What this does, is to tell ipXE to use DHCP to get IP and then go to option netboot to load main.ipxe file from your TFTP server. The variable \${next-server} is automatically replaced by the IP of server ipXE booted for, so no need to hardcore IP in.

Source of embed.ipxe: [Cesetxebernia GitHub](#)

### Enable NTFS and PING support

This is optional, but can help in the future. We will enable NTFS support for ipXE, so it can work with NTFS instead of HTTP if needed.

```
cd ~/ipxe/src

# NTFS support
sed -i 's/#undef\tdOWNLOAD_PROTO_NFS/#define\tdOWNLOAD_PROTO_NFS/' config/general.h

# Ping support
sed -i 's/\V\#define\ PING_CMD/#define\ PING_CMD/' config/general.h
sed -i 's/\V\#define\ IPSTAT_CMD/#define\ IPSTAT_CMD/' config/general.h
sed -i 's/\V\#define\ REBOOT_CMD/#define\ REBOOT_CMD/' config/general.h
sed -i 's/\V\#define\ POWEROFF/#define\ POWEROFF/' config/general.h
```

Source for these changes: [Rikka0w0 GitHub](#)

### Build the ipXE

Time to compile !

```
cd ~/ipxe/src
make bin-x86_64-efi/ipxe.efi EMBED=embed.ipxe
```

This should compile the source code into our ROM file.

### Copy ipxe.efi to place

Our main TFTP folder is /pxe-boot/ remember ? That's the place where the NW card will look for the ipxe.efi. Let's put it there.

```
cp ~/ipxe/src/bin-x86_64-efi/ipxe.efi /pxe-boot/ipxe.efi
```

### Create your first menu

As you remember, we embedded a file to ipXE that tells it to load main.ipxe so let's create one. Just a basic one, so we can test if everything work.

Create file /pxe-boot/main.ipxe and add this:

```
#!/ipxe
:MENU
menu
item --gap -- ----- IPXE boot menu -----
item hello      Hello world
item shell      ipxe shell
choose --default return --timeout 5000 target && goto ${target}

:hello
echo "hello world"
boot ||
goto MENU

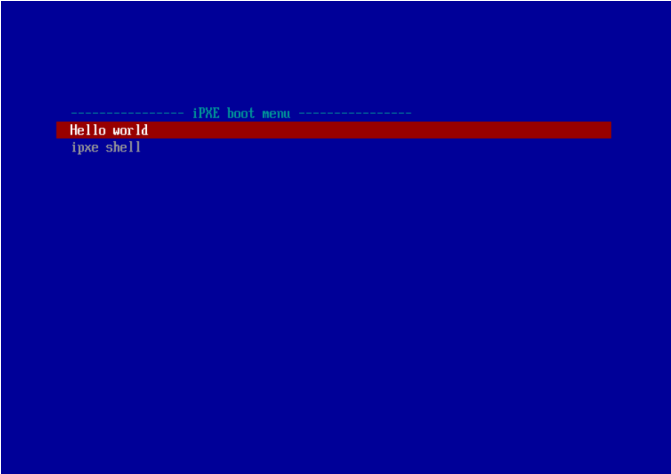
:shell
shell ||
goto MENU

autoboot
```

Don't worry about the content of the file now, we will change it when we will be adding entries for Windows and Linux. I will explain them then.

### Test!

We are at another test stage, you should be now able to do network boot to basic blue menu that look like this:



In your BIOS make sure you set Network Boot, or PXE boot as first booting option and make sure you select UEFI option for it. Do not use secure boot, I think that does not work.

If everything works, let's move on the next step, either Windows or Linux installs.

This was not that hard was it ? Now take a breather, walk for 5 min, get some coffee. Or help me to get some coffee ☺

👉 Liked it ? Buy me a drink ☺

Last update: May 26, 2022

Comments

What do you think?  
1 Response

👍

Upvote

😂

Funny

❤️

Love

😮

Surprised

😡

Angry

😞

Sad

1 Comment

<https://rpi4cluster.com>

Disqus' Privacy Policy

Login

📖 Favorite

🐦 Tweet

📱 Share

Sort by Best

Join the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name