

Chainloading iPXE

If you have a large number of machines which already have a [PXE](#) implementation (such as network cards with an existing Intel [PXE](#) ROM), then you may want to avoid having to reflash each machine's network card individually. You can achieve this by using chainloading.

To enable chainloading, you need to place a copy of iPXE on your TFTP server. Your machines will download this copy of iPXE from the TFTP server each time that they boot.

Setting up PXE chainloading

Download <http://boot.ipxe.org/undionly.kpxe> [<http://boot.ipxe.org/undionly.kpxe>] and save it to your TFTP server directory.

Configure your [DHCP](#) server to hand out `undionly.kpxe` as the boot file. If you are using [ISC dhcpd](#) then you need to edit `/etc/dhcpd.conf` to contain

```
next-server X.X.X.X;
filename "undionly.kpxe";
```

where **X.X.X.X** is the IP address of your TFTP server.

At this point, you should be able to boot one of your [PXE](#)-capable machines, and see it download iPXE from the TFTP server. If everything has worked, then you should see the iPXE startup banner appear:

```
iPXE -- Open Source Boot Firmware -- http://ipxe.org
```



Breaking the infinite loop

When the chainloaded iPXE starts up, it will issue a fresh [DHCP](#) request and boot whatever the [DHCP](#) server hands out. The [DHCP](#) server is currently set up to hand out the iPXE image, which means that you will be stuck in an infinite loop: [PXE](#) will load iPXE which will load iPXE which will load iPXE which will load iPXE...

Breaking the loop with the DHCP server

One way that you can break this infinite loop is to configure the [DHCP](#) server to hand out iPXE only for the first [DHCP](#) request; the second [DHCP](#) request will return the "real" boot filename.

- [Breaking the infinite loop with ISC dhcpd](#)
- [Breaking the infinite loop with the Microsoft DHCP server](#)

Breaking the loop with an embedded script

Another way to break the infinite loop is to build iPXE with an embedded [script](#) that directs iPXE to boot from a fixed [URL](#). For example, if you create the script file `demo.ipxe` containing:

```
#!ipxe
dhcp
chain http://boot.ipxe.org/demo/boot.php
```

and then [build](#) your own version of iPXE with this script embedded:

```
make bin/undionly.kpxe EMBED=demo.ipxe
```

then the chainloaded iPXE will always boot from `http://boot.ipxe.org/demo/boot.php` regardless of the [DHCP](#) server configuration.

UEFI

If you have machines which attempt to perform a UEFI network boot, then download <http://boot.ipxe.org>

/ipxe.efi [<http://boot.ipxe.org/ipxe.efi>] and save it to your TFTP server directory.

You will need to configure your DHCP server to hand out `undionly.kpxe` as the boot file to BIOS machines and `ipxe.efi` as the boot file to UEFI machines. If you are using ISC dhcpd then edit `/etc/dhcpd.conf` to contain

```
option client-arch code 93 = unsigned integer 16;
if option client-arch != 0 {
    filename "ipxe.efi";
} else {
    filename "undionly.kpxe";
}
next-server X.X.X.X;
```

where **X.X.X.X** is the IP address of your TFTP server.

Note that UEFI network booting tends to be substantially slower than BIOS network booting, due to fundamental architectural limitations in UEFI. Most UEFI systems provide the ability to perform a network boot in a BIOS compatibility mode. You may wish to upgrade your system to use BIOS mode for network booting.

[howto/chainloading.txt](#) · Last modified: 2016/01/28 13:27 by mcb30

All uses of this content must include an attribution to the iPXE project and the URL <http://ipxe.org>
References to "iPXE" may not be altered or removed.