FreeBSD Jumpstart Guide

Alfred Perlstein

alfred@FreeBSD.org

$FreeBSD: doc/en_US.ISO8859-1/articles/pxe/article.sgml,v 1.27 2006/08/29 19:45:44 blackend Exp $

This article details the method used to allow machines to install FreeBSD using the Intel® PXE method of booting a machine over a network.

1 Introduction

Warning: This procedure will make the "Server" both insecure and dangerous, it is best to just keep the "Server" on its own hub and not in any way accessible by any machines other than the "Clients".

Terminology:

Server	The machine offering netboot and install options.
Client	The machine that will have FreeBSD installed on it.

Requires: Clients supporting the Intel PXE netboot option, an Ethernet connection.

Please let me know if you come across anything you have problems with or suggestions for additional documentation.

If you would like someone to train/implement a specific netinstall system for you, please send email so that we can discuss terms.

I would also like to thank Paul Saab <ps@FreeBSD.org> and John Baldwin <jhb@FreeBSD.org> for doing most of the programming work on pxeboot, the interface to the Intel PXE (netboot) system.

2 Server Configuration

1. Install DHCP: Install `net/isc-dhcp3-server` you can use this config file `dhcpd.conf` (dhcpd.conf), stick it in `/usr/local/etc/`.
2. Enable tftp:

    1. Make a directory `/usr/tftpboot`
    2. Add this line to your `/etc/inetd.conf`:

    ```
    tftp    dgram   udp     wait    nobody  /usr/libexec/tftpd    tftpd /usr/tftpboot
    ```

3. Enable NFS:

    1. Add this to `/etc/rc.conf`:

    ```
    nfs_server_enable="YES"
    ```

    2. Add this to `/etc/exports`:

    ```
    /usr -alldirs -ro
    ```

4. Reboot to enable the new services or start them manually.

3 Bootstrap Setup

1. Download bootfiles: Download the kern.flp (ftp://snapshots.jp.freebsd.org/pub/FreeBSD/snapshots/i386/4-LATEST/floppies/kern.flp) and mfsroot.flp (ftp://snapshots.jp.freebsd.org/pub/FreeBSD/snapshots/i386/4-LATEST/floppies/mfsroot.flp) floppy images.
2. Set up tftp/pxe-boot directory:

    1. Put pxeboot in the boot directory:

    ```
    # rm -rf /usr/obj/*
    # cd /usr/src/sys/boot
    # make
    # cp /usr/src/sys/boot/i386/pxeldr/pxeboot /usr/tftpboot
    ```

    2. Using the vndevice mount the `kern.flp` file and copy its contents to `/usr/tftpboot`:

    ```
    # mdconfig -a -t vnode -f kern.flp -u 0 # (vnconfig vn0 kern.flp) associate a vndevice with the f
    # mount /dev/md0 /mnt # (mount /dev/vn0 /mnt) mount it
    # cp -R /mnt /usr/tftpboot # copy the contents to /usr/tftpboot
    # umount /mnt               # unmount it
    # vnconfig -u vn0           # disassociate the vndevice from the file
    ```

3. Compile a custom kernel for the clients (particularly to avoid the device config screen at boot) and stick it in `/usr/tftpboot`.
4. Make a special `loader.rc` to and install it in `/usr/tftpboot/boot/loader.rc` so that it does not prompt for the second disk, here is mine (loader.rc).

5. Extract the installer and helper utilities from the mfsroot disk and uncompress them, put them in `/usr/tftpboot` as well:

```
# vnconfig vn0 mfsroot.flp          # associate a vndevice with the file
# mount /dev/vn0 /mnt               # mount it
# cp /mnt/mfsroot.gz /usr/tftpboot # copy the contents to /usr/tftpboot
# umount /mnt                        # unmount it
# vnconfig -u vn0                    # disassociate the vndevice from the file
# cd /usr/tftpboot                   # get into the pxeboot directory
# gunzip mfsroot.gz                  # uncompress the mfsroot
```

6. Make your sysinstall script `install.cfg`, you can use mine (install.cfg) as a template, but you must edit it.

7. Copy the sysinstall script into the extracted and uncompressed mfsroot image:

```
# cd /usr/tftpboot
# vnconfig vn0 mfsroot
# mount /dev/vn0 /mnt
# cp install.cfg /mnt
# umount /mnt
# vnconfig -u vn0
```

4 Install Setup

1. Put the install files in an NFS accessible location on the Server. Make a directory corresponding the 'nfs' directive in the `install.cfg` file and mirror the FreeBSD install files there, you will want it to look somewhat like this:

```
ABOUT.TXT       TROUBLE.TXT     compat20        floppies        ports
ERRATA.TXT      UPGRADE.TXT     compat21        games           proflibs
HARDWARE.TXT    XF86336         compat22        info            src
INSTALL.TXT     bin             compat3x        kern.flp
LAYOUT.TXT      catpages        crypto          manpages
README.TXT      cdrom.inf       dict            mfsroot.flp
RELNOTES.TXT    compat1x        doc             packages
```

2. Copy the compressed packages into the packages/All directory under `nfs`.

3. Make sure you have an `INDEX` file prepared in the packages directory. You can make your own `INDEX` entries like so:

```
alfred-1.0||/|Alfred install bootstrap||alfred@FreeBSD.org||||
```

   Then you can install custom packages, particularly your own custom post-install package.

5 Custom Post-Install Package

You can use the script `pkgmaker.sh` (pkgmaker.sh) to create a custom package for post install, the idea is to have it install and configure any special things you may need done. `pkgmaker` is run in the directory above the package you wish to create with the single argument of the package (i.e., mypkg) which will then create a mypkg.tgz for you to include in your sysinstall package.

Inside your custom package dir you will want a file called `PLIST` which contains all the files that you wish to install and be incorporated into your package.

You will also want files called `pre` and `post` in the directory, these are shell scripts that you want to execute before and after your package is installed.

Since this package is in your `install.cfg` file it should be run and do the final configuration for you.