NAME

INSTALL – Installation procedure for NetBSD/macppc.

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5 on the *macppc* platform. It is available in four different formats titled `INSTALL.`*ext*, where *ext* is one of `.ps`, `.html`, `.more`, or `.txt`:

- `.ps` PostScript.
- `.html` Standard Internet HTML.
- `.more` The enhanced text format used on UNIX-like systems by the `more`(1) and `less`(1) pager utility programs. This is the format in which the on-line *man* pages are generally presented.
- `.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on thirty-one different system architectures featuring twelve distinct families of CPUs, and is being ported to more. The NetBSD 1.5 release contains complete binary releases for fifteen different machine types. (The sixteen remaining are not fully supported at this time and are thus not part of the binary distribution. For information on them, please see the NetBSD web site at **http://www.netbsd.org/**)

NetBSD is a completely integrated system. In addition to its highly portable, high performance kernel, NetBSD features a complete set of user utilities, compilers for several languages, the X Window System, firewall software and numerous other tools, all accompanied by full source code.

NetBSD is a creation of the members of the Internet community. Without the unique cooperation and coordination the net makes possible, it's likely that NetBSD wouldn't exist.

Changes Since The Last Release

The NetBSD 1.5 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many userland enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize over one year of development that went into the NetBSD 1.5 release. Some highlights include:

Kernel

- Ports to new platforms including: arc, cobalt, hpcmips, news68k, sgimips, and sparc64.
- Improved performance and stability of the UVM virtual memory subsystem.
- Implementation of generic kernel locking code, as well as a restructure and re-tuning of the scheduler, to be used by the future symmetric multi-processing (SMP) implementation.
- Improved compatibility support for Linux, OSF1, and SVR4 programs.

- New compatibility support for Win32 programs.
- Support for dynamically loaded ELF kernel modules.
- Kernel process tracing using `ktruss`(1).
- Deletion of swap devices using `swapctl`(8).
- Easier hot-pluggability of keyboards and mice using a new wscons device - wsmux.
- Improved PCMCIA and Cardbus support, including support for detaching of devices and cards, resulting in better support for notebooks and PDA devices.
- Numerous hardware improvements, including areas such as: audio, UDMA/66 support for ATA drives, USB, and wireless networking.

Networking

- Addition of IP version 6 (IPv6) and IPsec to the networking stack, from the KAME project. This includes addition of kernel code for IPv6/IPsec, IPv4/v6 dual-stack user applications and supporting libraries. Due to this, the shlib major version for `pcap`(3) is incremented and you may need to recompile userland tools. The KAME IPv6 part includes results from the unified-ipv6 effort.

File system

- Significant Fast file system (FFS) performance enhancements via integration of Kirk McKusick's soft updates and trickle sync code.
- Support for the Windows NT 'NTFS' file system (read-only at this stage).
- Support for revision 1 of the Linux 'ext2fs' file system.
- Enhanced stability and usability of LFS (the BSD log-structured file system).
- Various RAIDframe enhancements including: auto-detection of RAID components and auto-configuration of RAID sets, and the ability to configure the root file system (`/`) on a RAID set.
- Support for Microsoft Joliet extensions to the ISO9660 CD file system.
- Improved file system vnode locking mechanisms, thus resolving a source of several panics in the past.
- Support for NFS and RPC over IPv6.
- Server part of NFS locking (implemented by `rpc.lockd`(8)) now works.

Security

- Strong cryptographic libraries and applications integrated, including the AES cipher Rijndael, the OpenSSL library, more complete Kerberos IV and Kerberos V support, and an SSH server and client.
- `sysctl`(3) interfaces to various elements of process and system information, allowing programs such as `ps`(1), `dmesg`(1) and the like to operate without recompilation after kernel upgrades, and remove the necessity to run setgid kmem (thus improving system security).
- Disable various services by default, and set the default options for disabled daemons to a higher level of logging.
- Several code audits were performed. One audit replaced string routines that were used without bound checking, and another one to identify and disable places where format strings were used in an unsafe way, allowing arbitrary data entered by (possibly) malicious users to overwrite application code, and leading from Denial of Service attacks to compromised system.

System administration and user tools

- Conversion of the `rc`(8) system startup and shutdown scripts to an 'rc.d' mechanism, with separate control scripts for each service, and appropriate dependency ordering provided by `rcorder`(8).
- `postfix`(1) provided as alternative mail transport agent to `sendmail`(8).
- User management tools `useradd`(8), `usermod`(8), `userdel`(8), `groupadd`(8), `groupmod`(8), and `groupdel`(8) added to the system.
- Incorporation of a login class capability database (`/etc/login.conf`) from BSD/OS.
- Improved support for usernames longer than eight characters in programs such as `at`(1) and `w`(1).
- Many enhancements to `ftpd`(8) providing features found in larger and less secure FTP daemons, such as user classes, connection limits, improved support for virtual hosting, transfer statistics, transfer rate throttling, and support for various IETF ftpext working group extensions.
- The `ftp`(1) client has been improved even further, including transfer rate throttling, improved URL support, command line uploads. See the man page for details.

Miscellaneous

- Updates to the NetBSD source code style code (located in `/usr/share/misc/style`) to use ANSI C only (instead of K&R) and reflect current (best) practice, and begin migrating the NetBSD source code to follow it.
- Implementation of many SUSv2 features to the `curses`(3) library, including support for color.
- Updates of most third party packages that are shipped in the base system, including `file`(1), `ipfilter`(4), `ppp`(4), and `sendmail`(8) to the latest stable release.
- Many new packages in the *pkgsrc* system, including standard desktops like KDE and GNOME as well as latest Tcl/Tk and perl and many of the components of the Java Enterprise platform. The package framework itself now has full wildcard dependency support.

As has been noted, there have also been innumerable bug fixes.

Kernel interfaces have continued to be refined, and more subsystems and device drivers are shared among the different ports. You can look for this trend to continue.

NetBSD 1.5 is the first official release of NetBSD for the Apple Power Macintosh and their compatibles.

The Future of NetBSD

The NetBSD Foundation has been incorporated as a non-profit organization. Its purpose is to encourage, foster and promote the free exchange of computer software, namely the NetBSD Operating System. The foundation will allow for many things to be handled more smoothly than could be done with our previous informal organization. In particular, it provides the framework to deal with other parties that wish to become involved in the NetBSD Project.

The NetBSD Foundation will help improve the quality of NetBSD by:

- providing better organization to keep track of development efforts, including co-ordination with groups working in related fields.
- providing a framework to receive donations of goods and services and to own the resources necessary to run the NetBSD Project.

- providing a better position from which to undertake promotional activities.
- periodically organizing workshops for developers and other interested people to discuss ongoing work.

We intend to begin narrowing the time delay between releases. Our ambition is to provide a full release every six to eight months.

We hope to support even *more* hardware in the future, and we have a rather large number of other ideas about what can be done to improve NetBSD.

We intend to continue our current practice of making the NetBSD-current development source available on a daily basis.

We intend to integrate free, positive changes from whatever sources submit them, providing that they are well thought-out and increase the usability of the system.

Above all, we hope to create a stable and accessible system, and to be responsive to the needs and desires of NetBSD users, because it is for and because of them that NetBSD exists.

Sources of NetBSD

Refer to **http://www.netbsd.org/Sites/net.html**.

NetBSD 1.5 Release Contents

The root directory of the NetBSD 1.5 release is organized as follows:

`.../NetBSD-1.5/`

`CHANGES` Changes since earlier NetBSD releases.

`LAST_MINUTE` Last minute changes.

`MIRRORS` A list of sites that mirror the NetBSD 1.5 distribution.

`README.files`
README describing the distribution's contents.

`TODO` NetBSD 's todo list (also somewhat incomplete and out of date).

`patches/` Post-release source code patches.

`source/` Source distribution sets; see below.

In addition to the files and directories listed above, there is one directory per architecture, for each of the architectures for which NetBSD 1.5 has a binary distribution. There are also `README.export-control` files sprinkled liberally throughout the distribution tree, which point out that there are some portions of the distribution that may be subject to export regulations of the United States, e.g. code under `src/crypto` and `src/sys/crypto`. It is your responsibility to determine whether or not it is legal for you to export these portions and to act accordingly.

The source distribution sets can be found in subdirectories of the `source` subdirectory of the distribution tree. They contain the complete sources to the system. The source distribution sets are as follows:

gnusrc This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.
22.3 MB gzipped, 98.8 MB uncompressed

pkgsrc This set contains the "pkgsrc" sources, which contain the infrastructure to build third-party packages.
5.6 MB gzipped, 57.0 MB uncompressed

sharesrc This set contains the "share" sources, which include the sources for the man pages not associated with any particular program, the sources for the typesettable document set, the dictionaries, and more.
3.3 MB gzipped, 13.2 MB uncompressed

src This set contains all of the base NetBSD 1.5 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
24.2 MB gzipped, 120.6 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.5 kernel for all architectures, `config`(8), and `dbsym`(8).
17.6 MB gzipped, 88.6 MB uncompressed

xsrc This set contains the sources to the X Window System.
35.2 MB gzipped, 176.8 MB uncompressed

All the above source sets are located in the `source/sets` subdirectory of the distribution tree.

The source sets are distributed as compressed tar files. They may be unpacked into `/usr/src` with the command:

```
# ( cd / ; tar -zxpf - ) < set_name.tgz
```

The `sets/Split/` subdirectory contains split versions of the source sets for those users who need to load the source sets from floppy or otherwise need a split distribution. The split sets are named `set_name.`*`xx`* where `set_name` is the distribution set name, and *`xx`* is the sequence number of the file, starting with "aa" for the first file in the distribution set, then "ab" for the next, and so on. All of these files except the last one of each set should be exactly 240,640 bytes long. (The last file is just long enough to contain the remainder of the data for that distribution set.)

The split distributions may be reassembled and extracted with **`cat`** as follows:

```
# cat set_name.?? | ( cd / ; tar -zxpf - )
```

In each of the source distribution set directories, there are files which contain the checksums of the files in the directory:

`BSDSUM` Historic BSD checksums for the various files in that directory, in the format produced by the command: **`cksum -o 1`** *`file`*

`CKSUM` POSIX checksums for the various files in that directory, in the format produced by the command: **`cksum`** *`file`*.

`MD5` MD5 digests for the various files in that directory, in the format produced by the command: **`cksum -m`** *`file`*.

`SYSVSUM` Historic AT&T System V UNIX checksums for the various files in that directory, in the format produced by the command: **`cksum -o -2`** *`file`*.

The MD5 digest is the safest checksum, followed by the POSIX checksum. The other two checksums are provided only to ensure that the widest possible range of system can check the integrity of the release files.

NetBSD/macppc subdirectory structure

The macppc-specific portion of the NetBSD 1.5 release is found in the `macppc` subdirectory of the distribution: `.../NetBSD-1.5/macppc/`
`INSTALL.html`

`INSTALL.ps`
`INSTALL.txt`
`INSTALL.more` Installation notes in various file formats, including this file. The `.more` file contains underlined text using the `more`(1) conventions for indicating italic and bold display.

`binary/`
- `kernel/`
  - `netbsd.GENERIC.gz` A gzipped NetBSD kernel containing code for everything supported in this release.
- `sets/` macppc binary distribution sets; see below.

`installation/`
- `floppy/` macppc boot and installation floppies; see below.
- `misc/` Miscellaneous macppc installation utilities; see installation section, below.

Binary distribution sets

The NetBSD macppc binary distribution sets contain the binaries which comprise the NetBSD 1.5 release for the macppc. There are eight binary distribution sets. The binary distribution sets can be found in the `macppc/binary/sets` subdirectory of the NetBSD 1.5 distribution tree, and are as follows:

base The NetBSD 1.5 macppc **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional. It includes shared library support, and excludes everything described below.
17.1 MB gzipped, 46.2 MB uncompressed

comp Things needed for compiling programs. This set includes the system include files (`/usr/include`) and the various system libraries (except the shared libraries, which are included as part of the **base** set). This set also includes the manual pages for all of the utilities it contains, as well as the system call and library manual pages.
14.4 MB gzipped, 55.4 MB uncompressed

etc This distribution set contains the system configuration files that reside in `/etc` and in several other places. This set *must* be installed if you are installing the system from scratch, but should *not* be used if you are upgrading. (If you are upgrading, it's recommended that you get a copy of this set and *carefully* upgrade your configuration files by hand.)
0.1 MB gzipped, 0.6 MB uncompressed

games This set includes the games and their manual pages.
3.0 MB gzipped, 7.2 MB uncompressed

kern This set contains a NetBSD/macppc 1.5 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
1.3 MB gzipped, 3.1 MB uncompressed

man This set includes all of the manual pages for the binaries and other software contained in the **base** set. Note that it does not include any of the manual pages that are included in the other sets.
5.0 MB gzipped, 19.9 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
2.6 MB gzipped, 10.1 MB uncompressed

text This set includes NetBSD 's text processing tools, including `groff`(1), all related programs, and their manual pages.
1.2 MB gzipped, 4.6 MB uncompressed

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on XFree86, and tightly track XFree86 releases. They are currently equivalent to XFree86 3.3.6. Binary sets for the X Window system are distributed with NetBSD. The sets are:

xbase The basic files needed for a complete X client environment. This does not include the X servers.
2.9 MB gzipped, 8.9 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.

xcontrib Programs that were contributed to X.
0.2 MB gzipped, 0.6 MB uncompressed

xfont Fonts needed by X.
6.2 MB gzipped, 7.5 MB uncompressed

xserver The Xmacppc server which supports 256 colors with man pages.
1.7 MB gzipped, 4.6 MB uncompressed

The macppc binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

The instructions given for extracting the source sets work equally well for the binary sets, but it is worth noting that if you use that method, the files are `/`-relative and therefore are extracted *below* the current directory. That is, if you want to extract the binaries into your system, i.e. replace the system binaries with them, you have to run the **`tar -xpf`** command from `/`.

Note: Each directory in the macppc binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/macppc System Requirements and Supported Devices

Currently, NetBSD/macppc requires the use of Open Firmware to boot. Open Firmware is a command environment using the FORTH language which the NetBSD kernel uses to gether information about your system, and to control some of your devices. This means that PowerMacs and clones that lack Open Firmware cannot boot NetBSD on the macppc platform. Most machines introduced by Apple and the clone-makers after August 17, 1995 have Open Firmware and are supported.

Apple made several revisions of this Open Firmware environment, and each has various quirks and problems that we must work around. The single hardest step of installing NetBSD/macppc is to set up Open Firmware properly. Open Firmware versions 1.x and 2.0.x act similarly and the same set of instructions applies to them. Open Firmware version 2.4 is slightly different with regards to booting. Open Firmware version 3 is altogether different.

At present, NetBSD/macppc does not support the PPC 601 microprocessor, which means that the PowerMacintosh 7200 and 7500 models are not supported. The PowerMacintosh 7500 may be upgraded to a PPC 604, G3, or G4 microprocessor via a daughtercard replacement, in which case NetBSD will run on this system.

The minimal configuration requires 8 MB of RAM and ˜80 MB of disk space. To install the entire system requires much more disk space, and to run X or compile the system, more RAM is recommended, as NetBSD with 8 MB of RAM is very slow. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.

Supported models

Find your model from the list below and take note of its Open Firmware version:

Open Firmware 1.0.5 or 2.0.x

- Apple PowerBook (2400, 3400, G3, and G3 Series)
- Apple PowerMacintosh/Performa (4400, 54xx, 5500, 6300/160, 6360, 6400, and 6500)
- Apple PowerMacintosh (7300, 7500 (with CPU upgrade), 7600, 8500, 8600, 9500, and 9600)
- Apple PowerMacintosh (G3 "Beige" models with ATI RageII+: Desktop, Mini Tower, and All-in-One)
- APS Tech (M*Power 604e/200)
- Motorola StarMax (3000, 4000, 5000, and 5500)
- Power Computing (PowerBase, PowerCenter, PowerCenter Pro, PowerCurve, PowerTower, PowerTower Pro, and PowerWave)
- UMAX (J700, S900)
- UMAX (Apus 2000, Apus 3000, C500, and C600)

Open Firmware 1.1.22

- Apple Network Server (500 and 700)

  Note: Users have reported that the NetBSD kernel does not work well with the on-board MACE-based ethernet (the one that requires an Apple AAUI dongle). If your system does not have the Apple PCI Ethernet Card, you may need to purchase and install another ethernet card in your Apple Network Server.

Open Firmware 2.4

- Apple PowerMacintosh (G3 "Beige" models with ATI Rage Pro: Desktop, Mini Tower, and All-in-One)

Open Firmware 3

- Apple iMac; Bondi Blue (Rev A and Rev B, 233 MHz), 5 Flavors (Rev C and Rev D, 266 MHz and 333 MHz), and Slot Loading
- Apple PowerBook (G3 Series (bronze keyboard))
- Apple PowerMacintosh (G3 (Blue and White), G4 (PCI), G4 (AGP), and G4 Cube)

Unsupported models

- Systems with Open Firmware, but using a PPC 601 microprocessor (PowerMacintosh 7200 and 7500 (without CPU upgrade))
- Systems with a PowerPC microprocessor, but lacking Open Firmware
  - Apple PowerBook (1400, 2300, and 5300)
  - Apple PowerMacintosh/Performa (52xx, 53xx, 62xx, and 63xx (except 6300/160 and 6360 which are supported))
  - Apple PowerMacintosh/Performa (61xx, 71xx, and 81xx)
  - PowerComputing (Power 100 and Power 120)
- Systems with broken Open Firmware (Twentieth Anniversary Macintosh)
- Systems with a Motorola 68k microprocessor (these systems are supported by NetBSD/mac68k)
- Systems upgraded from any of the above (unless the motherboard is replaced as part of the upgrade)
- Systems released during the year 2000 (future releases of NetBSD/macppc will support these systems)

Supported devices

- Ethernet
  - On-board 79C950-based MACE Ethernet interface (*mc0*)
  - On-board bmac Ethernet interface (*bm0*)
  - On-board gmac Ethernet interface (*gm0*)
  - Apple PCI Ethernet Card (option for Apple Network Server) (*tlp*)
  - Asante Mac 10/100 PCI Rev A, part number 09-00169-01 (*de*)
  - Farallon Fast EtherTX 10/100, part number PN996L-TX (*de*)
  - SMC Etherpower II (9432TX) (*epic*)
  - SMC 83c170 (*epic*)
  - 3Com 3c905 (*ex*)
  - Intel EtherExpress PRO/10+ PCI LAN Adapter (*fxp*)
  - RealTek 8029 Ethernet (*ne*)
  - VIA Technologies VT86C926 (*ne*)
  - RealTek 8139 (*rtk*)
  - Lite-On PNIC (*tlp*)
  - D-Link DFE-530TX (*vr*)
  - Many other PCI and Cardbus Ethernet interfaces, such as Tulip-compatable (*de* and *tlp*), 3Com (*ep*), SMC (*epic*), Intel (*fxp*), NE2000-compatable (*ne*), and RealTek (*rtk*)
  - Many USB Ethernet interfaces (*aue*, *cue*, and *kue*)

- SCSI
  - On-board NCR 53c94 SCSI controller (*esp*)
  - On-board MESH SCSI controller (*mesh*)
  - Adaptec PCI controllers 291x, 2920, 2930C, 294x, 295x, 39xx, 19160, 29160 and AIC-78xx (*ahc*)
  - AdvanSys 1200[A,B], 9xx[U,UA] SCSI controller (*adv*)
  - AMD 53c974 (*pcscp*)
  - NCR/Symbios 53C8xx (*ncr* or *siop*)
  - Many other PCI SCSI controllers should work, but no one has tried them
  - Most SCSI disk/tape/CD-ROM devices should work

- IDE
  - On-board IDE controlers
  - Many PCI IDE controllers should work, although no one has tried them
  - Most IDE disk/CD-ROM/ATAPI devices should work

- Input devices
  - Most ADB keyboards, mice, trackballs, and trackpads
  - Most USB keyboards, mice, trackballs, and trackpads
  - Most PS/2 keyboards, mice, and trackballs (middle button on 3-button mice may not work)

- Video
  - On-board video on most models (*ofb*)

    Note: Several models have been reported to not work with NetBSD if the on-board video is in use, such as the Performa 54xx, 6360, 6400, PowerMacintosh 9500/150, 9500/180, 9500/200, PowerComputing PowerTower, and UMAX C600 and Apus 3000

  - PCI frame buffers which have Open Firmware support (ATI, IMS, and Matrox have several models which work)

- Serial ports
  - On-board serial ports (the modem and printer ports) (*ttya* and *ttyb*)

    Note: The on-board serial ports can be used for console (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking), although many users have reported problems trying to run ppp or other high speed serial applications

  - The Gee Three Stealth Serial Port, possibly the Griffin Technology gPort

    Note: These serial ports can be used for console (57600 bps, 8 bits, no parity, 1 stop bit, no handshaking). Use (*scca*) for the Open Firmware 'input-device' and 'output-device' variables.

  - Some USB, PCI, and Cardbus serial ports should work, but no one has tried them

- USB devices
  - Most MI USB devices should work (such as disks, printers, input devices, and ethernet interfaces)

    http://www.netbsd.org/Hardware/usb.html

- PCMCIA and Cardbus cards
  - Most MI PCMCIA and Cardbus cards should work, although very few have been tested with NetBSD/macppc

    http://www.netbsd.org/Hardware/cardbus.html
    http://www.netbsd.org/Hardware/pcmcia.html

- PCI cards
  - Most MI PCI cards should work, although very few have been tested with NetBSD/macppc

    http://www.netbsd.org/Hardware/pci.html

  Note: While the NetBSD kernel may support the various Cardbus, PCI, PCMCIA, and USB devices you may have, Open Firmware does *not* unless it has a specific Open Firmware ROM. This means you cannot boot from these devices. Some Adaptec SCSI controllers have bootable Open Firmware ROMs.

Unsupported devices

- On-board audio
- Floppy disk

  Note: Though NetBSD/macppc can boot from a floppy, there is no kernel support yet

- FireWire (IEEE 1394)
- Advanced power management (cannot put system to 'sleep')
- Multiple processors

  Note: Although NetBSD/macppc can boot with more than one processor present, it will not use the additional CPUs

- AirPort
- On-board video capture

Supported boot devices and media

Each version of Open Firmware supports different devices and media that you may boot from.

Go to the NetBSD/macppc Model Support webpage and look up your system. Take note of the comments about your model and keep these in mind during the rest of this installation procedure.

http://www.netbsd.org/Ports/macppc/models.html

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- CD-ROM
- FTP
- Remote NFS partition

The steps necessary to prepare the distribution sets for installation depend upon which installation medium you choose. The steps for the various media are outlined below.

CD-ROM Find out where the distribution set files are on the CD-ROM.

Proceed to the instruction on installation.

FTP The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. You need to know the numeric IP address of that site, and, if it's not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself. If you don't have access to a functioning nameserver during installation, the IP address of **ftp.netbsd.org** is `204.152.184.75` (as of October, 2000).

Once you have this information, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

NFS Place the NetBSD distribution sets you wish to install into a directory on an NFS server, and make that directory mountable by the machine on which you are installing or upgrading NetBSD. This will probably require modifying the `/etc/exports` file on of the NFS server and resetting its mount daemon (mountd). (Both of these actions will probably require superuser privileges on the server.)

You need to know the numeric IP address of the NFS server, and, if the server is not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

Once the NFS server is set up properly and you have the information mentioned above, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

Preparing your System for NetBSD installation

Open Firmware System Preparation

Most macppc systems have Open Firmware bugs. Luckily, Open Firmware has a small Non-Volatile RAM variable (NVRAM) which is reserved for FORTH commands which will be run before booting an operating system. Apple has released a freeware MacOS tool called System Disk, which patches most of these bugs. We *strongly* recommend that you use this tool to patch your Open Firmware, as several systems cannot boot without these patches.

Download System Disk from Apple's site:
ftp://ftp.apple.com/developer/macosxserver/utilities/SystemDisk2.3.1.smi.bin

For a brief tutorial on how to use System Disk, see:
http://www.netbsd.org/Ports/macppc/SystemDisk-tutorial/

Make sure you click the ''Save'' button, or the patches will not be saved to NVRAM.

Note: NVRAM patches and Open Firmware settings will be erased if you 'zap your PRAM' by holding down `COMMAND-OPTION-P-R` keys during the boot chimes

Updating your firmware

Open Firmware 3 systems have a rewritable firmware. Go to the 'Apple Software Updates' web site at **http://asu.info.apple.com/** and search for 'firmware' and install the most recent version for your model.

Getting to the Open Firmware Prompt

There are various methods to get to the Open Firmware prompt. They may or may not work, depending on which system model you have. Most models can use the on-board video and an ADB or USB keyboard to access Open Firmware. Those that cannot must use a serial console. Likewise, most systems can be set up to use a serial console if you do not wish to use the on-board video.

Open Firmware has two variables, 'input-device' and 'output-device ', which specify which devices to use for input and output. Most early models (Open Firmware versions 1.x and 2.0.x) will default to using a serial console. The later models (Open Firmware versions 2.4 and 3) will default to using the on-board video and keyboard. The instructions below describe how to set your system up to access Open Firmware from the on-board video and keyboard.

If, for some reason, you need to use a serial console, you can use a normal 'printer' cable (mini-DIN 8 to mini-DIN 8) and a MacOS tool, such as ZTerm to connect a MacOS system to your NetBSD/macppc system.
http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/comm/term/zterm-101.hqx

Getting to the Open Firmware Prompt (Easy method using MacOS)

Launch the MacOS System Disk tool. Click on ''Power User (Open Firmware)'' then click on the ''Advanced Options'' button. Now, click on the checkbox that says ''Stop Boot at Open Firmware prompt'' and select ''OK''. Click the ''Save'' button and reboot your system.

If your 'output-device' is '`/chaos/control`' (i.e. you have a PowerMacintosh 7300 - 8600 system), there is a chance that your monitor will not sync. See
http://www.netbsd.org/Ports/macppc/SystemDisk-tutorial/of105patch.html

You should now see the Open Firmware command prompt:

```
0 >
```

If your screen is black, then your model does not support using the on-board video in Open Firmware. You will need to connect up a serial console (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking).

Note: The "Stop Boot at Open Firmware prompt" setting is persistent. Your system will always stop at the Open Firmware prompt unless you uncheck this checkbox or tell Open Firmware

```
0 > setenv auto-boot? true
```

Note: Unfortunately, there are a few models that are better off without the System Disk patches. If you find that your machine doesn't boot, then try:

```
0 > setenv use-nvramrc? false
0 > reset-all
```

Getting to the Open Firmware Prompt (Harder method using MacOS)

If System Disk doesn't work because your version of MacOS is too old or because System Disk says that it doesn't support your model, then you may try using the MacOS BootVars tool on your Open Firmware 1 or 2 system (it will crash Open Firmware 3 systems).

ftp://ftp.netbsd.org/pub/NetBSD/arch/macppc/macos-utils/bootvars/bootvars.sit.hqx

Note: BootVars does *not* apply the (possibly critical) NVRAM patches that System Disk does. Expect devices to not work.

Look up the proper 'output-device' for your model on the NetBSD/macppc Model Support webpage.
http://www.netbsd.org/Ports/macppc/models.html
If the NetBSD/macppc Model support webpage does not list an 'output-device' for your model, then your system will default to using the on-board video. You needn't fill in the 'output-device' and 'input-device' variables.

Launch the MacOS BootVars tool. Click on the "auto-boot?" checkbox, then click on the "All Variables" checkbox and type **kbd** into the 'input-device' box, and the proper device name into the 'output-device' box. Click on the "write" button, and restart.

If your 'output-device' is '`/chaos/control`' (i.e. you have a PowerMacintosh 7300 - 8600 system), there is a chance that your monitor will not sync. See
http://www.netbsd.org/Ports/macppc/SystemDisk-tutorial/of105patch.html

You should now see the Open Firmware command prompt:

```
0 >
```

If your screen is black, then your model has defaulted to using a serial console. This is fairly common on Open Firmware 1 and 2 models if you do not use the System Disk tool to set up Open Firmware. You must hook up a serial console (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking).

Note: The "auto-boot?" setting is persistent. Your system will always stop at the Open Firmware prompt unless you uncheck this checkbox or tell Open Firmware

```
0 > setenv auto-boot? true
```

Getting to the Open Firmware Prompt (Without using MacOS)

If you don't have MacOS, then you need to hold down a special key combination when your system boots.

Note: Your Open Firmware 1 or 2 system will *not* have the (possibly critical) NVRAM patches that System Disk applies. Expect devices to not work.

After the chime starts, but before it stops, hold down the `COMMAND-OPTION-O-F` keys (the `COMMAND` key looks like a four-leaf clover or an open apple, and the `OPTION` key may look like a two-way switch with four straight line segments).

You should now see the Open Firmware command prompt:

```
0 >
```

If your screen is black, then your system has defaulted to using a serial console. This is fairly common on Open Firmware 1 and 2 models if you do not use the System Disk tool to set up Open Firmware. You must hook up a serial console (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking).

The Apple Network Server uses 'Port 2' (the '`ttya`' device in Open Firmware) as the console port.

Now, set your system to always stop at the Open Firmware prompt.

```
0 > setenv auto-boot? false
```

To allow Open Firmware to automatically boot the specified operating system when the system is booted, type:

```
0 > setenv auto-boot? true
```

To use your on-board video and keyboard, look up the proper 'output-device' for your model on the NetBSD/macppc webpage
http://www.netbsd.org/Ports/macppc/models.html
Run the following commands to use your screen instead of a serial console (replace '`screen`' with the correct 'output-device' for your model):

```
0 > setenv output-device screen
0 > setenv input-device kbd
0 > reset-all
```

Now you should see the Open Firmware prompt on your screen.

If your 'output-device' is '`/chaos/control`' (i.e. you have a PowerMacintosh 7300 - 8600 system), there is a chance that your monitor will not sync. See
http://www.netbsd.org/Ports/macppc/SystemDisk-tutorial/of105patch.html

Setting Open Firmware up to boot NetBSD

Check your Open Firmware version; you will need to know this in order to make decisions further on:

```
0 > dev /openprom
0 > .properties
```

If your system has Open Firmware prior to version 3, then you must set some Open Firmware variables before NetBSD can boot. Do *not* run these commands on Open Firmware 3 machines, as you may overwrite your firmware requiring a trip to Apple for repairs.

```
0 > setenv load-base 600000
0 > setenv real-base F00000
0 > reset-all
```

The last command reboots your machine so that the `real-base` setting takes effect.

If you will be netbooting your system, you can look up your MAC address.

```
0 > dev enet .properties
[...]
local-mac-address   CCCCCCCC CCCC
[...]
```

Note: Some early Open Firmware 1.0.5 machines had their MAC address stored incorrectly on the motherboard (little- vs. big-endian problems). The patches the System Disk installs will correct this. Without the patch, the machine will still work, but its MAC address may conflict with another ethernet device on your network.

Note: Some Open Firmware 3 machines have their MAC address stored incorrectly (little- vs. big-endian problem). If you look up your MAC address in MacOS, it will be different than what Open Firmware 3 uses to contact your netboot server. Your machine will still work, but its MAC address may conflict with another ethernet device on your network.

For future reference, when you want to boot into MacOS, type:

- Open Firmware 1, Open Firmware 2

  ```
  0 > bye
  ```

- Open Firmware 3

  ```
  0 > mac-boot
  ```

Note: Open Firmware 1.0.5 settings and nvram patches will be erased if you boot into MacOS. You will need to re-enter them before booting NetBSD again.

Note: Open Firmware 2.x systems will set the `real-base` environment variable to its default value (which doesn't work) if you boot into MacOS.

Available Boot Media

Each version of Open Firmware is capable of booting from a variety of media. To boot the NetBSD kernel, Open Firmware must first open a 'bootloader' which knows how to load the NetBSD kernel. There are two approaches, the `ofwboot.xcf` bootloader and the 'partition zero' bootloader. In the former case, the user tells Open Firmware to load `ofwboot.xcf` and it will load the kernel. In the latter case, the user tells Open Firmware to boot from a particular device, and a primary bootloader `bootxx` is loaded, which then loads the secondary bootloader `ofwboot` which then loads the kernel.

Open Firmware 3 systems can load the kernel without the use of a bootloader, although, some kernels do not "just work", and you will need to load the bootloader to load the kernel. Feel free to experiment and load the kernel with the command you use to load the bootloader, bypassing a step, however we recommend using a bootloader.

The following bootable media are available for loading the bootloader (`ofwboot` or `ofwboot.xcf`):

- *Partition zero*
  (Open Firmware 1.x, Open Firmware 2.x)

  This method loads a primary bootloader called `bootxx` which then loads the secondary bootloader `ofwboot` (functionally identical to `ofwboot.xcf` (q.v.)) which then loads the kernel. This is the method used by the boot floppies, and when you're installing NetBSD on the entire disk. Unfortunately, it appears that the Open Firmare 2.4 floppy disk driver is broken.

- *Partition zero*
  (Open Firmware 3)

  While Open Firmware 3 supports this boot method, NetBSD/macppc does not yet support it on these systems.

- *HFS or HFS+ file system*
  (Open Firmware 2.4, Open Firmware 3)

  Whilst you can load `ofwboot.xcf` from an HFS or HFS+ partition, you will not be able to load the kernel from HFS or HFS+ filesystems. This will be resolved in a future release of NetBSD/macppc

- *MS-DOS file system*
  (All Open Firmware versions)

  This cannot be used, as neither the `ofwboot.xcf` bootloader nor the kernel can read files directly from a MS-DOS file system.

- *Ethernet (network boot)*
  (All Open Firmware versions)

  You can run your entire system diskless or netboot only the files necessary to boot (i.e. the bootloader and the installation kernel), but you must have root access on another UNIX machine on your subnet.

  Note: Some Open Firmware 1.0.5 machines do not retrieve the BOOTP information correctly. These machines cannot netboot.

- *ISO9660 CD-ROM file system*
  (Open Firmware 1.x, Open Firmware 2.x)

  Load the `ofwboot.xcf` bootloader from an ISO 9660 CD-ROM. It can then load a NetBSD kernel from the ISO 9660 CD-ROM.

  Open Firmware 1.0.5 and 2.0.x machines will *not* boot from the ISO 9660 file system if you create a hybrid HFS/ISO format CD-R. Make sure that it the CD *only* has ISO 9660 data and does not have an Apple Partition Map.

  Note: Open Firmware can only open files on the first session of a multi-session CD-R

- *Hybrid HFS/ISO 9660 CD-ROM file system*
  (Open Firmware 2.4, Open Firmware 3)

  Load the `ofwboot.xcf` bootloader from the CD-ROM's HFS filesystem. It can then load a NetBSD kernel from the ISO 9660 filesystem.

  Make sure that the CD has an Apple Partition Map.

  Note: Open Firmware can only open files on the first session of a multi-session CD-R

Once `ofwboot` or `ofwboot.xcf` is loaded, it can open the kernel from one of the following sources:

- *Ethernet (NFS)*
- *NetBSD FFS file system*

  This is not available the first time you install NetBSD on your system.

- *ISO 9660 CD-ROM file system*

The boot floppy image provided with the distribution sets has a 'partition zero' bootloader and a NetBSD file system with an installation kernel.

Installation techniques

Run the installer to place NetBSD on a hard drive on your macppc system.

If you will be running your system diskless (i.e. entirely over NFS, not using any local hard drives), then you do not need to run the installer although you may still do so.

Note: Some MacOS **ftp** clients default to downloading files in 'ASCII' mode. This will render the NetBSD files useless. Make sure to set your ftp program to download in 'binary' mode.

Here are some common techniques for booting the NetBSD install kernel:

- *Boot off a floppy*
  (Open Firmware 1.x, Open Firmware 2.0.x)

  Use some tool to write the boot floppy image `installation/boot.fs` to a floppy disk, and boot from that. The primary bootloader (in 'partition zero') will load `ofwboot` which will load the installation kernel.

  – *From a UNIX machine*

  ```
  # dd if=boot.fs of=/dev/rfd0a bs=36k
  ```

  – *From a Windows machine*

  Get either the **rawrite.exe** program from the NetBSD/i386 distribution or **rawrite32.exe** from **http://www.duskware.com/rawrite32/**

  – *From a MacOS machine*

  Get **suntar** from
  http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/cmp/suntar-222.hqx

  1. Open the 'Preferences' menu and pick 'Expert Mode'
  2. Open the 'Special' menu and pick 'Overwrite Sectors ...'
  3. Insert a floppy disk
  4. Hit return (i.e. start at sector number zero)
  5. Select `boot.fs` and click "Open"
  6. Open the 'File' menu and pick 'Eject'

- *Boot off an ISO 9660 CD-ROM*
  (Open Firmware 1.x, Open Firmware 2.x)

  Place `ofwboot.xcf` and the installation kernel at the top level of the CD. You may also place the NetBSD 1.5 distribution sets on the disk. Have Open Firmware load `ofwboot.xcf` from the disk, and it will then load the installation kernel.

  – *From a UNIX machine*

  ```
  # mkhybrid -o output.iso -a -l -J -r -L /cdsources
  ```

  See the NetBSD Bootable CD-ROM HOW-TO for more detail:
  http://www.netbsd.org/Documentation/bootcd.html

– *From a MacOS machine (using Toast)*

1. Open the 'Format' menu and pick 'ISO 9660'
2. Click the "Data..." button and drag `installation/ofwboot.xcf` and `installation/netbsd.ram.gz` into the window
3. Click the "Settings" tab, open the "Naming" popup menu, and pick "Macintosh Names"
4. Click the "Done" button
5. Click the "Write CD..." button

- *Boot off a hybrid HFS/ISO 9660 CD-ROM*
  (Open Firmware 2.4, Open Firmware 3)

  Place `ofwboot.xcf` and the installation kernel at the top level of the CD. Make sure that the bootloader is present on the HFS partition, and the kernel is on the ISO partition. You may also place the NetBSD 1.5 distribution sets on the ISO partition. Have Open Firmware load `ofwboot.xcf` from the HFS partition, and it will load the installation kernel from the ISO partition.

– *From a UNIX machine*

```
# mkhybrid -o output.iso -hfs -part -a -l -J -r -L /cdsources
```

See the NetBSD Bootable CD-ROM HOW-TO for more detail:
http://www.netbsd.org/Documentation/bootcd.html

– *From a MacOS machine (using Toast)*

1. Open the 'Format' menu and pick 'Mac/ISO Hybrid'
2. Open the 'Utilities' menu and pick 'Create Temporary Partition...'
3. Make this temporary partition large enough for all of the files you will be placing on the HFS portion of the CD (only `ofwboot.xcf` is necessary).
4. Copy `installation/ofwboot.xcf` to the temporary partition
5. Click the "Data..." button and select the temporary partition you just created
6. Click the "ISO..." button and drag `installation/netbsd.ram.gz` into the window
7. Click the "Settings" tab, open the "Naming" popup menu, and pick "Macintosh Names"
8. Click the "Done" button
9. Click the "Write CD..." button

- *Boot off an HFS or HFS+ partition*
  (Open Firmware 3)

  Download the install kernel `installation/netbsd.ram.gz` and place it at the top level of any partition. Use a MacOS utility such as **`Stuffit Expander`** to uncompress the kernel. Get to the Open Firmware prompt. Boot the kernel directly (i.e. without the use of a bootloader) and use it to install NetBSD on your hard drive.

- *Boot over the ethernet*
  (All Open Firmware versions)

  Set up the DHCP daemon in BOOTP compatable mode, the TFTP daemon, and the NFS daemons on your netboot server. Tell Open Firmware to boot over ethernet, and it will send a BOOTP request, which will tell Open Firmware what your system's IP address is, and where its bootloader can be downloaded via TFTP. Once Open FIrmware has downloaded the bootloader via TFTP, it will then load the installation kernel via an NFS connection.

Follow these instructions to configure a NetBSD machine as your netboot server. We will use '`CC:CC:CC:CC:CC:CC`' as the MAC address (ethernet hardware address) of your netboot client machine that you wish to install NetBSD on. We use '`192.168.1.10`' as the IP address of your client, and '`client.test.net`' as the name. The server name is '`server.test.net`', and the path on the server to the NFS exported directory is `/export/client/root` although you may choose to use a different path.

Now, set up your **dhcpd** server. Make sure the following lines are in `/etc/dhcpd.conf`:

```
allow bootp;    # Allow bootp requests, thus the dhcp server
                # will act as a bootp server.

subnet 192.168.1.0 netmask 255.255.255.0 {
                # Which network interface the server will
                # listen on.  The zeros indicate which
                # range of addresses are allowed to connect.
}

group {
                # Set of parameters common to all clients
                # in this "group".
                #
        option broadcast-address      192.168.1.255;
        option domain-name            "test.net";
        option domain-name-servers    dns.test.net;
        option routers                router.test.net;
        option subnet-mask            255.255.255.0;

                # An individual client.
                #
        host client.test.net {
                hardware ethernet     CC:CC:CC:CC:CC:CC;
                fixed-address         192.168.1.10;

                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name      "client";

                # Name of the bootloader or kernel
                # to download via tftp.
                #
                filename              "ofwboot.xcf";

                # The path on the NFS server.
                #
                option  root-path     "/export/client/root";
        }
}
```

You will need to make sure that the `dhcpd.leases` file exists.

```
# touch /var/db/dhcpd.leases
```

You will need to **kill** the **dhcpd** and restart it to get it to re-read its configuration file. If the server is running NetBSD 1.5, you can achieve this with:

```
# /etc/rc.d/dhcpd restart
```

Now, place a copy of the `installation/ofwboot.xcf` bootloader in `/tftpboot`

```
# cp ofwboot.xcf /tftpboot
```

Now, you should edit `/etc/inetd.conf` and make sure that the line starting with `tftp` is uncommented. You will need to **kill -HUP** the **inetd** process to enable tftp if this line was previously commented out. The next step is to set up NFS exports. Create the directory you are exporting for the netboot client:

```
# mkdir -p /export/client/root
```

Put the following line in `/etc/exports` to enable NFS sharing:

```
/export/client/root -maproot=root client.test.net
```

If your server is currently running NFS services, you only need to **kill -HUP** the **mountd** process. Otherwise, you need to run:

```
# /usr/sbin/mountd
# /usr/sbin/nfsd -tun 4
```

Now, you need to place the files your netboot client will need. As noted above in the section `Preparing your System for NetBSD Installation` you have several options when choosing a location to store the distribution filesets. However, the easiest way is to put the distribution files into the exported directory for your client on the server.

```
# mv *.tgz /export/client/root
```

Your client will need a kernel to boot (use `installation/netbsd.ram.gz`).

```
# gunzip netbsd.ram.gz
# mv netbsd.ram /export/client/root/netbsd
```

• *Boot the floppy image off a bootable hard drive*
(Open Firmware 1.x, Open Firmware 2.x)

Find a spare bootable drive (i.e. SCSI or IDE), and use some tool to write the floppy disk image `installation/boot.fs` to your spare drive, and boot from that drive. The floppy image has a 'partition zero' bootloader which ultimately loads the installation kernel from the FFS filesystem from the disk image.

– *From a UNIX machine*

```
# dd if=boot.fs of=/dev/rsd0c
```

where `/dev/rsd0c` is the 'whole disk' partition for the drive you will be using. Be certain you have typed this correctly, as it will erase the disk.

– *From a Windows machine*

Get **rawrite32.exe** from **http://www.duskware.com/rawrite32/**

Be certain you have selected the correct disk, as it will erase the contents.

– *From a MacOS machine*

Get **suntar** from
http://hyperarchive.lcs.mit.edu/HyperArchive/Archive/cmp/suntar-222.hqx

1. Open the 'Preferences' menu and pick 'Expert Mode'
2. Open the 'File' menu and pick 'Open Device ...'
3. Click on the "scan SCSI" button to get a list of which drives are attached.
4. Pick the correct drive. Be certain you have selected the correct disk, as it will erase the contents.
5. Open the 'Special' menu and pick 'Overwrite Sectors ...'
6. Hit return (i.e. start at sector number zero)
7. Select `boot.fs` and click "Open"
8. Open the 'File' menu and pick 'Close Device'

Partitioning the drive NetBSD will be installed on

You have two options. You may use the NetBSD installer to partition your drive, in which case you may use a 'partition zero' bootloader, but you cannot have HFS or HFS+ filesystems; or you may use a MacOS tool to partition your drive, in which case you cannot use a 'partition zero' bootloader, but you can have HFS and HFS+ filesystems.

If you have an Open Firmware 1.x or 2.0.x system, you must use the NetBSD installer to partition your drive if you wish it to be bootable. You may partition your drive in MacOS but you will have to load `ofwboot.xcf` from some other media (such as over the network, from a floppy, or from a CD-ROM).

If you have an Open Firmware 2.4 system, either partitioning method will produce a bootable disk. If you use a MacOS tool, you must manually place `ofwboot.xcf` on one of your HFS or HFS+ partitions.

If you have an Open Firmware 3 system, you must use a MacOS tool to create a bootable disk. You will need to manually place `ofwboot.xcf` on one of your HFS or HFS+ partitions. If you use the NetBSD installer to partition your drive, you will need to load `ofwboot.xcf` from some other media (such as over the network or from a CD-ROM).

Unfortunately, the partitioning tools for NetBSD are not advanced enough to create HFS or HFS+ file systems. Therefore, if you wish to share this drive with MacOS, you will need to use the MacOS Drive Setup tool (versions 1.9.1 and later are known to work) to divide the hard drive you will be using for NetBSD into the following partitions:

HFS or *HFS+* Must be large enough to hold the bootloader, over 100 KB.

A/UX Root Must be at least 20 MB. Alternatively, you may decide to use one partition for your entire NetBSD installation, in which case it should be at least 200 MB.

A/UX Swap Any size. The recommenation is 1.5 times your RAM, although this is not strictly necessary for machines with a lot of RAM.

A/UX User, A/UX Free1, A/UX Free2, A/UX Free3
Use these for any additional partitions you may want to use under NetBSD.

Installing the NetBSD System

Booting the installer

The syntax of the Open Firmware **boot** command is:

```
boot device[:partition][,filename] [kernel_location] [-as]
```

where the **-a** flag will ask you for the location of the next item to load (i.e. `ofwboot.xcf` will ask where the kernel is, or the kernel will ask where the root file system is). The **-s** flag will boot into 'single-user' mode. `kernel_location` is either a filename if the kernel is on the same partition as the bootloader, or another complete '`device:partition,filename`'. And, there's no reason (other than Open Firmware bugginess) that you can't specify an entirely different device for the kernel. For instance, you could have a MacOS drive on your *ultra0* (ATA) bus, and a NetBSD drive on your *ultra1* bus.

The exact command you will be using depends on which version of Open Firmware your machine has and which device you will be booting from. You can find out what device aliases Open Firmware has on your machine by typing:

```
0 > devalias
```

To get a complete listing of what hardware is recognized by Open Firmware, type:

```
0 > dev / ls
```

To determine if a device is bootable, type:

```
0 > dev /path/to/device
0 > words
```

If the word 'open' is in the list of words, then this device might be bootable.

SCSI disks are usually of the form:

```
boot scsi_devalias/sd@i:p
```

where '`scsi_devalias`' is the Open Firmware '`devalias`' for your SCSI interface, '`i`' is the SCSI ID, and '`p`' is the partition.

- *Floppy disk*
  (Open Firmware 1.x, Open Firmware 2.0.x)

  The floppy disk has a 'partition zero' bootloader. All you need to do is:

  ```
  0 > boot fd:0
  ```

  You can eject a floppy by typing:

  ```
  0 > eject fd
  ```

- *Ethernet*
  (All Open Firmware versions)

  You only specify that you are booting from the ethernet:

  - Open Firmware 1, Open Firmware 2

    ```
    0 > boot enet
    ```

  - Open Firmware 3

    ```
    0 > boot enet:0
    ```

To make configuring your netboot server easier, you may specify the bootloader filename and the kernel name:

- Open Firmware 1, Open Firmware 2

  0 > **boot enet:,ofwboot.xcf enet:,netbsd.ram.gz**

- Open Firmware 3

  0 > **boot enet:0,ofwboot.xcf enet:0,netbsd.ram.gz**

• *CD-ROM*
(All Open Firmware versions)

Open Firmware doesn't understand long filenames (created with the RockRidge and Joliet extensions), so you may need to figure out what your file is called. For instance, **Toast** for MacOS creates CDs with long filenames, but uses MS-DOS style short names as well. Instead of referencing netbsd-GENERIC you would need to open NETBSD-G.ENE

The **mkisofs** and **mkhybrid** programs have a similar problem. For them, netbsd.ram.gz becomes NETBSD_RAM.GZ

Another thing to note is that you *must* use capital letters when specifying the filename to load. Also, keep in mind what format your CD-R has to be for your version of Open Firmware (pure ISO versus hybrid ISO/HFS) before giving up.

Here are some examples of what you might use to boot from CD-ROM:

```
0 > boot scsi-int/sd@3:0,OFWBOOT.XCF NETBSD.RAM
0 > boot scsi/sd@3:0,OFWBOOT.XCF NETBSD.RAM
0 > boot scsi-ext/sd@3:0,OFWBOOT.XCF NETBSD.RAM
0 > boot ata/atapi-disk:0,OFWBOOT.XCF NETBSD.RAM
0 > boot ide1/disk@0:0,OFWBOOT.XCF NETBSD.RAM
0 > boot cd:0,OFWBOOT.XCF NETBSD.RAM
```

• *IDE or SCSI drive with a* 'partition zero' bootloader
(Open Firmware 1, Open Firmware 2)

You do not specify a file to load, since the 'partition zero' booloader knows what to do. You would boot such a system if you have dumped the boot floppy image to your drive, or if you have an already-installed NetBSD/macppc system

Remember, that SCSI Zip disks are usually ID 5 or 6. Internal hard drives are usually SCSI ID 0.

Here are some examples of what you might use to boot from such a drive:

```
0 > boot scsi-int/sd@0:0
0 > boot scsi/sd@0:0
0 > boot scsi-ext/sd@0:0
0 > boot ata/ata-disk@0:0
0 > boot ata/ATA-Disk@0:0
0 > boot ide0/disk@0:0
0 > boot zip:0
```

• *IDE or SCSI drive from an HFS or HFS+* IDE0
(Open Firmware 2.4, Open Firmware 3)

MacOS drives have several system-level 'partitions reserved for' MacOS drivers. You may find that your first HFS or HFS+ partition might be as high as partition 9. You may need to keep trying higher partition numbers until you find the one that has your bootloader.

Also, since the bootloader is on a different partition from the NetBSD kernel, you will need to specify where to find the kernel.

Remember, that SCSI Zip disks are usually ID 5 or 6. Internal hard drives are usually SCSI ID 0.

Here are some examples of what you might use to boot from such a drive:

```
0 > boot scsi/sd@0:9,ofwboot.xcf scsi/sd@0:10,netbsd
0 > boot ide0/disk@0:7,ofwboot.xcf ide0/disk@0:9,ofwboot.xcf
0 > boot hd:8,ofwboot.xcf ultra1:3,netbsd
```

Common Problems and Error Messages

See the NetBSD/macppc FAQ for a thorough list.
http://www.netbsd.org/Ports/macppc/faq.html

Note: You may find it necessary to remove all non-Apple devices in your machine. Some users have found this necessary.

- "`DEFAULT CATCH!`"

  If your machine is Open Firmware version 1.0.5 or 2.X, this error does sometimes appear randomly. You might try the boot command a second time (this is known to work on some models). Otherwise, if you are not using a serial console, there might be a conflict between the screen driver and Open Firmware which prevents Open Firmware from loading `ofwboot.xcf` There are several models that cannot be booted while using the screen and keyboard. The only known workaround is to use a serial console (i.e. set your input and output devices to ttya or ttyb).

  ```
  0 > setenv output-device ttya
  0 > setenv input-device ttya
  0 > reset-all
  ```

- "`CLAIM failed`"

  This is a general message from Open Firmware to the effect that it failed to allocate some memory. There are several known causes for this to happen.

  This often means that the memory is messed up. If you tried something and it failed, then you tried another and got a "`CLAIM failed`" message, then this is an indication that you should reboot between attempts

  If your machine is Open Firmware version 1.0.5 or 2.X, this error does sometimes appear randomly. You might try the boot command a second time (this is known to work on some models). Otherwise, if you are not using a serial console, there might be a conflict between the screen driver and Open Firmware which prevents Open Firmware from loading `ofwboot.xcf` There are several models that cannot be booted while using the screen and keyboard. The only known workaround is to use a serial console (i.e. set your input and output devices to ttya or ttyb).

  ```
  0 > setenv output-device ttya
  0 > setenv input-device ttya
  0 > reset-all
  ```

- "`can't OPEN`"

  Open Firmware either can't open the device you specified (because it is not present or the device path is mistyped) or the file you specified. Check your typing and check to make sure that the media has the files you think it has.

- "`unrecognized Client Program formatstate not valid`"

  This is a general Open Firmware error message indicating that the filename you tried to open either doesn't exist or is in the wrong format. For Open Firmware 1 and 2 machines, it must be an XCOFF file (such as `ofwboot.xcf`) and for Open Firmware 3 machines, it must be either XCOFF or ELF (such as a kernel).

- Black screen

  If your system is a PowerBook, NetBSD may have turned down the brightness of the backlight. Use the buttons to turn your brightness back up.

  Otherwise, you need a serial console. See the section entitled **Getting to the Open Firmware Prompt**

- Grey screen with flashing question mark

  You mistyped the path to a device, and your system defaulted to booting into MacOS. The flashing question mark means that the MacOS ROM has loaded and is looking for a bootable MacOS file system.

- `ofwboot.xcf` hangs before the copyright notice and the kernel configuration

  You forgot to set "`real-base`" in Open Firmware or it got erased by your booting into MacOS.

  ```
  0 > setenv load-base 600000
  0 > setenv real-base F00000
  0 > reset-all
  ```

- After the kernel loads, the first key you press repeats forever

  Press the `SHIFT` key first.

- Hang after configuring devices, but before doing anything else

  Plug the USB keyboard directly into the computer (i.e. not through a hub)

- "`pciide0:0:0: lost interrupt`"

  See
  http://www.netbsd.org/Ports/macppc/faq.html#pciide-lost-interrupt

Running the sysinst installation program

1. *Introduction*

   Using **`sysinst`**, installing NetBSD is a relatively easy process. You still should read this document and have it in hand when doing the installation process. This document tries to be a good guideline for the installation and as such covers many details to be completed. Do not let this discourage you, the install program is not hard to use.

2. *General*

   The following is a walk-through of the steps you will take while getting NetBSD installed on your hard disk. **`sysinst`** is a menu driven installation system that allows for some freedom in doing the installation. Sometimes, questions will be asked and in many cases the default answer will be displayed in brackets ("[ ]") after the question. If you wish to stop the installation, you may press `CONTROL-C` at any time, but if you do, you'll have to begin the installation process again from scratch.

3. *Quick install*

   First, let's describe a quick install. The other sections of this document go into the installation procedure in more detail, but you may find that you do not need this. If you want detailed instructions, skip to section 3. This section describes a basic installation, using a CD-ROM install as an example.

- What you need.
  - The distribution sets (in this example, they are on CD).
  - Some form of bootable media, described above.
  - A CD-ROM drive (SCSI or ATAPI), a harddisk and a minimum of 8 MB of memory installed.
  - The harddisk should have at least 200 + *n* megabytes of space free, where *n* is the number of megabytes of main memory in your system. If you wish to install the X window system as well, you will need at least 60 MB more.
- The Quick Installation
  - Boot the system as described above. You should be at the **sysinst** main menu.
  - If you wish, you can configure some network settings immediately by choosing the **utilities** menu and then **configure network**. It isn't actually required at this point, but it may be more convenient. Go back to the main menu.
  - Choose **install**
  - You will be guided through some steps regarding the setup of your disk, and the selection of distributed components to install. When in doubt, refer to the rest of this document for details.
  - After your disk has been prepared, choose **CD-ROM** as the medium. The default values for the path and device should be ok.
  - After all the files have been unpacked, go back to the main menu and select **reboot.**
  - Once the system reaches the Open Firmware prompt, you will need to type the correct command to boot from your hard drive. NetBSD will now boot. If you haven't already done so in sysinst, you should log in as `root`, and set a password for that account. You are also advised to edit the file `/etc/rc.conf` to match your system needs.
  - Your installation is now complete.
  - For configuring the X window system, if installed, see the files in `/usr/X11R6/lib/X11/doc`.

4. *Booting NetBSD*

   Boot the system as described in the previous section.

   You may want to read the boot messages, to notice your disk's name and geometry. Its name will be something like `sd0` or `wd0` and the geometry will be printed on a line that begins with its name. As mentioned above, you may need your disk's geometry when creating NetBSD 's partitions. You will also need to know the name, to tell **sysinst** on which disk to install. The most important thing to know is that `wd0` is NetBSD 's name for your first IDE disk, `wd1` the second, etc. `sd0` is your first SCSI disk, `sd1` the second, etc.

   Once NetBSD has booted and printed all the boot messages, you will be presented with a welcome message and a main menu. It will also include instructions for using the menus.

5. *Network configuration*

   If you will not use network operation during the installation, but you do want your machine to be configured for networking once it is installed, you should first go to the utilities menu, and select **Configure network option**. If you only want to temporarily use networking during the installation, you can specify these parameters later. If you are not using Domain Name Service (DNS), you can give an empty response in reply to answers relating to this.

6. *Installation drive selection and parameters*

   To start the installation, select the menu option to install NetBSD from the main menu.

   The first thing is to identify the disk on which you want to install NetBSD. **sysinst** will report a list of disks it finds and ask you for your selection. Depending on how many disks are found, you may get a different message. You should see disk names like `wd0`, `wd1`, `sd0` or `sd1`.

7. *Partitioning the disk*

   - Which portion of the disk to use.

     You will be asked if you want to use the entire disk or only part of the disk. If you decide to use the entire disk for NetBSD, it will be checked if there are already other systems present on the disk, and you will be asked to confirm whether you want to overwrite these.

   - *Editing the NetBSD disklabel*

     The partition table of the NetBSD part of a disk is called a *disklabel*. In actuality, NetBSD uses the Apple Partition Map on your disk. If you use NetBSD to partition your drive, you will not be able to install HFS or HFS+ partitions on your disk. You may partition the drive using a MacOS tool, such as 'Drive Setup' 1.9.1 or later, however you may not be able to boot from this drive. See above. There are 3 layouts for the NetBSD part of the disk that you can pick from: **Standard, Standard with X** and **Custom**. The first two use a set of default values (that you can change) suitable for a normal installation, possibly including X. The last option lets you specify everything yourself.

     You will be presented with the current layout of the NetBSD disklabel, and given a chance to change it. For each partition, you can set the type, offset and size, block and fragment size, and the mount point. The type that NetBSD uses for normal file storage is called **4.2BSD**. A swap partition has a special type called **swap**. Some partitions in the disklabel have a fixed purpose.

     `a` Root partition. (/),

     `b` Swap partition.

     `c` The entire disk.

     `d-h` Available for other use. Traditionally, `d` is the partition mounted on `/usr`, but this is historical practice and not a fixed value.

     You will then be asked to name your disk's disklabel. The default response is **mydisk**. For most purposes this will be OK. If you choose to name it something different, make sure the name is a single word and contains no special characters. You don't need to remember this name.

8. *Preparing your hard disk*

   You are now at the point of no return. Nothing has been written to your disk yet, but if you confirm that you want to install NetBSD, your hard drive will be modified. If you are sure you want to proceed, enter `yes` at the prompt.

   The install program will now label your disk and make the file systems you specified. The file systems will be initialized to contain NetBSD bootstrapping binaries and configuration files. You will see messages on your screen from the various NetBSD disk preparation tools that are running. There should be no errors in this section of the installation. If there are, restart from the beginning of the installation process. Otherwise, you can continue the installation program after pressing the return key.

9. *Getting the distribution sets*

The NetBSD distribution consists of a number of *sets*, that come in the form of gzipped tarfiles. A few sets must be installed for a working system, others are optional. At this point of the installation, you will be presented with a menu which enables you to choose from one of the following methods of installing the sets. Some of these methods will first load the sets on your hard disk, others will extract the sets directly.

For all these methods, the first step is making the sets available for extraction, and then do the actual installation. The sets can be made available in a few different ways. The following sections describe each of those methods. After reading the one about the method you will be using, you can continue to section labeled 'Extracting the distribution sets'

10. *Installation using ftp*

    To be able to install using ftp, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you to provide some data, like IP number, hostname, etc. If you do not have name service set up for the machine that you are installing on, you can just press `RETURN` in answer to these questions, and DNS will not be used.

    You will also be asked to specify the host that you want to transfer the sets from, the directory on that host, and the account name and password used to log into that host using ftp. If you did not set up DNS when answering the questions to configure networking, you will need to specify an IP number instead of a hostname for the ftp server.

    sysinst will proceed to transfer all the default set files from the remote site to your hard disk.

11. *Installation using NFS*

    To be able to install using NFS, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you to provide some data, like IP number, hostname, etc. If you do not have name service set up for the machine that you are installing on, you can just press `RETURN` in answer to these questions, and DNS will not be used.

    You will also be asked to specify the host that you want to transfer the sets from, and the directory on that host that the files are in. This directory should be mountable by the machine you are installing on, i.e. correctly exported to your machine.

    If you did not set up DNS when answering the questions to configure networking, you will need to specify an IP number instead of a hostname for the NFS server.

12. *Installation from CD-ROM*

    When installing from a CD-ROM, you will be asked to specify the device name for your CD-ROM player (usually `cd0`), and the directory name on the CD-ROM where the distribution files are.

    sysinst will then check if the files are indeed available in the specified location, and proceed to the actual extraction of the sets.

13. *Installation from an unmounted file system*

    In order to install from a local file system, you will need to specify the device that the file system resides on (for example `sd1e`) the type of the file system, and the directory on the specified file system where the sets are located. **sysinst** will then check if it can indeed access the sets at that location.

14. *Installation from a local directory*

    This option assumes that you have already done some preparation yourself. The sets should be located in a directory on a file system that is already accessible. **sysinst** will ask you for the name of this directory.

15. *Extracting the distribution sets*

    After the install sets containing the NetBSD distribution have been made available, you can either extract all the sets (a full installation), or only extract sets that you have selected. In the latter case you will be shown the currently selected sets, and given the opportunity to select the sets you want. Some sets always need to be installed (**kern, base** and **etc**) they will not be shown in this selection menu.

    Before extraction begins, you can elect to watch the files being extracted; the name of each file that is extracted will be shown. This can slow down the installation process considerably, especially on machines with slow graphics consoles or serial consoles.

    After all the files have been extracted, all the necessary device node files will be created. If you have already configured networking, you will be asked if you want to use this configuration for normal operation. If so, these values will be installed in the network configuration files. The next menu will allow you to select the time zone that you're in, to make sure your clock has the right offset from GMT. Finally you can set a password for the "root" account, to prevent the machine coming up without access restrictions.

16. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 1.5.

Now, you can reboot to get to the Open Firmware prompt. Once there, you can set Open Firmware to always boot into NetBSD from the media of your choice. First, make sure you know what command you need to use to boot (see above). If you are using a 'partition zero' style boot scheme, you would type something like the following:

```
0 > setenv auto-boot? true
0 > setenv boot-device scsi/sd@0:0
0 > setenv boot-file netbsd
```

replacing **scsi/sd@0:0** with the actual device you will be booting from. If you find that your system tries booting before your hard drive has spun up, you may need to also enter something like:

```
0 > setenv boot-command begin ['] boot catch 1000 ms cr again
```

If you are not using a 'partition zero' style boot scheme, then you can still plug in all the information you need:

```
0 > setenv auto-boot? true
0 > setenv boot-device ide0/disk@0:8,ofwboot.elf
0 > setenv boot-file ide0/disk@0:13,/netbsd
0 > setenv boot-command boot
```

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state, with the most important ones described below.

1. Configuring `/etc/rc.conf`

   If you or the installation software haven't done any configuration of `/etc/rc.conf`, the system will drop you into single user mode on first reboot with the message

   ```
   /etc/rc.conf is not configured. Multiuser boot aborted.
   ```

   and with the root file system (/) mounted read-write. When the system asks you to choose a shell, simply press RETURN to get to a prompt. If you are asked for a terminal type, respond with **vt220** (or whatever is appropriate for your terminal type) and press RETURN. At this point, you need to configure at least one file in the `/etc` directory. Change to the `/etc` directory and take a look at the

/etc/rc.conf file. Modify it to your tastes, making sure that you set rc_configured=YES so that your changes will be enabled and a multi-user boot can proceed. Default values for the various programs can be found in /etc/defaults/rc.conf, where some in-line documentation may be found. More complete documentation can be found in rc.conf(5).

If your /usr directory is on a separate partition and you do not know how to use **ed**, you will have to mount your /usr partition to gain access to **ex** or **vi**. Do the following:

```
# mount /usr
# export TERM=vt220
```

If you have /var on a separate partition, you need to repeat that step for it. After that, you can edit /etc/rc.conf with vi(1). When you have finished, type **exit** at the prompt to leave the single-user shell and continue with the multi-user boot.

Other values that need to be set in /etc/rc.conf for a networked environment are *hostname* and possibly *defaultroute*, furthermore add an *ifconfig_int* for your interface <int>, where your on-board interfaces may be one of mc0, bm0, or gm0

```
ifconfig_mc0="inet 123.45.67.89 netmask 255.255.255.0"
```

or, if you have *myname.my.dom* in /etc/hosts:

```
ifconfig_mc0="inet myname.my.dom netmask 255.255.255.0"
```

To enable proper hostname resolution, you will also want to add an /etc/resolv.conf file or (if you are feeling a little more adventurous) run named(8). See resolv.conf(5) or named(8) for more information.

Other files in /etc that may require modification or setting up include /etc/mailer.conf, /etc/nsswitch.conf, and /etc/wscons.conf.

2. Logging in

   After reboot, you can log in as root at the login prompt. There is no initial password, but if you're using the machine in a networked environment, you should create an account for yourself (see below) and protect it and the "root" account with good passwords. Unless you have connected an unusual terminal device as the console you can just press RETURN when it prompts for Terminal type? [...]

3. Adding accounts

   Use the useradd(8) command to add accounts to your system, *do not* edit /etc/passwd directly. See useradd(8) for more information on how to add a new user to the system.

4. The X Window System

   If you have installed the X window system, look at the files in /usr/X11R6/lib/X11/doc for information.

   Don't forget to add /usr/X11R6/bin to your path in your shell's dot file so that you have access to the X binaries.

5. Installing third party packages

   If you wish to install any of the software freely available for UNIX-like systems you are strongly advised to first check the NetBSD package system. This automatically handles any changes necessary to make the software run on NetBSD, retrieval and installation of any other packages on which the software may depend, and simplifies installation (and deinstallation), both from source and precompiled binaries.

- More information on the package system is at
  http://www.netbsd.org/Documentation/software/packages.html
- A browsable listing of available packages is at
  ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/README.html
- Precompiled binaries can be found at
  ftp://ftp.netbsd.org/pub/NetBSD/packages/
- Package sources for compiling packages can be obtained by retrieving the file
  ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/tar_files/pkgsrc.tar.gz
  They are typically extracted into `/usr/pkgsrc` (though other locations work fine), as with the command:

  ```
  # mkdir /usr/pkgsrc; tar -C /usr/pkgsrc -zxpf pkgsrc.tar.gz
  ```

  After extracting, then see the `README` file in the extraction directory (e.g. `/usr/pkgsrc/README`) for more information.

6. Misc

   - Edit `/etc/mail/aliases` to forward root mail to the right place (run `newaliases`(1) afterwards.)
   - The `/etc/mail/sendmail.cf` file will almost definitely need to be adjusted; files aiding in this can be found in `/usr/share/sendmail`. See the README file there for more information.
   - Edit `/etc/rc.local` to run any local daemons you use.
   - Many of the `/etc` files are documented in section 5 of the manual; so just invoking

     ```
     # man 5 filename
     ```

     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

The upgrade to NetBSD 1.5 is a binary upgrade; it can be quite difficult to advance to a later version by recompiling from source due primarily to interdependencies in the various components.

To do the upgrade, you must boot from the installer kernel using one of the methods described above. To do the upgrade, you must have the boot floppy available. You must also have at least the **base** and **kern** binary distribution sets available, so that you can upgrade with them, using one of the upgrade methods described above. Finally, you must have sufficient disk space available to install the new binaries. Since the old binaries are being overwritten in place, you only need space for the new binaries, which weren't previously on the system. If you have a few megabytes free on each of your root `( / )` and `/usr` partitions, you should have enough space.

Since upgrading involves replacing the boot blocks on your NetBSD partition, the kernel, and most of the system binaries, it has the potential to cause data loss. You are strongly advised to *back up any important data on your disk*, whether on the NetBSD partition or on another operating system's partition, before beginning the upgrade process.

The upgrade procedure using the **`sysinst`** tool is similar to an installation, but without the hard disk partitioning. Another difference is that existing configuration files in `/etc` are backed up and merged with the new files. Getting the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for how to do this. Also, some sanity checks are done, i.e. file systems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 1.5 system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in /dev. If you've changed the contents of /dev by hand, you will need to be careful about this, but if not, you can just cd into /dev, and run the command

```
# sh MAKEDEV all
```

You must also deal with certain changes in the formats of some of the configuration files. The most notable change is that the options given to many of the file systems in /etc/fstab have changed, and some of the file systems have changed names. To find out what the new options are, it's suggested that you read the manual page for the file system's mount commands, for example mount_nfs(8) for NFS.

Finally, you will want to delete old binaries that were part of the version of NetBSD that you upgraded from and have since been removed from the NetBSD distribution.

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 1.5.

General issues

- /etc/rc modified to use /etc/rc.d/*

  In previous releases of NetBSD, /etc/rc was a traditional BSD style monolithic file. As of NetBSD 1.5, each discrete program or substem from /etc/rc and /etc/netstart has been moved into separate scripts in /etc/rc.d/.

  At system startup, /etc/rc uses rcorder(8) to build a dependency list of the files in /etc/rc.d and then executes each script in turn with an argument of 'start'. Many rc.d scripts won't start unless the appropriate rc.conf(5) entry in /etc/rc.conf is set to 'YES.'

  At system shutdown, /etc/rc.shutdown uses rcorder(8) to build a dependency list of the files in /etc/rc.d that have a "KEYWORD: shutdown" line, reverses the resulting list, and then executes each script in turn with an argument of 'stop'. The following scripts support a specific shutdown method: cron, inetd, local, and xdm.

  Local and third-party scripts may be installed into /etc/rc.d as necessary. Refer to the other scripts in that directory and rc(8) for more information on implementing rc.d scripts.

Issues affecting an upgrading from NetBSD 1.4 or later

- named(8) leaks version information

  Previous releases of NetBSD disabled a feature of named(8) where the version number of the server could be determined by remote clients. This feature has not been disabled in NetBSD 1.5, because there is a named.conf(5) option to change the version string:

  ```
  option {
          version "newstring";
  };
  ```

- sysctl(8) pathname changed

  sysctl(8) is moved from /usr/sbin/sysctl to /sbin/sysctl. If you have hardcoded references to the full pathname (in shell scripts, for example) please be sure to update those.

- sendmail(8) configuration file pathname changed

Due to sendmail(8) upgrade from 8.9.x to 8.10.x, /etc/sendmail.cf is moved to /etc/mail/sendmail.cf. Also, the default sendmail.cf(5) refers different pathnames than before. For example, /etc/aliases is now located at /etc/mail/aliases, /etc/sendmail.cw is now called /etc/mail/local-host-names, and so forth. If you have customized sendmail.cf(5) and friends, you will need to move the files to the new locations. See /usr/share/sendmail/README for more information.

Using online NetBSD documentation

Documentation is available if you first install the manual distribution set. Traditionally, the ''man pages'' (documentation) are denoted by 'name(section)'. Some examples of this are

- intro(1),
- man(1),
- apropros(1),
- passwd(1), and
- passwd(5).

The section numbers group the topics into several categories, but three are of primary interest: user commands are in section 1, file formats are in section 5, and administrative information is in section 8.

The *man* command is used to view the documentation on a topic, and is started by entering **man** [*section*] *topic*. The brackets [] around the section should not be entered, but rather indicate that the section is optional. If you don't ask for a particular section, the topic with the lowest numbered section name will be displayed. For instance, after logging in, enter

```
# man passwd
```

to read the documentation for passwd(1). To view the documentation for passwd(5), enter

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter **apropos** *subject-word*

where *subject-word* is your topic of interest; a list of possibly related man pages will be displayed.

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at majordomo@netbsd.org. To get help on using the mailing list server, send mail to that address with an empty body, and it will reply with instructions.

There are various mailing lists set up to deal with comments and questions about this release. Please send comments to: netbsd-comments@netbsd.org.

To report bugs, use the send-pr(1) command shipped with NetBSD, and fill in as much information about the problem as you can. Good bug reports include lots of details. Additionally, bug reports can be sent by mail to: netbsd-bugs@netbsd.org.

Use of send-pr(1) is encouraged, however, because bugs reported with it are entered into the NetBSD bugs database, and thus can't slip through the cracks.

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **http://www.netbsd.org/MailingLists/**. If you're interested in doing a serious amount of work on a specific port, you probably should contact the 'owner' of that port (listed below).

If you'd like to help with this effort, and have an idea as to how you could be useful, send us mail or subscribe to: netbsd-help@netbsd.org.

As a favor, please avoid mailing huge documents or files to these mailing lists. Instead, put the material you would have sent up for FTP or WWW somewhere, then mail the appropriate list about it, or, if you'd rather not do that, mail the list saying you'll send the data to those who want it.

Thanks go to

- The former members of UCB's Computer Systems Research Group, including (but not limited to):

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

  for their ongoing work on BSD systems, support, and encouragement.
- Also, our thanks go to:

  Mike Hibler
  Rick Macklem
  Jan-Simon Pendry
  Chris Torek

  for answering lots of questions, fixing bugs, and doing the various work they've done.
- UC Berkeley's Experimental Computing Facility provided a home for sun-lamp in the past, people to look after it, and a sense of humor. Rob Robertson, too, has added his unique sense of humor to things, and for a long time provided the primary FTP site for NetBSD.
- Vixie Enterprises for hosting the NetBSD FTP, SUP, and WWW servers.
- Redback Networks, Inc. for hosting the NetBSD mail and GNATS server.
- The Helsinki University of Technology in Finland for hosting the NetBSD CVS server.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.
- The many organisations that provide NetBSD mirror sites.
- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.
- Dave Burgess `burgess@cynjut.infonet.net` has been maintaining the 386BSD/NetBSD/FreeBSD FAQ for quite some time, and deserves to be recognized for it.
- The following individuals and organizations (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Steve Allen
  Jason Birnschein
  Mason Loring Bliss
  Jason Brazile
  Mark Brinicombe
  David Brownlee
  Simon Burge
  Dave Burgess
  Ralph Campbell
  Brian Carlstrom
  James Chacon
  Bill Coldwell

Charles Conn
Tom Coulter
Charles D. Cranor
Christopher G. Demetriou
Scott Ellis
Hubert Feyrer
Castor Fu
Greg Gingerich
William Gnadt
Michael Graff
Guenther Grau
Ross Harvey
Charles M. Hannum
Michael L. Hitch
Kenneth Alan Hornstein
Jordan K. Hubbard
Søren Jørvang
Scott Kaplan
Noah M. Keiserman
Harald Koerfgen
John Kohl
Chris Legrow
Ted Lemon
Norman R. McBride
Neil J. McRae
Perry E. Metzger
Toru Nishimura
Herb Peyerl
Mike Price
Dave Rand
Michael Richardson
Heiko W. Rupp
Brad Salai
Chuck Silvers
Thor Lancelot Simon
Bill Sommerfeld
Paul Southworth
Eric and Rosemary Spahr
Ted Spradley
Kimmo Suominen
Jason R. Thorpe
Steve Wadlow
Krister Walfridsson
Jim Wise
Christos Zoulas

AboveNet Communications, Inc.
Advanced System Products, Inc.
Avalon Computer Systems
Bay Area Internet Solutions
Brains Corporation, Japan
Canada Connect Corporation

Co-operative Research Centre for Enterprise Distributed Systems Technology
Demon Internet, UK
Digital Equipment Corporation
Distributed Processing Technology
Easynet, UK
Free Hardware Foundation
Innovation Development Enterprises of America
Internet Software Consortium
MS Macro System GmbH, Germany
Numerical Aerospace Simulation Facility, NASA Ames Research Center
Piermont Information Systems Inc.
Salient Systems Inc.
VMC Harald Frank, Germany
Warped Communications, Inc.
Whitecross Database Systems Ltd.

(If you're not on that list and should be, tell us! We probably were not able to get in touch with you, to verify that you wanted to be listed.)

- Finally, we thank all of the people who've put sweat and tears into developing NetBSD since its inception in January, 1993. (Obviously, there are a lot more people who deserve thanks here. If you're one of them, and would like to mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

Alistair Crooks	`agc@netbsd.org`
Jun-ichiro itojun Hagino	`itojun@netbsd.org`
Frank van der Linden	`fvdl@netbsd.org`
Luke Mewburn	`lukem@netbsd.org`
Christos Zoulas	`christos@netbsd.org`

The portmasters (and their ports):

Mark Brinicombe	`mark@netbsd.org`	**arm32**
Jeremy Cooper	`jeremy@netbsd.org`	**sun3x**
Ross Harvey	`ross@netbsd.org`	**alpha**
Jun-ichiro itojun Hagino	`itojun@netbsd.org`	**sh3**
Ben Harris	`bjh21@netbsd.org`	**arm26**
Eduardo Horvath	`eeh@netbsd.org`	**sparc64**
Darrin Jewell	`dbj@netbsd.org`	**next68k**
Søren Jørvang	`soren@netbsd.org`	**cobalt**
Søren Jørvang	`soren@netbsd.org`	**sgimips**
Wayne Knowles	`wdk@netbsd.org`	**mipsco**
Paul Kranenburg	`pk@netbsd.org`	**sparc**
Anders Magnusson	`ragge@netbsd.org`	**vax**
Minoura Makoto	`minoura@netbsd.org`	**x68k**
Phil Nelson	`phil@netbsd.org`	**pc532**
Tohru Nishimura	`nisimura@netbsd.org`	**luna68k**

NONAKA Kimihiro	nonaka@netbsd.org	**prep**
Scott Reynolds	scottr@netbsd.org	**mac68k**
Kazuki Sakamoto	sakamoto@netbsd.org	**bebox**
Noriyuki Soda	soda@netbsd.org	**arc**
Wolfgang Solfrank	ws@netbsd.org	**ofppc**
Ignatios Souvatzis	is@netbsd.org	**amiga**
Jonathan Stone	jonathan@netbsd.org	**pmax**
Shin Takemura	takemura@netbsd.org	**hpcmips**
Jason Thorpe	thorpej@netbsd.org	**alpha**
Jason Thorpe	thorpej@netbsd.org	**hp300**
Tsubai Masanari	tsubai@netbsd.org	**macppc**
Tsubai Masanari	tsubai@netbsd.org	**newsmips**
Izumi Tsutsui	tsutsui@netbsd.org	**news68k**
Frank van der Linden	fvdl@netbsd.org	**i386**
Leo Weppelman	leo@netbsd.org	**atari**
Nathan Williams	nathanw@netbsd.org	**sun3**
Steve Woodford	scw@netbsd.org	**mvme68k**

The NetBSD 1.5 Release Engineering team:

Chris G. Demetriou	cgd@netbsd.org
Havard Eidnes	he@netbsd.org
Ted Lemon	mellon@netbsd.org
John Hawkinson	jhawk@netbsd.org
Perry Metzger	perry@netbsd.org
Curt Sampson	cjs@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Todd Vierling	tv@netbsd.org

Developers and other contributors:

Steve Allen	wormey@netbsd.org
Julian Assange	proff@netbsd.org
Lennart Augustsson	augustss@netbsd.org
Christoph Badura	bad@netbsd.org
Robert V. Baron	rvb@netbsd.org
Erik Berls	cyber@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org
Allen Briggs	briggs@netbsd.org
Aaron Brown	abrown@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@netbsd.org
Simon Burge	simonb@netbsd.org
Dave Burgess	burgess@cynjut.infonet.net
Robert Byrnes	byrnes@netbsd.org

D'Arcy J.M. Cain	darcy@netbsd.org
Dave Carrel	carrel@netbsd.org
James Chacon	jmc@netbsd.org
Bill Coldwell	billc@netbsd.org
Julian Coleman	jdc@netbsd.org
Chuck Cranor	chuck@netbsd.org
Aidan Cully	aidan@netbsd.org
Johan Danielsson	joda@netbsd.org
Matt DeBergalis	deberg@netbsd.org
Rob Deker	deker@netbsd.org
Chris G. Demetriou	cgd@netbsd.org
Jaromir Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Havard Eidnes	he@netbsd.org
Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Thorsten Frueauf	frueauf@netbsd.org
Castor Fu	castor@netbsd.org
Ichiro Fukuhara	ichiro@netbsd.org
Brian R. Gaeke	brg@dgate.org
Thomas Gerner	thomas@netbsd.org
Simon J. Gerraty	sjg@netbsd.org
Justin Gibbs	gibbs@netbsd.org
Adam Glass	glass@netbsd.org
Michael Graff	explorer@netbsd.org
Brad Grantham	grantham@tenon.com
Brian C. Grayson	bgrayson@netbsd.org
Matthew Green	mrg@netbsd.org
Juergen Hannken-Illjes	hannken@netbsd.org
Charles M. Hannum	mycroft@netbsd.org
Eric Haszlakiewicz	erh@netbsd.org
John Hawkinson	jhawk@netbsd.org
HAYAKAWA Koichi	haya@netbsd.org
René Hexel	rh@netbsd.org
Michael L. Hitch	mhitch@netbsd.org
Christian E. Hopps	chopps@netbsd.org
Ken Hornstein	kenh@netbsd.org
Marc Horowitz	marc@netbsd.org
Nick Hudson	skrll@netbsd.org
Martin Husemann	martin@netbsd.org
Dean Huxley	dean@netbsd.org
Bernardo Innocenti	bernie@netbsd.org
ITOH Yasufumi	itohy@netbsd.org
IWAMOTO Toshihiro	toshii@netbsd.org

Matthew Jacob	mjacob@netbsd.org
Lonhyn T. Jasinskyj	lonhyn@netbsd.org
Chris Jones	cjones@netbsd.org
Takahiro Kambe	taca@netbsd.org
Antti Kantee	pooka@netbsd.org
Lawrence Kesteloot	kesteloo@cs.unc.edu
Thomas Klausner	wiz@netbsd.org
Klaus Klein	kleink@netbsd.org
Wayne Knowles	wdk@netbsd.org
John Kohl	jtk@netbsd.org
Kevin Lahey	kml@netbsd.org
Johnny C. Lam	jlam@netbsd.org
Martin J. Laubach	mjl@netbsd.org
Ted Lemon	mellon@netbsd.org
Joel Lindholm	joel@netbsd.org
Mike Long	mikel@netbsd.org
Warner Losh	imp@netbsd.org
Federico Lupi	federico@netbsd.org
Brett Lymn	blymn@netbsd.org
Paul Mackerras	paulus@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Neil J. McRae	neil@netbsd.org
Perry Metzger	perry@netbsd.org
der Mouse	mouse@netbsd.org
Joseph Myers	jsm@netbsd.org
Ken Nakata	kenn@netbsd.org
Bob Nestor	rnestor@netbsd.org
NONAKA Kimihiro	nonaka@netbsd.org
Masaru Oki	oki@netbsd.org
Atsushi Onoe	onoe@netbsd.org
Greg Oster	oster@netbsd.org
Herb Peyerl	hpeyerl@netbsd.org
Matthias Pfaller	matthias@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Waldi Ravens	waldi@moacs.indiv.nl.net
Darren Reed	darrenr@netbsd.org
Michael Richardson	mcr@netbsd.org
Tim Rightnour	garbled@netbsd.org
Gordon Ross	gwr@netbsd.org
Heiko W. Rupp	hwr@netbsd.org
SAITOH Masanobu	msaitoh@netbsd.org
Curt Sampson	cjs@netbsd.org
Wilfredo Sanchez	wsanchez@netbsd.org
Ty Sarna	tsarna@netbsd.org
SATO Kazumi	sato@netbsd.org
Matthias Scheler	tron@netbsd.org

Karl Schilke (rAT)	rat@netbsd.org
Konrad Schroder	perseant@netbsd.org
Reed Shadgett	dent@netbsd.org
Tim Shepard	shep@netbsd.org
Takao Shinohara	shin@netbsd.org
Takuya SHIOZAKI	tshiozak@netbsd.org
Chuck Silvers	chs@netbsd.org
Thor Lancelot Simon	tls@netbsd.org
Jeff Smith	jeffs@netbsd.org
Bill Sommerfeld	sommerfeld@netbsd.org
Bill Squier	groo@netbsd.org
Bill Studenmund	wrstuden@netbsd.org
Kevin Sullivan	sullivan@netbsd.org
SUNAGAWA Keiki	kei@netbsd.org
Kimmo Suominen	kim@netbsd.org
Matt Thomas	matt@netbsd.org
Christoph Toshok	toshok@netbsd.org
UCHIYAMA Yasushi	uch@netbsd.org
Shuichiro URATA	ur@netbsd.org
Todd Vierling	tv@netbsd.org
Aymeric Vincent	aymeric@netbsd.org
Paul Vixie	vixie@netbsd.org
Krister Walfridsson	kristerw@netbsd.org
Lex Wennmacher	wennmach@netbsd.org
Assar Westerlund	assar@netbsd.org
Todd Whitesel	toddpw@netbsd.org
Rob Windsor	windsor@netbsd.org
Dan Winship	danw@netbsd.org
Jim Wise	jwise@netbsd.org
Michael Wolfson	mbw@netbsd.org
Colin Wood	ender@netbsd.org

Legal Mumbo-Jumbo

All product names mentioned herein are trademarks or registered trademarks of their respective owners.

The following notices are required to satisfy the license terms of the software that we have mentioned in this document:

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by The NetBSD Foundation, Inc.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.

This product includes software developed by Adam Glass and Charles Hannum.

This product includes software developed by Adam Glass and Charles M. Hannum.

This product includes software developed by Adam Glass.

This product includes software developed by Alistair G. Crooks.

This product includes software developed by Amancio Hasty and Roger Hardiman.

This product includes software developed by Berkeley Software Design, Inc.

This product includes software developed by Bill Paul.

This product includes software developed by Charles D. Cranor and Washington University.

This product includes software developed by Charles D. Cranor.

This product includes software developed by Charles Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.

This product includes software developed by Charles Hannum.

This product includes software developed by Charles M. Hannum.

This product includes software developed by Chris Provenzano.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Christopher G. Demetriou.

This product includes software developed by Christos Zoulas.

This product includes software developed by David Jones and Gordon Ross.

This product includes software developed by Dean Huxley.

This product includes software developed by Eric S. Hvozda.

This product includes software developed by Ezra Story.

This product includes software developed by Gardner Buchanan.

This product includes software developed by Gordon Ross.

This product includes software developed by Gordon W. Ross and Leo Weppelman.

This product includes software developed by Gordon W. Ross.

This product includes software developed by Hauke Fath.

This product includes software developed by HAYAKAWA Koichi.

This product includes software developed by Hellmuth Michaelis and Joerg Wunsch.

This product includes software developed by Herb Peyerl.

This product includes software developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems.

This product includes software developed by Hubert Feyrer for the NetBSD Project.

This product includes software developed by Iain Hibbert.

This product includes software developed by Ian W. Dall.

This product includes software developed by Ignatios Souvatzis for the NetBSD Project.

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/.

This product includes software developed by Joachim Koenig-Baltes.

This product includes software developed by Jochen Pohl for The NetBSD Project.

This product includes software developed by John Polstra.

This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.

This product includes software developed by Jonathan Stone for the NetBSD Project.

This product includes software developed by Jonathan Stone.

This product includes software developed by Julian Highfield.

This product includes software developed by Kenneth Stailey.

This product includes software developed by Leo Weppelman.

This product includes software developed by Lloyd Parkes.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Marc Horowitz.

This product includes software developed by Mark Brinicombe.

This product includes software developed by Mark Tinguely and Jim Lowe.

This product includes software developed by Markus Wild.

This product includes software developed by Martin Husemann and Wolfgang Solfrank.

This product includes software developed by Mats O Jansson and Charles D. Cranor.

This product includes software developed by Mats O Jansson.

This product includes software developed by Matthias Pfaller.

This product includes software developed by Michael L. Hitch.

This product includes software developed by Niels Provos.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Paul Mackerras.

This product includes software developed by Peter Galbavy.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Rodney W. Grimes.

This product includes software developed by Roland C. Dowdeswell.

This product includes software developed by Rolf Grossmann.

This product includes software developed by Scott Bartram.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed by Tatoku Ogaito for the NetBSD Project.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Theo de Raadt and John Brezak.

This product includes software developed by Theo de Raadt.

This product includes software developed by Tohru Nishimura for the NetBSD Project.

This product includes software developed by TooLs GmbH.

This product includes software developed by Winning Strategies, Inc.

This product includes software developed by Zembu Labs, Inc.

This product includes software developed by the Center for Software Science at the University of Utah.

This product includes software developed by the Computer Systems Laboratory at the University of Utah.

This product includes software developed by the University of Calgary Department of Computer Science and its contributors.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed for the FreeBSD project.

This product includes software developed for the Internet Software Consortium by Ted Lemon.

This product includes software developed for the NetBSD Project by Frank van der Linden.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed for the NetBSD Project by Matthias Drochner.

This product includes software developed for the NetBSD Project by Matthieu Herrb.

This product includes software developed for the NetBSD Project by Perry E. Metzger.

This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.

This product includes software developed for the NetBSD Project by Ted Lemon.

This product includes software developed by LAN Media Corporation and its contributors.

This product includes software developed by Michael Graff for the NetBSD Project.

This product includes software developed by Niklas Hallqvist, C Stone and Job de Haas.

This product includes software developed by Eric Young (eay@mincom.oz.au).

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/).

This product includes software developed by the University of Oregon.

This product includes software developed by the University of Southern California and/or Information Sciences Institute.

This product includes software developed by Internet Initiative Japan Inc.

This product includes software developed by Allen K. Briggs.

This product includes software developed by Apple Computer, Inc.

This product includes software developed by Bill Studenmund.

This product includes software developed by Bradley A. Grantham.

This product includes software developed by Carnegie-Mellon University.

This product includes software developed by Chris P. Cputo.

This product includes software developed by Colin Wood.

This product includes software developed by David Huang.

This product includes software developed by Internet Research Institute, Inc.

This product includes software developed by John P. Wittkoski.

This product includes software developed by Open Software Foundation, Inc.

This product includes software developed by Per Fogelstrom, Opsycon AB and RTMX Inc, USA.

This product includes software developed by Takashi Hamada.

This product includes software developed by Tsubai Masanari.

This product includes software developed by University of Utah.

This product includes software developed by Wolfgang Solfrank.