NAME

INSTALL – Installation procedure for NetBSD/macppc.

CONTENTS

About this Document...3
Quick install notes for the impatient...................................3
What is NetBSD?...4
Upgrade path to NetBSD 1.5.3..4
Major Changes Between 1.5.2 and 1.5.3...................................5
Known Caveats with 1.5.3..6
Major Changes Between 1.5.1 and 1.5.2...................................6
Major Changes Between 1.5 and 1.5.1.....................................6
Changes Between The NetBSD 1.4 and 1.5 Releases.........................8
  Kernel 9
  Networking 9
  File system 9
  Security 9
  System administration and user tools 10
  Miscellaneous 10
The Future of NetBSD..11
Sources of NetBSD...11
NetBSD 1.5.3 Release Contents...12
  NetBSD/macppc subdirectory structure 13
  Binary distribution sets 13
NetBSD/macppc System Requirements and Supported Devices..................15
  Supported models 15
  Unsupported models 16
  Supported devices 17
  Unsupported devices 19
  Supported boot devices and media 19
Getting the NetBSD System on to Useful Media............................19
Preparing your System for NetBSD installation...........................20
  Prepare yourself 20
  Updating your BootROM 22
  Getting to the Open Firmware Prompt on Newer Models 22
  Getting to the Open Firmware Prompt on Apple Network 23
  Getting to the Open Firmware Prompt on Older Models 23
  Older Open Firmware System Preparation 24
  Getting to the Open Firmware Prompt (Recommended method using 24
  Getting to the Open Firmware Prompt (Harder method using 25
  Getting to the Open Firmware Prompt (Without using MacOS) 26
  Setting up pre-Open Firmware 3 to boot NetBSD 27
  Setting up Open Firmware 3 to boot NetBSD 28
  Available Boot Media 29
  Description of bootable media 31
  Partitioning your hard drive for NetBSD 32
  Preparing the Bootable Media 34
    Booting off an MS-DOS floppy..35
    Booting off the NetBSD install floppy...............................36

Booting off an HFS or HFS+ partition....................................37
Booting off the NetBSD/macppc CD-R image................................37
Booting off an ISO9660 CD-ROM...37
Booting off a hybrid HFS/ISO9660 CD-ROM.................................38
Booting over the ethernet...39
Booting the floppy image off a SCSI or IDE drive........................42
Installing the NetBSD System..42
Open Firmware boot syntax 43
Examples of Open Firmware boot commands 46
Booting off an MS-DOS floppy..46
Booting off the NetBSD install floppy...................................46
Booting off the NetBSD/macppc install CD-R..............................47
Booting off the NetBSD/macppc install CD-R..............................48
Booting off a home-made CD-ROM ISO9660 or Hybrid........................48
Booting off an IDE or SCSI drive with 'partition zero'..................48
Booting off an IDE or SCSI drive with an HFS partition..................49
Example of a normal boot 49
Common Problems and Error Messages 50
Black screen..50
Grey screen with flashing question mark.................................51
Information on your screen seems garbled or out of sync.................51
DEFAULT CATCH!..51
CLAIM failed..51
can't OPEN..52
unrecognized Client Program formatstate not valid.......................52
bad partition number, using 0no bootable HFS partition..................52
TFTP timeout..52
Bootloader hangs before the copyright notice............................52
Hang after configuring devices..53
Milestone 53
Running the sysinst installation program 53
Introduction..53
General...53
Quick install...54
Booting NetBSD..55
Network configuration...55
Preparing a disk which will be used for MacOS and NetBSD................55
Installation drive selection and parameters.............................57
Partitioning the disk..57
Preparing your hard disk..58
Getting the distribution sets..58
Installation using ftp..58
Installation using NFS..59
Installation from CD-ROM..59
Installation from an unmounted file system..............................59
Installation from a local directory.....................................59
Extracting the distribution sets..60
Making the device nodes...60
Finalizing your installation..60
Post installation steps...61
Upgrading a previously-installed NetBSD System..............................64
Compatibility Issues With Previous NetBSD Releases..........................65

General issues 65
Issues affecting an upgrade from NetBSD 1.4 or prior 65
Using online NetBSD documentation.. 66
Administrivia.. 66
Thanks go to... 67
We are... 69
Legal Mumbo-Jumbo.. 74
The End.. 76

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5.3 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.

Quick install notes for the impatient

This section contains some brief notes describing what you need to install NetBSD 1.5.3 on a machine of the macppc architecture.

- Fetch the bootloader `installation/ofwboot.xcf` and the installation kernel `installation/netbsd.ram.gz`. Alternatively, you can fetch the CD image, `macppc_bootable.iso` or the boot floppy image `installation/boot.fs` which include the bootloader and installation kernel. You only need the installation kernel and bootloader, or the CD image to boot your system. Alternatively, you may netboot the installation kernel, which requires several local netboot services. The details are not covered here, as setting up a netboot server is hardly "quick".
- The actual binary distribution is in the `binary/sets/` directory. When you boot the install kernel from floppy, hard drive, or CD-ROM, the installation program can fetch these files for you (using e.g. ftp), if you have a network connection. There are several other methods to get the binary sets onto your machine.

  You will at a minimum need the following sets: `kern.tgz`, `base.tgz` and `etc.tgz`. In a typical workstation installation you will probably want all the installation sets.

- If your mac has a floppy drive, copy `ofwboot.xcf` and `netbsd.ram.gz` to an MS-DOS formatted floppy disk and rename `netbsd.ram.gz` to `netbsd.gz`. Otherwise, drag `ofwboot.xcf` and `netbsd.ram.gz` to your hard drive icon (the top level of the drive, not the desktop). If you are making a CD, burn it now.

  The disk(s) you just prepared will be used to boot the installation kernel, which contains all the tools required to install NetBSD.

- Determine your machine's model, quirks, and Open Firmware version from the NetBSD/macppc Model Support webpage.

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

  At present, NetBSD/macppc cannot exist on the same hard drive as MacOS unless you partition your disk using a MacOS partitioning utility. Open Firmware versions prior to 3 cannot boot into NetBSD on a drive partitioned this way -- you must use the entire disk, partitioned with the installation tools. Open Firmware version 3 cannot boot into NetBSD on a drive partitioned with the installation tools, you must use a MacOS partitioning utility and the "**Re-install sets or install additional sets**" option in the installer (selecting the "**Install NetBSD to hard disk**" or "**Upgrade NetBSD on a hard disk**" options will render your drive unbootable). If you are unsure, you may want to read the section below on **Partitioning your hard drive for NetBSD**

- For systems with Open Firmware versions prior to 3, you may need to use Apple's System Disk utility to enter Open Firmware and use your screen and keyboard. To enter Open Firmware, hold down the `COMMAND-OPTION-O-F` keys after the boot chime starts, but before it ends. Entering Open Firmware versions prior to 3 is usually the most frustrating part of installation -- you may want to read the section below on **Older Open Firmware System Preparation**

  You should have the Open Firmware "`0 >`" prompt on your screen before attempting to boot NetBSD/macppc.

- At the Open Firmware prompt, type the command to boot. To boot from a floppy, the command is "**`boot fd:,\ofwboot.xcf netbsd.gz`**". For the install kernel and bootloader on your hard drive (Open Firmware 3), the command is "**`boot hd:\,ofwboot.xcf netbsd.ram.gz`**".

  For boot CDs, the command is something like "**`boot cd:,\ofwboot.xcf netbsd.macppc`**" (for Open Firmware 3) or "**`boot scsi-int/sd@3:0 NETBSD.MACPPC`**" (for earlier Open Firmware versions). You will need to use the correct case for **`OFWBOOT.XCF`** and **`NETBSD.MACPPC`** depending on how your version of Open Firmware interprets the ISO filesystem. You may need to replace **`cd`** with **`scsi/sd@3`**, **`scsi-int/sd@3`**, **`scsi-ext/sd@3`**, **`ata/atapi-disk`**, or some other device alias. You should also use the Open Firmware **`dir`** command to confirm that the NetBSD/macppc kernel is called `NETBSD.MACPPC`.

- For third-party programs which are not part of the base NetBSD distribution, you will want to explore the **`pkgsrc`** system with its more than 2100 program packages.

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 1.5.3 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.

Upgrade path to NetBSD 1.5.3

If you are not installing your system "from scratch" but instead are going to upgrade an existing system already running NetBSD you need to know which versions you can upgrade with NetBSD 1.5.3.

NetBSD 1.5.3 is an upgrade of NetBSD 1.5, 1.5.1, 1.5.2, and earlier major and maintenance releases of NetBSD.

The intermediate development versions of code available on the main trunk in our CVS repository (also known as "NetBSD-current") from *after* the point where the release cycle for 1.5 was started are designated by version identifiers such as 1.5A, 1.5B, etc. These identifiers do not designate releases, but indicate major changes in internal kernel APIs. Note that the kernel from NetBSD 1.5.3 can *not* be used to upgrade a system running one of those intermediate development versions. Trying to use the NetBSD 1.5.3 kernel on such a system *will* probably result in problems.

Please also note that it is not possible to do a direct "version" comparison between any of the intermediate development versions mentioned above and 1.5.3 to determine if a given feature is present or absent in 1.5.3. The development of 1.5 and the subsequent maintenance releases is done on a separate branch in the CVS repository. The branch was created when the release cycle for 1.5 was started, and during the release cycle of 1.5 and its maintenance releases, selected fixes and enhancements have been imported from the main development trunk. So, there are features in 1.5.3 which were not in, e.g. 1.5B, and vice versa.

Major Changes Between 1.5.2 and 1.5.3

Some highlights are:

- `lpd`(8) has been fixed to deal with a security issue (SA2001-018).
- Drivers for Mylex DAC960, Compaq EISA array controllers, and I2O block/SCSI devices added. This has caused `ca`(4) to be replaced by `ld`(4). Drivers for the Mylex DAC960 management interface and DPT/Adaptec I2O RAID management interface has also been added. See `dpti`(4) and `mlxctl`(8).
- A driver for the 3ware Escalade 5000 and 6000 series RAID controllers has been added, see `twe`(4).
- Various different fixes have been applied to the network device drivers `ep`(4), `ex`(4), `rtk`(4), `sip`(4), `ti`(4), `tl`(4), and `wi`(4).
- Support for some more variants of `rtk`(4) on CardBus has been added.
- The `ne`(4) driver has been extended to support some more pcmcia cards.
- Support for more `pciide`(4) controllers added: HPT-370A, Acard ATP-850/860, and AMD-768. UltraDMA 100 support added for CMD 0649.
- A problem with the NFS server code, exposed on NetBSD/alpha, where use of ".." would return info for "." has been fixed.
- Support for "other-endian" file systems has been improved.
- The layout algorithm for FFS file systems has been substantially improved, resulting in better performance, both due to improved locality between files and their corresponding directory, and lessened pressure on the buffer cache.
- The performance of soft dependencies in FFS has been significantly improved in some circumstances.
- Handling of init/fini section support and DWARF2 exception handling has been added to the C runtime startup code.
- OpenSSH has been upgraded to version 3.0.2.

- A security issue arising from a race between set-uid execution and use of ptrace has been fixed (SA2002-001).
- The package tools has been extended to provide for optional digital signatures on binary packages. Additionally, binary package version number handling has been rewritten.
- A driver for Creative Labs SBLive! EMU10000 has been added; see `emuxki`(4).
- A driver for ESS Allegro-1 / Maestro-3 has been added; see `esa`(4).
- Support for the 53c1010-33 and 53c1510D has been added to the `siop`(4) driver.
- IPFilter upgraded to 3.4.23.
- The boot code on NetBSD/macppc has been improved.
- `sendmail`(8) has been upgraded to version 8.11.6.
- A long file name buffer overrun in gzip has been fixed (SA2002-002).
- IPSEC policy check has been fixed for forwarded IPv4 packets (SA2002-003).
- Two security problems in OpenSSH has been fixed (SA2002-004, SA2002-005).
- A buffer overrun in the resolver part of the C library has been fixed (SA2002-006). To fix this problem in the BIND user utilities, BIND has been upgraded to version 8.3.3.
- Various other kernel stability and robustness fixes has been applied.

In addition, many bugs have been fixed—more than 70 problems reported through our problem tracking system have been fixed, and several other non-reported problems have also been found and fixed. See the CHANGES-1.5.3 file for the complete list.

Known Caveats with 1.5.3

The following are the major known issues with NetBSD 1.5.3.

- The upgrade of IPFilter to version 3.4.23 is not backward compatible. This means that the user-land tools used to manage IPFilter will have to be upgraded when the kernel is upgraded.
- Due to xsrc changes which are not backward compatible on the arm32 port, the X bits for arm32 is built from the 1.5.2 sources.

Major Changes Between 1.5.1 and 1.5.2

The main purpose of NetBSD 1.5.2 is to fix some problems discovered shortly after the release of NetBSD 1.5.1. Some highlights are:

- `telnetd`(8) has been fixed to deal with a security issue (SA2001-012).
- A weakness in the OpenSSL libcrypto's pseudo random number generator has been fixed (SA2001-013).
- Floating point emulation on i386 (to run on FPU-less CPUs) which was broken late in 1.5.1's cycle has been restored to working order.
- Bugs in the compiler have been worked around to produce a working miniroot image on sparc64.
- The script-based installers on amiga, atari and mvme68k have been updated to deal with new behaviour from `mount`(8).
- `dump`(8) has been fixed to deal with a local security issue (SA2001-014).

- Several system calls have had defensive length checks applied on their arguments (SA2001-015).
- The DHCP software has been upgraded to ISC Version 3, Release Candidate 10.

Additionally, a few more bugs have been fixed. See the CHANGES-1.5.2 file for the complete list of changes.

Major Changes Between 1.5 and 1.5.1

The complete list of changes between NetBSD 1.5 and 1.5.1 can be found in the file CHANGES-1.5.1 in the top directory of the source tree. The following are highlights only:

- A driver for the Aironet/Cisco wireless PCMCIA cards has been added; see `an`(4).
- NFS client performance has been improved, typically by 40% for writes but possibly up to 100% in certain setups.
- The `siop`(4) driver has improved in performance and robustness.
- Support for cloning pseudo-interfaces has been added. See `ifconfig`(8).
- Support for 802.1Q virtual LANs has been added. See `vlan`(4).
- The `isp`(4) driver has been upgraded to (among other things) work on MacPPC.
- BIND has been upgraded to version 8.2.3 (SA2001-001).
- Support for booting from RAIDframe RAID1 mirrors on i386 added.
- The `lfs`(4) file system has again been substantially updated, but is still experimental.
- Ultra/66 support has been added for capable VIA chipsets, and Ultra/100 support has been added for the HPT370, Promise and Intel ICH2 controllers in the `pciide`(4) driver. Support for Intel 82801BAM controllers has also been added, and handling of Ali controllers has been improved.
- OpenSSH has been updated to deal with a security issue (SA2001-003).
- Sendmail has been upgraded to version 8.11.3.
- The `ex`(4) driver has added support for 3Com 3c555, 3c556 and 3c556B MiniPCI Ethernet cards.
- A driver for the on-board audio hardware found on many Apple PowerMacs has been added; see `awacs`(4).
- The `sip`(4) driver has been fixed to properly support the dp83815, as found in current Netgear FA311 10/100 cards.
- `ftpd`(8) has been updated to deal with two security issues (SA2000-018 and SA2001-005).
- `ntpd`(8) has been updated to deal with a security issue (SA2001-004).
- `telnetd`(8) has been updated to deal with a security issue (SA2000-017).
- A vulnerability on i386 related to USER_LDT has been fixed (SA2001-002).
- The Linux emulation has been enhanced to prepare for the support of using the Linux version of VMware.
- IP checksumming speed has been improved on i386 compared to NetBSD 1.5 by about 10%.
- Support for the Socket Communications LP-E Type II PCMCIA NE2000 clone card has been added to `ne`(4).

- The DHCP software has been upgraded to ISC version 3, Beta 2, patchlevel 24, to fix core dumps in `dhclient`(8), among other things. Please note that the new `dhcpd`(8) forces you to configure a "ddns-update-style" of either "ad-hoc", "interim" or "none".
- Various fixes and enhancements to INET6 and IPSEC code; among them improved interaction between IPF/Nat and IPSEC.
- The Heimdal `kerberos`(8) implementation has been upgraded to version 0.3e.
- Support for Accton EN2242 and other AmdTek AN985 cards added to the `tlp`(4) driver.
- Several country-specific keyboard mappings have been added for USB keyboards.
- A driver for Yamaha YMF724/740/744/745-based sound cards has been added, see `yds`(4).
- The maximum number of BSD disklabel partitions on the i386 port has been increased from 8 to 16.
- Drivers for the AC'97 based audio sound chips ESS Technology Maestro 1, 2, and 2E (see `esm`(4)), Neo-Magic 256 (see `neo`(4)), and Cirrus Logic CrystalClear PCI Audio CS4281 (see `clct`(4)) have been added.

In addition, many bugs have been fixed—more than 95 problems reported through our problem tracking system have been fixed, and some other non-reported problems have also been found and fixed. See the CHANGES-1.5.1 file for the complete list.

The NetBSD Packages Collection (pkgsrc) which is used to maintain, build, track dependencies, and maintain NetBSD-specific fixes to third-party programs, has received a major overhaul for NetBSD 1.5.1. Some highlights are:

- Many new packages were added to the collection, which now includes about 2100 packages. Many of them are also available as pre-compiled binaries on ftp.netbsd.org and its mirrors. Many packages have been modified and enhanced to compile and function properly on big-endian (m68k, sparc), and LP64 architectures (alpha, sparc64).
- KDE2 and KOffice together consitutes a fully integrated office environment with no license problems, available for i386, alpha and many other architectures.
- Mozilla 0.9, KDE2's Konqueror, and Links 0.95 are just a few examples of the web browsers available.
- A support package for running VMware on NetBSD/i386 was added, it's called suse_vmware. The official VMware code, a valid license, and Wasabi Systems' compatibility package are all needed to run VMware.
- Internal changes of the pkgsrc system include strong checksums to prevent malicious distribution files, as well as restructuring of the package structure in pkgsrc for faster extraction of pkgsrc tar files and upgrades via SUP and CVS.

Please note that at the moment, **sysinst** will not assist you in installing pre-built third-party binary packages or the pkgsrc system itself, so you will have to manually install packages using **pkg_install** or fetch and extract the pkgsrc.tgz tar file to get started.

Lastly, it should be noted that the X11 binaries shipped in NetBSD 1.5.3 is still based on XFree86 version 3.3.6. Several newer graphics cards are inadequately supported by that code base, but on the other hand support for several older graphics cards is not available in newer XFree86 code. NetBSD is in the process of moving to XFree86 version 4, and is currently maintaining both the XFree86 3.3.6 and the XFree86 version 4 code in the **xsrc** source set, and you may at compile time pick which sources to build and install. To ease installation, testing and use of the XFree86 version 4 code, a binary snapshot based on XFree86 version 4.0.3 will be made available for at least the i386 architecture shortly after the release of NetBSD 1.5.3.

Changes Between The NetBSD 1.4 and 1.5 Releases

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 1.5.3 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.
- The 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 bounds checking, and another one identified and disabled places where format strings were used in unsafe ways, allowing arbitrary data to be entered by (possibly) malicious users to overwrite application code, and leading from Denial of Service attacks to compromised systems.
- `sshd`(8) and `ssh`(1) now require `rnd`(4) kernel random number devices.

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.

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.

This is the second major release of NetBSD for the macppc platform, including the Apple iBook, iMac, PowerMacintosh, and PowerBook models, as well as clones.

As is usual between releases, the macppc port has had many improvements made to it - too many to detail all of them here.

Numerous new drivers have been added. See the supported hardware list for details.

Some (but not all!) notable macppc-specific improvements include:

- A simpler install procedure for models released after 1996, which have Open Firmware version 3 and only one hard drive. The procedure described in this document will no longer render your drive unbootable on these models.
- Migration to the ELF binary format from a.out. You can still use your pre-1.5 a.out applications, though.
- Support for the latest laptop models which previously required a special ADB keyboard patch (the iBook, iBook SE, iBook (Dual USB), PowerBook (FireWire), and the PowerBook G4 (Titanium)).
- Support for the latest iMac machines, the iMac (Early 2001).

NetBSD 1.5.3 on macppc is, as usual, also fully backward compatible with old NetBSD/macppc binaries, so you don't need to recompile all your local programs provided you set the appropriate binary compatibility options in your kernel configuration.

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 1.5.3 Release Contents

The root directory of the NetBSD 1.5.3 release is organized as follows:

`.../NetBSD-1.5.3/`

`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 1.5.3 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 1.5.3 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.
7.4 MB gzipped, 73.0 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 1.5.3 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
24.8 MB gzipped, 123.1 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.5.3 kernel for all architectures; `config`(8); and `dbsym`(8).
18.0 MB gzipped, 90.9 MB uncompressed

xsrc This set contains the sources to the X Window System.
78.1 MB gzipped, 393.6 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 1.5.3 release is found in the macppc subdirectory of the distribution: .../NetBSD-1.5.3/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 1.5.3 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 1.5.3 distribution tree, and are as follows:

base The NetBSD 1.5.3 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.
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 1.5.3 `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.
5.6 MB gzipped, 22.7 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
2.2 MB gzipped, 7.5 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. The binaries shipped with NetBSD 1.5.3 are based on XFree86 version 3.3.6. NetBSD is in the process of moving to XFree86 version 4, and the X source set actually contains source for both XFree86 3.3.6 and XFree86 4, and the ability to decide at compile-time which one to build and install. The X Window System binary sets distributed with NetBSD 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. It is part of the boot ROMs in most PowerPC-based Macintosh systems. Until late 1996, Apple never intended to use Open Firmware for anything other than internal debugging and hardware support. It was not intended to be used to boot an operating system. This is why earlier machines have so much trouble with Open Firmware. This also 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 1.0.5 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 16 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 16 MB of RAM is very slow. Note that until you have around 32 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

- Apple PowerMacintosh (7300, 7500 (with CPU upgrade), 7600, 8500, 8600, 9500, and 9600)
- Power Computing (PowerBase, PowerCenter, PowerCenter Pro, PowerCurve, PowerTower, PowerTower Pro, and PowerWave)
- UMAX (J700, S900)

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 2.0.x

- Apple PowerBook (2400, 3400, G3, and G3 Series)
- Apple PowerMacintosh/Performa (4400, 54xx, 5500, 6300/160, 6360, 6400, and 6500)
- 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)
- UMAX (Apus 2000, Apus 3000, C500, and C600)

Open Firmware 2.4

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

Open Firmware 3

- Apple original iBook, iBook SE, and iBook (Dual USB)
- Apple iMac; Bondi Blue (Rev A and Rev B, 233 MHz), 5 Flavors (Rev C and Rev D, 266 MHz and 333 MHz), iMac (Slot Loading), iMac (Summer 2000), and iMac (Early 2001)
- Apple PowerBook (G3 Series (bronze keyboard) and G3 (FireWire))
- Apple PowerBook G4 (Titanium)
- Apple PowerMacintosh G3 (Blue and White)
- Apple PowerMacintosh (G4 (PCI), G4 (AGP), G4 (Gigabit Ethernet), G4 (Digital Audio), and G4 Cube)

Note: The PowerBook (FireWire) and PowerBook G4 (Titanium) models that have BootROM version 4.1.8 are not supported by 1.5.3. You will need to use a kernel from the -current branch of NetBSD. See

http://www.netbsd.org/Documentation/kernel

Note: Some iMac (Bondi Blue) and PowerMacintosh G3 (Blue and White) machines have keyboard problems. There is a workaround for the PowerMacintosh described in the NetBSD/macppc FAQ. There is no known workaround for the iMac machines encountering this problem. See

http://www.netbsd.org/Ports/macppc/faq.html#usb-keyboard-problems

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 in the second half of 2001 and later (future releases of NetBSD/macppc will support these systems)
- Systems released before the second half of 2001, but not listed above. These models might work, but have not been tested as of the release of NetBSD 1.5.3.

Supported devices

- CPU upgrades
  – Most CPU upgrades are supported. Some older models will not automatically enable the L2 cache on a G3 or G4 processor. See the FAQ entry

    http://www.netbsd.org/Ports/macppc/faq.html#cpu-upgrade
- 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) (*de*)
  – 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*)
  – D-Link DFE-530TX+ (*rtk*)
  – RealTek 8139 (*rtk*)
  – Netgear FA-311 (*sip*)
  – Lite-On PNIC (*tlp*)
  – D-Link DFE-530TX (*vr*)
  – Many other PCI and Cardbus Ethernet interfaces, such as Tulip-compatible (*de* and *tlp*), 3Com (*ep*), SMC (*epic*), Intel (*fxp*), NE2000-compatible (*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*)

    Note: The 294x models are not bootable in Open Firmware, even though they can boot MacOS .
  – 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
- Promise Ultra66 (*pciide*) (this, and other PC-based cards are not bootable on macppc)
- Acard ATP-850/860 based IDE controlers (*pciide*) (including the Acard AEC-6210/6260 and the Sonnet Tempo ATA/66 cards)
- Some other PCI IDE controllers should work, although no one has had much success
- 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, PowerComputing PowerCenter and PowerTower, and UMAX C600 and Apus 3000. Also, the video card that shipped with the PowerMacintosh 9500/150, 9500/180, and 9500/200 systems is reported to have the same problem.

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

• Audio
- On-board audio on most models (*awacs*)

  Note: This driver has only recently been introduced and is considered "untested" meaning that it may cause your system to crash and/or is lacking some features you may expect

- USB audio devices (*uaudio*)
- PCI audio cards, although only the Ensoniq AudioPCI (*eap*) has been thoroughly tested

• Serial ports
- On-board serial ports (the modem and printer ports) (*ttya* and *ttyb*)
  ttya is the mini-DIN 8 serial port with the 'Modem' icon, and *ttyb* is has the 'Printer' icon. Some models with internal modems have the serial port with the 'Modem' icon covered over, and the modem is *ttya*.

  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

- 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/AirMac 802.11b wireless interface

  Note: This interface is supported in the -current branch of NetBSD.

- On-board video capture

Supported boot devices and media

Each version of Open Firmware supports different devices and media that you may boot from. We define the terms "bootable media" as the media (hard drive, floppy, CD-R, ethernet) that will be used to bootstrap your macppc system into NetBSD, and "distribution sets" or "distribution media" as the media (hard drive, CD-R, ethernet) that contains the files that will be installed to generate a working NetBSD system onto your destination media.

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

You should wait to decide where to put the NetBSD distribution sets until you have figured out how you are going to boot your system. Refer back to this section after you have done 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.

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 April, 2001).

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

Prepare yourself

Take a deep breath.

Good. Now, make sure you are reading the PDF (if available), PostScript, or HTML version of this document, as the `.txt` and `.more` versions lack important formatting information that will prevent you from following the twisted path you must follow. OK, good. Now, print out this document.

While it's printing, get some coffee, relax a bit, and mentally prepare yourself for something that promises to be confusing, frustrating, and annoying. If you assume the worst, you'll be pleasantly surprised when everything works easier than you expected. Also, forget everything you've been told about installing NetBSD/macppc. That's right, flush your knowledge cache -- some of it is almost certainly dirty.

Some of this document assumes familiarity with MacOS, e.g. how to download BinHex files and extract things with StuffIt Expander. If you've never heard of those terms before, it is possible to install NetBSD/macppc without booting or knowing how to use MacOS, but depending on your model it may be almost as painful as learning a little MacOS. See the FAQ for more help:

http://www.netbsd.org/Ports/macppc/faq.html#macos-newbie

It's done printing? Fine, now get a dark writing implement.

The recommended installation procedure is as follows:

1. Go to the NetBSD/macppc model support page and look up your model information and issues (I can't stress this enough times).
   http://www.netbsd.org/Ports/macppc/models.html
2. Spend about 15 minutes and read through this document from this point onward. While you're reading, use the writing implement to cross out the sections that do not apply to your model (or hilight the sections that do depending on your preferences).
3. Now spend another 5 minutes and re-read this document, and mentally piece together how you think you'll be installing onto your machine. It may help to take a blank piece of paper and write out an outline of the procedure you think you'll be going through.
4. Now create your bootable media and media for the distribution sets.
5. Prepare your machine, depending on the instructions for your model (for example: partition your drive, download and run System Disk, or hook up a serial console).
6. Boot into Open Firmware, and verify that it has the version of Open Firmware you think it does. Also make sure that the other variables are set correctly.
7. Figure out the correct boot command, and boot your machine from the bootable media you just created. If you're having trouble, be sure to read the section on **Common Problems and Error Messages**
8. Celebrate! The worst is over, but you've still got some work to do. Take a break, maybe more coffee, maybe a quick walk around the block, whatever turns your fancy.
9. Now use the installer to install the distribution sets onto your system and do some initial configurations.
10. Figure out how to boot from the installed partition. Boot into NetBSD for the first time.
11. Configure to your preferences, install your favorite packages, and have fun with your new NetBSD/macppc machine!

Note: You really actually truly do need to follow the procedure listed in this document in the order that we describe. These systems are rather tricky to boot for the novice and expert alike. Once you cross off the sections that don't apply to you, it will make more sense.

Note: If the instructions in a subsection below do not apply to *all* versions of Open Firmware, there will be a line listing which versions of Open Firmware they apply to, such as:
(Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4, Open Firmware 3)

Updating your BootROM

(Open Firmware 3)

Open Firmware 3 systems have a rewritable "firmware", also called the BootROM. When you use an Apple firmware updater, it updates the BootROM. This will *not* change the version of Open Firmware in your machine - it will still be Open Firmware 3. The BootROM is what is first executed when you power on or reset your system. The BootROM then loads Open Firmware, which boots your operating system.

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. You will need to run the FirmWare updater from MacOS 9.

Note: If you accidentally set the **`load-base`** or **`real-base`** Open Firmware variables and reset your machine you will, in effect, rewrite the BootROM with garbage, rendering your machine unbootable. We recommend not doing this.

Note: The most recent BootROMs available (4.1.7 and later) are known to have some problems. Some PowerMacintosh users find that their third-party RAM has been disabled. See the discussion at
http://www.macfixit.com/archives/april.01.a.shtml#firm-04-07

The PowerBook (FireWire) and PowerBook G4 (Titanium) models with BootROM 4.1.8 are not supported in this release. If your system has this BootROM, you must use a kernel from the -current branch.

There are reports that FirmWare Update 4.1.9 on iMac (Summer 2000) will prevent the CD-ROM and the hard drive from operating together. You may get `wdc0:0:1: lost interrupt` problems.

Getting to the Open Firmware Prompt on Newer Models

(Open Firmware 3)

Hold down a special four-key combination when your system boots.

After the chime starts, but before it stops, hold down the four 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 or say ALT) until you see the Open Firmware command prompt on your screen:

```
Apple PowerBook3,1 2.1f1 BootROM built on 01/29/00 at 22:38:07
Copyright 1994-2000 Apple Computer, Inc.
All Rights Reserved

Welcome to Open Firmware.
To continue booting, type "mac-boot" and press return.
To shut down, type "shut-down" and press return.
 ok
0 >
```

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

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

Skip down to the section on **Setting up Open Firmware 3 to boot NetBSD** since the next several pages are instructions for older models.

Getting to the Open Firmware Prompt on Apple Network Servers

(Open Firmware 1.1.22)

The version of Open Firmware in the Apple Network Servers can only use a serial console. You must first hook up a serial console (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking) to 'Port 2' (the 'ttya' device in Open Firmware).

Hold down a special four-key combination on the keyboard attached to the ADB port on your system (not the serial console) when your system boots.

After the chime starts, but before it stops, hold down the four 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 or say ALT) until you see some introductory text and the Open Firmware command prompt on your serial terminal:

0 >

Your Apple Network Server's screen will remain black.

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

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

Skip down to the section on **Setting up pre-Open Firmware 3 to boot NetBSD** since the next several pages are instructions for MacOS models.

Getting to the Open Firmware Prompt on Older Models

(Open Firmware 1.0.5, Open Firmware 2.0.x, Open Firmware 2.4)

These models are significantly trickier than the newer machines. Open Firmware has two variables, 'input-device' and 'output-device ', which specify which devices to use for input and output. All Open Firmware 1.0.5 and most Open Firmware 2.0.x systems will default to using the 'Modem' serial port for the console instead of the ADB keyboard and the monitor attached to the on-board video.

Unless you use a MacOS-based utility to set these variables correctly, you *will* need to hook up a serial console temporarily to configure Open Firmware to use your keyboard and screen. Some models (such as the Performa 54xx, 6360, 6400, and 6500) have the 'Modem' serial port covered with a piece of plastic since the internal modem usurps that serial port. You will either need to use Boot Variables to set the 'input-device' and 'output-device' variables to 'ttyb' (which is the Printer serial port) or remove the internal modem.

Now would be a good time to look at the NetBSD/macppc Model Support webpage to determine the issues with your model.

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

In particular, some models *must* use a serial console, or they will be unable to boot NetBSD at all. All models can be set to use a serial console, if you desire to bypass the keyboard and screen.

If, after re-reading the next several sections, you still need help figuring out your 'input-device' and 'output-device' settings, see the FAQ:

http://www.netbsd.org/Ports/macppc/faq.html#ofw-input-output-devices

If 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

See the NetBSD Serial Port Primer for additional help and references:

http://www.netbsd.org/Documentation/Hardware/Misc/serial.html

Older Open Firmware System Preparation

(Open Firmware 1.0.5, Open Firmware 2.0.x, Open Firmware 2.4)

All pre-Open Firmware 3 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. Instructions for using System Disk are covered in the next section.

Unfortunately, some models are broken by or are unsupported by System Disk. If you have one of the following models, then skip down to the section on **Getting to the Open Firmware Prompt (Harder method using**
Apple Performa 4400, 5500, 6500, 54xx, 6400, and 6360,
Motorola Starmax 3000, 4000, 5000, and 5500,
APS Tech M*Power 604e/200,
PowerComputing PowerBase,
Umax Apus 2000, Apus 3000, C500, and C600
Umax S900

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

(Open Firmware 1.0.5, Open Firmware 2.0.x, Open Firmware 2.4)

Download System Disk:

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/

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.

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, or if you accidentally boot into MacOS .

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 on your screen:

```
Open Firmware, 1.0.5
To continue booting the MacOS type:
BYE<return>
To continue booting from the default boot device type:
BOOT<return>
 ok
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 a serial console to the 'Modem' port of your system (38400 bps, 8 bits, no parity, 1 stop bit, no handshaking).

Note: The ''Stop Boot at Open Firmware prompt'' setting is persistent. It is equivalent to the Open Firmware command

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

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)

(Open Firmware 1.0.5, Open Firmware 2.0.x, Open Firmware 2.4)

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.

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 some devices to not work (such as booting from hard drives and CD-Rs).

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. Uncheck the ''auto-boot?'' checkbox, then check 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 then 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 on your screen:

```
Open Firmware, 1.0.5
To continue booting the MacOS type:
BYE<return>
To continue booting from the default boot device type:
BOOT<return>
 ok
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. It is equivalent to the Open Firmware command

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

Getting to the Open Firmware Prompt (Without using MacOS)

(Open Firmware 1.0.5, Open Firmware 2.0.x, Open Firmware 2.4)

If you don't have MacOS, then you need to hold down a special four-key combination when your system boots. Do this on the keyboard attached to the ADB port on your system (not the serial console or PS/2 port) when your system boots.

Note: Your system will *not* have the (possibly critical) NVRAM patches that System Disk applies. Expect some devices to not work (such as booting from hard drives and CD-Rs).

After the chime starts, but before it stops, hold down the four 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 or say ALT) until you see the Open Firmware command prompt on your screen or serial console:

```
Open Firmware, 1.0.5
To continue booting the MacOS type:
BYE<return>
To continue booting from the default boot device type:
BOOT<return>
 ok
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).

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

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

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 up pre-Open Firmware 3 to boot NetBSD

(Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

This section describes some steps you must take to prepare Open Firmware to boot NetBSD. Additional resources are available in the FAQ regarding how to use the Open Firmware command environment, and the Open Firmware variables you may be using:
http://www.netbsd.org/Ports/macppc/faq.html#ofw-access
http://www.netbsd.org/Ports/macppc/faq.html#ofw-variables

Double-check your Open Firmware version:

0 > **dev /openprom**
0 > **.properties**
name   openprom

```
model                   Open Firmware, 1.0.5
relative-addressing
 ok
```

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 settings are stored.

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

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

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.

You can check your Open Firmware settings with the printenv command:

0 > **printenv**
```
little-endian?      false               false
real-mode?          false               false
auto-boot?          false               true
diag-switch?        false               false
[...]
use-nvramrc?        true                false
real-base           F00000              -1
[...]
load-base           600000              4000
[...]
input-device        kbd                 ttya
output-device       /chaos/control      ttya
```

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 2.0.x and Open Firmware 2.4 systems will set the real-base environment variable to its default value (which prevents NetBSD frombooting) if you boot into MacOS .

Setting up Open Firmware 3 to boot NetBSD

(Open Firmware 3)

This section describes some steps you must take to prepare Open Firmware to boot NetBSD. Additional resources are available in the FAQ regarding how to use the Open Firmware command environment, and the Open Firmware variables you may be using:

http://www.netbsd.org/Ports/macppc/faq.html#ofw-access
http://www.netbsd.org/Ports/macppc/faq.html#ofw-variables

Double-check your Open Firmware version:

```
0 > dev /openprom
0 > .properties
name                    openprom
device_type             BootROM
model                   OpenFirmware 3
relative-addressing
supports-bootinfo
 ok
```

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

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

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.

You can check your Open Firmware settings with the printenv command:

```
0 > printenv
------------- Partition: common -------- Signature: 0x70 ---------------
little-endian?      false               false
real-mode?          false               false
auto-boot?          false               true
diag-switch?        false               false
[...]
use-nvramrc?        true                false
real-base           -1                  -1
[...]
input-device        keyboard            keyboard
output-device       screen              screen
```

Available Boot Media

(All Open Firmware versions)

Each version of Open Firmware is capable of booting from a variety of media (such as hard drives, floppy disks, CD-ROMs, and ethernet). Open Firmware is able to boot files from a variety of filesystems (such as ISO9660, MS-DOS, and on some models HFS and HFS+). Unfortunately, Open Firmware is *not* able to use the NetBSD filesystem (FFS), so we must put the bootloader in a location that Open Firmware is capable of using.

Therefore, 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.

Note: ofwboot.elf is obsoleted. All users should be using ofwboot.xcf instead of ofwboot.elf now.

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

- *Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x*
  - Partition zero
  - Ethernet (netboot)
  - ISO9660 CD-ROM filesystem
  - MS-DOS filesystem
- *Open Firmware 2.4*
  - Partition zero
  - Ethernet (netboot)
  - ISO9660 CD-ROM filesystem
  - HFS or HFS+ filesystems
  - Hybrid (both ISO9660 and HFS) filesystem
  - MS-DOS filesystem
- *Open Firmware 3*
  - HFS, HFS+, or Hybrid (ISO9660 and HFS) filesystem
  - Ethernet (netboot)
  - Partition zero (not supported by NetBSD)
  - MS-DOS filesystem

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

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

  Such as the boot floppy, or an installed NetBSD partition.
- *HFS or HFS+ filesystem*

Only on Open Firmware 2.4 or Open Firmware 3 systems.

- *ISO9660 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. The boot CD-R images provided with the distribution sets has both a 'partition zero' bootloader and ofwboot.xcf on a hybrid partition so it can be booted on all Open Firmware versions. The boot CD-R image also has an ISO9660 filesystem with an installation kernel and the distribution sets.

Description of bootable media

(All Open Firmware versions)

This section gives a quick description of the various media that can be used by Open Firmware to load the bootloader.

- *Partition zero*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  This method loads a primary bootloader called bootxx located in the Apple Partition Map, which then loads the secondary bootloader ofwboot (functionally identical to ofwboot.xcf (q.v.)) which then loads the kernel. This is what **sysinst** will place on your hard drive during the installation procedure. Additionally, the NetBSD/macppc CD-R images and boot floppies use this method to load a kernel from the floppy's FFS partition or the CD-R's ISO9660 partition.

- *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)

  This method loads ofwboot.xcf from an HFS or HFS+ partition which then loads the kernel from an acceptable source (such as an FFS partition or an HFS or HFS+ partition). The FFS-based kernel may be on the boot floppy image (boot.fs) located on an HFS or HFS+ partition, or located on a full FFS partition.

  Alternatively, the kernel (compressed or non-compressed) may be on an HFS or HFS+ partition. Due to Open Firmware pickiness, it must be on the same partition as ofwboot.xcf .

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

  Load the ofwboot.xcf bootloader from an MS-DOS filesystem (such as a floppy or Zip disk). It can then load a NetBSD kernel from the same MS-DOS filesystem or elsewhere.

- *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: You must use your on-board ethernet device for netbooting. While there may be some PCI or Cardbus ethernet interfaces with Open Firmware ROMs, no one has reported success netbooting using these devices

- *ISO9660 CD-ROM file system*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  Load the ofwboot.xcf bootloader from an ISO9660 CD-ROM. It can then load a NetBSD kernel from the ISO9660 CD-ROM.

  Open Firmware 1.0.5 and 2.0.x machines will *not* boot from the ISO9660 file system if you create a hybrid HFS/ISO format CD-R. Make sure that the CD *only* has ISO9660 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/ISO9660 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 HFS or ISO9660 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

Partitioning your hard drive for NetBSD

(All Open Firmware versions)

- *Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x*
  You must use the NetBSD installer to partition your disk if you want it to be bootable. With this release of NetBSD, there is no way to dual-boot MacOS and NetBSD on one hard drive.

  You can use the instructions in this section to partition a disk that may also be used with MacOS , although a disk prepared in this way *will not boot* NetBSD. That means, your root partition (/) must be on a drive prepared with the NetBSD installer, but the partitions not necessary to boot (for example /usr , /home , or /export) may be on the same disk as MacOS .

  Unless you are planning to use partitions on the same disk as MacOS , skip forward to **Preparing the Bootable Media**

- *Open Firmware 2.4*
  We recommend using the NetBSD installer to partition your disk, but you will not be able to use partitions on this drive with MacOS.

  If you'd like to have MacOS and NetBSD partitions on one hard drive (and for both to be bootable), then read the rest of this section.

- *Open Firmware 3*
  You must have at least one disk that was partitioned using a MacOS tool before installing NetBSD on your system.

  This is the drive that will be holding the bootloader, ofwboot.xcf . Your NetBSD partitions may either be on this same disk (using the method described in the rest of this section), or on a separate disk accessible only to NetBSD. If you are dedicating an entire disk to NetBSD, then your MacOS partitioned disk only needs HFS or HFS+ partitions.

  The rest of this section describes how to make a single disk dual-bootable which is necessary for machines which have only one hard drive (such as the iBook, iMac, PowerBook, and PowerMacintosh Cube systems).

Using the

There are several contradictory partition numbering schemes. You may want to look at the explanation in the FAQ.

Unfortunately, the partitioning tools for are not advanced enough to create HFS or HFS+ file systems or a partition map that can be shared with You will need to use the Drive Setup tool (versions 1.8.1 and later are known to work) to partition the hard drive you will be using for Drive Setup will erase the contents of your drive -- it does not preserve data from any of your partitions. Make the following partitions:

HFS or *HFS+* Must be large enough to hold the bootloader, over 100 KB. May be as large as you desire for MacOS usage.

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 without X, or 250 MB with X. NetBSD interprets an *A/UX Root* partition as the first partition (*a*) on the disk. This partition is not readable from MacOS .

A/UX Swap Any size. The recommendation is 1.5 times your RAM, although this is not strictly necessary for machines with a lot of RAM. NetBSD interprets an *A/UX Swap* partition as the second partition (*b*) on the disk. This partition is not readable from MacOS .

A/UX User and *A/UX Free1* Use these for any additional partitions you may want to use under NetBSD, such as interprets these partitions as normal partitions. These partitions are not readable from

Em Any size. You may want to leave an additional partition available to transfer files between MacOS and NetBSD. If you think you'll be doing this often, then see the FAQ entry on sharing files:

http://www.netbsd.org/Ports/macppc/faq.html#disk-sharing

UFS Any size. This filesystem type is not currently supported by NetBSD/macppc although it might be in the future. *UFS* partitions are not readable from MacOS versions prior to X 10.0.0.

If you need further help, there is a partitioning HOW-TO on the NetBSD website:

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

Preparing the Bootable Media

The purpose of this section is to prepare the media from which your system will boot the installer. We'll describe how to put the files in the right places on your disk(s) or netboot server and prepare it for use on your 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, you only need to extract the distribution sets on the diskless server.

To get the distribution sets (which will be placed on your hard drive) onto appropriate media, see the above section entitled **Getting the NetBSD System on to Useful Media** . You may wish to do this at the same time you create the bootable media (such as netbooting and placing distribution sets on the netboot server, or booting from CD-R and placing distribution sets on the same disk)

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.

These are the recommended techniques for booting the NetBSD install kernel:

- *Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x*
  - Booting off an MS-DOS floppy
  - Booting off the NetBSD install floppy
  - Booting off the NetBSD/macppc install CD-R image
  - Booting off an ISO9660 CD-ROM
  - Booting over the ethernet
  - Booting the floppy image off a SCSI or IDE drive (not recommended)
- *Open Firmware 2.4*
  - Booting off an MS-DOS floppy (may not be supported by Open Firmware 2.4)
  - Booting off the NetBSD install floppy
  - Booting off the NetBSD/macppc install CD-R image
  - Booting off an HFS or HFS+ partition
  - Booting off an ISO9660 CD-ROM
  - Booting off a Hybrid HFS/ISO9660 CD-ROM
  - Booting over the ethernet
  - Booting the floppy image off a SCSI or IDE drive (not recommended)
- *Open Firmware 3*
  - Booting off an HFS or HFS+ partition
  - Booting off the NetBSD/macppc install CD-R image
  - Booting off a Hybrid HFS/ISO9660 CD-ROM
  - Booting over the ethernet

What follows are the steps to create different types of bootable media. You should only need to create one of these to get your system to boot the installer

- *Booting off an MS-DOS floppy*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  1. First, make sure you have done a low-level format on the disk. A bad floppy can produce Open Firmware errors similar to other problems.
  2. Mount the floppy on your computer.
     - Insert it into a Windows or DOS machine.
     - Use 'File Exchange or 'PC Exchange'' with MacOS versions prior to X 10.0.0
     - Use mount_msdos(8) on any unix-like machine.
  3. Copy netbsd.ram.gz and ofwboot.xcf to the floppy.
  4. Rename netbsd.ram.gz to netbsd.gz, since OpenFirmware can't deal with MS-DOS filenames longer than eight characters.

  Skip forward to **Installing the NetBSD System**

- *Booting off the* NetBSD*install floppy*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  1. First, make sure you have done a low-level format on the disk. A bad floppy can produce Open Firmware errors similar to other problems.
  2. If the boot floppy image is compressed, then you will need to uncompress it first.
  3. Use a tool to write the boot floppy image installation/boot.fs to a floppy disk
     - *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 **DiskCopy** 6.3.3 from
       http://downloAd.info.apple.com/Apple_Support_Area/Apple_Software_Updates/English-North_American/Macintosh/Utilities/Disk_Copy/Disk_Copy_6.3.3.smi.bin

       If you are running a MacOS version prior to 9.0, get
       ftp://ftp.netbsd.org/pub/NetBSD/arch/macppc/macos-utils/MakeDiskCopyFloppyType.hqx
       and un-BinHex it. Then, drag the boot.fs file onto this application, which will change the type and creator to dimg and ddsk, which is necessary for DiskCopy to work.

       4. Open the 'Utilities' menu and pick 'Make Floppy'

5. In the dialog box, select boot.fs

6. At the prompt, insert a blank floppy disk

– *From a MacOS machine (Alternate)*

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'

Skip forward to **Installing the NetBSD System**

- *Booting off an HFS or HFS+ partition*
(Open Firmware 2.4, Open Firmware 3)

Place ofwboot.xcf and the installation kernel netbsd.ram.gz at the top level of your Macintosh filesystem. That is, drag the two icons onto your hard drive icon (not the desktop). Make sure they're both on the same partition.

Skip forward to **Installing the NetBSD System**

- *Booting off the NetBSD/macppc CD-R image*
(All Open Firmware versions)

Go to one of the NetBSD mirror sites and download CD-R image from the pub/NetBSD/iso directory. **http://www.netbsd.org/mirrors/#iso**

– *From a UNIX machine*

Get and install cdrecord. NetBSD users should install it from the package collection. Other unix systems should get it from the official website:
ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/sysutils/cdrecord/README.html
http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

```
# cdrecord -v speed=4 dev=/dev/cd1c output.iso
```

You will need to substitute the correct name of the disk image file, speed for your CD writer, and the correct device for your system (for i386 it would be /dev/cd1d).

– *From a MacOS machine (using Toast)*

1. Open the 'Format' menu and pick 'Disk Image'

2. Click the ''Data...'' button and select the disk image you downloaded.

3. Click the ``Write CD...'' button

Skip forward to **Installing the NetBSD System**

- *Booting off an ISO9660 CD-ROM*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  This section describes how to create your own bootable NetBSD/macppc CD-R. We recommend that you use the official NetBSD/macppc-1.5.3 CD-R image, as described in the section above.

  Place ofwboot.xcf and the installation kernel netbsd.ram.gz at the top level of the CD. You may also place the NetBSD 1.5.3 distribution sets on the disk.

  – *From a UNIX machine*

    Get and install mkhybrid. This is now part of the cdrecord package. NetBSD users should install it from the package collection. Other unix systems should get it from the official website:
    ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/sysutils/cdrecord/README.html
    http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

    ```
    # mkhybrid -o output.iso -a -l -J -r
    # cdrecord -v speed=4 dev=/dev/cd1c output.iso
    ```

    You will need to substitute the correct speed for your CD writer, and the correct device for your system (for i386 it would be /dev/cd1d).

    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

  Skip forward to **Installing the NetBSD System**

- *Booting off a hybrid HFS/ISO9660 CD-ROM*
  (Open Firmware 2.4, Open Firmware 3)

  This section describes how to create your own bootable NetBSD/macppc CD-R. We recommend that you use the official NetBSD/macppc-1.5.3 CD-R image, as described two sections above.

  Place ofwboot.xcf and the installation kernel netbsd.ram.gz at the top level of the CD. Make sure that the bootloader is present on the HFS partition, and the kernel is on both the ISO9660 and HFS partitions.

You may also place the NetBSD 1.5.3 distribution sets on the ISO9660 partition (*not* only on the HFS partition).

– *From a UNIX machine*

  Get and install mkhybrid. This is now part of the cdrecord package. NetBSD users should install it from the package collection. Other unix systems should get it from the official website:
  ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/sysutils/cdrecord/README.html
  http://www.fokus.gmd.de/research/cc/glone/employees/joerg.schilling/private/cdrecord.html

```
# mkhybrid -o output.iso -hfs -part -a -l
# cdrecord -v speed=4 dev=/dev/cd1c output.iso
```

  You will need to substitute the correct speed for your CD writer, and the correct device for your system (for i386 it would be /dev/cd1d).

  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. You may also want to drag the NetBSD 1.5.3 distribution sets to this window as well.
  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

Skip forward to **Installing the NetBSD System**

• *Booting over the ethernet*
  (All Open Firmware versions)

  Set up the DHCP daemon in BOOTP compatible 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 net-boot server. If you intend to use a different platform as your net-boot server, follow the directions in the Diskless HOW-TO:

http://www.netbsd.org/Documentation/network/netboot

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 .

You should replace all of these with the names, addresses, and paths appropriate to your environment.

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

```
ddns-update-style none;
                # Do not use any dynamic DNS features
                #
allow bootp;    # Allow bootp requests, thus the dhcp server
                # will act as a bootp server.
                #
authoritative;  # This is the authoritative DHCP server for this subnet
                #
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
```

Skip forward to **Installing the NetBSD System**

- *Booting the floppy image off a SCSI or IDE drive*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  If the boot floppy image is compressed, then you will need to uncompress it first.

  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. For example, you could use a Zip drive, a Jaz drive, a Compact Flash drive, or even a spare hard 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'

Installing the NetBSD System

Open Firmware boot syntax

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

```
boot boot-device [boot-file] [-as]
```

where the boot-device describes where to find the bootloader, boot-file describes where to find the NetBSD kernel, and the options specify how you want to boot.

You use the boot-device to tell Open Firmware where to find ofwboot(7) by listing the device, the partition (if it's a disk), and the filename of the bootloader (if using ofwboot.xcf).

If the boot-file is on the same device and (if it's a disk) partition as the boot-device then you can just specify the kernel filename. Otherwise, you need to specify the full Open Firmware path to the kernel.

The **-a** flag will ask you for the location of the next item to load (i.e. the bootloader will ask where the kernel is (if unspecified), or the kernel will ask where the root file system is). The **-s** flag will boot into 'single-user' mode.

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. Sometimes you may have to guess as we don't know all of the combinations of models, device names, and file names. In general the format is: device:[partition][,\filename]. Keep in mind for the future that you may be able to have your boot-device and boot-file on entirely different devices (such as the bootloader netbooted from enet and the kernel loaded from a hard drive on the ultra0 ATA/IDE bus).

We'll try to walk you through the process of figuring out what Open Firmware calls your device, partition, and file names. To start with, Open Firmware keeps a ''device tree'' with all of the devices it finds in your system. You can get a listing of the nodes in this device tree with the **dev** and **ls** commands. **dev** is similar to the unix **cd** command and is used to change between the nodes in the Open Firmware device tree (similar to a filesystem). **ls** of course is similar to the unix **ls** command and is used to list the contents of the current device node. To get a listing of all the devices available in your system, use the following commands:

```
0 > dev /
0 > ls
```

Open Firmware has device aliases which are simple names for the full hardware path to a device (similar to **alias** in csh(1)). You can find out what device aliases Apple created on your machine with the **devalias** command. For example, here are the devaliases on a PowerMacintosh 7300:

```
0 > devalias
vci0                /chaos@F0000000
pci1                /bandit@F2000000
pci2                /bandit@F4000000
fd                  /bandit/gc/swim3
kbd                 /bandit/gc/via-cuda/adb/keyboard
ttya                /bandit/gc/escc/ch-a
ttyb                /bandit/gc/escc/ch-b
enet                /bandit/gc/mace
scsi                /bandit/gc/53c94
scsi-int            /bandit/gc/mesh
 ok
```

On most systems, you'll find the devices you're looking for. Typical aliases are:

Dl hd	internal hard drive
Dl cd	CD-ROM drive
Dl zip	internal Zip drive
Dl enet	ethernet
Dl fd	floppy drive
Dl scsi	SCSI bus
Dl scsi-int	internal SCSI bus (on systems with multiple SCSI busses)
Dl scsi-ext	external SCSI bus (on systems with multiple SCSI busses)
Dl ata	ATA/IDE bus
Dl ideN	ATA/IDE bus number N
Dl ultraN	Ultra/66 or Ultra/100 IDE bus number N

Note that some of these items are the device itself, and some are a bus. When you only have the devalias to a bus, you need to specify which device on that bus you want to use. You can use the Open Firmware **dev** and **ls** commands. For example, here are the devices on the internal SCSI bus of a PowerMacintosh 7300:

```
0 > dev scsi-int
0 > ls
FF83C850: /sd@0,0
FF83D480: /st@0,0
 ok
```

In this case, Open Firmware seems to be saying there are two devices, both at address zero (one is a SCSI disk 'sd@0,0' and the other is a SCSI tape 'st@0,0).' Unfortunately, older systems will only list the naming convention and not the actual devices currently connected, but that's OK -- we've got more tricks up our sleeve.

If you've got ATA/IDE drives, you have all the device information you need (since Apple only ever ships drives as ''master'' which is typically something like ata-disk@0, ATA-Disk@0, atapi-disk, or ata-disk@000ata-disk@011ata-disk@02

You can find out the devices on your SCSI bus with the **show-children** command:

```
0 > dev scsi-int
0 > show-children
Target 0
  Unit 0  Disk     IBM     DCAS-32160      S65A
Target 3
  Unit 0  Removable Read Only device    SONY    CD-ROM CDU-8005 1.0j
 ok
```

Open Firmware calls SCSI IDs ``Target''. The ``Unit'' number is the Logical Unit Number (LUN). This is almost always zero. Thus, this PowerMacintosh system has an IBM hard drive (DCAS-32160) at SCSI ID 0, and a Sony CD-ROM drive (CDU-8005) at SCSI ID 3.

Now, we've got enough information to construct the device name for Open Firmware. Just stick everything together to describe to Open Firmware what you want. For example, Open Firmware calls the CD-ROM drive in this system scsi-int/sd@3.

To determine if a device is bootable, type:

```
0 > dev scsi-int/sd@3
0 > words
load          write        read          seek        close     open
write-blocks  read-blocks  max-transfer  block-size  dma-sync  dma-map-out
dma-map-in    dma-free     dma-alloc
 ok
```

If the **word** ``open'' is present in the list, then the device is almost certainly bootable.

Next, you need to figure out what partition Open Firmware thinks your bootloader is located on if you're going to boot from a disk. If you're using a ``partition zero'' bootloader, the answer is obvious: 0. Thus, your boot-device for the NetBSD/macppc CD-R image on an Open Firmware 1.0.5 system would be scsi-int/sd@3:0 since the image has a 'partition zero' bootloader.

Other situations get a little trickier, as we know of no way to get a partition map from within Open Firmware, and it uses a different numbering scheme than either NetBSD or MacOS 9.x and earlier. See the FAQ for an explanation:

http://www.netbsd.org/Ports/macppc/faq.html#partitions

Typically, MS-DOS and ISO9660 formatted disks have their filesystems at partition 1. Typically, Drive Setup formatted disks have their filesystems starting at partition number 9. Often, if you omit the partition number, Open Firmware looks in the first partition it understands holding a valid filesystem.

Open Firmware uses a comma (,) to separate the partition number from the filename. It uses a backslash (the \ character) to separate directories. The bootloader uses forward slashes (the / character) to separate directories when specifying the boot-file. Thus, to specify the top of the filesystem on a CD-ROM in the example PowerMacintosh 7300 system, you'd use: scsi-int/sd@3:,\

Now, to confirm that you and Open Firmware are looking at the same files, you can get a directory listing of the filesystem on your device with the **dir** command. This command is only useful with filesystems that Open Firmware understands and is able to boot from. On Open Firmware 1.0.5, 1.1.22, and 2.0.x systems, you can use **dir** on ISO9660 (not hybrid) and MS-DOS filesystems. On Open Firmware 2.4, you can use it on HFS, HFS+, hybrid, ISO9960, and MS-DOS filesystems. On Open Firmware 3, you can use it on HFS, HFS+, hybrid (not pure ISO9660), and MS-DOS filesystems.

```
0 > dir fd:,\
FINDER  .DAT 022 2 2B8
DESKTOP .    022 0 0
RESOURCE.FRK 012 3 0
NETBSD~1.GZ  020 5 1FDFCA
TRASH   .    010 B00 0
OFWBOOT .XCF 020 A75 D8F4
 ok
```

You can see that this MacOS formatted floppy has a bunch of stuff, as well as the two important files: NETBSD~1.GZ and OFWBOOT.XCF. Note that MacOS shortened netbsd.ram.gz to NETBSD~1.GZ since MS-DOS filesystems can only natively hold 8 characters for the filename. Keep in mind that Open Firmware is often case-sensitive when it comes to filenames.

If the **dir** command showed you the files you're looking for, then you've figure out how to tell Open Firmware to look for your bootloader! In this case, your boot-device is fd:,\OFWBOOT.XCF and your boot-file is fd:,/NETBSD~1.GZ.

For additional help, see ofwboot(7) and the FAQ on topics like how to use the Open Firmware command environment and how to boot from a device attached to a PCI card which has Open Firmware support:

http://www.netbsd.org/Ports/macppc/faq.html#ofw-use
http://www.netbsd.org/Ports/macppc/faq.html#boot-pci

Examples of Open Firmware boot commands

Here are some examples of the commands you might use to boot your system:

- *Booting off an MS-DOS floppy*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  Note: This is not a normal MacOS boot floppy -- you must enter Open Firmware and type a boot command.

  Use the Open Firmware **dir** command to get a listing of the files on the floppy. Typically you'll find filenames like OFWBOOT.XCF and NETBSD~1.GZ.

  ```
  0 > boot fd:,\OFWBOOT.XCF NETBSD~1.GZ
  ```

  You can eject a floppy by typing:

0 > **eject fd**

- *Booting off the NetBSD install floppy*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  The floppy disk has a 'partition zero' bootloader.

  Note: This is not a normal MacOS boot floppy -- you must enter Open Firmware and type a boot command.
  All you need to do is:

  0 > **boot fd:0**

  You can eject a floppy by typing:

  0 > **eject fd**

- *Boot over the ethernet*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  You can try the simple form (i.e. that you are booting from ethernet):

  0 > **boot enet:,\ofwboot.xcf**

  Or you may be more specific, specifying the bootloader filename and the kernel name:

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

  Note: Some Open Firmware 1.0.5 machines have their MAC address stored incorrectly. Make sure that your netboot server is using the same MAC address that your macppc client is using. See the section on **Setting up pre-Open Firmware 3 to boot NetBSD** to figure out your MAC address.

  Note: Some Open Firmware 1.0.5 machines, using ofwboot.xcf version 1.2 cannot load compressed kernels. Uncompress them first.

- *Boot over the ethernet*
  (Open Firmware 3)

  You can try the simple form (i.e. that you are booting from ethernet):

  0 > **boot enet:0**

  Or you may be more specific, specifying the bootloader filename and the kernel name:

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

- *Booting off the NetBSD/macppc install CD-R*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  Here are some examples of what you might use to boot from CD-ROM (Apple usually sets their CD-ROM drives to SCSI ID 3):

  0 > **boot scsi-int/sd@3:0 NETBSD.MACPPC**
  0 > **boot scsi/sd@3:0 NETBSD.MACPPC**
  0 > **boot scsi-ext/sd@3:0 NETBSD.MACPPC**
  0 > **boot ata/atapi-disk:0 NETBSD.MACPPC**
  0 > **boot ide1/disk@0:0 NETBSD.MACPPC**
  0 > **boot cd:0 NETBSD.MACPPC**

- *Booting off the NetBSD/macppc install CD-R*
  (Open Firmware 3)

  0 > **boot cd:,\ofwboot.xcf netbsd.macppc**

- *Booting off a home-made CD-ROM ISO9660 or Hybrid*
  (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. You should use the Open Firmware **dir** command to get a listing of the files on your CD.

  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 the same case when specifying the filename to load that Open Firmware uses. Also, keep in mind what format your CD-R has to be for your version of Open Firmware (pure ISO versus hybrid ISO/HFS).

  Here are some examples of what you might use to boot from CD-ROM (Apple usually sets their CD-ROM drives to SCSI ID 3):

  0 > **boot scsi-int/sd@3:,\OFWBOOT.XCF NETBSD.MACPPC**
  0 > **boot scsi/sd@3:,\OFWBOOT.XCF NETBSD.MACPPC**
  0 > **boot scsi-ext/sd@3:,\OFWBOOT.XCF NETBSD.MACPPC**
  0 > **boot ata/atapi-disk:,\OFWBOOT.XCF NETBSD.MACPPC**
  0 > **boot ide1/disk@0:,\OFWBOOT.XCF NETBSD.MACPPC**
  0 > **boot cd:,\ofwboot.xcf netbsd.macppc**

- *Booting off an IDE or SCSI drive with a* 'partition zero *bootloader'*
  (Open Firmware 1.0.5, Open Firmware 1.1.22, Open Firmware 2.0.x, Open Firmware 2.4)

  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**

- *Booting off an IDE or SCSI drive with an HFS or HFS+ partition*
  (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. If you're running MacOS X you can run the following command to print out the partition table on your drive:

  % **sudo pdisk /dev/disk0 -dump**

  If you are confused by the various partition numbering schemes, see the explanation in the FAQ
  http://www.netbsd.org/Ports/macppc/faq.html#partitions

  You should use the Open Firmware **dir** command to get a directory listing of the files on your hard drive.

  If 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 the installer kernel located on an HFS or HFS+ partition:

  0 > **boot scsi/sd@0:9,\ofwboot.xcf netbsd.ram.gz**
  0 > **boot ide0/disk@0:10,\ofwboot.xcf netbsd.ram.gz**
  0 > **boot hd:9,\ofwboot.xcf netbsd.ram.gz**
  0 > **boot ultra1:9,ofwboot.xcf netbsd.ram.gz**

Example of a normal boot

Of course, a lot of the information in this example depends on your model and what your boot method is, but we'll include this anyways just so you get an idea of what to expect (user-typed commands are in **bold**).

```
Apple PowerBook3,1 2.1f1 BootROM built on 01/29/00 at 22:38:07
Copyright 1994-2000 Apple Computer, Inc.
All Rights Reserved

Welcome to Open Firmware.
To continue booting, type "mac-boot" and press return.
To shut down, type "shut-down" and press return.
 ok
0 > boot enet:,\ofwboot.xcf netbsd.ram.gz
loading XCOFF
tsize=C280 dsize=14AC bsize=2620 entry=600000
SECTIONS:
.text    00600000 00600000 0000C280 000000E0
.data    0060D000 0060D000 000014AC 0000C360
.bss     0060E4B0 0060E4B0 00002620 00000000
loading .text, done..
loading .data, done..
clearing .bss, done..

>> NetBSD/macppc OpenFirmware Boot, Revision 1.3
>> (tsubai@mint.iri.co.jp, Sun Nov 26 01:41:27 JST 2000)
1701508+177748 [100+68176+55886]=0x1e9468

 start=0x100000
Copyright (c) 1996, 1997, 1998, 1999, 2000, 2001
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5.2 (INSTALL) #0: Thu Mar 15 00:52:56 PST 2001
    mw@al:/usr/src/sys/arch/macppc/compile/INSTALL
total memory = 192 MB
avail memory = 172 MB
using 2483 buffers containing 9932 KB of memory
[...]
erase ^H, werase ^W, kill ^U, intr ^C, status ^T
Terminal type? [vt100]
Erase is backspace.
(I)nstall, (S)hell or (H)alt ?
```

Common Problems and Error Messages

This is a brief list of some of the Open Firmware problems you may run into. See the NetBSD/macppc FAQ for a thorough list.

http://www.netbsd.org/Ports/macppc/faq.html#boot-trouble

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

- 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 on Older Models**

- Grey screen with flashing question mark

  This means that your system is trying to boot MacOS. You might get to this state if you didn't properly enter Open Firmware, or if your Open Firmware boot command is misformed.

  If you typed a bad boot command, Open Firmware tries to boot from the value stored in the boot-device
  variable, whose default value is the MacOS ROM.

  The grey screen with the icon is generated by the MacOS ROM. A flashing question-mark or broken folder means that your machine is looking for a bootable MacOS filesystem, but can't find one. A happy face means it's trying to boot MacOS.

  The boot floppy does not have a bootable MacOS filesystem, it has a bootable NetBSD filesystem.

  Reboot, re-enter Open Firmware and check your syntax carefully.

  Note: Don't forget to check your Open Firmware environment variables, as they may have been changed by your brief excursion into MacOS
  .

- Information on your screen seems garbled or out of sync

  If you have a PowerMacintosh 7300 through 8600, then you need to read the section on System Disk and the patches it applies. See the section entitled **Getting to the Open Firmware Prompt on Older Models**

  If you have a Twentieth Anniversary Macintosh, Open Firmware does not work with the internal display, you will need to set up a serial console.

- ''DEFAULT CATCH!''

  This is a general message from Open Firmware to the effect that it had trouble loading a file.

  If your machine is Open Firmware version 1.0.5, 2.0.x, or 2.4, this error does sometimes appear randomly. You might try the boot command a second time (this is known to work on some models). It's also an indication that either your floppy disk is bad, or the floppy drive is bad. Try doing a low-level format on the floppy, re-copy the files, and try again.

There are several models that cannot be booted while using the on-board video and keyboard. Try using a serial console.

- ``CLAIM failed''

  This is a general message from Open Firmware to the effect that it failed to allocate some memory or memory is messed up.

  First, make sure you have the Open Firmware variable load-base set correctly. If your system is Open Firmware version 1.0.5, 1.1.22, 2.0.x, or 2.4, see the section above on **Setting up pre-Open Firmware 3 to boot NetBSD**

  Do not change load-base on an Open Firmware 3 system.

  If you tried one boot command and it failed, then you tried a second and got the CLAIM failed message, then this is an indication that you should reboot between attempts. Use the Open Firmware **reset-all** command.

  Contradictorily, if your machine is Open Firmware version 1.0.5, 2.0.x or 2.4, this error does sometimes appear randomly. You might try the boot command a second time (this is known to work on some models).

  There are several models that cannot be booted while using the on-board video and keyboard. Try using a serial console.

- ``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.

  Use the Open Firmware **dir** command to verify that the file(s) you tried accessing are really there.

- ``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). Make sure that you have use binary mode to FTP the files, and that they are properly uncompressed.

- ``bad partition number, using 0no bootable HFS partition

  If you're trying to boot an Open Firmware 1.0.5, 1.1.22, or 2.0.x system, this probably means that your media (i.e. hard drive or CD-ROM) has an HFS filesystem on it (such as a hybrid CD-R or a hard drive with MacOS partitions).

- ``TFTP timeout''

  Either the server's TFTP server isn't running, or you're using a model with Open Firmware 1.0.5 and not specifying the location of the bootloader. Unfortunately, on the early models, ofwboot.xcf gets confused and doesn't work right unless you load it explicitly from the boot-device, even if the bootp or DHCP server provides the correct information. You need to boot with a command like:

0 > **boot enet:,\ofwboot.xcf**

- Bootloader 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.

  Make sure you have the Open Firmware variable real-base set correctly. If your system is Open Firmware version 1.0.5, 1.1.22, 2.0.x, or 2.4, see the section above on **Setting up pre-Open Firmware 3 to boot NetBSD**

  Do not set real-base on an Open Firmware 3 system.

- Hang after configuring devices, but before doing anything else

  Actually, this can have many causes. The most likely is a keyboard problem. First, try plugging the USB keyboard directly into the computer (i.e. not through a hub) and unplugging the mouse.

  This is also a known problem on some PowerMacintosh G3 (Blue and White) and iMac (Bondi Blue) machines. If you have a PowerMacintosh G3 (Blue and White) with this problem, then you can try the following procedure:

  1. Unplug the USB keyboard and plug in an ADB keyboard
  2. Boot NetBSD with the ADB keyboard
  3. Enter root, swap and filesystem responses on the ADB keyboard
  4. When the system gets to the shell prompt (or login if you went multi-user), unplug the ADB and plug in the USB keyboard

  The USB will be usable at this point and you can use the system. You'll notice that the ADB is dead even before you unplugged it. Also you MUST boot without the USB installed or this won't work.

  Alternatively, much as it pains us to say this, NetBSD 1.4.3 works fine on both the iMac (Bondi Blue) and the PowerMacintosh G3 (Blue and White) models and does not have this USB keyboard problem.

Milestone

If you've reached this point, then you must've gotten the NetBSD installer to boot. Congratulations! That was the hard part. From now through the rest of this document, there should be no more Open Firmware specific problems, so read everything because it applies to all models.

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, by running the /sysinst program from the command prompt, you do not need to reboot.

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.

```
.*********************************************.
* NetBSD-1.5.3 Install System                 *
*                                             *
*>a: Install NetBSD to hard disk              *
* b: Upgrade NetBSD on a hard disk            *
* c: Re-install sets or install additional sets *
* d: Reboot the computer                      *
* e: Utility menu                             *
* x: Exit Install System                      *
.*********************************************.
```

  - If you wish, you can configure some network settings immediately by choosing the **Utility 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. Further information can be found on
  http://www.netbsd.org/Ports/macppc/x11.html

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 capacity. 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 **Utility menu** , and select the **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 the Domain Name System (DNS), you can give an empty response in reply to answers relating to this.

6. *Preparing a disk which will be used for* MacOS *and NetBSD*

   Skip this step if you are installing NetBSD onto a dedicated drive.

   Go to the **Utility Menu** , and select the **Run /bin/sh** option which will give you a shell prompt. From this shell prompt, you will do some of the steps that the normal install procedure runs automatically. Unfortunately, at the moment, our install tools aren't smart enough to deal with drives shared with MacOS and will overwrite important information describing your partitions.

You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

```
# stty erase '^h'
# stty erase '^?'
```

Type the following command (replacing wd0 with the name of your destination hard drive):

```
# disklabel wd0
```

This will print out the partition info that was generated by Drive Setup. Note that, as discussed above in the **Partitioning your hard drive for NetBSD** section, your *A/UX Root* typically is the first partition (*a*) and your *A/UX Swap* typically is the second partition (*b*). You may also find that your *A/UX User* partition is the seventh partition (*g*). For example:

```
     # disklabel wd0
[...]
#        size   offset     fstype   [fsize bsize cpg/sgs]
  a:   426613   837432     4.2BSD        0     0     0   # (Cyl. 1622*- 2449*)
  b:   204800   632632       swap                        # (Cyl. 1226*- 1622*)
  c:  2134305        0     unused        0     0         # (Cyl.    0 - 4136*)
  d:   426616     1216        HFS                        # (Cyl.    2*- 829*)
  e:   204800   427832        HFS                        # (Cyl.  829*- 1226*)
  f:       21  2134284    unknown                        # (Cyl. 4136*- 4136*)
  g:   870239  1264045     4.2BSD        0     0     0   # (Cyl. 2449*- 4136*)
disklabel: boot block size 0
disklabel: super block size 0
```

Now, you need to create filesystems on the partitions that NetBSD will be using.

Do not modify any partitions labeled HFS *or* unknown . The partitions you will be using have their fstype listed as 4.2BSD .

Run the **newfs** command on the 4.2BSD partitions:

```
     # newfs /dev/wd0a
newfs: /dev/sd1a: not a character-special device
Warning: 120 sector(s) in last cylinder unallocated
/dev/sd1a:      426612 sectors in 827 cylinders of 4 tracks, 129 sectors
        208.3MB in 52 cyl groups (16 c/g, 4.03MB/g, 1024 i/g)
super-block backups (for fsck -b #) at:
     32,   8432,  16832,  25232,  33056,  41456,  49856,  58256,  66080,
  74480,  82880,  91280,  99104, 107504, 115904, 124304, 132128, 140528,
 148928, 157328, 165152, 173552, 181952, 190352, 198176, 206576, 214976,
 223376, 231200, 239600, 248000, 256400, 264224, 272624, 281024, 289424,
 297248, 305648, 314048, 322448, 330272, 338672, 347072, 355472, 363296,
 371696, 380096, 388496, 396320, 404720, 413120, 421520,
newfs: ioctl (WDINFO): Invalid argument
newfs: /dev/sd1a: can't rewrite disk label
```

You can ignore the 'not a character-special device', 'sector(s) in last cylinder unallocated', 'ioctl (WDINFO): Invalid argument', and 'can't rewrite disk label' warnings.

Now you need to mount your destination root partition:

```
# mount /dev/wd0a /mnt
```

Make an fstab file for your new system (right now, you only really need to include / , /usr , and swap), for example:

```
# mkdir /mnt/etc
# cat > /mnt/etc/fstab
/dev/wd0a / ffs rw 1 1
/dev/wd0b none swap sw 0 0
/dev/wd0g /usr ffs rw 1 2
```

If you mess up while typing, you can press CONTROL-U to erase everything on the current line, or CONTROL-C to cancel the file creation, so you can start over.

Great, now create the mountpoints for the filesystems you listed in the fstab :

```
# mkdir /mnt/usr
```

Clean up and return to **sysinst**:

```
# cd /;umount /mnt
# exit
```

7. *Installation drive selection and parameters*

   To start the installation onto a dedicated NetBSD drive, select **Install NetBSD to hard disk** from the main menu. To start the installation onto a drive which will also be used with MacOS, select **Re-install sets or install additional sets** 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.

   If **sysinst** reports

   ```
   I can not find any hard disk for use by NetBSD
   ```

   or the drive you wish to install onto is missing, then you should look at the FAQ entry

   http://www.netbsd.org/Ports/macppc/faq.html#nodisk

8. *Partitioning the disk*

   You can skip a few steps, down to 'Getting the distribution sets', if you are installing onto a drive that will be used with MacOS (i.e. you selected **Re-install sets or install additional sets** from the main menu).

   • *Editing the NetBSD disklabel*

     The partition table of the NetBSD part of a disk is called a *disklabel*. In actuality, NetBSD/macppc uses an Apple Partition Map. The installer creates something like a real Apple Partition Map, but it is not compatible with MacOS, which is one of the reasons why you cannot use this installer to partition a disk that can be used with MacOS.

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, g 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.

9. *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.

   Note: The bootstrapping code installed in this step will *not* boot a machine with Open Firmware 3. You will still need to have ofwboot.xcf on an HFS or HFS+ partition.

10. *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'.

11. *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 address, 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 address 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.

12. *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 address, 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 address instead of a hostname for the NFS server.

13. *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.

14. *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. Remember, NetBSD/macppc doesn't grok HFS or HFS+ partitions

15. *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.

16. *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.

    If you are installing using the **Re-install sets or install additional sets** option, then you will need to create the device nodes in /dev, otherwise, 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 UTC. Finally you can set a password for the "root" account, to prevent the machine coming up without access restrictions.

17. *Making the device nodes*

    If you are installing using the **Re-install sets or install additional sets** option, then you will need to create the device nodes in /dev now. Otherwise, skip this step.

    Go to the main installation menu, and select **Utility menu** and then select the **Run /bin/sh** option, which will give you a shell prompt. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

    ```
    # stty erase '^h'
    # stty erase '^?'
    ```

    Type the following command (replacing wd0a with the partition name of your destination root partition): Now you need to mount your destination root partition:

    ```
    # mount /dev/wd0a /mnt
    # cd /mnt/dev
    # sh MAKEDEV all
    # cd /;umount /mnt
    ```

```
# exit
```

18. *Finalizing your installation*

Congratulations, you have successfully installed NetBSD 1.5.3. 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 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
0 > reset-all
```

The last command resets the system so that these settings are stored. Replace **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
```

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.xcf
0 > setenv boot-file ide0/disk@0:13,/netbsd
0 > reset-all
```

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 (**sysinst** usually will), 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 /bin/sh prompt. If you are asked for a terminal type, respond with **vt100** (or whatever is appropriate for your terminal type) **vt220** (or whatever is appropriate for your terminal type) and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

   ```
   # stty erase '^h'
   # stty erase '^?'
   ```

At this point, you need to configure at least one file in the /etc directory. You will need to mount your root filesystem read/write with:

/sbin/mount -u -w /

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=vt100

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 <int> network interface, where your on-board interfaces may be one of mc0, bm0, de0 or gm0 mc00 For example:

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. Unless you've set a password in **sysinst**, there is no initial password. 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. Also, you may want to read through the NetBSD/macppc X11R6 FAQ.

   http://www.netbsd.org/Ports/macppc/x11.html

   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,
     usually in the 1.5.3/macppc/All subdir. You can install them with the following commands:

     ```
     # export PKG_PATH=\
     ftp://ftp.netbsd.org/pub/NetBSD/packages/1.5.3/macppc/All\;\
     ftp://ftp.netbsd.org/pub/NetBSD/packages/1.5/macppc/All
     # pkg_add -v tcsh
     # pkg_add -v cvs
     # pkg_add -v apache
     # pkg_add -v perl
     ...
     ```

     The above commands will install the tcsh shell, the CVS source code management system, the Apache web server and the perl programming language as well as all the packages they depend on.
   - Package sources for compiling packages on your own 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), with the commands:

     ```
     # mkdir /usr/pkgsrc
     # ( cd /usr/pkgsrc ; tar -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. Don't forget to 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 1.5.3 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. Since installation of the bootloader will prevent MacOS from using the disk and will prevent Open Firmware 3 machines from booting, there is an option to bypass this step. Make sure you know whether or not to install the bootloader.

The upgrade procedure using the **sysinst** tool is similar to an installation, but without the hard disk partitioning. The original /etc directory is renamed to /etc.old, and no attempt is made to merge any of the previous configuration into the new system except that the previous /etc/fstab file is copied into the new configuration. 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 1.5.3 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 1.5.3.

General issues

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

  Prior to NetBSD 1.5, /etc/rc was a traditional BSD style monolithic file; 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 upgrade from NetBSD 1.4 or prior

- 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 sunlamp 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
Reinoud Zandijk
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.
Precedence Technologies Ltd
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:

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**
Simon Burge	simonb@netbsd.org	**pmax**
Jeremy Cooper	jeremy@netbsd.org	**sun3x**
Matt Fredette	fredette@netbsd.org	**sun2**
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 1.5.3 Release Engineering team:

Chris G. Demetriou	cgd@netbsd.org
Håvard Eidnes	he@netbsd.org
Ted Lemon	mellon@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:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@netbsd.org
Julian Assange	proff@netbsd.org
Lennart Augustsson	augustss@netbsd.org
Christoph Badura	bad@netbsd.org
Bang Jun-Young	junyoung@netbsd.org
Dieter Baron	dillo@netbsd.org
Robert V. Baron	rvb@netbsd.org
Jason Beegan	jtb@netbsd.org
Erik Berls	cyber@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Rafal Boni	rafal@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org
Allen Briggs	briggs@netbsd.org
Aaron Brown	abrown@netbsd.org
Andrew Brown	atatat@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@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
Alistair Crooks	agc@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
Tracy Di Marco White	gendalia@netbsd.org
Jaromír Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Emmanuel Dreyfus	manu@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Håvard Eidnes	he@netbsd.org
Stoned Elipot	seb@netbsd.org
Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Jason R. Fink	jrf@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
Tetsuya Isaki	isaki@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
Masanori Kanaoka	kanaoka@netbsd.org
KAWAMOTO Yosihisa	kawamoto@netbsd.org
Mario Kemper	magick@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
Martti Kuparinen	martti@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
Tomasz Luchowski	zuntum@netbsd.org
Federico Lupi	federico@netbsd.org
Brett Lymn	blymn@netbsd.org
Paul Mackerras	paulus@netbsd.org
MAEKAWA Masahide	gehenna@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Jared D. McNeill	jmcneill@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
NISHIMURA Takeshi	nsmrtks@netbsd.org
NONAKA Kimihiro	nonaka@netbsd.org
Jesse Off	joff@netbsd.org
Tatoku Ogaito	tacha@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
Chris Pinnock	cjep@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Michael Rauch	mrauch@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
SOMEYA Yoshihiko	someya@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
TAMURA Kent	kent@netbsd.org
Shin'ichiro TAYA	taya@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
YAMAMOTO Takashi	yamt@netbsd.org
Yuji Yamano	yyamano@netbsd.org
Reinoud Zandijk	reinoud@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:

er 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 Reinoud Zandijk.

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.

The End