NAME

INSTALL – Installation procedure for NetBSD/alpha.

CONTENTS

About this Document....2
What is NetBSD?....2
Upgrade path to NetBSD 1.5.3....2
Major Changes Between 1.5.2 and 1.5.3....3
Known Caveats with 1.5.3....4
Major Changes Between 1.5.1 and 1.5.2....4
Major Changes Between 1.5 and 1.5.1....4
Changes Between The NetBSD 1.4 and 1.5 Releases....6
  Kernel....6
  Networking....7
  File system....7
  Security....7
  System administration and user tools....8
  Miscellaneous....8
The Future of NetBSD....9
Sources of NetBSD....9
NetBSD 1.5.3 Release Contents....9
  NetBSD/alpha subdirectory structure....11
  Bootable installation/upgrade floppies....11
  Binary distribution sets....12
NetBSD/alpha System Requirements and Supported Devices....13
  Supported PCI bus devices....13
  Supported ISA bus devices....14
  Supported EISA bus devices....14
  Supported Turbochannel bus devices....14
Getting the NetBSD System on to Useful Media....16
Preparing your System for NetBSD installation....18
Installing the NetBSD System....18
  Running the sysinst installation program....21
    Introduction....21
    Possible PCMCIA issues....21
    General....22
    Quick install....22
    Booting NetBSD....23
    Network configuration....24
    Installation drive selection and parameters....24
    Partitioning the disk....24
    Preparing your hard disk....25
    Getting the distribution sets....25
    Installation using ftp....25
    Installation using NFS....25
    Installation from CD-ROM....25
    Installation from an unmounted file system....26
    Installation from a local directory....26
    Extracting the distribution sets....26

Finalizing your installation.....26
Manual and script-assisted installation 26
Post installation steps.....30
Upgrading a previously-installed NetBSD System.....32
Compatibility Issues With Previous NetBSD Releases.....33
General issues 33
Issues affecting an upgrade from NetBSD 1.4 or prior 33
Using online NetBSD documentation.....33
Administrivia.....34
Thanks go to.....34
We are.....37
Legal Mumbo-Jumbo.....42
The End.....43

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5.3 on the *alpha* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

`.more` The enhanced text format used on UNIX-like systems by the `more`(1) and `less`(1) pager utility programs. This is the format in which the on-line *man* pages are generally presented.

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on thirty-one different system architectures featuring twelve distinct families of CPUs, and is being ported to more. The NetBSD 1.5.3 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. Ultra-DMA 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)), NeoMagic 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.

There have been many alpha-specific enhancements since the 1.4 release. These include:

- The following new system types are supported:
  - API UP1000 (AMD 751-based) EV6 systems
  - 264DP, XP1000, DS10, DS20, API UP2000 and other Tsunami-based EV6 systems
  - DECpc AXP 150 (Jensen)
- Ability to boot off RAIDframe RAID-1 (mirrored) FFS partitions.
- Ability to boot off LFS partitions.

- ddb(4) traceback code added.
- Emulate user program use of BWX instructions on CPUs which don't support them. Handle unaligned accesses caused by BWX instructions.
- Improved TGA graphics support, accelerating the text mode and adding support for the: 8bpp TGA2, and 32bpp ZLXp-E2 and -E3.

NetBSD 1.5.3 on alpha is, as usual, also fully backward compatible with old NetBSD/alpha binaries, so you don't need to recompile all your local programs.

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/alpha subdirectory structure

The alpha-specific portion of the NetBSD 1.5.3 release is found in the alpha subdirectory of the distribution:

```
.../NetBSD-1.5.3/alpha/
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/        alpha binary distribution sets; see below.
installation/
             floppy/      alpha boot and installation floppies; see below.
             diskimage/
             tapeimage/   an image file cdhdtape is included for the case where the installer is
                          written to a CD, hard drive, or tape. This image file is the same for the
                          CD, HD, and tape cases, but a separate tapeimage/ directory exists
                          to hold a copy of the README file and to meet the NetBSD re-
                          lease(7) standard.
             instkernel/  contains a netbsd.gz installation kernel; this is the same installer
                          kernel as in all the other install images, but without the various boot
                          program and file system wrappers. It can be netbooted or diskbooted
```

from a previous installation. no need to **gunzip** this image.

`misc/` contains `GENERIC.fs`, a `GENERIC` kernel in a bootable file system image. This is used in some unusual installations as described in the next section. Miscellaneous alpha installation utilities; see installation section, below.

Bootable installation/upgrade floppies

There are three bootable images in the NetBSD alpha distribution. One is for a dual-floppy boot and is split into two separate files. The other is a single-file image containing the same install kernel, but intended to be written to a CD, tape, or hard drive. The third image is a `GENERIC` kernel intended for production use in unusual cases. This can be useful at some sites when:

- You want to run diskless but SRM bugs prevent the firmware from netbooting. You can work around this problem by always booting the generic kernel from the floppy.
- SRM doesn't recognize your (hard) disk controller but NetBSD does. This happens more frequently than you might think. SRM will usually only boot from `ncr`(4) or `isp`(4) SCSI devices, and on most platforms will not boot from an IDE drive. NetBSD will happily operate with almost any SCSI root or an IDE root; the solution here is to netboot a kernel or always boot from floppy.

Binary distribution sets

The NetBSD alpha binary distribution sets contain the binaries which comprise the NetBSD 1.5.3 release for the alpha. There are eight binary distribution sets. The binary distribution sets can be found in the `alpha/binary/sets` subdirectory of the NetBSD 1.5.3 distribution tree, and are as follows:

base The NetBSD 1.5.3 alpha **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.
22.7 MB gzipped, 59.8 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.
18.1 MB gzipped, 81.2 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.
3.1 MB gzipped, 7.6 MB uncompressed

kern This set contains a NetBSD/alpha 1.5.3 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
2.1 MB gzipped, 5.0 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.
1.4 MB gzipped, 5.1 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.
3.6 MB gzipped, 13.7 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
2.6 MB gzipped, 13.9 MB uncompressed

xcontrib Programs that were contributed to X.
0.2 MB gzipped, 0.8 MB uncompressed

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

xserver
2.3 MB gzipped, 9.2 MB uncompressed

The alpha 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 alpha binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/alpha System Requirements and Supported Devices

NetBSD/alpha 1.5.3 runs on most of the DEC Alpha PCI platforms, on all of the TURBOChannel models, on the high end 8200 and 8400 systems, and on the 4100 series.

The SRM console is required. This console can be distinguished from the ARC console (which is used to boot Windows NT) by the fact that it has a command line interface, rather than a menu-driven interface. The SRM prompt is '>>>'.

Some platforms have both the SRM console and the ARC console, and can switch between them, and other platforms have only one type of console loaded at any one time. If your system comes up with the ARC firmware, it may be possible to switch it to SRM with a menu or to download SRM from
ftp://gatekeeper.dec.com/pub/Digital/Alpha/firmware/index.html.
You may want to buy a firmware update CD from Compaq Computer Corporation.

More information on supported platforms and devices can be found on the alpha port web pages at **http://www.netbsd.org/**.

A basic system will fit on a 200 MB disk (including swap) without too much difficulty, but you will probably want at least 500 MB of disk to have any level of comfort.

Although it is actually possible to boot and install NetBSD/alpha in only 16 MB of RAM, you will want to have at least 32 MB. We support add-in devices on the PCI, ISA, EISA and TurboChannel buses. Because NetBSD has an extremely machine-independent device driver system, many device drivers are the same as used in other ports that use the same bus. For example, the *de* network card driver is shared by the i386 and alpha ports. Some drivers on inspection appear as if they will work on the alpha but have not been tested because that hardware was not available to NetBSD testers; these are marked as *UNTESTED* below. If you have one of these devices, and it does work, please get in touch with `port-alpha-maintainer@netbsd.org` and let us know that it works. If it doesn't work, do the same thing and we can probably fix it pretty easily.

Supported PCI bus devices

- Graphics Adapters
  - VGA-compatible video (*pcivga*)
  - ZLXp-E1 DECchip 21030-based video (*tga*)
  - ZLXp-E2 and ZLXp-E3 video (*tga*)
- Network Cards
  - DECchip 21x40-family 10 and 100 Mbps Ethernet (*de*, *tlp*)
  - DEC DEFPA FDDI (*fpa*)
  - PCI LANCE Ethernet (*le*; *UNTESTED*)
  - Efficient Networks ENI-155p ATM (*en*; *UNTESTED*)
  - 3Com 3c59x and 3c90x (except 3c906) 10 and 100 Mbps Ethernet (*ep*)
  - Intel EtherExpress Pro 10/100B PCI Ethernet (*fxp*)
  - SMC EPIC/100 Fast Ethernet boards (*epic*)
- SCSI Controllers
  - Adaptec 291x, 2920, 2930C, 294x, 295x, 39xx, 19160, 29160 and AIC-78xx SCSI (*ahc*)
  - BusLogic 9xx SCSI (*bha*, Works on alpha PC164)
  - Qlogic ISP 10x0-family SCSI (*isp*)
  - NCR/Symbios 53c8xx-family SCSI (*ncr*, *siop*; The latter is preferred. NCR825 Doesn't always work)
- Miscellaneous Devices
  - Cyclades Cyclom-Y serial boards (*cy*; *UNTESTED*)
  - PCI-PCI bridges (*ppb*; Tested with the DECchip 21050, but should work with all bridges and system firmware revisions that comply with the PCI-PCI bridge specification)

Supported ISA bus devices

- Graphics Adapters
  - VGA-compatible video (*vga*; Text console only)
- Network Cards
  - 3Com 3c509 Ethernet (*ep*)
  - DEC DE200,DE201,DE202 (*le*)
  - DEC DE203,DE204,DE205 (*lc*)
- Miscellaneous Devices
  - PC-style parallel ports (*lpt*)
  - NS16450 and NS16550 UARTs (*com*)
  - ISA multi-port 16x50 boards (*ast*, *boca*; Only the latter has been tested)

Supported EISA bus devices

- Network Cards
  - DEC DEFEA FDDI (*fea*)
  - 3Com 3c5xx series (*ed*; *UNTESTED*)
- SCSI Controllers
  - Adaptec 274x and aic7770 SCSI (*ahc*; *UNTESTED*)
  - BusLogic 7xx SCSI (*bha*; *UNTESTED*)

Supported Turbochannel bus devices

- Graphics Adapters
  - CFB video (PMAG-BA, *cfb*)
  - SFB video (PMAGB-BA, *sfb*)

  Note: Although these boards are supported by NetBSD/alpha since there is no keyboard or mouse support available for the TurboChannel systems, they aren't very useful. XXX: is this still true now that the MI sfb.c is used???
- Network Cards
  - DEC LANCE Ethernet (PMAD-AA, *le*; *UNTESTED*)
  - DEC DEFTA FDDI (PMAF-F, *fta*)

Note that some devices, especially ISA-based devices, have to have certain settings set properly for the install and `GENERIC` kernels to detect them. (Once installed, you can always rebuild your own kernel to detect them anywhere you wish, of course.) Here is a list of such devices and the necessary settings:

```
Device          Name    Port    IRQ     DRQ     Misc
------          ----    ----    ---     ---     ----
Serial ports    com0    0x3f8   4               [8250/16450/16550/clones]
                com1    0x2f8   3               [8250/16450/16550/clones]
                com2    0x3e8   5               [8250/16450/16550/clones]

Parallel ports  lpt0    0x378   7               [interrupt-driven or polling]
                lpt1    0x278                   [polling only]
                lpt2    0x3bc                   [polling only]

AHA-174x SCSI host adapters (in enhanced mode)
                ahb0    any     any     any

AHA-2X4X or AIC-7xxx-based SCSI host adapters
                ahc0    any     any     any

Bus Logic BT445, BT74x, or BT9xx SCSI host adapters
                bha0    0x330   any     any
                bha1    0x334   any     any

MFM/ESDI/IDE/RLL hard disk controllers
                wdc0    0x1f0   14              [supports two devices]
                wdc1    0x170   15              [supports two devices]

ATA disks       wd0, wd1, ...
SCSI disks      sd0, sd1, ...
SCSI tapes      st0, st1, ...
```

```
SCSI and ATAPI CD-ROMs cd0, cd1, ...
        For each SCSI and IDE controller found, the SCSI or ATA(PI) devices
        present on the bus are probed in increasing ID order for SCSI and
        master/slave order for ATA(PI). So the first SCSI drive found will
        be called sd0, the second sd1, and so on ...

3COM 3x59X or 3COM 3x90X PCI Ethernet boards
                ep0     any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

Intel EtherExpress 100 Fast Ethernet adapters
                fxp0    any     any     [you must assign an interrupt in your
                                         PCI BIOS, or let it do so for you]

DEC DE200,201,202 EtherWORKS II/Turbo ISA Ethernet boards
                le?    0x300   5       memory at D0000-DFFFF
                le?    0x200   10      memory at D8000-DFFFF

        You should enter the following SRM console command to enable the
        le device:
        >>> isacfg -mk -slot ? -dev 0 -handle DE200-LE -irq0 5
            -membase0 d0000 -memlen0 10000 -iobase0 300 -etyp 1 -enadev 1

DEC DE203,204,205 EtherWORKS III ISA Ethernet boards
                lc0    0x300   any
                lc1    0x320   any

        You should enter the following SRM console command to enable
        the device:
        >>> add_de205
```

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- CD-ROM
- MS-DOS floppy
- FTP
- Remote NFS partition
- Tape
- Existing NetBSD partitions, if performing an upgrade

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.

MS-DOS floppy Count the number of `set_name.`*xx* files that make up the distribution sets you want to install or upgrade. You will need one sixth that number of 1.44 MB floppies.

Format all of the floppies with MS-DOS. Do *not* make any of them bootable MS-DOS floppies, i.e. don't use `format /s` to format them. (If the floppies are bootable, then the MS-DOS system files that make them bootable will take up some space, and you

won't be able to fit the distribution set parts on the disks.) If you're using floppies that are formatted for MS-DOS by their manufacturers, they probably aren't bootable, and you can use them out of the box.

Place all of the `set_name.xx` files on the MS-DOS disks.

Once you have the files on MS-DOS disks, 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.

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.

Tape

To install NetBSD from a tape, you need to make a tape that contains the distribution set files, in 'tar' format.

If you're making the tape on a UNIX-like system, the easiest way to do so is probably something like:

```
# tar -cf tape_device dist_directories
```

where *tape_device* is the name of the tape device that describes the tape drive you're using; possibly `/dev/rst0`, or something similar, but it will vary from system to system. (If you can't figure it out, ask your system administrator.) In the above example, *dist_directories* are the distribution sets' directories, for the distribution sets you wish to place on the tape. For instance, to put the **misc, base,** and **etc** distributions on tape (in order to do the absolute minimum installation to a new disk), you would do the following:

```
# cd .../NetBSD-1.5.3
# cd alpha/binary
# tar -cf tape_device misc etc kern
```

Note: You still need to fill in *tape_device* in the example.

Once you have the files on the tape, 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.

Upgrade

If you are upgrading NetBSD, you also have the option of installing NetBSD by putting the new distribution sets somewhere in your existing file system, and using them from there. To do that, you must do the following:

Place the distribution sets you wish to upgrade somewhere in your current file system tree. Please note that the `/dev` on the floppy used for upgrades only knows about `wd0`, `wd1`, `sd0`, `sd1`, and `sd2`. If you have more than two IDE drives or more than three SCSI drives, you should take care not to place the sets on the high-numbered drives.

At a bare minimum, you must upgrade the **base** and **kern** binary distributions, and so must put the **base** and **kern** sets somewhere in your file system. If you wish, you can do the other sets, as well, but you should *not* upgrade the **etc** distribution; it contains contains system configuration files that you should review and update by hand.

Once you have done this, you can proceed to the next step in the upgrade process, actually upgrading your system.

Preparing your System for NetBSD installation

If you have any data on your disks that you want to keep, *back it up* before starting. Note that NetBSD/alpha does not support booting more than one operating system from a single disk, although it's fine to have multiple operating systems on your machine if you have a separate disk for NetBSD, or if one of them uses a network boot.

Installing the NetBSD System

Installation of NetBSD/alpha is now easier than ever! For the latest news, problem reports, and discussion, join the port-alpha mailing list by mailing a line saying

```
subscribe port-alpha
```

to `majordomo@netsbd.org`. Also, see **http://www.netbsd.org** for more information.

If you encounter any problems, please report them via the mailing list or the `send-pr`(1) program so that they can be fixed for the next release.

To install or upgrade NetBSD, you need to first boot an installation program and then interact with the screen-menu program **sysinst**. The installation program actually consists of the NetBSD kernel plus an in-memory file system of utility programs.

The usual procedure is to write the installation system to a floppy disk set and then boot from the floppies, however, there are now six ways to boot the NetBSD/alpha installation system! Each approach loads the exact same installation bits. The six paths are:

- Floppy disk boot
- CD boot
- Hard Drive Boot
- Magnetic Tape Boot
- Existing Root FS Boot
- Network boot

In all cases, you need to transfer a bootable image of the installation system from the NetBSD CD or from an ftp site to the chosen media type. Although booting from floppy is the usual path, the hard drive boot is useful if you have another operating system (and a spare drive) already installed, or if you don't mind swapping hard drives from box to box. CD and tape boots are nice and fast if you have a CD writer or a tape format in common with another previously installed UNIX-like system. Finally, most versions of SRM can locate the NetBSD boot program **netboot** via bootp and download it via tftp, **netboot** then mounts the root file system (/) via NFS and loads the kernel.

Note that if you are installing or upgrading from a writable media, the media can be write-protected if you wish. These systems mount a root image from inside the kernel, and will not need to write to the media. If you booted from a floppy, the floppy disk may be removed from the drive after the system has booted.

- Floppy disk boot

  The 3.5", 1.44 MB boot floppy set is found under the NetBSD/alpha 1.5.3 distribution directory in `alpha/installation/floppy/` as two files called `disk1of2` and `disk2of2`. You need to put these two disk images on two floppy disks.

  If you have a UNIX-like system handy, you can do this with commands like the following:

  ```
  # dd if=disk1of2 of=/dev/rfd0a bs=18k
  # dd if=disk2of2 of=/dev/rfd0a bs=18k
  ```

  If the UNIX-like system you are using is not a NetBSD system, you will probably need to replace `/dev/rfd0a` with the name of the floppy device on your particular system.

  If you have an MS-DOS or Windows system available, you can use the **rawrite.exe** utility to transfer the image to a floppy disk. This utility is provided with the NetBSD/i386 install tools, under `i386/installation/misc`; a documentation file, **rawrite.doc** is available there as well.

  Once the floppy has been made, you simply need to put it in the drive and type

  ```
  >>> B DVA0
  ```

- CD boot
- Hard Drive boot
- Magnetic Tape Boot

  All three of these media types use the same initial image: `.../installation/diskimage/cdhdtape` The image can be written to a hard drive partition

with a command like:

```
# dd if=cdhdtape bs=16k of=/dev/rsd0c
```

To boot from a magnetic tape device such as DAT or DLT, it is important to create the tape image with 512-byte records. Use a command like:

```
# dd if=cdhdtape bs=512 of=/dev/rst0
```

If the host system is not NetBSD, the names of the destination devices are likely to be different. Be sure to use a "raw partition" device that doesn't skip over labels!

The use of CD-R devices varies greatly depending on the host OS and host software; it isn't possible to give typical instructions here.

- Existing Root FS Boot

  The installation subdirectory `instkernel/` contains `netbsd.gz`; this is the same install kernel but without a bootable file system image wrapped around it. You can perform an complete reinstall by beginning it as an upgrade, and booting this kernel in the normal way off the root file system (`/`) of a previous installation.

  The gzipped image can be booted directly; it is not necessary to uncompress it first.

- Network Boot

  Booting NetBSD/alpha 1.5.3 over a network requires a BOOTP or DHCP server, a TFTP server and an NFS server. (These are usually all run on the same machine.) There are three basic stages to the boot:

  – alpha console software sends a BOOTP request to get its own address, the address of the TFTP server and the file to download. It downloads this file, which is the second stage bootstrap, via TFTP and then executes it.

  – The secondary boot program resends the BOOTP request, this time also locating the NFS server and root path. It mounts the root path via NFS and reads in and transfers to the kernel: `/netbsd`.

  – The kernel probes and configures the devices, and then sends out another BOOTP request so it can find out its address, the NFS server, and path. It then mounts its root (`/`) via NFS and continues.

  You will need to set up servers for BOOTP, TFTP and NFS.

  If you want to run a full system from the network, untar the NetBSD snapshot or distribution into a directory on your server and NFS export that directory to the client. Make sure you put a kernel there as well, and create the device nodes in `/dev` with **`sh ./MAKEDEV all`**. Detailed instructions on netbooting can be found by visiting the NetBSD alpha platform page: **http://www.netbsd.org/Ports/alpha**

  At the time of this release, the URL for the netbooting instructions is:
  http://www.netbsd.org/Documentation/network/netboot/

  You'll want to map root to `root` (rather than the default `nobody`) when you export your root file system (`/`). A typical `/etc/exports` line on a NetBSD system would be:

  ```
  /usr/export/alpha -maproot=0 myclient.mydom.com
  ```

  One option is to load just the install kernel over the network but then proceed to a normal disk-based install and disk-based operation. (Perhaps the alpha doesn't have a floppy drive, or you just don't want to use a Windows system to make the floppy; we understand.)

  For this case, you still need to export an NFS root, but the only thing it needs to have in it is the `instkernel` image from the distribution.

The gzipped image can be booted directly; it is not necessary to uncompress it first.

The console will be using TFTP to load the NetBSD boot program, so for the TFTP setup, you need to copy the second stage bootstrap, **netboot**, into an appropriately named file such as `boot.netbsd.alpha` in the directory used by your TFTP server. If you extracted a full snapshot, you can get the netboot program from `/usr/mdec/netboot`; if not, you can get this from the `installation/netboot` directory where you found the alpha distribution.

For the BOOTP server you need to specify the:

- hardware type (Ethernet)
- hardware address (Ethernet MAC address)
- IP address of the client
- subnet mask of the client
- address of of the TFTP/NFS server
- name of the second stage bootstrap loaded via TFTP
- path to the root for the client (mounted via NFS)

Here's an example for a UNIX-like system running **bootpd**:

```
myhost.mydom.com:        :ht=ethernet:ha=0000c0391ae4:\
        :ip=192.168.1.2:sm=255.255.255.0:\
        :sa=192.168.1.1:bf=boot.netbsd.alpha:rp=/usr/export/alpha:
```

And here's an example for a UNIX-like system running the ISC **dhcpd**:

```
host axp {
        hardware ethernet 0:0:c0:39:1a:e4;
        fixed-address 192.168.1.2;
        option host-name "myhost.mydom.com";
        filename "boot.netbsd.alpha";
        option root-path "/usr/export/alpha";
        option domain-name-servers 192.168.1.1;
        option broadcast-address 255.255.255.0;
        option domain-name "my.domain";
}
```

The only Ethernet device the console on most alpha systems knows how to boot from is the onboard Ethernet interface or a DEC Tulip (21040, 21041, 21140) PCI Ethernet card. Some older SMC 100 Mbps cards that use this chip have been known to work as well. Many older systems will not be able to use the newer 2.0 stepping of the 21140, however. If your system appears not to be receiving packets, this may be the problem. (You may or may not be able to update your firmware to fix this; see the alpha port pages on www.netbsd.org for more information on this.) In general, 10 Mb cards from manufacturers other than DEC will work, and 100 Mb cards not from DEC will not.

Once you're set up, you should be able to boot with:

```
>>> boot -proto bootp ewa0
```

You should permanently set your protocol to BOOTP with:

```
>>> set ewa0_protocols bootp
```

The 3000 series of Turbochannel systems and certain other models use *old SRM*, do not have a **-proto** option and use different device names. They also tend to not netboot very well so you probably don't need to worry about this section. However, if you want to give it a try, note the following differences:

- There is no **-proto** argument, or **ewa0_protocols** variable. Old SRM uses bootp if the device name is given as **ez0**.
- The use of the setnetbootinfo(8) program will probably also be necessary, as it is unlikely that an SRM from that era will properly communicate the ethernet HW address to the boot program.
- Example:

```
>>> boot ez0
```

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. *Possible* PCMCIA *issues*

   Machines with PCMCIA slots may have problems during installation. With the improvements of the PCMCIA code in this release, this will not happen very frequently. If you do not have PCMCIA on your machine (PCMCIA is only really used on laptop machines), you can skip this section, and ignore the "[PCMCIA]" notes. If you do have PCMCIA in your machine, you can safely ignore this section and the "[PCMCIA]" the first time, as you are likely to not have problems. Should troubles occur during floppy boot, they may be PCMCIA specific. You should then re-read this section and try again, following the instructions in the "[PCMCIA]" notes.

   This section explains how to work around the installation problem.

   The kernel keeps careful track of what interrupts and I/O ports are in use during autoconfiguration. It then allows the PCMCIA devices to pick unused interrupts and I/O ports. Unfortunately, the INSTALL kernel may not detect all devices in your system. This may be because the INSTALL kernel only supports the minimum set of devices to install NetBSD on your system, or it may be that NetBSD does not have support for the device causing the conflict.

   For example, suppose your laptop has a soundblaster device built in; the INSTALL kernel has no sound support. The PCMCIA code might allocate your soundblaster's IRQ and I/O ports to PCMCIA devices, causing them not to work, or to lock up the system. This is especially bad if one of the devices in question is your ethernet card.

   As of NetBSD 1.5, the kernel attempts to probe for available interrupts that are usable by the PCIC (PCMCIA interrupt controller). Assuming that this functions correctly, it should alleviate interrupt conflicts; however, I/O port conflicts are still possible.

   This problem will impact some, but not all, users of PCMCIA. If this problem is affecting you, watch the "[PCMCIA]" notes that will appear in this document.

3. *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.

4. *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).
     - Two floppy disks. One 1.44 MB 3.5" floppy.
     - A CD-ROM drive (SCSI or ATAPI), a harddisk and a minimum of 32 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.
   - Creating the boot floppies. You can create the floppies needed for installation under MS-DOS or Windows. Supposing your 1.44 MB floppy drive is drive A:, and your CD is drive E: do the following from an MS-DOS command prompt:

     ```
     e:
     cd \NetBSD-1.5.3\i386\installation\misc
     rawrite
     ```

     When asked for a source filename, answer
     (...alpha installation root)`\floppy\disk1of2`

     When asked for a destination drive answer '**a**'.
   - To create a bootfloppy under NetBSD or other UNIX-like system, you would type something like:

     ```
     # dd if=.../boot1.fs bs=18k of=/dev/rfd0a
     ```
   - The Quick Installation
     - Insert the first boot floppy you just created. Boot the computer. Type

       ```
       >>> B DVA0
       ```

       The main menu will be displayed.

       ```
       .*********************************************.
       * 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**, after you have removed the bootfloppy from the 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.xfree86.org/**.

5. *Booting NetBSD*

   [PCMCIA] Unplug your PCMCIA devices, so that they won't be found by NetBSD.

   Boot your machine. The boot loader will start, and will print a countdown and begin booting.

   If the boot loader messages do not appear in a reasonable amount of time, you either have a bad boot floppy or a hardware problem. Try writing the install floppy image to a different disk, and using that.

   It will take a while to load the kernel from the floppy, probably around a minute or so, then, the kernel boot messages will be displayed. This may take a little while also, as NetBSD will be probing your system to discover which hardware devices are installed. 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.

   Note that once the system has finished booting, you need not leave the floppy in the disk drive.

   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.

6. *Network configuration*

   [PCMCIA] You can skip this section, as you will only get data from floppy in the first part of the install.

   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.

7. *Installation drive selection and parameters*

   To start the installation, select **Install NetBSD to hard disk** 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`.

8. *Partitioning the disk*

   - Which portion of the disk to use.

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

   - *Editing the NetBSD disklabel*

     The partition table of the NetBSD part of a disk is called a *disklabel*. There are 3 layouts for the NetBSD part of the disk that you can pick from: **`Standard, Standard with X`** and **`Custom`**. The first two use a set of default values (that you can change) suitable for a normal installation, possibly including X. The last option lets you specify everything yourself.

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

     `a` Root partition. (`/`),

     `b` Swap partition.

     `c` The entire disk.

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

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

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.

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.

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.

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

17. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 1.5.3. You can now reboot the machine, and boot NetBSD from harddisk.

Manual and script-assisted installation

All of the installation procedures consist of putting a label on the disk to provide information on the sizes and placement of the partitions into which the disk is divided, putting the boot blocks on the disk, creating the file systems on the partitions, and unpacking the distribution tar archives.

1. Disk prep: label, boot block, and file system setup

   A. Manual Install from the Shell Prompt

      The normal installation involves running the install shell script and interactively configuring the file systems, and then simply unpacking the tar files into these followed by running MAKEDEV.

      However, as stated above it is also possible to do the installation yourself from the shell, and in any case it is helpful to understand what the install script does. The procedure is:

      - create `/etc/disktab`, see `disktab`(5)
      - run `disklabel`(8),
      - run `newfs`(8)
      - `mount`(8) the new root on `/mnt`
      - cd to `/usr/mdec` and run `installboot`(8)

      If you are reviewing man pages on NetBSD platforms other than alpha, be sure that when reading `installboot`(8) you read the alpha version by typing:

      ```
      # man 8 alpha/installboot
      ```

      At this point you need only unpack the distribution sets by running `tar`(1) as described below.

   B. `/install` and `/upgrade` traditional installation scripts

      The `install` and `upgrade` scripts are still there, so by exiting the sysinst program you can type `install` or `upgrade` at the shell prompt and run them as you did in the good old days.

      You may install on either a SCSI or an IDE disk; you will be prompted for the disk to install on. The disks in your system will be numbered starting at *x*`d0` (where *x* is an '`s`' for SCSI disks, '`w`' for IDE disks) based on the SCSI-ID or IDE drive order; if you have more than one disk, watch the boot messages carefully to see which ones are probed as which numbers.

      Once you've selected a disk to install on, you'll be prompted for the geometry. This is also displayed in the boot messages, and you'll be given a chance to review the boot messages again to get the exact figures for the number of cylinders, heads and sectors.

      After this you must specify the size of your partitions. Generally you'll be giving the sizes in cylinders; the install program will tell you how many bytes there are in each cylinder.

      The swap partition is the second thing you specify, after the `/` (root) partition. Regardless of the size of your disk, you'll want to specify a swap partition that's at least as large as the amount of RAM you have, and probably not less than 64 MB in any case.

If you have a small disk (under 500 MB), it's probably best to devote all of the disk (excepting 64 MB or more for the swap) to the / (root) partition.

If you have more space, we recommend devoting at least 32 MB, and preferably 48 MB, to the / (root) partition. /usr will need 150 MB or so if you're not installing X, 200 MB or so if you are. A typical organization is 50 MB for / (root), 150-250 MB for swap, and the remaining space for /usr. With enough swap space configured, you can make /tmp a nice, fast mfs. See mount_mfs(8), and note that the mfs will require swap space for the largest planned amount of /tmp storage. It doesn't return space when files are deleted, but just keeps it its own freelist so the swap space required is equal to the highwater mark of /tmp use, plus space required to back up main memory and store inactive images.

Once you've specified this information, the install script will write the disklabel, install boot blocks to make the disk bootable, initialise the file systems, and mount them all under /mnt. You are now ready to go on to the next step.

2. Configuration: arranging access to the distribution sets

After doing the disk and file system setup with either shell commands or the script assist, you then need only unpack the distribution sets with the tar(1) command. To do this you will need access from the target host to the tar files that contain the operating system in order to extract them to your disk. This is done via an NFS or FTP transfer over a network, via a CD-ROM archive, a tape archive, or by preloading an accessible hard drive with the necessary tar files.

- Preparing to Install from a CD-ROM

  All you need to do is mount the CD-ROM, which will generally be device cd0. (The initial boot messages will tell you what the CD-ROM drive is probed as.) This would be done with:

  ```
  # mount -r -t cd9660 /dev/cd0a /mnt2
  ```

- Preparing to Install from the Network

  The first thing you need to do is configure the loopback network interface, which is done with the command

  ```
  # ifconfig lo0 127.0.0.1
  ```

  Then you will have to configure your Ethernet card. The command

  ```
  # ifconfig -l
  ```

  will give you a list of the network interfaces on your system. It will show you your ethernet cards first, followed by lo0 (the loopback interface that we configured above), ppp0 (the PPP interface) and sl0 (the SLIP interface).

  To configure your ethernet card, type **ifconfig** *if* inet *addr* [netmask] [media *media*] where *if* is the network card (interface), almost always de0, *addr* is the IP address, the optional *netmask* parameter is the network mask, and the optional *media* parameter is one of:

10base2	BNC connector	10 Mbps
AUI	AUI connector	10 Mbps
10baseT/UTP	Twisted pair connector	10 Mbps
100baseTX	Twisted pair connector	100 Mbps
100baseFX	Fibre-optic connector	100 Mbps
100baseT4	T4 twisted pair interface	100 Mbps

  If the host you are getting the data files from is not on the local network, you will also have to configure a gateway into your system. Do this with

```
# route add default <gateway-IP-address>
```

One improvement over the good old days is that the resolver is now present; by configuring `/etc/resolv.conf` you can get name resolution during any install NFS or FTP operations.

Once networking has been configured, you may mount the directory with the install files via NFS, or download them via FTP.

To mount them via nfs, type

```
# mount -t nfs <hostname:/path/to/nfs/volume> /mnt2
```

If this volume has been exported read-only, you may need the **-r** option to mount.

To download the install sets with ftp, create a directory in which to put them and then use the ftp client to download them. Mirror sites are listed at: **http://www.netbsd.org/Sites/net.html** A typical session might be:

```
# mkdir /mnt/usr/release
# cd /mnt/usr/release
# ftp ftp.netbsd.org
```

the following commands are given to the **ftp** *program after logging in*

```
ftp> prompt
ftp> cd /pub/NetBSD/NetBSD-1.5.3/alpha/binary/sets
ftp> mget *
ftp> quit
```

Feel free, of course, to leave off the sets that you don't need if you don't plan to install everything.

3. Unpack distribution sets: Extracting the Operating System Files

   Change to the root directory of your hard drive (which is `/mnt` if you've used the standard install script to this point) by typing

   ```
   # cd /mnt
   ```

   For this and the following commands, replace `/mnt/usr/release/` with the path to your NFS volume or CD-ROM if that's how you chose to access your install files instead.

   The sets and kernel are extracted with:

   ```
   # cd /mnt
   # for i in base kern comp etc games man misc text; do
       tar -zxpf /mnt/usr/release/$i.tgz;
   # done
   ```

   or perhaps:

   ```
   # cd /mnt
   # for i in /mnt/usr/release/*.tgz; do
       echo $i
       tar -zxpf $i
   # done
   ```

   Now make the device nodes:

   ```
   # cd /mnt/dev
   # sh ./MAKEDEV all
   ```

4. Restart your system

   Unmount the file systems and halt. The exact instructions to type here will depend on the file systems you created, but typically the commands are:

   ```
   # cd /
   # umount /mnt/usr
   # umount /mnt
   # sync
   ( sync is not strictly necessary but it is traditional )
   # halt
   ```

   You should now be at the SRM console's >>> prompt and can reboot into the new configuration (possibly after an optional power cycle) with a command such as:

   ```
   >>> boot dka0
   ```

   This command might be: **boot dka100** if your drive is on ID 1. You can usually use **show device** to see a full list of bootable devices in your system. Your system will come up in single-user mode, ready for you to configure it.

You can create the floppy needed for installation under MS-DOS or Windows. Supposing your 1.44 MB floppy drive is drive A:, and your CD is drive **E:**, do the following from an MS-DOS command prompt:

```
e:
cd \NetBSD-1.5.3\installation\misc
rawrite
```

When asked for a source filename, answer

```
...\floppy\disk1of2
```

When asked for a destination drive answer

```
a
```

(Repeat the procedure for `installation/floppy/disk2of2`.)

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 **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=vt220
```

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

Other values that need to be set in `/etc/rc.conf` for a networked environment are *hostname* and possibly *defaultroute*, furthermore add an *ifconfig_int* for your <int> network interface, along the lines of

```
ifconfig_de0="inet 123.45.67.89 netmask 255.255.255.0"
```

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

```
ifconfig_de0="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.

   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/alpha/All` subdir. You can install them with the following commands:

  ```
  # export PKG_PATH=\
  ftp://ftp.netbsd.org/pub/NetBSD/packages/1.5.3/alpha/All\;\
  ftp://ftp.netbsd.org/pub/NetBSD/packages/1.5/alpha/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 have the boot floppy set available. You must also have at least the **base** and **kern** binary distribution sets available, so that you can upgrade with them, using one of the upgrade methods described above. Finally, you must have sufficient disk space available to install the new binaries. Since the old binaries are being overwritten in place, you only need space for the new binaries, which weren't previously on the system. If you have a few megabytes free on each of your root (`/`) and `/usr` partitions, you should have enough space.

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

The upgrade procedure using the **sysinst** tool is similar to an installation, but without the hard disk partitioning. 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),
- `apropos`(1),
- `passwd`(1), and
- `passwd`(5).

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

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

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

Administrivia

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

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

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

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

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

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

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

  for their ongoing work on BSD systems, support, and encouragement.

- Also, our thanks go to:

  Mike Hibler
  Rick Macklem
  Jan-Simon Pendry
  Chris Torek

  for answering lots of questions, fixing bugs, and doing the various work they've done.

- UC Berkeley's Experimental Computing Facility provided a home for sun-lamp in the past, people to look after it, and a sense of humor. Rob Robertson, too, has added his unique sense of humor to things, and for a long time provided the primary FTP site for NetBSD.

- Vixie Enterprises for hosting the NetBSD FTP, SUP, and WWW servers.

- Redback Networks, Inc. for hosting the NetBSD mail and GNATS server.

- The Helsinki University of Technology in Finland for hosting the NetBSD CVS server.

- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.

- The many organisations that provide NetBSD mirror sites.

- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.

- Dave Burgess `burgess@cynjut.infonet.net` has been maintaining the 386BSD/NetBSD/FreeBSD FAQ for quite some time, and deserves to be recognized for it.

- The following individuals and organizations (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Steve Allen
  Jason Birnschein
  Mason Loring Bliss

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

Jim Wise
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:

This product includes software developed by the University of California, Berkeley and its contributors. This product includes software developed by The NetBSD Foundation, Inc. This product includes software developed by the NetBSD Foundation, Inc. and its contributors. This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. This product includes software developed by Adam Glass and Charles Hannum. This product includes software developed by Adam Glass and Charles M. Hannum. This product includes software developed by Adam Glass. This product includes software developed by Alistair G. Crooks. This product includes software developed by Amancio Hasty and Roger Hardiman. This product includes software developed by Berkeley Software Design, Inc. This product includes software developed by Bill Paul. This product includes software developed by Charles D. Cranor and Washington University. This product includes software developed by Charles D. Cranor. This product includes software developed by Charles Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors. This product includes software developed by Charles Hannum. This product includes software developed by Charles M. Hannum. This product includes software developed by Chris Provenzano. This product includes software developed by Christian E. Hopps. This product includes software developed by Christopher G. Demetriou for the NetBSD Project. This product includes software developed by Christopher G. Demetriou. This product includes software developed by Christos Zoulas. This product includes software developed by David Jones and Gordon Ross. This product includes software developed by Dean Huxley. This product includes software developed by Eric S. Hvozda. This product includes software developed by Ezra Story. This product includes software developed by Gardner Buchanan. This product includes software developed by Gordon Ross. This product includes software developed by Gordon W. Ross and Leo Weppelman. This product includes software developed by Gordon W. Ross. This product includes software developed by Hauke Fath. This product includes software developed by HAYAKAWA Koichi. This product includes software developed by Hellmuth Michaelis

and Joerg Wunsch. This product includes software developed by Herb Peyerl. This product includes software developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. This product includes software developed by Hubert Feyrer for the NetBSD Project. This product includes software developed by Iain Hibbert. This product includes software developed by Ian W. Dall. This product includes software developed by Ignatios Souvatzis for the NetBSD Project. This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/. This product includes software developed by Joachim Koenig-Baltes. This product includes software developed by Jochen Pohl for The NetBSD Project. This product includes software developed by John Polstra. This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project. This product includes software developed by Jonathan Stone for the NetBSD Project. This product includes software developed by Jonathan Stone. This product includes software developed by Julian Highfield. This product includes software developed by Kenneth Stailey. This product includes software developed by Leo Weppelman. This product includes software developed by Lloyd Parkes. This product includes software developed by Manuel Bouyer. This product includes software developed by Marc Horowitz. This product includes software developed by Mark Brinicombe. This product includes software developed by Mark Tinguely and Jim Lowe. This product includes software developed by Markus Wild. This product includes software developed by Martin Husemann and Wolfgang Solfrank. This product includes software developed by Mats O Jansson and Charles D. Cranor. This product includes software developed by Mats O Jansson. This product includes software developed by Matthias Pfaller. This product includes software developed by Michael L. Hitch. This product includes software developed by Niels Provos. This product includes software developed by Paul Kranenburg. This product includes software developed by Paul Mackerras. This product includes software developed by Peter Galbavy. This product includes software developed by Philip A. Nelson. This product includes software developed by Rodney W. Grimes. This product includes software developed by Roland C. Dowdeswell. This product includes software developed by Rolf Grossmann. This product includes software developed by Scott Bartram. This product includes software developed by SigmaSoft, Th. Lockert. This product includes software developed by Tatoku Ogaito for the NetBSD Project. This product includes software developed by Terrence R. Lambert. This product includes software developed by Theo de Raadt and John Brezak. This product includes software developed by Theo de Raadt. This product includes software developed by Tohru Nishimura for the NetBSD Project. This product includes software developed by TooLs GmbH. This product includes software developed by Winning Strategies, Inc. This product includes software developed by Zembu Labs, Inc. This product includes software developed by the Center for Software Science at the University of Utah. This product includes software developed by the Computer Systems Laboratory at the University of Utah. This product includes software developed by the University of Calgary Department of Computer Science and its contributors. This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman. This product includes software developed for the FreeBSD project. This product includes software developed for the Internet Software Consortium by Ted Lemon. This product includes software developed for the NetBSD Project by Frank van der Linden. This product includes software developed for the NetBSD Project by Jason R. Thorpe. This product includes software developed for the NetBSD Project by John M. Vinopal. This product includes software developed for the NetBSD Project by Matthias Drochner. This product includes software developed for the NetBSD Project by Matthieu Herrb. This product includes software developed for the NetBSD Project by Perry E. Metzger. This product includes software developed for the NetBSD Project by Piermont Information Systems Inc. This product includes software developed for the NetBSD Project by Ted Lemon. This product includes software developed by LAN Media Corporation and its contributors. This product includes software developed by Michael Graff for the NetBSD Project. This product includes software developed by Niklas Hallqvist, C Stone and Job de Haas. This product includes software developed by Eric Young (eay@mincom.oz.au). This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). This product includes software developed by the University of Oregon. This product includes software developed by the University of Southern California and/or Information Sciences Institute. This product includes software developed by Internet Initiative Japan Inc. This product includes software developed by Reinoud Zandijk.

In the following statement, "This software" refers to the parallel port driver:

> This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.

The End