NAME

INSTALL — Installation procedure for NetBSD/i386.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 4.0.1 on the *i386* platform. It is available in four different formats titled `INSTALL.`*ext*, where *.ext* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

Quick install notes for the impatient

This section contains some brief notes describing what you need to install NetBSD 4.0.1 on a machine of the i386 architecture.

- Fetch the appropriate pair of boot floppy images from the `i386/installation/floppy/` directory, or a CD-ROM image (the images from `i386/installation/cdrom/` are bootable, but do not contain binary sets - these are intended for network installs, or when different bootblocks are needed). Most people will need the `boot1.fs` and `boot2.fs` or `boot.iso` images for VGA console installation, or the `boot-com1.fs` and `boot-com2.fs` or `boot-com.iso` images for installation via serial console. You may also possibly (but not necessarily) want to use `bootlap1.fs` and `bootlap2.fs` if installing on a laptop. The INSTALL_LAPTOP kernel can be loaded from one of the cdrom images via the bootloader: **boot** *nblaptop*.

  The default kernel on ISO images have ACPI enabled. This is known to cause issues on a few older machines which have buggy ACPI tables. To boot with ACPI disabled, interrupt the NetBSD boot loader, and enter: **boot -c**. At the userconf prompt, enter: **disable acpi** followed by: **quit**. The kernel from the two floppy set is also present on the ISO images and can be loaded using: **boot** *nbnoacpi*.

- The actual binary distribution is in the `i386/binary/sets/` directory. When you boot the install floppies, the installation program can fetch these files for you (using e.g. ftp), if you have a network connection. There are several other methods to get the binary sets onto your machine.

  You will at a minimum need one of the kernel sets, typically `kern-GENERIC.tgz`, as well as `base.tgz` and `etc.tgz`. In a typical workstation installation you will probably want all the installation sets.

- Write the floppy images directly to a pair of floppies. If you have problems writing a raw image to a floppy, the **rawrite.exe** MS-DOS program or the **Rawrite32.exe** Windows32 program (inside `rawrite32.zip`) in the `i386/installation/misc/` directory may be of help.

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

- For third-party programs which are not part of the base NetBSD distribution, you will want to explore the **pkgsrc** system with its more than 7200 program packages.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on fifty four different system architectures (ports), featuring seventeen machine architectures across fifteen distinct CPU families, and is being ported to more. The NetBSD 4.0.1 release contains complete binary releases for many different system architectures. (A few ports are not fully supported at this time and are thus not part of the binary distribution. Please see the NetBSD web site at **http://www.NetBSD.org/** for information on them.)

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

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

Changes Between The NetBSD 4.0 release and 4.0.1 update

The NetBSD 4.0.1 update is the first security/critical update of the NetBSD 4.0 release branch. This represents a selected subset of fixes deemed critical in nature for stability or security reasons.

These fixes will also appear in future releases (NetBSD 4.1), together with other less-critical fixes and feature enhancements.

The complete list of changes can be found in the CHANGES-4.0.1:

ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-4.0.1/CHANGES-4.0.1

file in the top level directory of the NetBSD 4.0.1 release tree.

Security Advisories Fixes

- NetBSD-SA2008-004, `bzip2`(1) Multiple issues (CVE-2008-1372 and CVE-2005-0953), has been fixed by upgrading bzip2 to 1.0.5.
- NetBSD-SA2008-005, OpenSSH Multiple issues (CVE-2008-1483 and CVE-2008-1657), has been fixed by applying patches from upstream.
- NetBSD-SA2008-006, Integer overflow in `strfmon`(3) function (CVE-2008-1391), has been fixed.
- NetBSD-SA2008-008, OpenSSL Montgomery multiplication (CVE-2007-3108), has been fixed.
- NetBSD-SA2008-009, BIND cache poisoning (CVE-2008-1447 and CERT VU#800113), has been fixed by updating BIND to 9.4.2-P2. Note there are two related changes to this advisory:
  - The default behavior of ipfilter's Port Address Translation has been changed to using random port allocation rather than sequential mappings, to avoid decreasing the randomness of source ports used for DNS queries which affects the BIND cache poisoning problem.
  - A '`query-source`' statement, which could allow the BIND cache poisoning attack, has been commented out in the default `named.conf`(5) file.
- NetBSD-SA2008-010, Malicious PPPoE discovery packet can overrun a kernel buffer (CVE-2008-3584), has been fixed.
- NetBSD-SA2008-011, ICMPv6 MLD query (CVE-2008-2464), has been fixed.
- NetBSD-SA2008-012, Denial of Service issues in `racoon`(8) (CVE-2008-3652), has been fixed by upgrading ipsec-tools to release 0.7.1. Note this also fixes CVE-2008-3651.
- upcoming NetBSD-SA2008-013, IPv6 Neighbor Discovery Protocol routing vulnerability (CVE-2008-2476), has been fixed.
- upcoming NetBSD-SA2008-014, Remote cross-site request forgery attack issue in `ftpd`(8) (CVE-2008-4247), has been fixed.
- upcoming NetBSD-SA2008-015, Remote kernel panics on IPv6 connections (CVE-2008-3530), has been fixed.

Note: NetBSD-SA2008-007 and advisories prior to NetBSD-SA2008-004 don't affect NetBSD 4.0.

Other Security Fixes

- Fix a buffer overrun which could crash a FAST_IPSEC kernel.
- `tcpdump`(8): Fix CVE-2007-1218, CVE-2007-3798 and CAN-2005-1278 in base-tcpdump.
- Fix a buffer overflow of PCF font parser in X11 libXfont library (CVE-2008-0006).
- Fix a buffer overflow of Tektronix Hex Format support in binutils (CVE-2006-2362).
- `machfb`(4), `voodoofb`(4): Introduce two missing KAUTH_GENERIC_ISSUSER check in the `mmap`(2) code.

Networking

- Update root.cache to 2008020400 version.
- Fix IP packet forwording code to make sure to send a reasonable fragment size when IPSEC is configured.
- Fix a bug in TCP SACK code which causes data corruption.
- Fix an `rc.d`(8) script for `amd`(8) not to shutdown gracefully since it seems to cause problems for more people than the old (also broken) behavior.
- `ftpd`(8): Fix and reorganize PAM support.

Libraries

- Pthread support of BIND has been disabled for future binary compatibility after removal of the scheduler activations.
- Fix coredump of gdtoa (conversion between binary floating-point and ASCII string) functions on out of memory conditions.

Drivers

- `fxp`(4): Fix random pool corruption and hangup problems.
- `wd`(4): Handle more LBA48 bug quirks on some Hitachi's SATA/IDE drives.

Miscellaneous

- Disable a NULL pointer check in zlib for standalone programs. This fixes errors on loading a gzipped kernel (including installation kernels) on several ports (news68k etc.) whose kernels are loaded at address zero.
- `awk`(1): Bring back an accidentally removed fix to allow escape of a newline in string literals.
- `gcc`(1):
  - Fix compilation of native sh3 gcc on 64-bit build machines.
  - Fix an internal compiler error on compiling m68k softfloat or m68010 targets on 64-bit build machines.
- `zgrep`(1): Make '**`-h`**' option (suppress filenames on output when multiple files are searched) actually work.
- Fix parallel build failure on building hpcarm, hpcmips and hpcsh releases.

Platforms specific

- acorn32: Fix a bootloader problem on some RiscPCs.
- cobalt:
  - Add a workaround to avoid panic on probing a multi function PCI device on Qube's PCI slot.
  - Fix a bug in the interrupt handler which causes network freeze if more than one interfaces are used.
- hp700: Fix potential kernel / userland memory corruption in `copyinstr`(9) and `copyoutstr`(9).
- sparc64: Fix a bug in `locore.s` which causes unexpected behavior.
- sun3: Fix a bug which might cause an occasional panic during boot.

- vax: Make syscall handler use proper `copyin`(9) function on parsing syscall args.

Changes Between The NetBSD 3.0 and 4.0 Releases

The NetBSD 4.0 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 user-land 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 more than a year of development that went into the NetBSD 4.0 release. The complete list of changes can be found in the CHANGES:

ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-4.0/CHANGES

and CHANGES-4.0:

ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-4.0/CHANGES-4.0

files in the top level directory of the NetBSD 4.0 release tree. Some highlights include:

Networking

- `agr`(4): new pseudo-device driver for link level aggregation.
- IPv6 support was extended with an RFC 3542-compliant API and added for `gre`(4) tunnels and the `tun`(4) device.
- A NDIS-wrapper was added to use Windows binary drivers on the i386 platform, see `ndiscvt`(8).
- The IPv4 source-address selection policy can be set from a number of algorithms. See "IPSRCSEL" in `options`(4) and `in_getifa`(9).
- Imported `wpa_supplicant`(8) and `wpa_cli`(8). Utilities to connect and handle aspects of 802.11 WPA networks.
- Imported `hostapd`(8). An authenticator for IEEE 802.11 networks.
- `carp`(4): imported Common Address Redundancy Protocol to allow multiple hosts to share a set of IP addresses for high availability / redundancy, from OpenBSD.
- ALTQ support for the PF packet filter.
- `etherip`(4): new EtherIP tunneling device. It's able to tunnel Ethernet traffic over IPv4 and IPv6 using the EtherIP protocol specified in RFC 3378.
- `ftpd`(8) can now run in standalone mode, instead of from `inetd`(8).
- `tftp`(1) now has support for multicast TFTP operation in open-loop mode, server is in progress.
- `tcp`(4): added support for RFC 3465 Appropriate Byte Counting (ABC) and Explicit Congestion Notification as defined in RFC 3168.

File systems

- `scan_ffs`(8), `scan_lfs`(8): utilities to find FFSv1/v2 and LFS partitions to recover lost disklabels on disks and image files.
- tmpfs: added a new memory-based file system aimed at replacing mfs. Contrary to mfs, it is not based on a disk file system, so it is more efficient both in overall memory consumption and speed. See `mount_tmpfs`(8).
- Added UDF support for optical media and block devices, see `mount_udf`(8). Read-only for now.
- NFS export list handling was changed to be filesystem independent.

- LFS: lots of stability improvements and new cleaner daemon. It is now also possible to use LFS as root filesystem.
- `vnd`(4): the vnode disk driver can be used on filesystems such as smbfs and tmpfs.
- Support for System V Boot File System was added, see `newfs_sysvbfs`(8) and `mount_sysvbfs`(8).

Drivers

- Audio:
  - Support for new models on drivers such as Intel ICH8/6300ESB, NVIDIA nForce 3/4, etc.
  - Added support for AC'97 modems.
  - `auich`(4): added support to handle the AC'97 modem as audio device, enabled with the kernel option "AUICH_ATTACH_MODEM"
  - `azalia`(4): added support for S/PDIF.
- Hardware Monitors:
  - `amdpm`(4): added support for the i2c bus on the AMD-8111 used on many Opteron motherboards and for the Analog Devices ADT7464 hardware monitor chip.
  - `adt7467c`(4): new driver for Analog Devices ADT7467 and ADM1030 hardware monitor chips.
  - `ipmi`(4): new driver for motherboards implementing the Intelligent Platform Management Interface 1.5 or 2.0, from OpenBSD.
  - `it`(4): new driver for iTE 8705F/8712F and SiS 950 hardware monitors.
  - The `lm`(4) driver was rewritten and support for more chips was added, for example for Winbond W83627HF, W83627THF, W83627DHG and Asus AS99127F.
  - `owtemp`(4): new driver for the 1-Wire temperature sensors.
  - `tmp121temp`(4): new driver for the Texas Instruments TMP121 temperature sensor.
  - `ug`(4): new driver for Abit uGuru hardware monitor found on newer Abit motherboards.
- Miscellaneous:
  - `geodewdog`(4): new AMD Geode SC1100 Watchdog Timer driver.
  - `gscpcib`(4): new AMD Geode SC1100 PCI-ISA bridge that provides support for the GPIO interface.
- Networking:
  - `ath`(4): updated HALs with support for WiSOC (AR531x) and 32bit SPARC.
  - `bge`(4): added support for the following chips: BCM5753, BCM5753M, BCM5715, BCM5754, BCM5755 and BCM5787. Numerous improvements and bugfixes were made too.
  - `kse`(4): new driver for Micrel KSZ8842/8841 PCI network cards.
  - `msk`(4): new driver for Marvell Yukon 2 GigE PCI network cards, from OpenBSD.
  - `nfe`(4): new driver for NVIDIA nForce Ethernet network cards, from OpenBSD.
  - `ral`(4): new 802.11 driver for PCI/Cardbus Ralink RT2500, RT2501, RT2600, RT2661 and RT2500 USB chipsets, from OpenBSD.

  - rum(4): new 802.11 driver for USB Ralink RT2501 and RT2601 chipsets, from OpenBSD.
  - sip(4): now works on sparc64.
  - tlp(4): added support for ASIX AX88140A and AX88141.
  - vr(4): added support for the VIA Rhine III.
  - wm(4): added support for i8003, ICH8, ICH9 and others. Support for IPv6 Rx TCP/UDP Checksum Offloading and more.
  - wpi(4): new driver for Intel PRO/Wireless 3945ABG PCI 802.11 network cards, from OpenBSD.
- Security:
  - glxsb(4): new driver for the AMD Geode LX AES Security Block that provides random numbers and AES acceleration, from OpenBSD.
- Power Management:
  - Support for Intel Speedstep SMI on PIIX4 PCI-ISA for i386.
  - Support for AMD PowerNow and Cool'n'Quiet Technology on K7 and K8 CPUs (both in 32 and 64 bit mode), including Athlon Mobile, Athlon64, Opteron or X2. See options(4) for more information.
  - Support for more Enhanced Speedstep CPUs, including VIA C7/Eden and Intel Core Solo/Duo/Duo2. See options(4) for more information.
  - The Enhanced Speedstep and PowerNow drivers were modified to be able to be scaled in all CPUs available, saving power on SMP systems.
- Storage:
  - ahcisata(4): new driver for AHCI 1.0 and 1.1 compliant SATA controllers.
  - ataraid(4): added support to handle Adaptec HostRAID and VIA V-Tech software RAID.
  - ciss(4): new driver for HP/Compaq 5th+ generation Smart ARRAY controllers, from OpenBSD.
  - fdc(4): added support for SBus based sparc64 machines and fixed formatting on sparc.
  - gcscide(4): new driver for the AMD Geode CS5535 Companion Device IDE controller.
  - jmide(4): new driver for JMicron Technology JMB36x PCIe to SATA II/PATA controllers.
  - mfi(4): new driver for LSI Logic and Dell MegaRAID SAS controllers, from OpenBSD.
  - mpt(4): added support for newer SAS and similar devices.
  - njata(4): new driver for Workbit NinjaATA-32 CardBus IDE controller.
  - pdcsata(4): added support for the Promise PDC20775, PDC20771, PDC40518, PDC40718 and some bugfixes.
  - piixide(4): added support for some ICH8/ICH8-M/ICH9 IDE and SATA controllers.
  - svwsata(4): new driver for Serverworks K2 SATA controllers, from OpenBSD.
  - viaide(4) added support for the VIA VT8237A SATA controller and AMD CS5536 Companion Device IDE Controller.
- USB:

- `ucycom`(4): new driver for Cypress microcontroller based serial devices.
- `uipaq`(4): new driver for the iPAQ devices.
- `uslsa`(4): new driver for Silicon Labs CP210x series serial adapters.
- `utoppy`(4): new driver for the Topfield TF5000PVR range of digital video recorders.

Platforms

- i386:
  - Added support for the for Multiboot specification. This means much improved support for loading the kernel by GRUB, including passing in parameters to the kernel.
  - Added the unichromefb framebuffer driver that supports the VIA Unichrome Graphics adapter.
  - `vesafb`(4): added new framebuffer driver that supports VESA BIOS (VBE) 2.0 extensions and up.
  - Added cd9660 file system support to the BIOS bootloader.
- evbarm: new platform support for Arcom Viper PXA255-based single board, Atmark Techno Armadillo-9 and Armadillo-210, Certance CP-3100, Linksys NSLU2 (a.k.a. "Slug") and I-O DATA HDL-G Giga LANDISK NAS devices.
- evbmips: added support for Alchemy Au1550 processors, DBAu1550 boards, Alchemy Au15XX PCI host, (OMS-AL400/128) and Atheros AR5312 SoC.
- New port ews4800mips: NEC's MIPS based EWS4800 workstations.
- cobalt: added support for booting off raidframe RAID1 mirrors.
- hpcmips: added the `teliosio`(4) driver for the Sharp Telios LCD screen and Battery unit.
- New port landisk: port to the SH4 processor based NAS appliances, supporting models by I-O DATA (USL-5P, HDL-U, HDL-AV, HDL-W and HDLM-U series, SuperTank LAN Tank, UHDL-160U and UHDL-300U) and Plextor PX-EH16L, PX-EH25L and PX-EH40L.
- macppc: this port has gained support to use accelerated wsdisplay drivers by default (if possible), and uses the appropriate driver rather than the Generic Open Firmware Framebuffer.
- prep: this port has been modernized, and support for five additional machines has been added, among them the IBM 7024-E20 and 7025-F30 models and Motorola Powerstack E1. Additionally, sysinst support was added, and the bootloader process was improved, allowing easy installation and upgrade to future releases.
- sparc: added support for booting off raidframe RAID1 mirrors.
- Xen: support for Xen3 domU and dom0 (Unprivileged domain and domain 0), including support for hardware virtualization on CPUs that support it.

Kernel subsystems

- Improved Firewire (IEEE1394) support imported from FreeBSD.
- The `midi`(4) framework got a complete overhaul for better support of Active Sensing and improved handling of tempo and timebase changes.
- Added a Bluetooth protocol stack including:
  - hardware drivers: `ubt`(4) for USB controllers, and `bt3c`(4) for the 3Com Bluetooth PC-Card.

- socket based access to the HCI, L2CAP, RFCOMM and SCO protocols.
- pseudo drivers for integrating services on remote Bluetooth devices such as Keyboards, Mice and SCO Audio into the NetBSD device framework.

See `bluetooth`(4), `bthset`(1) and `btpin`(1).

- Imported the `bio`(4) framework from OpenBSD, to query/control block hardware RAID device controllers. Currently supporting the `mfi`(4) driver.
- Kernel uses stateful read-ahead algorithm.
- `dkctl`(8) can be used to switch buffer queuing strategies on the fly on `wd`(4) disks, see also `bufq`(9).
- `fileassoc`(9) is used by Veriexec, it adds in-kernel and file-system independent file meta-data association interface.
- `firmload`(9): an API for loading firmware images used by various hardware devices.
- `gpio`(4): imported General Purpose I/O framework from OpenBSD.
- `onewire`(4): imported Dallas Semiconductor 1-wire bus framework from OpenBSD.
- The `proplib`(3) protocol was added for sending property lists to/from the kernel using ioctls.
- `spi`(4): new SPI (Serial Peripherial Interface) framework.
- `timecounter`(9) adds a new time-keeping infrastructure along with NTP API 4 nanokernel implementation. Almost all platforms were changed to support this API.
- Start of 32bit-Linux-emulation for amd64 (COMPAT_LINUX32).
- `wscons`(4) console driver supports splash screens, scrolling, progress bar for kernel and boot messages.

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.

Security

- The FAST_IPSEC IPsec implementation was extended to use hardware acceleration for IPv6, in addition to the hardware accelerated IPv4 that was available before. See `fast_ipsec`(4) for more information.
- `mprotect`(2) got restrictions to enforce W^X policies, from PaX. See `options`(4), `sysctl`(3), and `paxctl`(1).
- GCC 4's support for stack smashing protection (SSP) was enabled by adding libssp, see `security`(8).
- The kernel authorization framework `kauth`(9) was added, replacing the traditional BSD credential management and privileged operation access control with an abstract layer, allowing the implementation of various security models either as part of the NetBSD distribution or as third-party LKMs.

  NetBSD's kernel authorization is a hybrid clean-room implementation of a similar interface developed by Apple, extending its capabilities and combining concepts of credential inheritance control.

Userland

- 3rd party software updates:
  - BIND 9.4.1-P1
  - OpenSSL 0.9.8e
  - CVS 1.11.22

  - OpenSSH 4.4
  - gettext 0.14.4
  - PF from OpenBSD 3.7
  - (n)awk 20050424
  - Postfix 2.4.5
  - am-utils 6.1.3
  - file 4.21
  - zlib 1.2.3
  - GNU binutils 2.16.1
  - GNU groff 1.19.2
  - IPFilter 4.1.23
  - GNU gcc 4.1.2 prerelease
  - GNU gdb 6.5 (some architectures)
  - NTP 4.2.4p2
  - pppd 2.4.4
- `cdplay`(1): added digital transfer mode support.
- `cksum`(1) can now verify checksums.
- `csplit`(1): new utility that splits a file into pieces. From FreeBSD/OpenBSD.
- `identd`(1): added support for forwarding ident queries and receiving of proxied ident queries.
- `getent`(1): added support for the ethers database.
- `gkermit`(1): new program for transferring files using the Kermit protocol.
- `mail`(1): added support for Mime and multi-character set handling, command line editing and completion.
- `utoppya`(1): new utility to interface to the `utoppy`(4) driver.
- `init`(8): added support for running multi-user in a chroot() environment. Allows / file system on e.g., `cgd`(4), `vnd`(4) or `ccd`(4) volumes.
- `gpt`(8): new GUID partition table maintenance utility, from FreeBSD.
- iSCSI target (server) code added, see `iscsi-target`(8); Initiator (client) code is underway.
- `lockstat`(8): new command to display a summary of kernel locking events recorded over the lifetime of a called program.
- `ofctl`(8): new command to display the OpenPROM or OpenFirmware device tree for the macppc, shark and sparc64.
- Various utilities to support Bluetooth were added:
  - `btconfig`(8) for controller configuration.
  - `btdevctl`(8) to manage pseudo devices relating to remote services.

- `bthcid`(8) and `btpin`(1) for authenticating radio connections.
- `sdpd`(8) for providing service discovery to remote devices.
- `sdpquery`(1) for querying services on remote devices.
- `rfcomm_sppd`(1) to access remote services over RFCOMM via stdio or pty.
- `bthset`(1) for making connections to Bluetooth headsets.

Besides this list, there have also been innumerable bug fixes and other miscellaneous enhancements of course.

Components removed from NetBSD

In this release of NetBSD, some software components known from previous releases were removed from the system. In some cases those were components that are not useful anymore or their utility does not justify maintenance overhead. Other components were not working properly and there was lack of interest in fixing them.

- Sushi was removed from the base system due to lack of interest and maintenance. If you really want it, it is available in the CVS repository at othersrc/usr.sbin/sushi. However, be warned that it is unmaintained and is most likely out of date.
- Vinum was removed due to lack of interest and maintenance. At the time of removal, it had several known serious issues (including not being compilable). RAIDframe provides similar functionality. If you were using Vinum you will need to back up your data, delete the Vinum partitions, create RAIDframe partitions, and restore your data to them. Details about RAIDframe can be found in `raid`(4), `raidctl`(8), and the NetBSD Guide:
  http://www.NetBSD.org/docs/guide/en/chap-rf.html.
- Sendmail was removed. Postfix is the MTA and provides the `sendmail`(1) command line tool. Postfix has been included with NetBSD since NetBSD 1.5 was released in December 2005. Details about Postfix can be found in the NetBSD Guide:
  http://www.NetBSD.org/docs/guide/en/chap-mail.html.
  For those who need Sendmail, it is available from pkgsrc in the `mail/sendmail` and `mail/sendmail813` packages.
- NETCCITT and NETNS were removed due to lack of interest and maintenance. They had known serious issues (including being out of date with respect to other network code) and there were no known users at the time of their removal. Unfortunately, there is no replacement or option for them.
- UUCP was removed. The NetBSD improvements were merged into the pkgsrc version. For those who use UUCP tools, they are available from pkgsrc in the `net/uucp` package. The `cu`(1) command is available as a frontend to `tip`(1).
- The Fortran 77 compiler (g77) has been removed with the transition from GCC 3 to GCC 4, which does not include it. For those who need it, it is available from pkgsrc in the `lang/gcc3-f77` package.
- The evbsh5 port has been removed from NetBSD due to lack of interest, compounded by a lack of available SH5 hardware.

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 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/mirrors/.**

NetBSD 4.0.1 Release Contents

The root directory of the NetBSD 4.0.1 release is organized as follows:

`.../NetBSD-4.0.1/`

`CHANGES`	Changes between NetBSD 3.0 and branching 4.0.
`CHANGES-4.0`	Changes since NetBSD 4.0 was branched.
`CHANGES-4.0.1`	Changes between NetBSD 4.0 and 4.0.1.
`CHANGES.prev`	Changes in earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 4.0.1 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 4.0.1 has a binary distribution.

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.
95 MB gzipped, 484 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.
6 MB gzipped, 25 MB uncompressed

src This set contains all of the base NetBSD 4.0.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
45 MB gzipped, 214 MB uncompressed

syssrc This set contains the sources to the NetBSD 4.0.1 kernel for all architectures; `config`(1); and `dbsym`(8).
33 MB gzipped, 165 MB uncompressed

xsrc This set contains the sources to the X Window System.
95 MB gzipped, 502 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. Except for the **pkgsrc** set, which is traditionally unpacked into `/usr/pkgsrc`, all sets may be unpacked into `/usr/src` with the command:

```
# cd / ; tar -zxpf set_name.tgz
```

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 -a MD5 *file*.

`SHA512` SHA512 digests for the various files in that directory, in the format produced by the command:
cksum -a SHA512 *file*.

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

The SHA512 digest is the safest checksum, followed by the MD5 digest, and finally the POSIX checksum. The other two checksums are provided only to ensure that the widest possible range of systems can check the integrity of the release files.

NetBSD/i386 subdirectory structure

The i386-specific portion of the NetBSD 4.0.1 release is found in the `i386` subdirectory of the distribution: `.../NetBSD-4.0.1/i386/`. It contains the following files and directories:

```
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.
  - `netbsd-GENERIC.MP.gz`
    A gzipped NetBSD kernel containing code for everything supported in this release. This kernel also supports SMP on systems with more than one CPU.
  - `netbsd-GENERIC.NOACPI.gz`
    A gzipped NetBSD kernel containing code for everything supported in this release but without ACPI support, for use on legacy hardware.
  - `netbsd-GENERIC_DIAGNOSTIC.gz`
    A gzipped NetBSD kernel containing code for everything supported in this release, with diagnostic messages enabled.
  - `netbsd-GENERIC_LAPTOP.gz`
    A version of `GENERIC` that has USB, PCMCIA and CardBus enabled to allow installing on laptop machines.
  - `netbsd-INSTALL.gz`
    A somewhat smaller installation kernel, which you can use to boot the system on memory-tight systems. This is the same kernel as present on the miniroot and on many install floppies.
  - `netbsd-INSTALL_LAPTOP.gz`
    A version of `INSTALL` that has USB, PCMCIA and CardBus enabled to allow installing on laptop machines.
  - `netbsd-INSTALL_LARGE.gz`
    A version of `INSTALL` that has ACPI and some large disk and network drivers, to allow installing on the newest machines. This is the default boot kernel from the ISO images.
- `sets/` i386 binary distribution sets; see below.

`installation/`
- `floppy/` i386 boot and installation floppies; see below.
- `cdrom/` i386 bootable cdrom images; see below.
- `misc/` Miscellaneous i386 installation utilities; see installation section, below.

Binary distribution sets

The NetBSD i386 binary distribution sets contain the binaries which comprise the NetBSD 4.0.1 release for the i386. The binary distribution sets can be found in the `i386/binary/sets` subdirectory of the NetBSD 4.0.1 distribution tree, and are as follows:

base The NetBSD 4.0.1 i386 **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.
24 MB gzipped, 69 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.
26 MB gzipped, 89 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.
1 MB gzipped, 1 MB uncompressed

games This set includes the games and their manual pages.
3 MB gzipped, 8 MB uncompressed

kern-GENERIC
This set contains a NetBSD/i386 4.0.1 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
5 MB gzipped, 10 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.
8 MB gzipped, 33 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
4 MB gzipped, 12 MB uncompressed

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.
3 MB gzipped, 11 MB uncompressed

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on XFree86, and tightly track XFree86 releases. They are currently equivalent to XFree86 4.5.0. Binary sets for the X Window System are distributed with NetBSD. The sets are:

xbase The basic files needed for a complete X client environment. This does not include the X servers.
6 MB gzipped, 18 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
11 MB gzipped, 39 MB uncompressed

xfont Fonts needed by the X server and by X clients.
31 MB gzipped, 39 MB uncompressed

xetc Configuration files for X which could be locally modified.
0.03 MB gzipped, 0.17 MB uncompressed

xserver The X server. This includes all XFree86 X servers. Because all of them are included, this set is large. However, you will only need one of the servers provided in this set. (Typically, **`XFree86`**).
9 MB gzipped, 25 MB uncompressed

The i386 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 filenames stored in the sets are relative and therefore the files are extracted *below the current directory*. Therefore, 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 the root directory (`/`) of your system. This utility is used only in a Traditional method installation.

Note: Each directory in the i386 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/i386 System Requirements and Supported Devices

NetBSD 4.0.1 runs on ISA (AT-Bus), EISA, MCA, PCI, and VL-bus systems with 386-family processors, with or without math coprocessors. The minimal configuration is said to require 4 MB of RAM and 50 MB of disk space, though we do not know of anyone running with a system quite this minimal today. To install the entire system requires much more disk space (the unpacked binary distribution, without sources, requires at least 65 MB without counting space needed for swap space, etc), and to run X or compile the system, more RAM is recommended. (4 MB of RAM will actually allow you to run X and/or compile, but it won't be speedy. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.)

Supported devices

Explanation of bracketted footnote tags [] follows this listing.

- Floppy controllers.
- MFM, ESDI, IDE, and RLL hard disk controllers.
  There is complete support (including IDE DMA or Ultra-DMA) for the following PCI controllers
  - Acard ATA-850 and 860 based IDE Controllers
  - Acer labs M5229 IDE Controller
  - Advanced Micro Devices AMD-756, 766, and 768 IDE Controllers
  - CMD Tech PCI0643, 0646, 0648, and 0649 IDE Controllers
  - Contaq Microsystems/Cypress CY82C693 IDE Controller
  - HighPoint HPT366, HPT370, HPT372, and HPT374.
  - IBM ESDI Fixed Disk Controllers [m]
  - Intel PIIX, PIIX3, and PIIX4 IDE Controllers
  - Intel 82801 (ICH/ICH0/ICH2/ICH4/ICH5/ICH6) IDE Controllers
  - Promise PDC20246 (Ultra/33), PDC20262 (Ultra/66), PDC20265/PDC20267 (Ultra/100), PDC20268 (Ultra/100TX2 and Ultra/100TX2v2), Ultra/133, Ultra/133TX2, and Ultra/133TX2v2.
  - Promise SATA150 serial-ATA controllers: PDC20318, PDC20319, PDC20371, PDC20375, PDC20376, PDC20377, PDC20378 and PDC20379.
  - Silicon Integrated System 5597/5598 IDE controller
  - VIA Technologies VT82C586, VT82C586A, VT82C596A, VT82C686A, and VT8233A IDE Controllers

  Most of these controllers are only available in multifunction PCI chips. Other PCI IDE controllers are supported, but performance may not be optimal. ISA, ISA Plug and Play and PCMCIA IDE controllers are supported as well.
- SCSI host adapters
  - Adaptec AHA-154xA, -B, -C, and -CF
  - Adaptec AHA-1640 cards (MCA variant of AHA-1540) [m]
  - Adaptec AHA-174x
  - Adaptec AIC-6260 and AIC-6360 based boards, including the Adaptec AHA-152x, Adaptec APA-1460 (PCMCIA) and APA-1480 (CardBus), and the SoundBlaster SCSI host adapter.

    Note: You cannot boot from these boards if they do not have a boot ROM; consequently only the AHA-152x and motherboards using this chip are likely to be bootable.
  - Adaptec AHA-2910, 2915, 2920, and 2930C adapters.
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.

- Adaptec AHA-394x[U][W] cards [b]
- Adaptec AHA-3950U2 cards
- Adaptec AHA-3960, 19160, and 29160 Ultra-160 adapters
- AdvanSys ABP-9x0[U][A] cards
- AdvanSys ABP-940UW[68], ABP-970UW[68], and ASB3940UW-00 cards
- AMD PCscsi-PCI (Am53c974) based SCSI adapters, including Tekram DC-390
- BusLogic 54x (Adaptec AHA-154x clones)
- BusLogic 445, 74x, 9xx (but not the new 'FlashPoint' series of BusLogic SCSI adapters)
- Qlogic ISP [12]0x0 SCSI/FibreChannel boards
- Seagate/Future Domain ISA SCSI adapter cards
  - ST01/02
  - Future Domain TMC-885
  - Future Domain TMC-950
- Symbios Logic (NCR) 53C8xx-based PCI SCSI host adapters
  - Acculogic PCIpport
  - ASUS SC-200 (requires NCR BIOS on motherboard to boot from disks)
  - ASUS SC-875
  - ASUS SP3[G] motherboard onboard SCSI
  - DEC Celebris XL/590 onboard SCSI
  - Diamond FirePort 40
  - Lomas Data SCSI adapters
  - NCR/SYM 8125 (and its many clones; be careful, some of these cards have a jumper to set the PCI interrupt; leave it on INT A!)
  - Promise DC540 (a particularly common OEM model of the SYM 8125)
  - Tekram DC-390U/F
  - Tyan Yorktown
- Symbios Logic (NCR) 5380/53C400-based ISA SCSI host adapters [∗]
- Ultrastor 14f, 34f, and (possibly) 24f
- Western Digital WD7000 SCSI and TMC-7000 host adapters (ISA cards only)

• MDA, CGA, VGA, SVGA, and HGC Display Adapters.

  Note: Not all of the display adapters NetBSD/i386 can work with are supported by X. See the XFree86 FAQ for more information.

• Serial ports
  - 8250/16450-based ports
  - 16550/16650/16750-based ports
  - AST-style 4-port serial cards [∗]
  - BOCA 8-port serial cards [∗]
  - BOCA 6-port (ioat) serial cards [∗]
  - IBM PC-RT 4-port serial cards [∗]
  - TCOM TC-400 (4-port), TC-800 (8-port) serial cards [∗]
  - Single-port Hayes ESP serial cards [∗]
  - Cyclades Cyclom-Y serial cards [∗] [+]
  - Addonics FlexPort 8S 8-port serial cards [∗]
  - Byte Runner Technologies TC-400 and TC-800 serial cards [∗]
  - PCI universal communication cards

• Parallel ports. [∗] [+]

• Ethernet adapters

- AMD LANCE and PCnet-based ISA Ethernet adapters [∗]
  - Novell NE1500T
  - Novell NE2100
  - Kingston 21xx
  - Digital EtherWORKS II ISA adapters (DE200/DE201/DE202)
- AMD LANCE and PCnet-based MCA Ethernet adapters [m]
  - SKNET Personal
  - SKNET MC+
- AMD PCnet-based PCI Ethernet adapters
  - Addtron AE-350
  - BOCALANcard/PCI
  - SVEC FD0455
  - X/Lan Add-On Adapter
  - IBM #13H9237 PCI Ethernet Adapter
- ATT StarLAN 10, EN100, and StarLAN Fiber
- 3COM 3c501
- 3COM 3c503
- 3COM 3c505 [∗]
- 3COM 3c507
- 3COM 3c509, 3c579, 3c589, and 3c59X
- 3COM 3c523 EtherLink/MC [m]
- 3COM 3c529 EtherLink III [m]
- 3COM 3c90X (including 3c905B), 3c450, 3c55X, 3c575, 3c980, 3cSOHO100
- Digital DC21x4x-based PCI Ethernet adapters
  - Accton EN2242
  - ASUS PCI-DEC100TX+
  - Cogent EM1X0, EM960 (a.k.a. Adaptec ANA-69XX)
  - Cogent EM964 [b]
  - Cogent EM4XX [b]
  - Compex Readylink PCI
  - DANPEX EN-9400P3
  - Digital Celebris GL, GLST on-board ethernet
  - DEC (Digital) PCI Ethernet/Fast Ethernet adapters (all)
  - DLINK DFE500-TX
  - JCIS Condor JC1260
  - Linksys PCI Fast Ethernet
  - SMC EtherPower 10, 10/100 (PCI only!)
  - SMC EtherPower^2 [b]
  - Sundance ST-201 based ethernet adapters (including DLINK DFE550-TX and DFE580-TX)
  - SVEC PN0455
  - SVEC FD1000-TP
  - Znyx ZX34X
- Digital EtherWORKS III ISA adapters (DE203/DE204/DE205) [∗]
- Digital DEPCM-BA (PCMCIA) and DE305 (ISA) NE2000-compatible cards
- BICC Isolan [∗ and not recently tested]
- Efficient Networks EN-155 and Adaptec AIC-590x ATM interfaces
- Essential Communications Hippi (800 Mbit/s)
- Fujitsu MB86960A/MB86965A based cards

    - Fujitsu FMV-180 series
    - Allied-Telesis AT1700 series
    - Allied-Telesis AT1700 series MCA [m]
    - Allied-Telesis RE2000 series
  - Intel EtherExpress 16
  - Intel EtherExpress PRO/10
  - Intel EtherExpress 100 Fast Ethernet adapters
  - Intel Intel PRO/1000 Gigabit Ethernet adapters
  - Novell NE1000, NE2000 (ISA, PCI, PCMCIA, ISA PnP)
  - Realtek 8129/8139 based boards
  - Realtek 8139C+/8169/8169S/8110S based boards
  - SMC/WD 8003, 8013, and the SMC 'Elite16' ISA boards
  - SMC/WD 8003, 8013 and IBM PS/2 Adapter/A MCA boards [m]
  - SMC/WD 8216 (the SMC 'Elite16 Ultra' ISA boards)
  - SMC 91C9x-based boards (ISA and PCMCIA)
  - SMC EPIC/100 Fast Ethernet boards
    - SMC Etherpower-II
  - Texas Instruments ThunderLAN based ethernet boards
    - Compaq Netelligent 10/100 TX
    - Compaq ProLiant Integrated Netelligent 10/100 TX
    - Compaq Netelligent 10 T (untested)
    - Compaq Integrated NetFlex 3/P
    - Compaq NetFlex 3/P in baseboard variant (the PCI variant doesn't use the same chip!)
    - Compaq Dual Port Netelligent 10/100 TX
    - Compaq Deskpro 4000 5233MMX (untested)
    - Texas Instruments TravelMate 5000 series laptop docking station Ethernet board
  - VIA VT3043 (Rhine) and VT86C100A (Rhine-II) based ethernet boards
    - D-Link DFE530TX

- FDDI adapters
  - Digital DEFPA PCI FDDI adapters [∗] [+]
  - Digital DEFEA EISA FDDI adapters [∗] [+]

- Token-Ring adapters
  - IBM Token-Ring Network PC Adapter [+]
  - IBM Token-Ring Network PC Adapter II [+]
  - IBM Token-Ring Network Adapter/A [+]
  - IBM Token-Ring Network 16/4 Adapter [+]
  - IBM Token-Ring Network 16/4 Adapter/A [m]
  - IBM 16/4 ISA Adapter [+]
  - IBM Auto 16/4 Token-Ring ISA Adapter [+]
  - 3COM 3C619 TokenLink [+]
  - 3COM 3C319 TokenLink Velocity [+]

- Wireless network adapters
  - 3Com AirConnect Wireless LAN
  - ATT/Lucent/Agere WaveLAN/ORiNOCO IEEE (802.11) PCMCIA cards
  - Aironet 4500/4800 and Cisco 340 series PCMCIA cards
  - BayStack 650 802.11FH PCMCIA cards [∗] [+]
  - Corega Wireless LAN PCC-11 cards [∗] [+]
  - DEC/Cabletron RoamAbout 802.11 DS High Rate cards [∗] [+]

- ELSA AirLancer MC-11 card [∗] [+]
- Intersil Prism II
- Melco AIR CONNECT WLI-PCM-L11 cards [∗] [+]
- NetWave AirSurfer PCMCIA cards [∗] [+]

• High Speed Serial
  - LAN Media Corporation SSI/LMC10000 (up to 10 Mbps) [∗] [+]
  - LAN Media Corporation HSSI/LMC5200 [∗] [+]
  - LAN Media Corporation DS3/LMC5245 [∗] [+]

• Tape drives
  - Most SCSI tape drives
  - Seagate and OnStream ATAPI tape drives, possibly others
  - QIC-02 and QIC-36 format (Archive- and Wangtek- compatible) tape drives [∗] [+]

• CD-ROM drives
  - Non-IDE Mitsumi CD-ROM drives [∗] [+]

    Note: The Mitsumi driver device probe is known to cause trouble with several devices!

  - Most SCSI CD-ROM drives
  - Most ATAPI CD-ROM drives.

    Note: Some low-priced IDE CD-ROM drives are known for being not or not fully ATAPI compliant, and thus require some hack (generally an entry to a quirk table) to work with NetBSD.

• Mice
  - "Logitech" -style bus mice [∗] [+]
  - Microsoft-style bus mice [∗] [+]
  - "PS/2" -style mice [∗] [+]
  - Serial mice (no kernel support necessary)

• Sound Cards
  - Aria based sound cards [∗]
  - Cirrus Logic CS461x, CS4280 and CS4281 audio [∗] [+]
  - Ensoniq AudioPCI [∗] [+]
  - ESS Technology ESS 1688 Audiodrive, ES1777/1868/1869/1887/1888/888, Maestro 1/2/2E and Solo-1 ES1938/1946 [∗] [+]
  - Gravis Ultrasound Plug and Play [∗] [+]
  - Gravis Ultrasound and Ultrasound Max [∗] [+]
  - NeoMagic MagicMedia 256AV / 256ZX AC'97 audio [∗] [+]
  - Personal Sound System [∗] [+]
  - ProAudio Spectrum [∗] [+]
  - S3 SonicVibes [∗] [+]
  - SoundBlaster, SoundBlaster Pro, SoundBlaster 16 [∗] [+]
  - VIA VT82C686A southbridge integrated AC97 audio [∗] [+]
  - Windows Sound System [∗] [+]
  - Yamaha YMF724/740/744/754 audio (DS-1 series) [∗] [+]
  - Yamaha OPL3-SA3 audio [∗] [+]

• Game Ports (Joysticks) [∗] [+]

• Miscellaneous

  - Advanced power management (APM) [∗]
  - Advanced Configuration and Power Interface (ACPI) [∗] [+]
- Universal Serial Bus (USB)
  - UHCI host controllers [∗] [+]
  - OHCI host controllers [∗] [+]
  - Hubs [∗] [+]
  - Keyboards using the boot protocol [∗] [+]
  - Mice [∗] [+]
  - Printers [∗] [+]
  - Modems using Abstract Control Model [∗] [+]
  - Generic support for HID devices [∗] [+]
  - Ethernet adapters [∗] [+]
  - Audio devices [∗] [+]
  - driver for FTDI based serial adapters [∗] [+]
  - Mass storage devices such as disks, ZIP drives and digital cameras [∗] [+]
  - driver for the Prolific host-to-host adapter [∗] [+]
  - Handspring Visor driver [∗] [+]
- PCMCIA Controllers.
  ISA, PCI, and ISA Plug and Play attachments
  - Intel 82365 PCIC, rev 0 and rev 1
  - Cirrus PD6710
  - Cirrus PD672X

  Note: This will work with most laptops as well as with ISA cards which provide PCMCIA slots for desktops.
- RAID Controllers
  - 3ware Escalade family of controllers
  - Compaq Integrated Array (PCI) [b]
  - Compaq IAES (EISA)
  - Compaq IDA, IDA-2 (EISA)
  - Compaq Smart Array 221, 3100ES, 3200, 4200, 4250ES (PCI) [b]
  - Compaq Smart Array 431, RAID LC2 [b]
  - Compaq SMART 2, 2/E (EISA)
  - Compaq SMART 2/E, 2/P, 2DH, 2SL (PCI) [b]
  - DELL RAID controllers
    - PERC 2/SC
    - PERC 2/DC
    - PERC 4/Di
    - PERC 4/SC
    - PERC 4e/Si
    - CERC 1.5
  - DPT SCSI RAID boards (ISA [∗], EISA and PCI)
    - SmartCache III
    - SmartCache IV
    - SmartRAID III
    - SmartRAID IV
  - MegaRAID controllers
    - 320-1

- 320-2
- Series 418
- Enterprise 1200 (Series 428)
- Enterprise 1300 (Series 434)
- Enterprise 1400 (Series 438)
- Enterprise 1500 (Series 467)
- Enterprise 1600 (Series 471)
- Elite 1500 (Series 467)
- Elite 1600 (Series 493)
- Express 100 (Series 466WS)
- Express 200 (Series 466)
- Express 300 (Series 490)
- Express 500 (Series 475)

Specific driver footnotes:

[*] Drivers are *not* present in kernels on the distribution floppies. Except as noted above, all drivers are present on all disks. Also, at the present time, the distributed kernels support only one SCSI host adapter per machine. NetBSD normally allows more, though, so if you have more than one, you can use all of them by compiling a custom kernel once NetBSD is installed.

[+] Support *is* included in the `GENERIC` kernels, although it is not in the kernels which are on the distribution floppies.

[b] Devices require BIOS support for PCI-PCI bridging on your motherboard. Most reasonably modern Pentium motherboards have this support, or can acquire it via a BIOS upgrade.

[m] Devices are only supported by MCA-enabled kernels.

Getting the NetBSD System on to Useful Media

If you are not booting off a CD-ROM, you will need to have some floppy disks to boot off; either three 1.44 MB floppies or one 1.2 MB floppy.

For laptops that have cardbus slots, you should use the `bootlap1.fs` and `bootlap2.fs` floppy images.

For older machines with little RAM, use `boot-tiny.fs`. This image is tailored towards old, small-memory systems, and thus does not contain any PCI or SCSI support. It should work on systems with 4M of RAM. Note that this means 4M available to NetBSD; systems that are said to have 4M may have 640k of base memory and 3072k of extended memory, which currently will not work, as this is a total of 3712k.

For old machines that may have EISA, SCSI and more RAM, but only have a 1.2M floppy drive, use `boot-small1.fs` and `boot-small2.fs`.

For old IBM PS/2 machines with MCA, use `boot-ps2-1.fs` and `boot-ps2-2.fs` floppy images.

For all other systems, use `boot1.fs` and `boot2.fs`

For the 2-floppy sets (and the CD boot image), utilities to repair a badly crashed systems are included. The `boot-tiny.fs` image has a separate `rescue-tiny.fs` rescue floppy image because of lack of space.

If you are using a UNIX-like system to write the floppy images to disks, you should use the `dd` command to copy the file system image(s) (.fs file) directly to the raw floppy disk. It is suggested that you read the `dd`(1) manual page or ask your system administrator to determine the correct set of arguments to use; it will be slightly different from system to system, and a comprehensive list of the possibilities is beyond the scope of this document.

If you are using MS-DOS to write the floppy image(s) to floppy disk, you should use the **`rawrite`** utility, provided in the `i386/installation/misc` directory of the NetBSD distribution. It will write a file system image (.fs file) to a floppy disk. A **rawrite32** is also available that runs under MS Windows.

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.

Installation is supported from several media types, including:

- CD-ROM / DVD
- 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 / DVD Find out where the distribution set files are on the CD-ROM or DVD. Likely locations are `binary/sets` and `i386/binary/sets`.

Proceed to the instruction on installation.

MS-DOS floppy NetBSD doesn't include split sets to keep the distribution size down. They can be created on a separate machine using the `split(1)` command, running e.g. **split -b 235k base.tgz base.** to split the `base.tgz` file from `i386/binary/sets` into files named `base.aa`, `base.ab`, and so on. Repeat this for all `set_name.tgz` files, splitting them into `set_name.`*xx* files. Count the number of `set_name.`*xx* files that make up the distribution sets you want to install or upgrade. You will need one fifth that number of 1.2 MB floppies, or one sixth that number of 1.44 MB floppies. You should only use one size of floppy for the install or upgrade procedure; you can't use some 1.2 MB floppies and some 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. If you don't have DHCP available on your network, you will 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 IPv4 address of **ftp.NetBSD.org** is `204.152.190.13` and the IPv6 address is `2001:4f8:4:7:230:48ff:fe31:43f2` (as of December, 2007).

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 for those 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 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 you don't have DHCP available on your network and 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 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 **kern-GENERIC, 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-4.0.1
# cd i386/binary
# tar -cf tape_device kern-GENERIC base etc
```

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.

Preparing your System for NetBSD installation

First and foremost, before beginning the installation process, *make sure you have a reliable backup* of any data on your hard disk that you wish to keep. Mistakes in partitioning your hard disk may lead to data loss.

Before you begin, you should be aware of the geometry issues that may arise in relation to your hard disk. First of all, you should know about sector size. You can count on this to be 512 bytes; other sizes are rare (and currently not supported). Of particular interest are the number of sectors per track, the number of tracks per cylinder (also known as the number of heads), and the number of cylinders. Together they describe the disk geometry.

The BIOS has a limit of 1024 cylinders and 63 sectors per track for doing BIOS I/O. This is because of the old programming interface to the BIOS that restricts these values. Most of the big disks currently being used have more than 1024 real cylinders. Some have more than 63 sectors per track. Therefore, the BIOS can be instructed to use a fake geometry that accesses most of the disk and the fake geometry has less than or equal to 1024 cylinders and less than or equal to 63 sectors. This is possible because the disks can be addressed in a way that is not restricted to these values, and the BIOS can internally perform a translation. This can be activated in most modern BIOSes by using *Large* or *LBA* mode for the disk.

NetBSD does not have the mentioned limitations with regard to the geometry. However, since the BIOS has to be used during startup, it is important to know about the geometry the BIOS uses. The NetBSD kernel should be on a part of the disk where it can be loaded using the BIOS, within the limitations of the BIOS geometry. The install program will check this for you, and will give you a chance to correct this if this is not the case.

If you have not yet installed any other systems on the hard disk that you plan to install NetBSD on, or if you plan to use the disk entirely for NetBSD, you may wish to check your BIOS settings for the 'Large' or 'LBA' modes, and activate them for the hard disk in question. While they are not needed by NetBSD as such, doing so will remove the limitations mentioned above, and will avoid hassle should you wish to share the disk with other systems. Do *not* change these settings if you already have data on the disk that you want to preserve!

In any case, it is wise to check your the BIOS settings for the hard disk geometry before beginning the installation, and write them down. While this should usually not be needed, it enables you to verify that the install program determines these values correctly.

The geometry that the BIOS uses will be referred to as the *BIOS geometry*, the geometry that NetBSD uses is the *real geometry*.

sysinst, the NetBSD installation program, will try to discover both the real geometry and BIOS geometry.

It is *important* that **sysinst** know the proper BIOS geometry to be able to get NetBSD to boot, regardless of where on your disk you put it. It is less of a concern if the disk is going to be used entirely for NetBSD. If you intend to have several OSes on your disk, this becomes a much larger issue.

Installing the NetBSD System

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 for the sake of completeness. Do not let this discourage you; the install program is not hard to use.

2. *Possible hardware problems*

   Should you encounter hardware problems during installation, try rebooting after unplugging removable devices you don't need for installation. Non-removable devices can be disabled with **userconf** (use **boot -c** to enter it).

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. It is not necessary 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 the next section. 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 1.44 MB 3.5" floppy disks if the CD is not bootable or if you cannot boot from CD.
     - A PC with a 386 or newer processor.
     - A CD-ROM drive (SCSI or ATAPI), a hard disk and a minimum of 4 MB of memory installed.
     - The hard disk 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 120 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-4.0.1\i386\installation\misc
     rawrite
     ```

     When asked for a source filename, answer
     `..\floppy\boot1.fs`
     for the first diskette and
     `..\floppy\boot2.fs`
     for the second diskette.

     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 of=/dev/rfd0a bs=18k
     ```
   - The Quick Installation
     - Insert the first boot floppy you just created. Restart the computer. When prompted, insert the second boot floppy. After language selection, the main menu will be displayed.

```
.*********************************************.
* NetBSD-4.0.1 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*

   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.

   If that doesn't work, try booting after disabling your CPU's internal and external caches (if any). If it still doesn't work, NetBSD probably can't be run on your hardware. This can probably be considered a bug, so you might want to report it. If you do, please include as many details about your system configuration as you can.

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

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*

   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`.

   `sysinst` next tries to figure out the real and BIOS geometry of your disk. It will present you with the values it found, if any, and will give you a chance to change them.

   Next, depending on whether you are using a `wd`*X* or `sd`*X* disk, you will either be asked for the type of disk (`wd`*X*) you are using or you will be asked if you want to specify a fake geometry for your SCSI disk (`sd`*X*). The types of disk are IDE, ST-506 or ESDI. If you're installing on an ST-506 or ESDI drive, you'll be asked if your disk supports automatic sector forwarding. If you are *sure* that it does, reply affirmatively. Otherwise, the install program will automatically reserve space for bad144 tables.

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.

   If you want to use the entire disk for NetBSD, you can skip the following section and go to *Editing the NetBSD disklabel*.

9. *Editing the Master Boot Record*

   First, you will be prompted to specify the units of size that you want to express the sizes of the partitions in. You can either pick megabytes, cylinders or sectors.

   After this, you will be presented with the current values stored in the MBR, and will be given the opportunity to change, create or delete partitions. For each partition you can set the type, the start and the size. Setting the type to **`unused`** will delete a partition. You can also mark a partition as active, meaning that this is the one that the BIOS will start from at boot time.

   Be sure to mark the partition you want to boot from as active!

   After you are done editing the MBR, a sanity check will be done, checking for partitions that overlap. Depending on the BIOS capabilities of your machine and the parameters of the NetBSD partition you have specified, you may also be asked if you want to install newer bootcode in your MBR. If you have multiple operating systems on the disk that you are installing on, you will also be given the option to install a bootselector, that will allow you to pick the operating system to start up when your computer is (re-)started.

If everything is ok, you can go on to the next step, editing the NetBSD disklabel.

10. *Editing the NetBSD disklabel*

    The partition table of the NetBSD part of a disk is called a *disklabel*. There are 4 layouts for the NetBSD part of the disk that you can pick from: **`Standard, Standard with X, Custom`** and **`Use Existing`**. The first two use a set of default values (that you can change) suitable for a normal installation, possibly including X. With the **`Custom`** option you can specify everything yourself. The last option uses the partition info already present on the disk.

    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**. You can also specify a partition as type **MSDOS**. This is useful if you share the disk with MS-DOS or Windows; NetBSD is able to access the files on these partitions. You can use the values from the MBR for the MS-DOS part of the disk to specify the partition of type **MSDOS** (you don't have to do this now, you can always re-edit the disklabel to add this once you have installed NetBSD, or use `mbrlabel`(8) to help you updating your disklabel with data from the MBR).

    Some partitions in the disklabel have a fixed purpose.

`a`	Root partition (`/`)
`b`	Swap partition.
`c`	The NetBSD portion of the disk.
`d`	The entire disk.
`e-p`	Available for other use. Traditionally, `e` 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 will be ok for most purposes. 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.

11. *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.

12. *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 the section labeled 'Extracting the distribution sets'.

13. *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 if you want to use DHCP, and if not to provide 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, the account name and password used to log into that host using ftp, and optionally a proxy server to use. 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.

14. *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 if you want to use DHCP, and if not to provide 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.

15. *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.

16. *Installation from a floppy set*

    Because the installation sets are too big to fit on one floppy, the floppies are expected to be filled with the split set files. The floppies are expected to be in MS-DOS format. You will be asked for a directory where the sets should be reassembled. Then you will be prompted to insert the floppies containing the split sets. This process will continue until all the sets have been loaded from floppy.

17. *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.

18. *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.

19. *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. Alternatively, you will be asked if you wish to have a progress bar. This is the preferred option as it shows progress without significantly slowing down the installation process.

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 will be asked to select a password encryption algorithm and can then set a password for the "root" account, to prevent the machine coming up without access restrictions.

20. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 4.0.1. You can now reboot the machine, and boot NetBSD from hard disk.

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-only. 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 file system 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. Instead of manually configuring network and naming service, DHCP can be used by setting dhclient=YES in /etc/rc.conf.

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. By default, root login from the network is disabled (even via ssh(1)). One way to become root over the network is to log in as a different user that belongs to group "wheel" (see group(5)) and use su(1) to become root.

   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 vipw(8) and pwd_mkdb(8) if you want to edit the password database.

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.

   You will need to set up a configuration file, see /usr/X11R6/lib/X11/XF86Config.eg for an example. The xf86cfg(1) and xf86config(1) utilities can interactively create a first version of such a configuration file for you. See **http://www.xfree86.org/** and the XFree86 manual page for more 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/docs/software/packages.html

- A list of available packages suitable for browsing is at
  ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc/README.html
- Precompiled binaries can be found at
  ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/
  usually in the `i386/4.0/All` subdir. You can install them with the following commands under `sh`(1):

  ```
  # PKG_PATH=ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/i386/4.0/All
  # export PKG_PATH
  # pkg_add -v tcsh
  # pkg_add -v bash
  # pkg_add -v perl
  # pkg_add -v apache
  # pkg_add -v kde
  # pkg_add -v firefox
  ...
  ```

  If you are using `csh`(1) then replace the first two lines with the following:

  ```
  # setenv PKG_PATH ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/i386/4.0/All
  ...
  ```

  Note: Some mirror sites don't mirror `/pub/pkgsrc` directory. If you would like to use such mirrors, you could also try the `/pub/NetBSD/packages/current-packages/NetBSD/i386/4.0/All` directory which may have the same contents.

  The above commands will install the Tenex-csh and Bourne Again shell, the Perl programming language, Apache web server, KDE desktop environment and the Firefox web browser as well as all the packages they depend on.

  Note: In some case the `pkg_add`(1) command will complain about a version mismatch of packages with a message like the following:

  ```
  Warning: package 'foo' was built for a different version
  of the OS:
  NetBSD/i386 4.0 (pkg) vs. NetBSD/i386 4.0.1 (this host),
  ```

  This warning would be harmless if the formal major release numbers are same between the pkg and your host. Please refer "the NetBSD release glossary and graphs":
  http://www.NetBSD.org/releases/release-map.html
  for details of the release numbering scheme of NetBSD.
- Package sources for compiling packages on your own can be obtained by retrieving the file
  ftp://ftp.NetBSD.org/pub/NetBSD/packages/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, see the `README` and `doc/pkgsrc.txt` files 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/postfix/main.cf` file will almost definitely need to be adjusted. If you prefer a different MTA, then install it using the NetBSD package system or by hand and adjust `/etc/mailer.conf`.
- 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 4.0.1 is a binary upgrade; it can be quite difficult to update the system from an earlier version by recompiling from source, primarily due 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 files already installed on the system are overwritten in place, you only need additional free space for files which weren't previously installed or to account for growth of the sets between releases. 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 kernel, the boot blocks on your NetBSD partition, 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 the NetBSD partition or on another operating system's partition on your disk before beginning the upgrade process.

The upgrade procedure using the **sysinst** tool is similar to an installation, but without the hard disk partitioning. **sysinst** will attempt to merge the settings stored in your `/etc` directory with the new version of NetBSD. 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 4.0.1 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`

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.

NetBSD/i386 has switched its executable format from the old a.out format to ELF, the now more commonly used and supported format. Your old binaries will continue to work just fine. The installation procedure will try to take the necessary steps to accomplish this. The most important step is to move the old a.out shared libraries in `/usr/lib` and `/usr/X11R6/lib` (if X was installed) to `/emul/aout`, where they will be automatically found if an older a.out binary is executed. **sysinst** will use an existing `/emul` and / or `/emul/aout` directory if available, and will create it (as a symbolic link to `/usr/aout`) if necessary.

If you already had a `/emul` directory, or a symbolic link by that name, **sysinst** should rename it and tell you about it.

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 4.0.1.

N.B. when using **sysinst** for upgrading, it will automatically invoke

```
postinstall fix
```

and thus all issues that are fixed by **postinstall** by default (see below) will be handled.

Issues affecting an upgrade from NetBSD 2.1 and older releases.

See the section below on upgrading from NetBSD 3.x as well.

It is **very important** that you populate the directory `/etc/pam.d` with appropriate configuration files for Pluggable Authentication Modules (PAM) because you will not be able to login any more otherwise. Using *postinstall* as described below will take care of this. Please refer to

http://www.NetBSD.org/docs/guide/en/chap-pam.html

for documentation about PAM.

The following issues can generally be resolved by running *postinstall* with the **etc** set :

```
postinstall -s /path/to/etc.tgz check
postinstall -s /path/to/etc.tgz fix
```

Issues fixed by *postinstall*:

- Various files in `/etc` need upgrading. These include:
  - `/etc/defaults/*`
  - `/etc/mtree/*`
  - `/etc/pam.d/*`
  - `/etc/daily`
  - `/etc/weekly`
  - `/etc/monthly`
  - `/etc/security`
  - `/etc/rc.subr`
  - `/etc/rc`
  - `/etc/rc.shutdown`
  - `/etc/rc.d/*`

The following issues need to be resolved manually:

- The user '_pflogd' and the groups '_pflogd' and 'authpf' need to be created.

Issues affecting an upgrade from NetBSD 3.x releases.

The following issues can generally be resolved by running *postinstall* with the **etc** set :

```
postinstall -s /path/to/etc.tgz check
postinstall -s /path/to/etc.tgz fix
```

Issues fixed by *postinstall*:

- Various files in `/etc` need upgrading. These include:
  - `/etc/defaults/*`
  - `/etc/mtree/*`
  - `/etc/daily`
  - `/etc/weekly`

- `/etc/monthly`
- `/etc/security`
- `/etc/rc.subr`
- `/etc/rc`
- `/etc/rc.shutdown`
- `/etc/rc.d/*`
- `/etc/envsys.conf`

The following issues need to be resolved manually:

- The users '_proxy', '_rwhod', and '_sdpd' and the groups '_proxy', '_rwhod' and '_sdpd' need to be created and the user 'uucp' needs to be updated.
- A number of things have been removed from the NetBSD 4.0 release including: the evbsh5 port, the Fortran 77 compiler (g77), NETCCITT, NETNS, Sendmail, Sushi, UUCP, and Vinum. If you were using any of these, then please see the "Components removed from NetBSD" section near the beginning of this document.
- The replacement of Sendmail by Postfix can be handled automatically by *postinstall* but it is not done by default. If you want to transition to Postfix, the command

  ```
  postinstall -s /path/to/etc.tgz fix mailerconf
  ```

  will update your `/etc/mailer.conf` file to use Postfix as the MTA. When using **`sysinst`** to upgrade the system, it will ask if you want this to be done.

  Note that if you have a customized Sendmail setup, you need to set up Postfix in an equivalent way; there is no tool for automatic conversion of Sendmail configuration to a Postfix one.

  Postfix will be started up automatically when the system boots. You may see messages like "$sendmail is not set properly" at boot. You can suppress them by removing `/etc/rc.d/sendmail` and `/etc/rc.d/smmsp`. Those files and other parts of sendmail configuration like files under `/usr/share/sendmail` are not removed by default while upgrading for those who want to continue using sendmail from outside the base system. If you want to delete them, *postinstall* can be used:

  ```
  postinstall -s /path/to/etc.tgz fix sendmail
  ```

Issues with GDB 6.5

Some architectures (arm, i386, powerpc and sparc64) have switched to a newer gdb version (6.5) in this release. Unfortunately support for debugging programs using the SA (scheduler activation) based thread library, is incomplete in this gdb version. Furthermore kernel crashdumps can not be debugged due to a missing identification in the kernel binaries.

Both issues have been addressed on the wrstuden-fixsa branch, but did not make it into the NetBSD release. Both will be fixed in the next patch release.

Using online NetBSD documentation

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

- `intro`(1),
- `man`(1),
- `apropros`(1),
- `passwd`(1), and

- `passwd`(5).

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter

```
# apropos subject-word
```

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

Administrivia

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

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

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

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

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

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

- The Internet Systems Consortium, Inc. for hosting the NetBSD FTP, CVS, AnonCVS, mail, mail archive, GNATS, SUP, Rsync and WWW servers.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.
- The Luleå Academic Computer Society for providing the backup services server.
- The Columbia University Computer Science Department for hosting the NYC build cluster.
- The Western Washington University Computer Science Department for running the WWU build cluster that produces daily snapshots.
- The many organizations 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.
- We list the individuals and organizations that have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it at **http://www.NetBSD.org/donations/.** (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 be mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

Alistair Crooks	`agc@NetBSD.org`
Quentin Garnier	`cube@NetBSD.org`
Matt Thomas	`matt@NetBSD.org`
YAMAMOTO Takashi	`yamt@NetBSD.org`
Christos Zoulas	`christos@NetBSD.org`

The portmasters (and their ports):

Erik Berls	`cyber@NetBSD.org`	**cobalt**
Manuel Bouyer	`bouyer@NetBSD.org`	**xen**
Simon Burge	`simonb@NetBSD.org`	**evbmips**
Simon Burge	`simonb@NetBSD.org`	**pmax**
Simon Burge	`simonb@NetBSD.org`	**sbmips**
Julian Coleman	`jdc@NetBSD.org`	**atari**
Marcus Comstedt	`marcus@NetBSD.org`	**dreamcast**
Andrew Doran	`ad@NetBSD.org`	**amd64**
Andrew Doran	`ad@NetBSD.org`	**i386**
Matthias Drochner	`drochner@NetBSD.org`	**cesfic**
Gavan Fantom	`gavan@NetBSD.org`	**iyonix**
Jaime A Fournier	`ober@NetBSD.org`	**zaurus**
Matt Fredette	`fredette@NetBSD.org`	**sun2**
Ichiro FUKUHARA	`ichiro@NetBSD.org`	**hpcarm**
Chris Gilbert	`chris@NetBSD.org`	**cats**

Ben Harris	`bjh21@NetBSD.org`	**acorn26**
Ross Harvey	`ross@NetBSD.org`	**alpha**
Nick Hudson	`skrll@NetBSD.org`	**hp700**
Martin Husemann	`martin@NetBSD.org`	**sparc64**
IWAMOTO Toshihiro	`toshii@NetBSD.org`	**hpcarm**
Darrin Jewell	`dbj@NetBSD.org`	**next68k**
Søren Jørvang	`soren@NetBSD.org`	**sgimips**
Wayne Knowles	`wdk@NetBSD.org`	**mipsco**
Takayoshi Kochi	`kochi@NetBSD.org`	**ia64**
Paul Kranenburg	`pk@NetBSD.org`	**sparc**
Michael Lorenz	`macallan@NetBSD.org`	**macppc**
Anders Magnusson	`ragge@NetBSD.org`	**vax**
Cherry G. Mathew	`cherry@NetBSD.org`	**ia64**
NISHIMURA Takeshi	`nsmrtks@NetBSD.org`	**x68k**
Tohru Nishimura	`nisimura@NetBSD.org`	**luna68k**
Tohru Nishimura	`nisimura@NetBSD.org`	**sandpoint**
Andrey Petrov	`petrov@NetBSD.org`	**sparc64**
Scott Reynolds	`scottr@NetBSD.org`	**mac68k**
Tim Rightnour	`garbled@NetBSD.org`	**ofppc**
Tim Rightnour	`garbled@NetBSD.org`	**prep**
Tim Rightnour	`garbled@NetBSD.org`	**rs6000**
Noriyuki Soda	`soda@NetBSD.org`	**arc**
Ignatios Souvatzis	`is@NetBSD.org`	**amiga**
Jonathan Stone	`jonathan@NetBSD.org`	**pmax**
Shin Takemura	`takemura@NetBSD.org`	**hpcmips**
Matt Thomas	`matt@NetBSD.org`	**alpha**
Matt Thomas	`matt@NetBSD.org`	**netwinder**
Jason Thorpe	`thorpej@NetBSD.org`	**algor**
Jason Thorpe	`thorpej@NetBSD.org`	**evbarm**
Jason Thorpe	`thorpej@NetBSD.org`	**shark**
Izumi Tsutsui	`tsutsui@NetBSD.org`	**ews4800mips**
Izumi Tsutsui	`tsutsui@NetBSD.org`	**hp300**
Izumi Tsutsui	`tsutsui@NetBSD.org`	**news68k**
Valeriy E. Ushakov	`uwe@NetBSD.org`	**landisk**
Nathan Williams	`nathanw@NetBSD.org`	**sun3**
Steve Woodford	`scw@NetBSD.org`	**evbppc**
Steve Woodford	`scw@NetBSD.org`	**mvme68k**
Steve Woodford	`scw@NetBSD.org`	**mvmeppc**
Reinoud Zandijk	`reinoud@NetBSD.org`	**acorn32**

The NetBSD 4.0.1 Release Engineering team:

Grant Beattie	`grant@NetBSD.org`
Manuel Bouyer	`bouyer@NetBSD.org`
David Brownlee	`abs@NetBSD.org`
James Chacon	`jmc@NetBSD.org`
Julian Coleman	`jdc@NetBSD.org`
Håvard Eidnes	`he@NetBSD.org`
Liam J. Foy	`liamjfoy@NetBSD.org`

John Heasley	heas@NetBSD.org
Geert Hendrickx	ghen@NetBSD.org
Soren Jacobsen	snj@NetBSD.org
Daniel de Kok	daniel@NetBSD.org
Phil Nelson	phil@NetBSD.org
Jeff Rizzo	riz@NetBSD.org

NetBSD Developers:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@NetBSD.org
Jukka Andberg	jandberg@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
Alan Barrett	apb@NetBSD.org
Grant Beattie	grant@NetBSD.org
Jason Beegan	jtb@NetBSD.org
Erik Berls	cyber@NetBSD.org
Hiroyuki Bessho	bsh@NetBSD.org
John Birrell	jb@NetBSD.org
Mason Loring Bliss	mason@NetBSD.org
Charles Blundell	cb@NetBSD.org
Rafal Boni	rafal@NetBSD.org
Stephen Borrill	sborrill@NetBSD.org
Sean Boudreau	seanb@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
John Brezak	brezak@NetBSD.org
Allen Briggs	briggs@NetBSD.org
Mark Brinicombe	mark@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
Simon Burge	simonb@NetBSD.org
Robert Byrnes	byrnes@NetBSD.org
Pavel Cahyna	pavel@NetBSD.org
D'Arcy J.M. Cain	darcy@NetBSD.org
Daniel Carosone	dan@NetBSD.org
Dave Carrel	carrel@NetBSD.org
James Chacon	jmc@NetBSD.org
Mihai Chelaru	kefren@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Ben Collver	ben@NetBSD.org

Marcus Comstedt	marcus@NetBSD.org
Jeremy Cooper	jeremy@NetBSD.org
Chuck Cranor	chuck@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Aidan Cully	aidan@NetBSD.org
Garrett D'Amore	gdamore@NetBSD.org
Johan Danielsson	joda@NetBSD.org
John Darrow	jdarrow@NetBSD.org
Jed Davis	jld@NetBSD.org
Matt DeBergalis	deberg@NetBSD.org
Arnaud Degroote	degroote@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
Jaime A Fournier	ober@NetBSD.org
Stoned Elipot	seb@NetBSD.org
Michael van Elst	mlelstv@NetBSD.org
Enami Tsugutomo	enami@NetBSD.org
Bernd Ernesti	veego@NetBSD.org
Erik Fair	fair@NetBSD.org
Gavan Fantom	gavan@NetBSD.org
Hauke Fath	hauke@NetBSD.org
Hubert Feyrer	hubertf@NetBSD.org
Jason R. Fink	jrf@NetBSD.org
Matt J. Fleming	mjf@NetBSD.org
Marty Fouts	marty@NetBSD.org
Liam J. Foy	liamjfoy@NetBSD.org
Matt Fredette	fredette@NetBSD.org
Thorsten Frueauf	frueauf@NetBSD.org
Castor Fu	castor@NetBSD.org
Ichiro Fukuhara	ichiro@NetBSD.org
Quentin Garnier	cube@NetBSD.org
Thomas Gerner	thomas@NetBSD.org
Simon J. Gerraty	sjg@NetBSD.org
Justin Gibbs	gibbs@NetBSD.org
Chris Gilbert	chris@NetBSD.org
Eric Gillespie	epg@NetBSD.org
Brian Ginsbach	ginsbach@NetBSD.org
Paul Goyette	pgoyette@NetBSD.org
Michael Graff	explorer@NetBSD.org
Brian C. Grayson	bgrayson@NetBSD.org
Matthew Green	mrg@NetBSD.org

Andreas Gustafsson	gson@NetBSD.org
Ulrich Habel	rhaen@NetBSD.org
Jun-ichiro itojun Hagino	itojun@NetBSD.org
HAMAJIMA Katsuomi	hamajima@NetBSD.org
Adam Hamsik	haad@NetBSD.org
Juergen Hannken-Illjes	hannken@NetBSD.org
Charles M. Hannum	mycroft@NetBSD.org
Ben Harris	bjh21@NetBSD.org
Ross Harvey	ross@NetBSD.org
Eric Haszlakiewicz	erh@NetBSD.org
John Hawkinson	jhawk@NetBSD.org
HAYAKAWA Koichi	haya@NetBSD.org
John Heasley	heas@NetBSD.org
Geert Hendrickx	ghen@NetBSD.org
René Hexel	rh@NetBSD.org
Iain Hibbert	plunky@NetBSD.org
Kouichirou Hiratsuka	hira@NetBSD.org
Michael L. Hitch	mhitch@NetBSD.org
Ádám Hóka	ahoka@NetBSD.org
Jachym Holecek	freza@NetBSD.org
David A. Holland	dholland@NetBSD.org
Christian E. Hopps	chopps@NetBSD.org
Ken Hornstein	kenh@NetBSD.org
Marc Horowitz	marc@NetBSD.org
Eduardo Horvath	eeh@NetBSD.org
Nick Hudson	skrll@NetBSD.org
Shell Hung	shell@NetBSD.org
Martin Husemann	martin@NetBSD.org
Dean Huxley	dean@NetBSD.org
Love Hörnquist Åstrand	lha@NetBSD.org
Roland Illig	rillig@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
Soren Jacobsen	snj@NetBSD.org
Lonhyn T. Jasinskyj	lonhyn@NetBSD.org
Darrin Jewell	dbj@NetBSD.org
Nicolas Joly	njoly@NetBSD.org
Chris Jones	cjones@NetBSD.org
Søren Jørvang	soren@NetBSD.org
Takahiro Kambe	taca@NetBSD.org
Masanori Kanaoka	kanaoka@NetBSD.org
Antti Kantee	pooka@NetBSD.org
Frank Kardel	kardel@NetBSD.org
Mattias Karlsson	keihan@NetBSD.org
KAWAMOTO Yosihisa	kawamoto@NetBSD.org
Mario Kemper	magick@NetBSD.org

Min Sik Kim	minskim@NetBSD.org
Thomas Klausner	wiz@NetBSD.org
Klaus Klein	kleink@NetBSD.org
John Klos	jklos@NetBSD.org
Wayne Knowles	wdk@NetBSD.org
Takayoshi Kochi	kochi@NetBSD.org
John Kohl	jtk@NetBSD.org
Daniel de Kok	daniel@NetBSD.org
Jonathan A. Kollasch	jakllsch@NetBSD.org
Paul Kranenburg	pk@NetBSD.org
Lubomir Kundrak	lkundrak@NetBSD.org
Jochen Kunz	jkunz@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Kentaro A. Kurahone	kurahone@NetBSD.org
Arnaud Lacombe	alc@NetBSD.org
Kevin Lahey	kml@NetBSD.org
David Laight	dsl@NetBSD.org
Johnny C. Lam	jlam@NetBSD.org
Martin J. Laubach	mjl@NetBSD.org
Greg Lehey	grog@NetBSD.org
Ted Lemon	mellon@NetBSD.org
Christian Limpach	cl@NetBSD.org
Frank van der Linden	fvdl@NetBSD.org
Joel Lindholm	joel@NetBSD.org
Tonnerre Lombard	tonnerre@NetBSD.org
Mike Long	mikel@NetBSD.org
Michael Lorenz	macallan@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
Anders Magnusson	ragge@NetBSD.org
Cherry G. Mathew	cherry@NetBSD.org
David Maxwell	david@NetBSD.org
Gregory McGarry	gmcgarry@NetBSD.org
Dan McMahill	dmcmahill@NetBSD.org
Jared D. McNeill	jmcneill@NetBSD.org
Neil J. McRae	neil@NetBSD.org
Julio M. Merino Vidal	jmmv@NetBSD.org
Perry Metzger	perry@NetBSD.org
Luke Mewburn	lukem@NetBSD.org
Brook Milligan	brook@NetBSD.org
Minoura Makoto	minoura@NetBSD.org
Simas Mockevicius	symka@NetBSD.org
der Mouse	mouse@NetBSD.org
Joseph Myers	jsm@NetBSD.org
Ken Nakata	kenn@NetBSD.org

Takeshi Nakayama	nakayama@NetBSD.org
Phil Nelson	phil@NetBSD.org
John Nemeth	jnemeth@NetBSD.org
Bob Nestor	rnestor@NetBSD.org
NISHIMURA Takeshi	nsmrtks@NetBSD.org
Tohru Nishimura	nisimura@NetBSD.org
NONAKA Kimihiro	nonaka@NetBSD.org
Takehiko NOZAKI	tnozaki@NetBSD.org
Tobias Nygren	tnn@NetBSD.org
OBATA Akio	obache@NetBSD.org
Jesse Off	joff@NetBSD.org
Tatoku Ogaito	tacha@NetBSD.org
OKANO Takayoshi	kano@NetBSD.org
Masaru Oki	oki@NetBSD.org
Atsushi Onoe	onoe@NetBSD.org
Greg Oster	oster@NetBSD.org
Rui Paulo	rpaulo@NetBSD.org
Jonathan Perkin	sketch@NetBSD.org
Andrey Petrov	petrov@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Peter Postma	peter@NetBSD.org
Dante Profeta	dante@NetBSD.org
Chris Provenzano	proven@NetBSD.org
Niels Provos	provos@NetBSD.org
Mindaugas Rasiukevicius	rmind@NetBSD.org
Michael Rauch	mrauch@NetBSD.org
Marc Recht	recht@NetBSD.org
Darren Reed	darrenr@NetBSD.org
Jeremy C. Reed	reed@NetBSD.org
Antoine Reilles	tonio@NetBSD.org
Tyler R. Retzlaff	rtr@NetBSD.org
Scott Reynolds	scottr@NetBSD.org
Michael Richardson	mcr@NetBSD.org
Tim Rightnour	garbled@NetBSD.org
Alan Ritter	rittera@NetBSD.org
Jeff Rizzo	riz@NetBSD.org
Hans Rosenfeld	hans@NetBSD.org
Gordon Ross	gwr@NetBSD.org
Steve Rumble	rumble@NetBSD.org
Ilpo Ruotsalainen	lonewolf@NetBSD.org
Heiko W. Rupp	hwr@NetBSD.org
Blair J. Sadewitz	bjs@NetBSD.org
David Sainty	dsainty@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org
Kazuki Sakamoto	sakamoto@NetBSD.org
Curt Sampson	cjs@NetBSD.org

Wilfredo Sanchez	wsanchez@NetBSD.org
Ty Sarna	tsarna@NetBSD.org
SATO Kazumi	sato@NetBSD.org
Jan Schaumann	jschauma@NetBSD.org
Matthias Scheler	tron@NetBSD.org
Silke Scheler	silke@NetBSD.org
Karl Schilke (rAT)	rat@NetBSD.org
Amitai Schlair	schmonz@NetBSD.org
Konrad Schroder	perseant@NetBSD.org
Georg Schwarz	schwarz@NetBSD.org
Lubomir Sedlacik	salo@NetBSD.org
Christopher SEKIYA	sekiya@NetBSD.org
Reed Shadgett	dent@NetBSD.org
John Shannon	shannonjr@NetBSD.org
Tim Shepard	shep@NetBSD.org
Takeshi Shibagaki	shiba@NetBSD.org
Naoto Shimazaki	igy@NetBSD.org
Takao Shinohara	shin@NetBSD.org
Takuya SHIOZAKI	tshiozak@NetBSD.org
Daniel Sieger	dsieger@NetBSD.org
Chuck Silvers	chs@NetBSD.org
Thor Lancelot Simon	tls@NetBSD.org
Jeff Smith	jeffs@NetBSD.org
Noriyuki Soda	soda@NetBSD.org
Wolfgang Solfrank	ws@NetBSD.org
SOMEYA Yoshihiko	someya@NetBSD.org
Bill Sommerfeld	sommerfeld@NetBSD.org
Jörg Sonnenberger	joerg@NetBSD.org
Ignatios Souvatzis	is@NetBSD.org
T K Spindler	dogcow@NetBSD.org
Bill Squier	groo@NetBSD.org
Jonathan Stone	jonathan@NetBSD.org
Bill Studenmund	wrstuden@NetBSD.org
Kevin Sullivan	sullivan@NetBSD.org
SUNAGAWA Keiki	kei@NetBSD.org
Kimmo Suominen	kim@NetBSD.org
Robert Swindells	rjs@NetBSD.org
Shin Takemura	takemura@NetBSD.org
TAMURA Kent	kent@NetBSD.org
Shin'ichiro TAYA	taya@NetBSD.org
Ian Lance Taylor	ian@NetBSD.org
Matt Thomas	matt@NetBSD.org
Jason Thorpe	thorpej@NetBSD.org
Christoph Toshok	toshok@NetBSD.org
Greg Troxel	gdt@NetBSD.org
Tsubai Masanari	tsubai@NetBSD.org
Izumi Tsutsui	tsutsui@NetBSD.org
UCHIYAMA Yasushi	uch@NetBSD.org
Masao Uebayashi	uebayasi@NetBSD.org

Shuichiro URATA	`ur@NetBSD.org`
Valeriy E. Ushakov	`uwe@NetBSD.org`
Todd Vierling	`tv@NetBSD.org`
Aymeric Vincent	`aymeric@NetBSD.org`
Paul Vixie	`vixie@NetBSD.org`
Mike M. Volokhov	`mishka@NetBSD.org`
Krister Walfridsson	`kristerw@NetBSD.org`
Lex Wennmacher	`wennmach@NetBSD.org`
Leo Weppelman	`leo@NetBSD.org`
Assar Westerlund	`assar@NetBSD.org`
Todd Whitesel	`toddpw@NetBSD.org`
Frank Wille	`phx@NetBSD.org`
Nathan Williams	`nathanw@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`
Steve Woodford	`scw@NetBSD.org`
YAMAMOTO Takashi	`yamt@NetBSD.org`
Yuji Yamano	`yyamano@NetBSD.org`
David Young	`dyoung@NetBSD.org`
Reinoud Zandijk	`reinoud@NetBSD.org`
S.P.Zeidler	`spz@NetBSD.org`
Maria Zevenhoven	`maria7@NetBSD.org`
Christos Zoulas	`christos@NetBSD.org`

Other contributors:

Dave Burgess	`burgess@cynjut.infonet.net`
Brian R. Gaeke	`brg@dgate.org`
Brad Grantham	`grantham@tenon.com`
Lawrence Kesteloot	`kesteloo@cs.unc.edu`
Waldi Ravens	`waldi@moacs.indiv.nl.net`

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:

NetBSD is a registered trademark of The NetBSD Foundation, Inc.
This product includes software developed by the University of California, Berkeley and its contributors.
This product includes software developed by the NetBSD Foundation.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See http://www.netbsd.org/ for information about NetBSD.
This product contains software developed by Ignatios Souvatzis for the NetBSD project.
This product contains software written by Ignatios Souvatzis and Michael L. Hitch for the NetBSD project.
This product contains software written by Michael L. Hitch for the NetBSD project.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
This product includes cryptographic software written by Eric Young (eay@mincom.oz.au)
This product includes software designed by William Allen Simpson.
This product includes software developed at Ludd, University of Lulea, Sweden and its contributors.
This product includes software developed at Ludd, University of Lulea.

This product includes software developed at the Information Technology Division, US Naval Research Laboratory.
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by David Jones and Gordon Ross
This product includes software developed by Gordon W. Ross and Leo Weppelman.
This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Internet Research Institute, Inc.
This product includes software developed by Leo Weppelman and Waldi Ravens.
This product includes software developed by Mika Kortelainen
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
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 Advanced Risc Machines Ltd.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Projet.
This product includes software developed by Alex Zepeda.
This product includes software developed by Alistair G. Crooks.
This product includes software developed by Alistair G. Crooks. for the NetBSD project.
This product includes software developed by Allen Briggs
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 Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Bodo Moeller. (If available, substitute umlauted o for oe)
This product includes software developed by Boris Popov.
This product includes software developed by Brad Pepers
This product includes software developed by Bradley A. Grantham.
This product includes software developed by Brini.
This product includes software developed by Causality Limited.
This product includes software developed by Charles D. Cranor and Seth Widoff.
This product includes software developed by Charles D. Cranor and Washington University.
This product includes software developed by Charles D. Cranor, Washington University, and the University of California, Berkeley and its contributors.
This product includes software developed by Charles D. Cranor, Washington University, the University of California, Berkeley and its contributors.
This product includes software developed by Charles D. Cranor.
This product includes software developed by Charles Hannum.
This product includes software developed by Charles M. 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 M. Hannum.
This product includes software developed by Christian E. Hopps, Ezra Story, Kari Mettinen, Markus Wild, Lutz Vieweg and Michael Teske.
This product includes software developed by Christian E. Hopps.
This product includes software developed by Christian Limpach
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 Chuck Silvers.
This product includes software developed by Colin Wood for the NetBSD Project.
This product includes software developed by Colin Wood.
This product includes software developed by Cybernet Corporation and Nan Yang Computer Services Limited
This product includes software developed by Daishi Kato
This product includes software developed by Dale Rahn.
This product includes software developed by Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by Darrin B. Jewell
This product includes software developed by David Miller.
This product includes software developed by Dean Huxley.
This product includes software developed by Eduardo Horvath.
This product includes software developed by Eric S. Hvozda.
This product includes software developed by Eric S. Raymond
This product includes software developed by Eric Young (eay@@mincom.oz.au)
This product includes software developed by Eric Young (eay@cryptsoft.com)
This product includes software developed by Eric Young (eay@mincom.oz.au)
This product includes software developed by Ezra Story and by Kari Mettinen.
This product includes software developed by Ezra Story, by Kari Mettinen and by Bernd Ernesti.

This product includes software developed by Ezra Story, by Kari Mettinen, Michael Teske and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, and Michael Teske.
This product includes software developed by Ezra Story.
This product includes software developed by Frank van der Linden for the NetBSD Project.
This product includes software developed by Gardner Buchanan.
This product includes software developed by Gary Thomas.
This product includes software developed by Gordon Ross
This product includes software developed by Gordon W. Ross
This product includes software developed by HAYAKAWA Koichi.
This product includes software developed by Harvard University and its contributors.
This product includes software developed by Harvard University.
This product includes software developed by Herb Peyerl.
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 F. Darwin and others.
This product includes software developed by Ian W. Dall.
This product includes software developed by Ichiro FUKUHARA.
This product includes software developed by Ignatios Souvatzis for the NetBSD Project.
This product includes software developed by Internet Initiative Japan Inc.
This product includes software developed by James R. Maynard III.
This product includes software developed by Jared D. McNeill.
This product includes software developed by Jason L. Wright
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 Joerg Wunsch
This product includes software developed by John Birrell.
This product includes software developed by John P. Wittkoski.
This product includes software developed by John Polstra.
This product includes software developed by Jonathan R. Stone for the NetBSD Project.
This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.
This product includes software developed by Jonathan Stone.
This product includes software developed by Juan Romero Pardines for the NetBSD Foundation, Inc. and its contributors.
This product includes software developed by Jukka Marin.
This product includes software developed by Julian Highfield.
This product includes software developed by Kazuhisa Shimizu.
This product includes software developed by Kazuki Sakamoto.
This product includes software developed by Kenneth Stailey.
This product includes software developed by Kiyoshi Ikehara.
This product includes software developed by Klaus Burkert,by Bernd Ernesti, by Michael van Elst, and by the University of California, Berkeley and its contributors.
This product includes software developed by LAN Media Corporation and its contributors.
This product includes software developed by Leo Weppelman for the NetBSD Project.
This product includes software developed by Leo Weppelman.
This product includes software developed by Lloyd Parkes.
This product includes software developed by Luke Mewburn.
This product includes software developed by Lutz Vieweg.
This product includes software developed by MINOURA Makoto, Takuya Harakawa.
This product includes software developed by Manuel Bouyer.
This product includes software developed by Marc Horowitz.
This product includes software developed by Marcus Comstedt.
This product includes software developed by Mark Brinicombe for the NetBSD project.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Murray
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 Masanobu Saitoh.
This product includes software developed by Masaru Oki.
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 Matt DeBergalis
This product includes software developed by Matthew Fredette.
This product includes software developed by Matthias Pfaller.
This product includes software developed by Michael Graff for the NetBSD Project.
This product includes software developed by Michael Graff.

This product includes software developed by Michael L. Hitch.
This product includes software developed by Michael Shalayeff.
This product includes software developed by Michael Smith.
This product includes software developed by Mike Glover and contributors.
This product includes software developed by Mike Pritchard.
This product includes software developed by Minoura Makoto.
This product includes software developed by Nan Yang Computer Services Limited.
This product includes software developed by Niels Provos.
This product includes software developed by Niklas Hallqvist, Brandon Creighton and Job de Haas.
This product includes software developed by Niklas Hallqvist.
This product includes software developed by Onno van der Linden.
This product includes software developed by Paul Kranenburg.
This product includes software developed by Paul Mackerras.
This product includes software developed by Per Fogelstrom
This product includes software developed by Peter Galbavy.
This product includes software developed by Phase One, Inc.
This product includes software developed by Philip A. Nelson.
This product includes software developed by Philip L. Budne.
This product includes software developed by RiscBSD.
This product includes software developed by Roar Thronæs.
This product includes software developed by Rodney W. Grimes.
This product includes software developed by Roger Hardiman
This product includes software developed by Roland C. Dowdeswell.
This product includes software developed by Rolf Grossmann.
This product includes software developed by Ross Harvey for the NetBSD Project.
This product includes software developed by Ross Harvey.
This product includes software developed by Scott Bartram.
This product includes software developed by Scott Stevens.
This product includes software developed by Shingo WATANABE.
This product includes software developed by Softweyr LLC, the University of California, Berkeley, and its contributors.
This product includes software developed by Soren S. Jorvang.
This product includes software developed by Stephan Thesing.
This product includes software developed by Steve Woodford.
This product includes software developed by Takashi Hamada
This product includes software developed by Takumi Nakamura.
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 Tetsuya Isaki.
This product includes software developed by Thomas Gerner
This product includes software developed by Thomas Klausner for the NetBSD Project.
This product includes software developed by Tobias Weingartner.
This product includes software developed by Todd C. Miller.
This product includes software developed by Tohru Nishimura and Reinoud Zandijk for the NetBSD Project.
This product includes software developed by Tohru Nishimura for the NetBSD Project.
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 Toru Nishimura.
This product includes software developed by Trimble Navigation, Ltd.
This product includes software developed by WIDE Project and its contributors.
This product includes software developed by Waldi Ravens.
This product includes software developed by Wasabi Systems for Zembu Labs, Inc. http://www.zembu.com/
This product includes software developed by Winning Strategies, Inc.
This product includes software developed by Wolfgang Solfrank.
This product includes software developed by Yasushi Yamasaki
This product includes software developed by Zembu Labs, Inc.
This product includes software developed by the Alice Group.
This product includes software developed by the Center for Software Science at the University of Utah.
This product includes software developed by the Charles D. Cranor, Washington University, University of California, Berkeley and its contributors.
This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
This product includes software developed by the David Muir Sharnoff.
This product includes software developed by the Harvard University and its contributors.
This product includes software developed by the Kungliga Tekniska Högskolan and its contributors.
This product includes software developed by the Network Research Group at Lawrence Berkeley Laboratory.
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 PocketBSD project and its contributors.

The End