NAME

INSTALL — Installation procedure for NetBSD/sparc64.

CONTENTS

About this Document....2
Quick install notes for the impatient....2
What is NetBSD?....3
Changes Between The NetBSD 3.0 and 4.0 Releases....3
  Networking....3
  File systems....4
  Drivers....4
  Platforms....6
  Kernel subsystems....7
  Security....8
  Userland....8
  Components removed from NetBSD....9
The Future of NetBSD....10
Sources of NetBSD....11
NetBSD 4.0 Release Contents....11
  NetBSD/sparc64 subdirectory structure....12
  Binary distribution sets....12
NetBSD/sparc64 System Requirements and Supported Devices....14
  Supported machines....14
  Unsupported machines....15
  Supported devices....15
  Unsupported devices....16
Getting the NetBSD System on to Useful Media....16
Preparing your System for NetBSD installation....18
  Deciding on partition sizes....18
  Setting up Open Firmware....18
  Determining how to boot from an SBUS or PCI....19
  Configuration of network interfaces....20
Installing the NetBSD System....20
  Installing NetBSD from CDROM....21
  Installing NetBSD by using the NetBSD miniroot....21
  Installing NetBSD by using a NetBSD kernel on a....21
  Installing NetBSD by using a netboot setup....22
  Running the sysinst installation program....27
    Introduction....27
    General....27
    Quick install....27
    Booting NetBSD....28
    Network configuration....28
    Installation drive selection and parameters....28
    Partitioning the disk....28
    Preparing your hard disk....29
    Getting the distribution sets....29
    Installation using ftp....29
    Installation using NFS....30
    Installation from CD-ROM....30

Installation from an unmounted file system.....30
Installation from a local directory.....30
Extracting the distribution sets.....30
Ensure you have the correct kernel installed.....31
Finalizing your installation.....31
Manual Installation of NetBSD using Solaris 31
Preparing the disk in Solaris.....31
Installing NetBSD Software from Solaris.....33
Creating NetBSD Device Nodes under Solaris.....33
Configuring the NetBSD system from Solaris.....33
Booting NetBSD for the first time.....33
Post installation steps.....33
Upgrading a previously-installed NetBSD System.....36
Compatibility Issues With Previous NetBSD Releases.....37
Issues affecting an upgrade from NetBSD 2.1 and older 37
Issues affecting an upgrade from NetBSD 3.x releases. 37
Issues with GDB 6.5.....38
Using online NetBSD documentation.....38
Administrivia.....39
Thanks go to.....39
We are.....40
Dedication.....48
Legal Mumbo-Jumbo.....48
The End.....52

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 4.0 on the *sparc64* 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 on a machine of the sparc64 architecture.

- Fetch the CD image, `sparc64cd-4.0.iso` or the installation kernel and bootloader, `sparc64/binary/kernel/netbsd-INSTALL.gz` and `sparc64/installation/misc/ofwboot` which can be booted from a Solaris or NetBSD partition. Alternatively, you may netboot the installation kernel. This process is covered below, in detail.
- The actual binary distribution is in the `sparc64/binary/sets/` directory. When you boot the install CD-ROM or installation kernel, the installation program can fetch these files for you (using e.g. ftp), if you have a network connection. There are several other methods to get the binary sets onto your machine.

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

- Burn the CD or put the installation kernel and bootloader at the root level of a bootable Solaris or NetBSD partition.

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

- You will need to get to the OpenFirmware "**ok**" prompt. After your system first powers on, and displays some initial information, press the `STOP-A` keys, or send a BREAK if you're on a serial console. At the "**ok**" prompt, type the command to boot your system into NetBSD. The command to boot from CD is: "**boot cdrom**". The command to boot the NetBSD kernel from a Solaris or NetBSD partition depends on which disk and partition it is on. To boot from the first partition of the first (primary) disk: "**boot disk:a /ofwboot -a**". When it asks you for a kernel, specify "**netbsd-INSTALL.gz**"

- 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 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 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 Release Contents

The root directory of the NetBSD 4.0 release is organized as follows:

`.../NetBSD-4.0/`

`CHANGES`	Changes between NetBSD 3.0 and branching 4.0.
`CHANGES-4.0`	Changes since NetBSD 4.0 was branched.
`CHANGES.prev`	Changes in earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 4.0 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 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 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 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/sparc64 subdirectory structure

The sparc64-specific portion of the NetBSD 4.0 release is found in the `sparc64` subdirectory of the distribution: `.../NetBSD-4.0/sparc64/`. 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-INSTALL.gz`
The install kernel.
`sets/` sparc64 binary distribution sets; see below.

`installation/`
`miniroot/`
sparc64 miniroot file system image; see below.
`misc/` Miscellaneous sparc64 installation utilities; see installation section, below.
`netboot/`
Netboot bootloader; see below.

Binary distribution sets

The NetBSD sparc64 binary distribution sets contain the binaries which comprise the NetBSD 4.0 release for the sparc64. The binary distribution sets can be found in the `sparc64/binary/sets` subdirectory of the NetBSD 4.0 distribution tree.

Note: If you want to install 32-bit only binaries, you will still need to boot using the sparc64 installation tools. When it asks you for the distribution set to install, provide the NetBSD /sparc `binary/sets`. Be sure to use the 32 bit sparc `kern-GENERIC_SUN4U.tgz` kernel distribution. Otherwise, continue to follow the sparc64 installation procedure, not the sparc instructions.

Most people will want the 64-bit sparc64 distribution tree:

base The NetBSD 4.0 sparc64 **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.
26 MB gzipped, 81 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.
28 MB gzipped, 106 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.
4 MB gzipped, 8 MB uncompressed

kern-GENERIC
This set contains a NetBSD/sparc64 4.0 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
4 MB gzipped, 8 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.
4 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.
7 MB gzipped, 21 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
13 MB gzipped, 52 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 Xsun, Xsun24, XsunMono and Xvfb servers with man pages.
14 MB gzipped, 41 MB uncompressed

The sparc64 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 sparc64 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/sparc64 System Requirements and Supported Devices

Supported machines

The minimal configuration requires 32 MB of RAM and ~60 MB of disk space. To install the entire system requires much more disk space, and to run X or compile the system, more RAM is recommended, as NetBSD with 32 MB of RAM feels like Solaris with 32 MB of RAM - slow. Note that until you have at least 64 MB of RAM, getting more RAM is more important than getting a faster CPU.

Not all of the machines listed here have been tested. Often Sun will use the same motherboard design in multiple models. For example, they'll release a Blade ("workstation") model, Enterprise or Fire ("server") model, and Netra or ft ("telco") model with essentially the same devices, as far as NetBSD is concerned. If one model in this scheme works, it is highly likely another will work.

Sun has also released a few models with names that might imply the systems are UltraSPARC-based but actually have an i386 class CPU. These systems might be supported by the i386 port of NetBSD. See the following site for more info on what system boards are in which models.

http://sunsolve.sun.com/handbook_pub/Devices/System_Board/SYSBD_TOC.html
(requires a SunSolve login with a valid Sun Service Plan)

- Sbus-based UltraSPARC I or II systems
  - Ultra 1 family (Ultra 1, Ultra 1 Creator, Ultra 1 Creator 3D, Netra i 150, Netra nfs 150, Enterprise 1, Enterprise 150)
  - Ultra 2 family (Ultra 2, Ultra 2 Creator, Ultra 2 Creator 3D, Enterprise 2)
  - Enterprise (3000, 3500, 4000, 4500, 500x, 5500, 600x, 6500, 10000)
- PCI-based UltraSPARC II, IIi, or IIe systems
  - Ultra 5/10 family (Ultra 5, Ultra 10, Enterprise 5, Enterprise 10)
  - Ultra 30 family (Ultra 30, Netra T 1100)
  - Ultra 60 family (Ultra 60, E220R, Netra T 1120 and T 1125)
  - Ultra 80 family (Ultra 80, E420R, Netra T 1400 and T 1405)
  - Ultra 250 family (Enterprise 250 and E250R)
  - Ultra 450 family (Ultra 450, Enterprise 450, Netra ft 1800)
  - Blade 100 and 150
  - SPARCengine CP1400 or SPARCengine CP1500 system boards (Netra T1 (models 100 and 105), CT400, CT800, CT1600)
  - SPARCengine Ultra AXi family (Netra X1, Fire v100, Fire v120, Netra 120, Netra AC 200, Netra DC 200)

- SPARCengine Ultra AXe system board

Unsupported machines

- Systems with an UltraSPARC III CPU
- Systems with an UltraSPARC IIIi CPU
- Systems with an UltraSPARC IV CPU
- Systems with an UltraSPARC T1 or T2

Supported devices

- Ethernet devices
  - Sbus LANCE ethernet (*le*)
  - Sbus BigMac ethernet (*be*)
  - On-board, PCI, and Sbus HME ethernet (*hme*)
  - PCI ERI (*gem*)
  - Many other PCI and PCMCIA Ethernet interfaces, such as Tulip-compatible (*tlp*), Intel (*fxp*), Intel gigabit (*wm*), NE2000-compatible (*ne*), and Realtek (*rtk*).
  - Several USB Ethernet interfaces, such as (*aue*, *cue*, and *kue*)
- Video devices
  - All on-board frame buffers
    Unless otherwise specified, only text console is supported
  - On-board and SBUS card (GX, GX+, TGX, TGX+) (*cgsix*)
    8 bpp accelerated framebuffer. Accelerated X via the integrated XFree.
  - UPA devices (ffb and afb, also called Creator3D) 24 bpp accelerated framebuffer. Accelerated X via the integrated XFree.
- SCSI host controllers
  - ncr53c9x based controllers (*esp*)
  - ncr53c8xx based controllers (*siop* or *esiop*)
  - ISP10x0 based controllers (*isp*)
  - 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.

    Note: Some Adaptec cards that support booting on MacOS may be bootable on sparc64 systems.
  - Many other PCI SCSI controllers should work, but no one has tried them
  - Most SCSI disk/tape/CD-ROM devices should work
- PCI IDE host controllers
  - Generic IDE controllers (*pciide*)
  - Acard ATP-850/860 based IDE controllers (*pciide*)
    (including the Acard AEC-6210/6260 and the Sonnet Tempo ATA/66 cards)
  - CMD Tech PCI064[3689] and Silicon Image 0680 IDE Controllers (*cmdide*)
  - Silicon Image 3112 SATA Controller (*satalink*)

    Note: Some IDE cards that support booting on MacOS may be bootable on sparc64 systems.
  - Many other PCI IDE controllers should work, but no one has tried them
  - Most IDE and ATAPI disk/tape/CD-ROM devices should work
- Floppy drives
  - On-board Floppy Disk Controller -- SBus machines only for now (*fdc*)

- Serial devices
  - On-board Sbus z8530 serial (*zs*)
  - On-board PCI 16550-based su and su-pnp serial (*com*)
  - On-board PCI SAB82532 dual UART serial (*sab*)
- Parallel devices
  - On-board and Sbus parallel ports (*bpp*)
  - On-board PCI/Ebus parallel ports (*lpt*)
- Audio devices
  - On-board, PCI, and SBus based CS4231-based 44 KHz, 16 bit, stereo, PCM audio (*audiocs*)

There are a large number of untested PCI drivers that have never been tested on UltraSPARC PCI systems that may 'just work'.

Unsupported devices

- SMP (NetBSD/sparc64 currently only uses one CPU if more than one is present)
- smart card reader

Getting the NetBSD System on to Useful Media

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.

Note: If you are installing the 32-bit sparc distribution sets, you will need to rename `sparc/binary/sets/kern-GENERIC_SUN4U.tgz` to `kern-GENERIC.tgz` since the sparc64 installation tools would otherwise attempt to install the kernel for 32-bit sparc computers which does not boot on sparc64 systems.

CD-ROM / DVD Find out where the distribution set files are on the CD-ROM or DVD. Likely locations are `binary/sets` and `sparc64/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 `sparc64/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 sixth that number of 1.44 MB floppies.

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

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

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

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

FTP

The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. 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
# cd sparc64/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

Deciding on partition sizes

If you're installing NetBSD/sparc64 for the first time it's a good idea to look at the partition sizes of the disk you plan to install NetBSD on. Will you be installing NetBSD onto the same disk as Solaris, on its own disk, or will you be netbooting? While NetBSD can work just fine on a disk shared with Solaris, the NetBSD installer does not currently support this. The limitation is that the NetBSD `disklabel`(8) writes partition info that Solaris is not familiar with. Therefore, if you are sharing a disk with Solaris, any time you change a partition table, you must do it from Solaris.

Assuming a classic partition scheme with `/` (root) and `/usr` file systems, a comfortable size for the NetBSD `/` partition is about 100 MB. A full binary installation including X11R6 takes nearly 350 MB in `/usr`. Since the pkgsrc binaries are typically installed in `/usr/pkg` you may want a significantly larger `/usr` partition. A good initial size for the swap partition is the amount of physical memory in your machine, if you've got more than 128 MB RAM. If you've got less RAM, you may want swap to be at least 128 MB.

Setting up Open Firmware

First, you need to stop your system from automatically booting when powered on. Pressing the `STOP` key (sometimes called the `L1` key, found on the left side of your keyboard) and the `a` key will halt your system and give you the "`ok`" prompt. If you are using a serial console, send a "BREAK" signal from your terminal (the method of sending "BREAK" varies from terminal to terminal).

If the ethernet address of your sparc64 system is `ff:ff:ff:ff:ff:ff` (check with the Open Firmware "banner" command), then your NVRAM battery is dead and you will have trouble using ethernet (among other problems). Read the Sun NVRAM/Hostid FAQ.

http://www.squirrel.com/squirrel/sun-nvram-hostid.faq.html

If you have a valid ethernet address and you plan to netboot, write down your system's ethernet address.

You cannot use the security modes of the SPARC OpenFirmware.

```
ok setenv security-mode none
```

If you are using a serial console, the NetBSD/sparc64 installer defaults to using 9600 bps, 8N1 settings. You may want to configure your system and serial terminal like this prior to booting the installer. Additionally, a new installation of NetBSD/sparc64 will default to these settings as well.

SCSI devices are specified by an Open Firmware **devalias** which provides simple mnemonics for the full path to the device. Type **devalias** to get a list of all of the available aliases. At a minimum, the alias and partition are necessary when booting.

Therefore, to boot from the swap partition on the internal hard drive one would use:

ok **boot disk:b**

To boot from a CD-ROM (Open Firmware assumes SCSI CD-ROMs are at target 6), one would use:

ok **boot cdrom**

And, to boot from a kernel named `netbsd-GENERIC` on the fourth partition ('d', often the `/usr` partition) on an external SCSI hard drive (target 2, partition 3), one would use:

ok **boot disk2:d netbsd-GENERIC**

If you get ".... Fast Data MMU Miss" when booting after the NetBSD installation, your OpenBoot PROM may need updating. It has been reported that version 3.31 lead to a successful boot on an Ultra Enterprise 420R, while version 3.23 did not. Exact values may vary, depending on your hardware, current OpenBoot PROM version and moon phase.

Determining how to boot from an SBUS or PCI card

Some SBUS and PCI cards have firmware that lets you use them as a boot device. These cards do not automatically create a **devalias** entry, so you must traverse the device tree to figure out what Open Firmware calls your card. You will be using Open Firmware commands at the 'ok' prompt. First 'cd' to the top of the device tree and list the nodes there. The following is the procedure to boot from an IDE card in an UltraSPARC 30.

```
ok cd /
ok ls
f006cf08 SUNW,ffb@1e,0
f006c32c SUNW,UltraSPARC-II@0,0
f006002c counter-timer@1f,1c00
f005f410 pci@1f,2000
f005eb54 pci@1f,4000
f004cf84 virtual-memory
f004c9a4 memory@0,0
f002ce38 aliases
f002cdc8 options
f002cc90 openprom
f002cc24 chosen
f002cbb4 packages
```

Usually, you can simply type in the name before the at (@) sign and OpenFirmware will fill in the rest.

```
ok cd pci@1f,4000
ok ls
f0081524 ide@2
f007be50 scsi@3
f0074688 network@1,1
f0060324 ebus@1
ok cd ide@2
ok ls
0081fe4 cdrom
f0081938 disk
ok cd disk
```

```
ok ls
ok pwd
/pci@1f,4000/ide@2/disk
```

OK, now we know the path to the IDE device in this example. Now, we need to determine if it's capable of booting. If it is, it will have the **word** 'open'.

```
ok words
close          load           write          read           seek
open           write-blocks   read-blocks    max-transfer
block-size     dma-free       dma-alloc      spin-down      spin-up
```

Great! Also, in case you're interested in further details about your hardware, you can use the '.properties' command.

So, when it's time to type in a boot command, use the shortened version of the **pwd** command. You need to be more specific if there are two devices with the same name (in this case, two /pci entries). In this example, you'd type:

```
ok boot /pci@1f,4000/ide/disk@0,0
```

You can also store this device path across reboots using the **nvalias** command.

```
ok nvalias wd0 /pci@1f,4000/ide/disk@0,0:a
```

And when the kernel is done booting, it may not automatically use your card as the root device -- you may need to type in the NetBSD/sparc64 name for that device:

```
root on sd0a dumps on sd0b
no file system for sd0 (dev 0x700)
cannot mount root, error = 79
root device (default sd0a): ?
use one of: hme0 sd0[a-h] wd0[a-h] halt
root device (default sd0a): wd0a
dump device: wd0a
file system (default generic): ffs
root on wd0a
```

Configuration of network interfaces

Some network devices (i.e. certain SBus cards) allow a choice between operating on a UTP or a AUI port. The `le` driver supports automatic detection of the port which is actually connected to the wire. If automatic detection is not available or not working properly in your environment, you may have to specify the type connection using the *media* parameter of `ifconfig`(8). During installation, you'll get the opportunity to specify the appropriate medium. Use `10base5/AUI` to select the AUI connector, or `10baseT/UTP` to select the UTP connector.

Installing the NetBSD System

Installing NetBSD is a relatively complex process, but if you have this document in hand it shouldn't be too much trouble.

There are several ways to install NetBSD onto a disk. The easiest way in terms of preliminary setup is to install from CDROM. If you don't have access to a CDROM or CDROM burner, you can use a miniroot image that can be booted off your local disk's swap partition. Alternatively, if your UltraSPARC is hooked up in a network you can find a server and arrange for a diskless setup which is a convenient way to install on a machine whose disk does not currently hold a usable operating system (see the section **Installing NetBSD by using a diskless setup** below).

If you have problems with these or you are installing NetBSD onto the same disk as Solaris, see the section below on **Manual Installation of NetBSD using Solaris**

Installing NetBSD from CDROM

Installing from CDROM, whether it has the full distribution or just a kernel and **sysinst** is the least painful way to install NetBSD. Simply insert the CD-ROM in the drive, power up the computer, and type:

ok **boot cdrom**

This Open Firmware boot command will cause the NetBSD kernel contained in the CD-ROM to be booted. After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the sysinst installation program** below.

Installing NetBSD by using the NetBSD miniroot

The miniroot is a self-contained NetBSD file system holding all utilities necessary to install NetBSD on a local disk. It is distributed as a plain file designed to be transferred to a raw disk partition from which it can be booted using the appropriate OpenFirmware command. Usually, the miniroot will be loaded into the swap partition of a disk. If needed, you can use any other unused partition, but remember that the partition will then not available during the installation process.

Loading the miniroot onto your raw partition is simple using the `dd`(1) command. Just remember to first uncompress the miniroot image and boot your OS with the `-s` flag so that it runs "single-user" and does not attempt to start swapping.

ok **boot -s**

On Solaris you use a command like:

gunzip miniroot.fs.gz
dd if=miniroot.fs of=/dev/rdsk/c0t0d0s1 bs=4k

On NetBSD the command is:

gunzip miniroot.fs.gz
dd if=miniroot.fs of=/dev/rsd0b bs=4k

Replace `/dev/rdsk/c0t0d0s1` or `/dev/rsd0b` with your swap partition.

After transferring the miniroot to disk, bring the system down by:

halt

Then boot the miniroot by typing the appropriate command at the OpenFirmware prompt:

ok **boot disk:b netbsd**

If you've loaded the miniroot onto some other disk than `sd0` use the correct `devalias`, such as

ok **boot disk1:b netbsd**

This Open Firmware boot command will cause the NetBSD kernel contained in the miniroot image to be booted. After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the sysinst installation program** below.

Installing NetBSD by using a NetBSD kernel on a Solaris partition

This procedure is very straightforward. You will be putting the NetBSD installation kernel (kernel with a RAM disk installer) on your Solaris partition and telling Open Firmware to boot the NetBSD kernel.

CAVEAT: this method is a bit fragile. Depending on physical partition layout, partition size, and parameters used to create the filesystem of your Solaris root partition, bugs in the bootloader might be triggered and cause this method to fail.

First, copy the `netbsd-INSTALL.gz` kernel and bootloader to the root level of your hard drive and halt your system

```
# cp binary/kernel/netbsd-INSTALL.gz /
# cp installation/misc/ofwboot /
# halt
```

At the Open Firmware prompt, boot NetBSD.

```
ok boot disk:a /ofwboot -a
```

The `-a` flag is needed so that the bootloader will ask you to find your installation kernel.

```
Rebooting with command: boot disk:a /ofwboot -a
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a  File and args: /ofwboot -a
Enter filename [/ofwboot]: return
 NetBSD/sparc64 OpenFirmware Boot, Revision 1.7
 (autobuild@tgm.netbsd.org, Thu May 20 16:29:20 UTC 2004)
Boot: netbsd-INSTALL.gz
```

After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the sysinst installation program** below.

Installing NetBSD by using a netboot setup

1. Introduction

   To netboot a sparc64, you must configure one or more servers to provide information and files to your sparc64 (the 'client'). If you are using NetBSD (any architecture) on your netboot server(s), the information provided here should be sufficient to configure everything. Additionally, you may wish to look at the `diskless`(8) manual page and the manual pages for each daemon you'll be configuring. If the server(s) are another operating system, you should consult the NetBSD Diskless HOW-TO, which will walk you through the steps necessary to configure the netboot services on a variety of platforms.

   http://www.NetBSD.org/docs/network/netboot/

   You may either netboot the installer so you can install onto a locally attached disk, or you may run your system entirely over the network.

   Briefly, the netboot process involves discovery, bootstrap, kernel and file system stages. In the first stage, the client discovers information about where to find the bootstrap program. Next, it downloads and executes the bootstrap program. The bootstrap program goes through another discovery phase to determine where the kernel is located. The bootstrap program tries to mount the NFS share containing the kernel. Once the kernel is loaded, it starts executing. For RAM disk kernels, it mounts the RAM disk file system and begins executing the installer from the RAM disk. For normal (non-RAM disk) kernels, the kernel tries to mount the NFS share that had the kernel and starts executing the installation tools or `init`(8). All sparc64 systems use a combination of RARP and DHCP for the discovery stage. TFTP is used in the bootstrap phase to download the bootstrap program, `ofwboot.net`, which has been linked to a file name appropriate to the client's IP address as described in the TFTP section below. NFS is used in both the kernel and file system stages to download the kernel, and to access files on the file server.

   We will use '`CC:CC:CC:CC:CC:CC`' as the MAC address (ethernet hardware address) of your netboot client machine. You should have determined this address in an earlier stage. In this example, we will use '`192.168.1.10`' as the IP address of your client and '`client.test.net`' as its name. We will assume you're providing all of your netboot services on one machine called '`server.test.net`' with the client's files exported from the directory `/export/client/root`. You should, of course, replace all of these with the names, addresses, and paths appropriate to your environment.

You should set up each netboot stage in order (i.e. discovery, bootstrap, kernel, and then file system) so that you can test them as you proceed.

2. dhcpd(8) in bootpd(8) compatible mode

   Put the following lines in your /etc/dhcpd.conf (see dhcpd.conf(5) and dhcp-options(5) for more information):

```
ddns-update-style none;
                # Do not use any dynamic DNS features
                #
allow bootp;    # Allow bootp requests, thus the dhcp server
                # will act as a bootp server.
                #
authoritative;  # master DHCP server for this subnet
                #
subnet 192.168.1.0 netmask 255.255.255.0 {
                # Which network interface to listen on.
                # The zeros indicate the range of addresses
                # that are allowed to connect.
}
group {
                # Set of parameters common to all clients
                # in this "group".
                #
        option broadcast-address        192.168.1.255;
        option domain-name              "test.net";
        option domain-name-servers      dns.test.net;
        option routers                  router.test.net;
        option subnet-mask              255.255.255.0;
                #
                # An individual client.
                #
        host client.test.net {
                hardware ethernet       CC:CC:CC:CC:CC:CC;
                fixed-address           192.168.1.10;
                #
                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name        "client";
                #
                # Name of the bootloader or kernel
                # to download via tftp.

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

                #
                # If your DHCP server is not your NFS server, supply the
                # address of the NFS server. Since we assume you run everythi
```

```
                    # on one server, this is not needed.
                    #
                    # next-server server.test.net;
            }
     #you may paste another "host" entry here for additional
     #clients on this network
    }
```

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

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

You will need to start the **dhcpd**. If it's already running, you will need to restart it to force it to re-read its configuration file. If the server is running NetBSD, you can achieve this with:

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

3. `rarpd`(8)

   Create an `/etc/ethers` file with the following line:

   ```
   CC:CC:CC:CC:CC:CC     client
   ```

   Add your client to the server's `/etc/hosts` file:

   ```
   192.168.1.10 client
   ```

   You will need to start the **rarpd**. If it's already running, you will need to restart it to force it to re-read its configuration file. If the server is running NetBSD, you can achieve this with:

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

4. `tftpd`(8)

   The default configuration of the TFTP server is to run in a `chroot`(8) environment in the `/tftpboot` directory. Thus, the first order of business is to create this directory:

   ```
   # mkdir -p /tftpboot
   ```

   Next, edit `/etc/inetd.conf` and uncomment the line with the TFTP daemon:

   ```
   tftp  dgram  udp  wait  root  /usr/libexec/tftpd tftpd -l -s /tftpboot
   ```

   Now, restart `inetd`(8). If the server is running NetBSD, you can achieve this with:

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

   Now, you need to copy the bootloader for your sparc64 machine to `/tftpboot`. Get `ofwboot.net` from the `installation/netboot` directory of the distribution.

   ```
   # cp ofwboot.net /tftpboot
   ```

   Now, you need to link `ofwboot.net` to the filename that your sparc64 will look for. It will look for a filename composed of the machine's IP address (in hexadecimal). For example, a machine which has been assigned IP address 192.168.1.10, will make a TFTP request for `C0A8010A`.

   You can use `bc`(1) to help calculate the filename:

   ```
   # bc
   obase=16
   192
   C0
   168
   ```

```
A8
1
1
10
A
quit
# cd /tftpboot
# ln -s ofwboot.net C0A8010A
```

Just to be sure, let's make everything readable.

```
# chmod -R a+rX /tftpboot
```

Sometimes, the arp(8) table gets messed up, and the TFTP server can't communicate with the client. In this case, it will write a log message (via syslogd(8)) to /var/log/messages saying: 'tftpd: write: Host is down'. If this is the case, you may need to force the server to map your client's ethernet address to its IP address:

```
# arp -s client CC:CC:CC:CC:CC:CC
```

5. nfsd(8), mountd(8), and rpcbind(8)

   Now your system should be able to load the bootstrap program and start looking for the kernel. Let's set up the NFS server. Create the directory you are exporting for the netboot client:

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

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

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

   If your server is currently running an NFS server, you only need to restart mountd(8). Otherwise, you need to start rpcbind(8) and nfsd(8). If the server is running NetBSD, you can achieve this with:

   ```
   # /etc/rc.d/rpcbind start
   # /etc/rc.d/nfsd start
   # /etc/rc.d/mountd restart
   ```

6. NetBSD kernel and installation tools

   Now, if you place a kernel named netbsd in /export/client/root your client should boot the kernel. Use binary/kernel/netbsd-GENERIC.gz).

   ```
   # gunzip netbsd-GENERIC.gz
   # mv netbsd-GENERIC /export/client/root/netbsd
   ```

   If you are netbooting the installer, copy the distribution files to the client's root directory and extract the tools from installation/misc/instfs.tgz.

   ```
   # cp *tgz /export/client/root
   # cd /export/client/root
   # tar -xpzf instfs.tgz
   ```

7. Client file system

   You can skip this step if you do not plan to run your client diskless after installation. Otherwise, you need to extract and set up the client's installation of NetBSD. The Diskless HOW-TO describes how to provide better security and save space on the NFS server over the procedure listed here.

   http://www.NetBSD.org/docs/network/netboot/nfs.html

- Extracting distribution sets

```
# cd /export/client/root
# tar -xpzf /path/to/files/base.tgz
# tar -xpzf /path/to/files/etc.tgz
```

  Continue with the other non-essential distribution sets if desired.

- Set up swap

```
# mkdir /export/client/root/swap
# dd if=/dev/zero of=/export/client/swap bs=4k count=4k
# echo '/export/client/swap -maproot=root:wheel client.test.net' | cat >> /
# /etc/rc.d/mountd restart
```

  This creates a 16 MB swap file and exports it to the client.

- Create device nodes

```
# cd /export/client/root/dev
# ./MAKEDEV all
```

  This procedure only works on NetBSD hosts.

- Set up the client's `fstab`(5)

  Create a file in `/export/client/root/etc/fstab` with the following lines:

```
server:/export/client/swap   none  swap  sw,nfsmntpt=/swap
server:/export/client/root   /     nfs   rw 0 0
```

- Set up the client's `rc.conf`(5)

  Edit `/export/client/root/etc/rc.conf`

```
rc_configured=YES
hostname="client"
defaultroute="192.168.1.1"
nfs_client=YES
auto_ifconfig=NO
net_interfaces=""
```

  Make sure rc does not reconfigure the network device since it will lose its connection to the NFS server with your root file system.

- Set up the client's `hosts`(5) file.

  Edit `/export/client/root/etc/hosts`

```
::1                     localhost
127.0.0.1               localhost
192.168.1.10 client.test.net client
192.168.1.5  server.test.net server
```

8. Setting up the server daemons

   If you want these services to start up every time you boot your server, make sure the following lines are present in your `/etc/rc.conf`:

```
dhcpd=YES        dhcpd_flags="-q"
rarpd=YES        rarpd_flags="-a"
nfs_server=YES         # enable server daemons
mountd=YES
```

```
rpcbind=YES      rpcbind_flags="-l"   # -l logs libwrap
```

Also, you'll need to make sure the `tftpd` line in `/etc/inetd.conf` remains uncommented.

Now, netboot your system from the server by entering the appropriate **boot** command at the Open Firmware prompt.

ok **boot net netbsd**

After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the sysinst installation program** below.

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. *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.

3. *Quick install*

   First, let's describe a quick install. The other sections of this document go into the installation procedure in more detail, but you may find that you do not need this. If you want detailed instructions, skip to 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).
     - A CD-ROM drive (SCSI or ATAPI), a hard disk and a minimum of 16 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.
   - The Quick Installation
     - Boot the system as described above. You should be at the **sysinst** main menu.

```
.*******************************************.
* NetBSD-4.0 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.

  WARNING: If you are installing onto a disk which you want to use with Solaris, stop here. You will need to perform a manual installation as **sysinst** overwrites the Solaris partition table. See the section on **Manual Installation of NetBSD using Solaris**
- After your disk has been prepared, choose **CD-ROM** as the medium. The default values for the path and device should be ok.
- After all the files have been unpacked, go back to the main menu and select **reboot.**
- Once the system reaches the Open Firmware prompt, you will need to type the correct command to boot from your hard drive. NetBSD will now boot. If you haven't already done so in **sysinst**, you should log in as `root`, and set a password for that account. You are also advised to edit the file `/etc/rc.conf` to match your system needs.
- Your installation is now complete.
- For configuring the X window system, if installed, see the files in `/usr/X11R6/lib/X11/doc`. Further information can be found on **http://www.xfree86.org/**.

4. *Booting NetBSD*

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

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

5. *Network configuration*

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

6. *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`.

7. *Partitioning the disk*

- Which portion of the disk to use.

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

8. *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**. Some partitions in the disklabel have a fixed purpose.

   `a` Root partition (`/`)

   `b` Swap partition.

   `c` The NetBSD portion of the disk.

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

   You will then be asked to name your disk's disklabel. The default response 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.

9. *Preparing your hard disk*

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

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

10. *Getting the distribution sets*

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

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

11. *Installation using ftp*

To be able to install using ftp, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you 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.

12. *Installation using NFS*

    To be able to install using NFS, you first need to configure your network setup, if you haven't already at the start of the install procedure. **sysinst** will do this for you, asking you 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.

13. *Installation from CD-ROM*

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

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

14. *Installation from an unmounted file system*

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

15. *Installation from a local directory*

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

16. *Extracting the distribution sets*

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

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

17. *Ensure you have the correct kernel installed*

    If you are installing from the 32-bit sparc distribution set, make sure that you installed the correct kernel. The sparc64 installation tools do not by default copy the correct 32-bit kernel. Unless you prepared ahead of time by renaming the `kern-GENERIC_SUN4U.tgz` to `kern-GENERIC.tgz` then you will need to follow the next few instructions.

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

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

    Type the following command (replacing `wd0a` with the partition name of your destination root partition):

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

    Now you ned to mount the location of your distribution sets:

    ```
    # mount /dev/cd0a /mnt2
    # tar xpzvf /mnt2/sparc/binary/kernel/kern-GENERIC_SUN4U.tgz
    # umount /mnt
    # umount /mnt2
    # exit
    ```

18. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 4.0. You can now reboot the machine, and boot NetBSD from hard disk.

Skip down to the section on **Booting NetBSD for the first time**

Manual Installation of NetBSD using Solaris

(Adapted from Murray Stokely's murray@osd.bsdi.com instructions)

You can use Solaris to prepare the NetBSD user-friendly installer or to perform a full manual installation of NetBSD. If you want to use the user-friendly miniroot installer or RAM disk installation kernel, follow the sections **Installing NetBSD by using the NetBSD miniroot** or **Installing NetBSD by using a NetBSD kernel on a Solaris partition**.

Manual installation from Solaris 10 is not possible because NetBSD cannot use the resulting UFS file system. It is possible to install Solaris 10 and NetBSD on the same disk. To do so, partition the disk with the Solaris format command, then boot NetBSD and perform a manual installation. Be careful not to write a NetBSD disklabel. Use the disklabel command to read the partition size, as constructed from the Solaris disklabel. By default the NetBSD newfs command writes a NetBSD disklabel. Avoid this by using the -F and -s arguments to newfs.

- *Preparing the disk in Solaris*

  The first step is to format and label the disk that you would like to use with NetBSD. This can be accomplished with the `format`(1M) command in Solaris, which allows you to partition a disk and write a

disklabel. It also is used to perform a low-level format on SCSI drives. You will want to create a root partition and a swap partition. Depending on your preferences, you may also wish to create separate `/usr` or `/var` partitions.

```
# /usr/sbin/format
Searching for disks...
Mode sense page(3) reports nsect value as 280, adjusting it to 218
done

c0t1d0: configured with capacity of 16.95GB

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 SUN4.2G cyl 3880 alt 2 hd 16 sec 135
          /pci@1f,4000/scsi@3/sd@0,0
       1. c0t1d0 IBM-DXHS18Y-0430 cyl 8152 alt 2 hd 20 sec 218
          /pci@1f,4000/scsi@3/sd@1,0
Specify disk (enter its number): 1
selecting c0t1d0
[disk formatted]
Disk not labeled.  Label it now? y

format format
Ready to format.  Formatting cannot be interrupted
and takes 114 minutes (estimated). Continue? y
Beginning format. The current time is Sat May 29 22:15:13 2004

Formatting...
done
Verifying media...
        pass 0 - pattern = 0xc6dec6de
   8151/19/208
        pass 1 - pattern = 0x6db6db6d
   8151/19/208

Total of 0 defective blocks repaired.
format partition
partition print
Current partition table (original):
Total disk cylinders available: 8152 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -   60      129.86MB    (61/0/0)     265960
  1       swap    wu      61 -  121      129.86MB    (61/0/0)     265960
  2     backup    wu       0 - 8151       16.95GB    (8152/0/0) 35542720
  3 unassigned    wm       0                0         (0/0/0)           0
  4 unassigned    wm       0                0         (0/0/0)           0
  5 unassigned    wm       0                0         (0/0/0)           0
  6        usr    wm     122 - 8151       16.69GB    (8030/0/0) 35010800
  7 unassigned    wm       0                0         (0/0/0)           0

partition label
Ready to label disk, continue? y
```

partition **quit**
format **quit**

After your disk has been labeled you need to create file systems on your slices. The Solaris `newfs`(1M) command will create ffs file systems that can be used by NetBSD.

```
# /usr/sbin/newfs /dev/dsk/c0t1d0s0
# /usr/sbin/newfs /dev/dsk/c0t1d0s6
```

- *Installing NetBSD Software from Solaris*

  You should now mount your NetBSD root and `/usr` partitions under Solaris so that you can populate the file systems with NetBSD binaries.

```
# /usr/sbin/mount /dev/dsk/c0t1d0s0 /mnt
# mkdir /mnt/usr
# /usr/sbin/mount /dev/dsk/c0t1d0s6 /mnt/usr
```

  Now extract the distribution file sets

```
# cd ~/netbsd/binary/sets
# gunzip *.tar.gz
# echo ~/netbsd/binary/sets/*.tar | (cd /mnt; xargs -n1 pax -rpe -f )
```

  Now you should copy the NetBSD second stage bootloader into your new root partition and install the bootblocks using Solaris's `installboot`(1M) command.

```
# cp ~/netbsd/installation/misc/ofwboot /mnt
# /usr/sbin/installboot ~/netbsd/installation/misc/bootblk /dev/rdsk/c0t1d0s0
```

- *Creating NetBSD Device Nodes under Solaris*

  This is not a necessary step. If your `/dev` directory is empty, `init`(8) will create a RAM disk with all of the essential device nodes each time the system boots. If you want to create the device nodes on disk, you will need to use the Solaris `mknod`(1M) command. Look in `/dev/MAKEDEV` for the correct names, major and minor numbers, ownership, and permissions.

- *Configuring the NetBSD system from Solaris*

  To save effort, you may want to use your favorite Solaris editor to configure some of the files in `/etc` before booting into NetBSD the first time. In particular, you should look at `/etc/fstab`, `/etc/rc.conf`, `/etc/resolv.conf`, and `/etc/hosts`. See the section below on **Post installation steps** before **Booting NetBSD for the first time**.

Booting NetBSD for the first time

Now it is time to boot NetBSD for the first time. You will boot from your disk using similar syntax as described above in **Setting up Open Firmware** and **Determining how to boot from an SBUS or PCI card** That is, boot from your first disk:

ok **boot 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 **sun** for a local console, or whatever is appropriate for your serial console (some systems display garbage with a **sun** terminal type, you may need to use **sun-ss5**) 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=sun
```

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_hme0="inet 123.45.67.89 netmask 255.255.255.0"
```

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

```
ifconfig_hme0="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. Select the proper terminal devices

   If you are using a serial console, you will have to edit the /etc/ttys file and change sun-ss5 to the appropriate terminal type, such as vt220.

3. 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? [...]`.

4. 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.

5. The X Window System

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

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

6. 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/NetBSD/packages/
     usually in the 4.0/sparc64/All subdir. You can install them with the following commands under sh(1):

     ```
     # PKG_PATH=ftp://ftp.NetBSD.org/pub/NetBSD/packages/4.0/sparc64/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/NetBSD/packages/4.0/sparc64/All
     ...
     ```

     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.

   - 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.

7. 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 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 boot from the installer kernel using one of the methods described above. 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 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.

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 .

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),
- apropos(1),
- passwd(1), and

- passwd(5).

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

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

```
# apropos subject-word
```

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

Administrivia

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

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

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

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

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

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

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

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

- 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
YAMAMOTO Takashi	yamt@NetBSD.org
Matt Thomas	matt@NetBSD.org
Christos Zoulas	christos@NetBSD.org

The portmasters (and their ports):

Anders Magnusson	ragge	**vax**
Andrew Doran	ad	**amd64**
Andrew Doran	ad	**i386**
Andrey Petrov	petrov	**sparc64**
Ben Harris	bjh21	**acorn26**
Cherry G. Mathew	cherry	**ia64**
Chris Gilbert	chris	**cats**
Darrin Jewell	dbj	**next68k**
Erik Berls	cyber	**cobalt**
Gavan Fantom	gavan	**iyonix**
IWAMOTO Toshihiro	toshii	**hpcarm**
Ichiro FUKUHARA	ichiro	**hpcarm**
Ignatios Souvatzis	is	**amiga**
Izumi Tsutsui	tsutsui	**ews4800mips**
Izumi Tsutsui	tsutsui	**hp300**

Izumi Tsutsui	tsutsui	**news68k**
Jaime A Fournier	ober	**bebox**
Jaime A Fournier	ober	**zaurus**
Jason Thorpe	thorpej	**algor**
Jason Thorpe	thorpej	**evbarm**
Jason Thorpe	thorpej	**shark**
Jonathan Stone	jonathan	**pmax**
Julian Coleman	jdc	**atari**
Manuel Bouyer	bouyer	**xen**
Marcus Comstedt	marcus	**dreamcast**
Martin Husemann	martin	**sparc64**
Matt Fredette	fredette	**sun2**
Matt Thomas	matt	**alpha**
Matt Thomas	matt	**netwinder**
Matthias Drochner	drochner	**cesfic**
Michael Lorenz	macallan	**macppc**
NISHIMURA Takeshi	nsmrtks	**x68k**
Nathan Williams	nathanw	**sun3**
Nick Hudson	skrll	**hp700**
Noriyuki Soda	soda	**arc**
Paul Kranenburg	pk	**sparc**
Phil Nelson	phil	**pc532**
Reinoud Zandijk	reinoud	**acorn32**
Ross Harvey	ross	**alpha**
Søren Jørvang	soren	**sgimips**
Scott Reynolds	scottr	**mac68k**
Shin Takemura	takemura	**hpcmips**
Simon Burge	simonb	**evbmips**
Simon Burge	simonb	**pmax**
Simon Burge	simonb	**sbmips**
Steve Woodford	scw	**evbppc**
Steve Woodford	scw	**mvme68k**
Steve Woodford	scw	**mvmeppc**
Takayoshi Kochi	kochi	**ia64**
Tim Rightnour	garbled	**prep**
Tohru Nishimura	nisimura	**luna68k**
Tohru Nishimura	nisimura	**sandpoint**
Wayne Knowles	wdk	**mipsco**
Wolfgang Solfrank	ws	**ofppc**

The NetBSD 4.0 Release Engineering team:

Grant Beattie	grant@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
David Brownlee	abs@NetBSD.org
Pavel Cahyna	pavel@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
Dave Carrel	carrel@NetBSD.org
Daniel Carosone	dan@NetBSD.org
James Chacon	jmc@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Ben Collver	ben@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
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
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
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
HAMAJIMA Katsuomi	hamajima@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
Jachym Holecek	freza@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
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
Antti Kantee	pooka@NetBSD.org
Masanori Kanaoka	kanaoka@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

Paul Kranenburg	pk@NetBSD.org
Lubomir Kundrak	lkundrak@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Kentaro A. Kurahone	kurahone@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
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
Anders Magnusson	ragge@NetBSD.org
MAEKAWA Masahide	gehenna@NetBSD.org
David Maxwell	david@NetBSD.org
Dan McMahill	dmcmahill@NetBSD.org
Gregory McGarry	gmcgarry@NetBSD.org
Jared D. McNeill	jmcneill@NetBSD.org
Neil J. McRae	neil@NetBSD.org
Perry Metzger	perry@NetBSD.org
Simas Mockevicius	symka@NetBSD.org
Juan Romero Pardines	xtraeme@NetBSD.org
Julio M. Merino Vidal	jmmv@NetBSD.org
Minoura Makoto	minoura@NetBSD.org
Luke Mewburn	lukem@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
Jonathan Perkin	sketch@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Rui Paulo	rpaulo@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
Gordon Ross	gwr@NetBSD.org
Steve Rumble	rumble@NetBSD.org
Ilpo Ruotsalainen	lonewolf@NetBSD.org
Heiko W. Rupp	hwr@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
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
T K Spindler	dogcow@NetBSD.org
SOMEYA Yoshihiko	someya@NetBSD.org
Bill Sommerfeld	sommerfeld@NetBSD.org
Jörg Sonnenberger	joerg@NetBSD.org
Ignatios Souvatzis	is@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
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
Arnaud Lacombe	alc@NetBSD.org
Jim Wise	jwise@NetBSD.org
Michael Wolfson	mbw@NetBSD.org

Steve Woodford	scw@NetBSD.org
Colin Wood	ender@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Yuji Yamano	yyamano@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

Dedication

NetBSD 4.0 is dedicated to the memory of Jun-Ichiro "itojun" Hagino, who died in October 2007. Itojun was a member of the KAME project, which provided IPv6 and IPsec support; he was also a member of the NetBSD core team (the technical management for the project), and one of the Security Officers. Due to Itojun's efforts, NetBSD was the first open source operating system with a production ready IPv6 networking stack, which was included in the base system before many people knew what IPv6 was. We are grateful to have known and worked with Itojun, and we know that he will be missed. This release is therefore dedicated, with thanks, to his memory.

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.
This product includes software developed by the RiscBSD kernel team
This product includes software developed by the RiscBSD team.
This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.
This product includes software developed by the University of California, Berkeley and its contributors, as well as the Trustees of Columbia University.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory.
This product includes software developed by the University of Illinois at Urbana and their contributors.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.
This product includes software developed 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 for the FreeBSD project

This product includes software developed for the NetBSD Project by Bernd Ernesti.
This product includes software developed for the NetBSD Project by Christopher G. Demetriou.
This product includes software developed for the NetBSD Project by Christos Zoulas
This product includes software developed for the NetBSD Project by Emmanuel Dreyfus.
This product includes software developed for the NetBSD Project by Frank van der Linden
This product includes software developed for the NetBSD Project by Ignatios Souvatzis.
This product includes software developed for the NetBSD Project by Jason R. Thorpe.
This product includes software developed for the NetBSD Project by John M. Vinopal.
This product includes software developed for the NetBSD Project by Matthias Drochner.
This product includes software developed for the NetBSD Project by Michael L. Hitch.
This product includes software developed for the NetBSD Project by Perry E. Metzger.
This product includes software developed for the NetBSD Project by Scott Bartram and Frank van der Linden
This product includes software developed for the NetBSD Project by Allegro Networks, Inc., and Wasabi Systems, Inc.
This product includes software developed for the NetBSD Project by Genetec Corporation.
This product includes software developed for the NetBSD Project by Jonathan Stone.
This product includes software developed for the NetBSD Project by Piermont Information Systems Inc.
This product includes software developed for the NetBSD Project by SUNET, Swedish University Computer Network.
This product includes software developed for the NetBSD Project by Shigeyuki Fukushima.
This product includes software developed for the NetBSD Project by Wasabi Systems, Inc.
This product includes software developed under OpenBSD by Per Fogelstrom Opsycon AB for RTMX Inc, North Carolina, USA.
This product includes software developed under OpenBSD by Per Fogelstrom.
This software is a component of "386BSD" developed by William F. Jolitz, TeleMuse.
This software was developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to "NetBSD", "FreeBSD", "Mach" (by CMU).
This software includes software developed by the Computer Systems Laboratory at the University of Utah.
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/).
This product includes software developed by Marshall M. Midden.
This product includes software developed or owned by Caldera International, Inc.
The Institute of Electrical and Electronics Engineers and The Open Group, have given us permission to reprint portions of their documentation.

In the following statement, the phrase ‘‘this text’’ refers to portions of the system documentation.

Portions of this text are reprinted and reproduced in electronic form in NetBSD, from IEEE Std 1003.1, 2004 Edition, Standard for Information Technology -- Portable Operating System Interface (POSIX), The Open Group Base Specifications Issue 6, Copyright (C) 2001-2004 by the Institute of Electrical and Electronics Engineers, Inc and The Open Group. In the event of any discrepancy between these versions and the original IEEE and The Open Group Standard, the original IEEE and The Open Group Standard is the referee document.

The original Standard can be obtained online at http://www.opengroup.org/unix/online.html.

This notice shall appear on any product containing this material

This product includes software developed by the Alice Group.
This product includes software developed by David Miller.

The End