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 4.0 and 5.0 Releases....3
  General kernel....3
  Networking....5
  File systems....5
  Security....6
  Drivers....6
  Platforms....10
  Userland....16
  Components removed from NetBSD....19
  Known Problems....20
Features to be removed in a later release....20
The NetBSD Foundation....20
Sources of NetBSD....21
NetBSD 5.0 Release Contents....21
  NetBSD/sparc64 subdirectory structure....22
  Binary distribution sets....22
NetBSD/sparc64 System Requirements and Supported Devices....24
  Supported machines....24
  Unsupported machines....25
  Supported devices....25
  Unsupported devices....26
Getting the NetBSD System on to Useful Media....26
Preparing your System for NetBSD installation....28
  Deciding on partition sizes....28
  Setting up Open Firmware....28
  Determining how to boot from an SBUS or PCI....29
  Configuration of network interfaces....30
Installing the NetBSD System....31
  Installing NetBSD from CD-ROM....31
  Installing NetBSD by using the NetBSD miniroot....31
  Installing NetBSD by using a NetBSD kernel on a....32
  Installing NetBSD by using a netboot setup....32
  Running the sysinst installation program....37
    Introduction....37
    General....37
    Quick install....37
    Booting NetBSD....38
    Network configuration....38
    Installation drive selection and parameters....38
    Selecting which sets to install....39
    Partitioning the disk....39
    Preparing your hard disk....39
    Getting the distribution sets....39

Installation from CD-ROM....40
Installation using ftp....40
Installation using NFS....40
Installation from an unmounted file system....40
Installation from a local directory....40
Extracting the distribution sets....40
Ensure you have the correct kernel installed....41
Finalizing your installation....41
Manual Installation of NetBSD using Solaris....41
Preparing the disk in Solaris....42
Installing NetBSD Software from Solaris....43
Creating NetBSD Device Nodes under Solaris....43
Configuring the NetBSD system from Solaris....43
Booting NetBSD for the first time....43
Post installation steps....44
Upgrading a previously-installed NetBSD System....46
Compatibility Issues With Previous NetBSD Releases....47
Issues affecting an upgrade from NetBSD 3.x releases....47
Issues affecting an upgrade from NetBSD 4.x releases....48
Using online NetBSD documentation....48
Administrivia....49
Thanks go to....49
We are....50
Legal Mumbo-Jumbo....59
The End....63

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 5.0 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 5.0 on a machine of the sparc64 architecture.

- Fetch files needed to install NetBSD. This is either a CD image, `sparc64cd-5.0.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 many thousands of third party software applications.

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 57 different system architectures (ports) across 15 distinct CPU families, and is being ported to more. The NetBSD 5.0 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 4.0 and 5.0 Releases

The NetBSD 5.0 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 the massive development that went into the NetBSD 5.0 release. The complete list of changes can be found in the CHANGES:

ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-5.0/CHANGES

and CHANGES-5.0:

ftp://ftp.NetBSD.org/pub/NetBSD/NetBSD-5.0/CHANGES-5.0

files in the top level directory of the NetBSD 5.0 release tree.

Some highlights include:

General kernel

- Introduced new synchronization primitives and made almost all core kernel subsystems use fine-grained locking, resulting in much greater performance on MP systems. See mutex(9), rwlock(9), and condvar(9).
- The threading system was rewritten and is now based on a 1:1 model.
- Made the process scheduler modular, allowing for different scheduling algorithms to be selected.
- The default scheduler implementation was rewritten, giving large performance improvements on SMP systems, and bringing support for real-time and time-sharing classes.
- Added support for SCHED_M2, a second scheduling algorithm, as an alternative to the original SCHED_4BSD algorithm.
- Added support for POSIX real-time scheduling extensions.
- Added support for POSIX asynchronous I/O.
- Added support for POSIX message queues.
- Added support for processor sets, CPU sets, and thread affinity. See pset(3), cpuset(3), and affinity(3).
- Added MI code to support in-kernel preemption and made i386 and amd64 use it. See kpreempt(9).
- Added a new implementation of software interrupts in a machine-independent way to provide software interrupts with thread context. All ports were made to use it. See softint(9).
- Added support for per-system call statistics.
- Added support for per-CPU work-queues. See workqueue(9).
- Added support for kernel core dumps to software RAID 1 sets.
- Added mremap(2), to remap virtual memory addresses.
- Added support for dynamic changing of SysV message queues and semaphore limits.
- Improved pagedaemon behavior on memory shortage.
- Added TFTPROOT option for TFTPing root ramdisk at root mount time.
- Added putter(9) (Pass-to-Userspace Transporter), a generic request-response handler for kernel-attached userspace daemons.
- Added a new power management framework, pmf(9).
- By default, the kernel no longer enters ddb(4) if it panics. To get the old behavior, set ddb.onpanic to 1 in /etc/sysctl.conf.
- ddb(4): Added a "whatis" command, inspired by Solaris.
- All ports made to use generic todr(9).
- All ports made to use timecounter(9).
- Added Mersenne Twister PRNG implementation.
- Added posix_madvise(2).
- Numerous improvements to compat_linux(8) and compat_linux32.

- Device attach and detach notifications are now exposed via /dev/drvctl.
- Added sockopt(9), a new kernel API for passing socket options.
- If no usable init(8) program can be found, set the RB_ASKNAME flag and prompt users for the init path.

Networking

- Added ipv6 fast forward support, which increases forwarding speed by hashing/caching flows.
- Added per-socket keepalive timer settings and the ability to change connection timeouts.
- Added support for automatic sizing of TCP socket buffers.
- Added a new socket option SO_NOHEADER to tell the network stack to strip the protocol header from packets received on a raw socket.
- Added accept filters. See accept_filter(9), accf_data(9), and accf_http(9).
- fast_ipsec(4): Added support for IPsec NAT-T.
- pf(4): Made "nat" and "rdr" translation rules obey state policy flags. Extended pf.conf(5) syntax to express the translation-state policies.
- Network statistics were changed to be per-CPU.
- ALTQ: Added support for source address hashing in the WFQ scheduler.
- Pulled in IANA-generated services and protocols files.

File systems

- Added wapbl(4), a preview of metadata journaling for FFS. Contributed by Wasabi Systems.
- Added read-only support for Apple HFS+ file systems. See mount_hfs(8).
- Added read-only support for Silicon Graphics EFS file systems. See mount_efs(8).
- Added write support for UDF file systems. See mount_udf(8).
- Added mount_psshfs(8), to support puffs(3) sshfs.
- Added mount_9p(8), to support 9P file services with puffs(3).
- Added rump_nfs(8), a userspace NFS client.
- Added mount_sysctlfs(8), to support browsing, querying, and modifying the sysctl(3) hierarchy.
- Added refuse(3) for FUSE compatibility functionality, layered on top of puffs(3).
- Added the user kernel file system library, ukfs(3), for standalone file system access.
- Added p2k(3), a puffs(3) to kernel vfs adaption library.
- puffs(4): Added support for NFS exporting puffs file servers.
- Added 32 bit UID/GID support for ext2fs.
- Added a newfs_ext2fs(8) utility to create Ext2 file systems.
- /proc is now mounted by default. Extended the Linux emulation of /proc to include /proc/stat, /proc/loadavg, and /proc/<pid>/statm. Linux compatibility mode is now the default. Use -o nolinux if Linux compatibility is not desired.

- Write speed to FAT file systems was significantly increased and the 255 head limit was removed.
- Imported FreeBSD's unionfs.
- `fsck_ext2fs`(8), `fsck_ffs`(8), `fsck_lfs`(8): Disable userid to username lookups by default and add -U flag to perform them.
- `scan_ffs`(8): Added -b option to search a partition for valid alternate superblocks.
- `fsck_ffs`(8): Added -x/-X options, which allow running fsck_ffs -n on a snapshot of a live filesystem.

Security

- Added support for per-user /tmp. See `security`(8).
- Added support for ASLR (Address Space Layout Randomization) in the kernel and dynamic linker. See `security`(8).
- Added an `opencrypto`(9) provider for VIA ACE (AES encryption instructions).
- Added `nsp`(4), a driver for NetOctave NSP2000, contributed by NBMK Encryption Technologies, ported from vendor FreeBSD SDK and integrated with opencrypto by Coyote Point Systems.
- `opencrypto`(9): Improved performance by adding asynchronous operation and batched submit/retrieve of requests/results. Contributed by Coyote Point Systems.
- `cgd`(4): Changed the default IV to encblkno1, which is faster without a real loss of security.
- `openssl`(1): Enabled support for Camellia.

Drivers

- Audio:
  - Added `pad`(4), a pseudo-audio device driver for feeding back raw PCM data to userland.
  - Added `sgsmix`(4), a driver for the SGS 7433 mixer found in some G3 Macs.
  - `umidi`(4): Added support for Roland UA25, UA4FX, and SonicCell devices.
  - `dbri`(4): Added support for audio input.
  - `auvia`(4) now works on big endian machines.
  - `azalia`(4): AD1984 support was greatly improved. Added support for Realtek ALC662-GR and ALC269 codecs, which are found in EeePCs. Added support for ALC268.
  - `auich`(4): Fixed a clock accounting problem that prevented detecting sample rates correctly.
- Hardware Monitoring:
  - Imported envsys2, which brings, among other things, the ability to send events to `powerd`(8). See `envsys`(4) and `sysmon_envsys`(9).
  - Added support for hardware monitoring on Sun Ultra Enterprise 450.
  - Added `aiboost`(4), a driver for the ASUS AI Booster ACPI Hardware monitor.
  - Added `smsc`(4), a driver for the hardware monitoring portion of the SMSC LPC47B397.
  - Added `aps`(4), a driver for the IBM Thinkpad Active Protection System.
  - Added `coretemp`(4), a driver for Intel Core (and newer) on-die thermal sensors.

  - Added finsio(4), a driver for various Fintek Super I/O chips.
  - Added amdtemp(4), a driver for AMD CPU on-die thermal sensors.
  - Added dbcool(4), a driver for Analog Devices dbCool chips including ADT7460, ADT7463, ADT7467, and ADM1030.
  - Added alipm(4), a driver for the Acer Labs M7101 Power Management Controller.
  - Added admtemp(4), a driver for the Analog Devices ADM1021 temperature sensor.
  - viaenv(4): Added support for VIA VT8231.
  - nsclpcsio(4): Added support for the VLM logical device.
  - lm(4): Added an i2c attachment for the LM78 family of temp sensor and fan controllers.
- Networking:
  - Added jme(4), a driver for the JMicron Technologies JME250 Gigabit Ethernet and JME260 Fast Ethernet controllers.
  - Added u3g(4), a driver for many multi-port 3G datacards.
  - Added iwn(4), a driver for the Intel Wireless LAN 4965AGN adapter.
  - Added zyd(4), a driver for ZyDAS ZD1211/ZD1211B USB IEEE 802.11b/g wireless network devices.
  - Added uhmodem(4), a driver for 3G wireless modems including Huawei E220 and E620, E-mobile D01HW and D02HW, and NTT DoCoMo a2502.
  - Added lii(4), a driver for the Atheros L2 Fast Ethernet controller.
  - Added btuart(4), a driver for Bluetooth HCI UART (H4).
  - Firmware images are now shipped for ipw(4), iwi(4), wpi(4), and iwn(4). For ipw(4) and iwi(4), the Intel EULA has to be accepted via sysctl(8).
  - fxp(4): Fixed some TX timeout and RX pool corruption problems. Added a workaround for a hardware ip4csum-tx bug.
  - bge(4): Added support for BCM5786 and BCM5906(M). Fixed fiber card support.
  - nfe(4): Added support for NVIDIA MCP67/73 Ethernet controllers. Fixed wakeup issues on some newer chips. Fixed a problem with receiving jumbo frames.
  - btbc(4): Added support for AnyCom BlueCard devices.
  - rum(4): Added support for MELCO WLI-U2-SG54HP, PLANEX GW-US54Mini2, COREGA CG-WLUSB2GL and K.K. CG-WLUSB2GPX, and ABOCOM WUG2700.
  - makphy(4): Added support for Marvell 88E1116 Gigabit PHY.
  - gem(4): Added support for Sun PCI SX fiber cards and Sun SBus SX fiber cards.
  - wm(4): Added support for the Intel PRO/1000 PT Quad Port Server Adapter. Fixed a bug on receiving a jumbo frame which lead to a panic in sbcompress(). Added support for more ICH9 devices. Fixed an EEPROM-trashing bug on ICH8 and ICH9 chipsets.
  - udav(4): Added support for Shantou ADM8515.
  - brgphy(4): Added support for BCM5708C.

  - re(4): Made hardware vlan(4) insertion/extraction work properly. Added support for the Realtek 8102E/8102EL PCIe 10/100 Ethernet adapters, as well as the 8111C chips that are found on many Intel-based motherboards.
  - sk(4), msk(4): Fixed a lock panic on receiving jumbo packets.
  - msk(4): Fixed a Yukon EC Ultra cold power up issue.
  - bnx(4): Added support for SerDes controllers.
  - vge(4): Added ifconfig down and ALTQ support.
- Storage:
  - Added arcmsr(4), a driver for Areca Technology Corporation SATA RAID controllers.
  - Added siisata(4), a driver for Silicon Image SteelVine SATA-II controllers (SiI3124, SiI3132, and SiI3531).
  - isp(4): Major update, including 4Gb (24XX) card support and new firmware sets.
  - piixide(4): Added support for ICH10.
  - ahcisata(4): Added support for ATAPI devices.
  - svwsata(4): Added support for ServerWorks HT-1000 SATA controller.
  - njata(4): Added support for Workbit CF32A CF adapter.
  - viaide(4): Added support for VIA CX700, CX700M2, NVIDIA MCP67, and MCP73/77 controllers.
  - mfi(4): Added support for LSI SAS1078 and Dell PERC 6 controllers.
  - mpt(4): Fixed performance problems for old revisions of the Symbios 53c1030.
  - cac(4): Added initial bio(4) support; only volume status is handled at this time.
  - siop(4): Added support for the non-PCI NCR 53c720/770 in big-endian mode.
  - twa(4): Added support for 3ware 9650 and 9690, based on contributions from Wasabi Systems.
  - ciss(4): Added bio(4) support.
  - ataraid(4): Added NVIDIA MediaShield, JMicron RAID, and Intel MatrixRAID support. Added support for status reports through bio(4).
  - ixpide(4): Added support for ATI SB700/SB800 controllers.
  - aac(4): Added support for raw I/O mode and >2TB.
  - umass(4): Added support for Sony GPS GPS-CS1. Fixed a panic on device removal.
  - aic(4): Worked around an rbus resource allocation problem so that aic PCMCIA cards work again.
  - Implemented DIOCACHESYNC for RAIDframe, vnd(4), and cgd(4).
  - rccide: Added support for the ServerWorks HT-1000 IDE controller.
  - wd33c93: Enabled Fast SCSI transfers for WD33C93B parts with appropriate input clocks.
- USB:
  - Added uvideo(4), a driver for USB Video Class capture devices, from Patrick Mahoney's Google Summer of Code 2008 project.

  - Added uslsa(4), a driver for CP210x USB-RS232 devices.
  - Added uchcom(4), a driver for WinChipHead CH341/340 and HL-340 USB-Serial adapters.
  - Added uberry(4), a driver to allow RIM BlackBerries to charge from the USB port.
  - Added appropriate bus_dmamap_sync(9) calls to uhci(4), ohci(4), and ehci(4) to prevent the CPU from reordering loads and stores against DMA descriptors. This fixes "host controller process error/host controller halted" errors.
  - ehci(4): Added isochronous transfer support, contributed by Jeremy Morse as part of his Google Summer of Code 2008 project.
  - ukbd(4): Added support for function keys F16 through F19.
  - uplcom(4): Added support for Willcom WS002IN PHS and SMART Technologies-badged devices. Recognize Corega CG-USBRS232R as a serial device.
  - ugensa(4): Added support for Novatel Wireless Merlin CMDA and Ovation U727.
  - ubsa(4): Added support for CDMA modems sold by Eurotel/O2.
  - uftdi(4): Added support for Sealevel SeaPORT+4 USB to Serial adapter.
  - slhci(4): Replaced with Matthew Orgass's driver.
- Graphics and Video:
  - Switched to X.Org on amd64, i386, macppc, sgimips, shark, and sparc64.
  - Added video(4), a video4linux2 compatible capture interface, part of Patrick Mahoney's Google Summer of Code 2008 project.
  - Added uvideo(4), a driver for USB Video Class capture devices, from Patrick Mahoney's Google Summer of Code 2008 project.
  - Added pseye(4), a driver for the Sony PLAYSTATION(R) Eye USB webcam.
  - Added genfb(4), a generic framebuffer console driver with PCI and SBus frontends.
  - Added AGPv3 support to VIA AGP driver.
  - Added support for Fujitsu AG-10e graphics cards.
  - Added r128fb, an accelerated console driver for ATI Rage 128 graphics controllers.
  - Added isv(4), a driver for the IDEC Supervision/16 image capture board.
  - wscons(4): Added scrollback support to vcons.
  - cgfourteen(4): Added support for wscons.
  - agp(4): Added support for ALI M1689, MB i965Q, Intel Q33/35/G33, Intel 945GME, and Intel 946GZ.
  - bktr(4) now works on amd64.
- Miscellaneous:
  - Added pud(4), a driver that makes it possible to implement block and character devices in userspace.
  - Added spdmem(4), a Serial Presence Detect driver that decodes technical specs stored in the eeprom on common types of memory modules.

- Added bcsp(4) to support the BlueCore Serial Protocol.
- Added thinkpad(4), a driver to support IBM/Lenovo Thinkpad hotkeys, brightness controls, and temperature and fan monitoring.
- Added gcscpcib(4), a driver for the AMD CS5535 and CS5536 Companion Device with support for the timecounter, watchdog timer, and GPIO.
- Added ichsmb(4), a driver for Intel ICH SMBus controllers.
- Added asus(4), a driver for ASUS ACPI hotkeys as found in the EeePC.
- Added acpidalb(4), a driver for PNP0C32 ACPI hotkeys, aka the Direct Application Launch Buttons.
- Added hpqlb(4), a driver for hotkeys on some HP notebooks.
- adb(4): New and simplified MI ADB drivers.
- ichlpcib(4): Added support for the TCO (watchdog) on ICH6 or newer chipsets. Now runs on EM64T systems as well.
- itesio(4): Added support for the watchdog timer.
- ulpt(4): Implemented non-blocking read.
- puc(4): Added support for the I-O DATA RSA-PCI 2 port serial board, Digi International 4 and 8 port boards, and B&B Electronics MIPort serial boards.
- piixpm(4): Added support for ATI SB600, SB700, and SB800 SMBus controllers.
- wscons(4): Added support for the Colemak keyboard layout.
- com(4): Added support for PCMCIA Sierra Wireless Aircard 850.
- nfsmb(4): Added support for numerous NVIDIA chipsets.

Platforms

- x86 (i386, amd64):
  - Removed GENERIC.MP and made MULTIPROCESSOR mandatory.
  - Enabled BUFQ_PRIOCSCAN in GENERIC for better responsiveness under heavy disk I/O.
  - Added fatboot, a sub-512 byte bootloader that can be written into the first sector of a FAT16 file system.
  - Added support for enhanced speedstep on Intel Atom CPUs.
  - Added a High Precision Event Timer driver.
  - Merged amd64 and i386 pmap. Large pages are always used if available.
  - Added support for the Intel 82G965 chipset.
  - Added boot.cfg(5) to configure the bootloader.
  - The boot menu now allows disabling ACPI and/or SMP.
  - Added support for ACPI S3 (suspend-to-RAM) on MP systems.
  - Added mbr(8) variants that directly access serial ports.

  - It is now possible on some systems to get to a ddb(4) session on a VGA console if the system crashed while X11 was running.
  - Added detection of errata for AMD Family 10h steppings A and 2.
  - Added checks for erratum 261 on AMD Family 10h stepping 3 processors.
  - boot(8): Added support for the multiboot protocol. This allows booting Xen without GRUB.
  - The kernel now recognizes CD-ROMs as booted devices.
  - Added an Intel On Demand Clock Modulation driver.
- i386:
  - Removed support for 80386 level CPUs.
  - Added support for Microsoft Xbox.
  - Added cmos(4), a driver for CMOS RAM.
  - Added support for VIA Esther.
  - i386 bootblocks are now able to boot amd64 kernels and support for a.out kernels has been removed.
  - Removed pccons.
  - Added support for sparse kernel core dumps.
  - Removed the MATH_EMULATE option.
  - Added the i386-specific COMPAT_30_PTHREAD option, which restores binary compatibility with netbsd-3 libpthread.
- amd64:
  - Added support for PCI_BUS_FIXUP and PCI_ADDR_FIXUP kernel options.
- acorn32:
  - Added support for Kinetic cards.
  - Fixed up boot32 to work on A7000+.
- algor:
  - Fixed a delay(9) issue and now the P5064 kernel works in gxemul.
- alpha:
  - Make dynamic executables work under compat_osf1(8) again.
- amiga:
  - Added a wdc(4) frontend for the buddha and catweazle Z2 hardware.
- arc:
  - Added support for booting from UFS2 partitions.
  - Added sysinst(8) support.
- arm:
  - Added armv6 kernel support.

  - Added arm32 support for kernel crashdumps.
  - Added VFP support.
  - Added AT91 support.
  - Added FA526 support.
  - Added the missing kernel bits that kept `eeprom`(8) from actually changing firmware settings.
- atari:
  - Fixed the sysinst miniroot image and retired the old miniroot.
  - Fixed a hang on loading md_root from floppy during installation.
  - Many pmap bugs were fixed and it was synced with amiga's pmap.
  - Fixed a lost interrupt problem on Falcon `wdc`(4).
  - Renamed the BOOT kernel to SMALL030 and removed BOOTX.
  - Added support for installation from GEMDOS partitions.
  - Fixed an ncr5380 SCSI freezing problem that occurred during boot.
- bebox:
  - Converted to generic powerpc PCI and ISA frameworks.
  - Added support for `genfb`(4).
- cobalt:
  - Added `sysinst`(8) support.
  - Added support for booting from UFS2 partitions.
  - Added support for booting from Ext2fs partitions.
  - Added netboot support for 21041 on Qube 2700.
  - Startup and shutdown messages, as well as a banner and kernel name are now printed on the LCD panel.
  - Added kernel and bootloader support for optional Z85C30 serial on Qube 2700.
  - Fixed a botched `spl`(9) bug which could cause a network freeze on traffic between two network interfaces.
- evbarm:
  - Added support for the Cortina Systems SL3516 evaluation board.
  - Added support for NPE Ethernet found on IXP425 boards.
  - Added a PXA2x0 RTC driver.
- evbmips:
  - Added support for Infineon ADM5120.
- evbppc:
  - Added support for Xilinx Virtex II-Pro/4-FX.

  - Converted PM/PPC to generic powerpc clock, PCI, and interrupt frameworks.
  - Fixed sysinst to refer to the proper WALNUT and OPENBLOCKS266 kernels.
- hp700:
  - Added support for booting from CD-ROM images.
  - Fixed kernel profiling.
  - Kernels are now compiled with -O2.
  - Added `gdb`(1) support.
  - Added bootinfo support.
  - Added support for NCR 53C720.
- hpcmips:
  - Fixed a silent hangup problem on TX3922 machines.
- ibmnws:
  - Converted to generic powerpc clock, PCI, and interrupt frameworks.
- landisk:
  - Added `sysinst`(8) support.
  - Fixed MD `bus_dmamap_load`(9) so that NFS write works with `re`(4).
- mac68k:
  - Added support for more NuBus video cards, including VillageTronic MacPicasso 340.
  - Added support for Creative Systems Inc Hurdler CPI NuBus parallel printer card.
  - Switched `sn`(4) to use the MI SONIC driver.
- macppc:
  - Switched to generic powerpc interrupt, clock, PCI, `bus_dma`(9), `bus_space`(9), SMP, and IPI frameworks.
  - `genfb`(4) is now the default framebuffer.
  - Switched to the new ADB subsystem.
  - Added support for bus speed control found in some Intrepid-based laptops like the 800MHz iBook G4.
  - Converted to shared ofw_autoconf.
  - Converted to shared powerpc major numbers.
  - Added an experimental Apple UniNorth AGP driver.
- mvmeppc:
  - Converted to generic powerpc clock/PCI/interrupt frameworks.
- m68k:
  - Fixed a panic in FPE code caused by 040/060 specific FDADD instructions on 020/030 machines.

  - gcc(1): Fix jump table addressing in the M68k codegen.
  - Fixed floating point handling regressions.
- ofppc:
  - Completely rewrote the ofppc port from scratch.
  - Added support for the bplan/Genesi Pegasos II.
  - Added support for the IBM 7044-270.
  - Added support for the IBM 7046-B50.
  - Added install support and booting from RAID.
  - Fixed ofwboot to work on an IBM CHRP-based RS/6000.
  - Added support for SMP.
- pmppc:
  - Removed pmppc as a port and moved it under evbppc.
- powerpc:
  - Added a generic interrupt handler framework.
  - Added generic shared timecounter-based clock routines.
  - Added shared PReP memory maps.
  - Added generic powerpc OpenFirmware code.
  - Added a generic powerpc IPI framework.
  - Added new generic powerpc SMP support framework.
  - Added tuned assembler from IBM for memcpy(3), memmove(3), and memcmp(3).
  - Added booted_kernel and booted_device sysctl.
  - Added the missing kernel bits that kept eeprom(8) from actually changing firmware settings.
- prep:
  - Switched to generic powerpc PCI, ISA, interrupt, clock, IPI, and SMP frameworks.
  - Added support for the MTX604 RAVEN based machine. Other RAVEN based machines are also likely to work now.
  - Added support for floppy controller/drive.
- sandpoint:
  - Switched to generic powerpc interrupt, clock, PCI, and ISA frameworks.
  - Added a DHCP/NFS bootloader.
  - Converted to shared powerpc major numbers.
- sgimips:
  - Added support for the SGI O2 PS/2 controller macekbc(4), onboard display adapter crmfb(4), and audio driver mavb(4).

  - Added support for Set Engineering's GIO Fast Ethernet board: `tl`(4).
  - Added support for the SGI Light/Entry/Starter LG1/LG2 framebuffers found in Indigo systems: `light`(4).
  - Added support for the `sq`(4) interface on the Challenge S's IOPLUS mezzanine.
  - Added support for booting from UFS2 partitions.
  - Many bug fixes and improvements to the MAC-110 Ethernet driver found in the O2.
- sh3:
  - Implemented fast path TLB miss handling.
  - Fixed compilation of native sh3 gcc on 64-bit build machines.
  - Added support for single-stepping in `ddb`(4).
- shark:
  - Added `dhclient`(8) to the install disk image.
  - The installation disk now uses wscons.
  - Removed pccons, which has been obsoleted by `wscons`(4).
  - X works again on revision 4 machines.
- sparc:
  - Added support for booting from UFS2 partitions.
  - Added support for Sun PGX32 / TechSource Raptor GFX 8P to `genfb`(4).
- sparc64:
  - SMP now works.
  - Fixed boot device matching on U5/U10 machines to properly distinguish cd0 from wd0.
- vax:
  - Turned on generation of PIC code.
  - Added an accelerated driver for SPX graphics cards found in some VAXstations.
  - Fixed Unibus support on VAX11/780 and 8600 models.
- xen:
  - Added support for amd64, both as dom0 and domU.
  - Added support for the i386 PAE extension to Xen3 domU.
  - Fixed PCI_BUS_FIXUP/PCI_ADDR_FIXUP support.
- zaurus:
  - Added the zaurus port, with support for Sharp C3x00 PDAs.
  - Allow Zaurus screen to rotate 90 degrees to a usable state with the keyboard.
  - Added support for detecting the AC adapter and charging the battery.
  - Added `sysinst`(8) support.

Userland

- 3rd party software updates:
  - ACPICA 20080321
  - am-utils 6.2a3
  - BIND 9.5.0-P2
  - file 4.26
  - GNU gcc 4.1.3-20080831
  - GNU gdb 6.5 on all ports
  - IPFilter 4.1.29
  - (n)awk 20070501
  - NTP 4.2.4p6
  - nvi 1.81
  - openpam 20071221
  - OpenSSH 5.0p1
  - OpenSSL SNAP-20080509
  - PF from OpenBSD 4.2
  - Postfix 2.5.4
  - texinfo 4.8
- Libraries:
  - `pthread`(3):
    - Major overhaul, resulting in many performance and stability improvements.
    - Added POSIX real-time extensions.
    - Diagnostic assertions are now disabled by default. Application errors will be reported only by error return from pthread APIs. The assertions can be re-enabled by setting the PTHREAD_DIAGASSERT environment variable as desired.
    - Added a PTHREAD__COMPAT build flag which creates a libpthread that can be dropped into a NetBSD 2/3/4 chroot and used on a kernel without scheduler activations support.
  - Replaced the default `malloc`(3) with jemalloc, bringing a significant performance boost for many threaded workloads that make heavy use of malloc.
  - `curses`(3):
    - Merged wide curses work done by Ruibiao Qui as a Google Summer of Code project.
    - Added `termattrs`(3) and `term_attrs`(3).
    - Added `getwin`(3) and `putwin`(3).
  - `util`(3):
    - Added `estrndup`(3).

    - Added raise_default_signal(3).
  - Added C99 complex support (float/double) to math(3).
  - proplib(3):
    - Implemented prop_dictionary_make_immutable(3).
    - Added prop_array_util(3) functions.
  - libc:
    - Added dehumanize_number(3).
    - Added posix_memalign(3).
    - Replaced O(nm) versions of strspn(3), strcspn(3), and strpbrk(3) with O(n+m) implementations.
    - Added getlogin_r(2).
    - Added the C99 functions imaxabs(3) and imaxdiv(3).
    - Enabled atomic_ops(3) in userspace.
    - queue(3): Added TAILQ_CONCAT() and STAILQ_CONCAT().
  - Added libgcov.
- Imported bozohttpd, a small HTTP server. See httpd(8).
- Imported dhcpcd 4.0.10.
- Imported openldap 2.4.11.
- Added rump(3), the Runnable Userspace Meta Program framework. Allows running kernel code in userspace applications.
- Added cpuctl(8), a utility that allows placing CPUs online/offline.
- Added schedctl(8), a program to control scheduling of processes and threads.
- Added psrset(8), a utility to control processor sets.
- Added atf(7), the Automated Testing Framework, Julio M. Merino Vidal's 2007 Google Summer of Code project.
- Added newgrp(1), a utility to change effective group ID.
- Added tcpdrop(8), a utility to drop tcp(4) connections.
- Added acpitools from FreeBSD: acpidump(8) and amldb(8).
- Added dkscan_bsdlabel(8) to scan disks for BSD disklabels.
- Added btkey(1), a utility to manage Bluetooth link keys in OS and device storage.
- Added svhlabel(8), a tool to update disklabel(5) from SGI Volume Header, like mbrlabel(8) for MBR labels.
- Added pcc(1) as an alternative compiler.
- Added btpand(8), a Bluetooth Personal Area Networking profile daemon.
- Added c99(1) as a wrapper to run cc(1) in C99 mode.

- Added support for Solaris $ORIGIN etc. expansions in paths.
- Added BUILDSEED support to build.sh to yield reproducible C++ builds.
- Added support for "cc -m32" on amd64 and sparc64 to generate 32 bit binaries. 32bit libraries are now provided on these two ports.
- `ld`(1) can now link 32bit objects on amd64.
- `vi`(1) has been updated to nvi 1.81, which supports internationalization. It also grew a new NetBSD-specific expandtab option.
- `pkill`(1): Added the -l (long format) option.
- `find`(1): Added the -delete and -E (extended regex) options.
- `xargs`(1): Replaced with FreeBSD's while keeping our GNU compatible exit values.
- `sdiff`(1): Replaced by OpenBSD's `sdiff`(1).
- `pax`(1): Added a -V flag for verbose summary without listing.
- `top`(1): Allow a single process to be selected by pid. Added a thread mode that displays LWPs.
- `scsictl`(8): Added a setspeed command.
- `split`(1): Added a new option "-n chunk_count" that splits the input into chunk_count smaller files.
- `df`(1): Fixed the -P option and added the -g (gigabytes) option.
- `wtf`(6) now searches pkgsrc's help database when called inside a package directory.
- `atactl`(8): Improved SATA support.
- `wlanctl`(8): Added a -p flag that only prints public nodes.
- `btconfig`(8): Added a new "rssi" option to toggle inquiry results with RSSI.
- `ifconfig`(8): Added "list scan" to ifconfig, which lists access points in the neighborhood.
- `newsyslog.conf`(5) gained a "J" flag to bzip2 logfiles.
- `fdisk`(8) now reports the first active partition.
- `bioctl`(8) was rewritten to handle new features like creating and removing hot-spares, pass-through disks and RAID volumes, start/stop consistency checks in volumes.
- `savecore`(8) now uses the raw device to read crashdumps.
- `make`(1): Implemented "-dl" (aka LOUD) to override "@" at the start of script lines.
- `monop`(6): The save and restore format changed, breaking compatibility with already broken previous save files.
- `iconv`(1) now allows SUSv3 syntax.
- `lint`(1): Added _Complex support.
- `ftp`(1): Added epsv6 and epsv to disable extended passive mode.
- `getent`(1): Added support for "netgroup" databases.
- `ypserv`(8): Disabled libwrap address to hostname lookups to avoid the chance of ypserv blocking for an extended period of time due to a long DNS timeout.

- postfix(1): Enabled LDAP support for tables.
- amd(8): Enabled LDAP support for maps.
- newfs(8): Added support for the "t" (terabytes) suffix.
- grep(1): A warning is now printed if "-r" is used without specifying an argument.
- db(1): Added support for encoding or decoding VIS_HTTPSTYLE, and for tuning the page size of the database.
- daily.conf(5): Added run_fsck_flags to allow passing extra options to the daily fsck -n.
- sysinst(8) now supports the Colemak and Dvorak keyboard layouts.
- od(1) Added support for the "-A addressformat" flag.
- etcupdate(8): Removed the "-b binarydir" and "-s srcdir/etc" options which were deprecated in NetBSD-4.0. Deprecate the "-s tgz1:tgz2" option; please use "-s tgz1 -s tgz2" instead.
- postinstall(8): Deprecated the "-s tgz1:tgz2" option; please use "-s tgz1 -s tgz2" instead.
- sed(1): Added the -r flag, which is an alias for -E, to be compatible with GNU sed.
- patch(1): Merged improved version from DragonFly. patch -b now behaves as specified by POSIX.
- rc.conf(5): ifconfig_xxN variables may now have multi-line values, just like /etc/ifconfig.xxN files, and semicolons may be used instead of line breaks.
- ls(1): -n now implies -l.
- ps(1): Added the -A option, to display information about all processes. Use "O" for LSONPROC like Solaris instead of bundling LSIDL, LSRUN, and LSONPROC to "R".
- ksh(1): Fixed POSIX mode interpretation of backslashes inside backquotes inside double quotes.
- makefs(8): Made the allow-multidot option for cd9660 useful.
- restore(8) now works on Linux dump volumes, by ignoring extended attribute records on these volumes.
- Added an rc.d(8) script for rndctl(8).
- MAKEDEV(8) now creates /dev rather than init(8).
- MAKEDEV(8) now uses mtree(8) in preference to pax(1) and mknod(8), making node creation more efficient, and mount_tmpfs(8) in preference to mount_mfs(8) when creating a memory file system.
- MAKEDEV.local(8) can now use functions defined in MAKEDEV(8).

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

Components removed from NetBSD

In this release of NetBSD, the following software components have been removed from the system. Some were not useful anymore, or their utility did not justify the maintenance overhead. Others were not working properly and there was a lack of interest in fixing them.

- The pc532 port.
- Support for systrace.
- HP-UX binary compatibility.

- The esl driver.
- Support for NQNFS.
- The TPCONS option in netiso.

Known Problems

Using block device nodes directly for I/O may cause a kernel crash when the file system containing `/dev` is FFS and is mounted with -o log. Workaround: use raw disk devices, or remount the file system without -o log.

Occassionally, gdb may cause a process that is being debugged to hang when "single stepped". Workaround: kill and restart the affected process.

gdb cannot debug running threaded programs correctly. Workaround: generate a core file from the program using `gcore`(1) and pass the core to gdb, instead of debugging the running program.

Statically linked binaries using pthreads are currently broken.

Certain early revision AMD Opteron and Athlon 64 processors contain a bug that may cause system instability when running with more than one CPU core active. An OS-level workaround for this issue has been prepared but was not ready in time for inclusion in NetBSD 5.0. It will be available as part of a later release in the 5.0 series.

Large filesystems (over 2TB) may sometimes falsely claim to be out of space. A fix for this is available, but was not made in time for 5.0. It will be available as part of a later release in the 5.0 series.

The sparc port does not have functional SMP support in this release.

Features to be removed in a later release

The following features are to be removed from NetBSD in the future:

- Support for soft dependencies, also knows as soft updates (see "softdep" in `mount`(8)) will be removed in the next major release.

  NetBSD 5.0 includes a preview of WAPBL (Write Ahead Physical Block Logging), which will replace soft dependencies in the next major release. See `wapbl`(4) and
  http://mail-index.netbsd.org/netbsd-announce/2008/12/14/msg000051.html
  for details.

- Support for Xen 2.0.x. The Xen-3 and hypervisor interface is diverging from Xen-2 as development is ongoing, increasing the maintenance cost for NetBSD. It is expected that the netbsd-5 branch will get support for PCI pass-though to domUs before the Xen-2 support is removed from HEAD.

  It should be considered as deprecated. Users are expected to not rely on it any more beyond this major release.

  Further, at least version 3.1 of Xen will be required to run NetBSD as Dom0 or DomU.

The NetBSD Foundation

The NetBSD Foundation is a tax exempt, not-for-profit 501(c)(3) corporation that devotes itself to the traditional goals and Spirit of the NetBSD Project and owns the trademark of the word "NetBSD". It supports the design, development, and adoption of NetBSD worldwide. More information on the NetBSD Foundation, its composition, aims, and work can be found at: **http://www.NetBSD.org/foundation/**

Sources of NetBSD

Refer to **http://www.NetBSD.org/mirrors/**

NetBSD 5.0 Release Contents

The root directory of the NetBSD 5.0 release is organized as follows:

`.../NetBSD-5.0/`

`CHANGES` Changes between the 4.0 and 5.0 releases.

`CHANGES-5.0` Changes between the initial 5.0 branch and final release of 5.0.

`CHANGES.prev` Changes in previous NetBSD releases.

`LAST_MINUTE` Last minute changes and notes about the release.

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

`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 5.0 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.
82 MB gzipped, 422 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.
7 MB gzipped, 29 MB uncompressed

src This set contains all of the base NetBSD 5.0 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
58 MB gzipped, 284 MB uncompressed

syssrc This set contains the sources to the NetBSD 5.0 kernel for all architectures as well as the `config`(1) utility.
36 MB gzipped, 180 MB uncompressed

xsrc This set contains the sources to the X Window System.
133 MB gzipped, 699 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 AT&T System V UNIX checksums for the various files in that directory, in the format produced by the command:
cksum -o 2 *file*.

The 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 5.0 release is found in the `sparc64` subdirectory of the distribution: `.../NetBSD-5.0/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-GENERIC.MP.gz`
A gzipped NetBSD kernel containing code for everything supported in this release. This kernel supports SMP on systems with more than one CPU.
`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 5.0 release for sparc64. The binary distribution sets can be found in the `sparc64/binary/sets` subdirectory of the NetBSD 5.0 distribution tree.

Note: If you want to install 32-bit only binaries, you will still need to boot using the sparc64 installation tools. When sysinst 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 5.0 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.
33 MB gzipped, 103 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.
59 MB gzipped, 225 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, 9 MB uncompressed

kern-GENERIC
This set contains a NetBSD/sparc64 5.0 `GENERIC` kernel named `/netbsd`. You *must* install this or the kern-GENERIC.MP distribution set.
4 MB gzipped, 10 MB uncompressed

kern-GENERIC.MP
This set contains a NetBSD/sparc64 5.0 `GENERIC.MP` kernel, which will use multiple processors (if present), named `/netbsd`.
4 MB gzipped, 10 MB uncompressed

man This set includes all of the manual pages for the binaries and other software contained in the **base** set. Note that it does not include any of the manual pages that are included in the other sets.
12 MB gzipped, 50 MB uncompressed

misc This set includes the system dictionaries, the typesettable document set, and other files from `/usr/share`.
4 MB gzipped, 13 MB uncompressed

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

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on X.Org. 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.
8 MB gzipped, 25 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
16 MB gzipped, 65 MB uncompressed

xfont Fonts needed by the X server and by X clients.
32 MB gzipped, 74 MB uncompressed

xetc Configuration files for X which could be locally modified.
1 MB gzipped, 1 MB uncompressed

xserver The X server. This includes the modular Xorg and Xvfb servers.
12 MB gzipped, 42 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 -xzpf`** command from the root directory (`/`) of your system.

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 ~120 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, including:
    - Tulip-compatible (*tlp*)
    - Intel (*fxp*)
    - Intel gigabit (*wm*)
    - NE2000-compatible (*ne*)
    - Realtek (*rtk* or *re*)
    - SysKonnect XMAC II (*sk*)
    - Marvell GMAC (*msk*)
  - Several USB Ethernet interfaces, such as (*aue*, *cue*, and *kue*)
- Wireless network adapters
  - Atheros wireless cards (*ath*)
- 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 Xorg.
  - UPA devices (ffb and afb, also called Creator3D) 24 bpp accelerated framebuffer. Accelerated X via the integrated Xorg.
  - On-board ATI mach64 framebuffers (found in Ultra 5/10).
- 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, 3114, and 3512 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*)

- Other PCI devices
  - PCI "universal" communications cards (*puc*)

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

Unsupported devices

Smart card readers.

Getting the NetBSD System on to Useful Media

Note that if you are installing or upgrading from writable media, it 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 instructions on installation.

MS-DOS floppy NetBSD does not include split distribution sets for installation by floppy. However, 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 `2001:4f8:3:7:230:48ff:fe31:43f2` (as of April, 2009).

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.

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.

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-5.0
# 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 200 MB. A full binary installation including X takes over 600 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.

Note that the OBP on Ultra 1 and Ultra 2 machines can only boot from the first 4Gb of the disk, so this limits the size of the root partition on these models.

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. 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 boot devices. These cards do not automatically create **devalias** entries, 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
```

The root device can also be specified in your kernel config file.

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 CD-ROM. If you don't have access to a CD-ROM or CD-ROM 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 CD-ROM

Installing from CD-ROM 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. See

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

   for details.

   - 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. Still, you should read this document and have it in hand when doing the installation process. This document tries to be a good guide to 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 32 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 215 MB more.
   - The Quick Installation
     - Boot the system as described above. You should be at the **sysinst** main menu.

```
.*********************************************.
* NetBSD-5.0 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 `/etc/rc.conf` to match your needs.
- Your installation is now complete.

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** which disk to use. 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 do not intend to use networking 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 when asked to provide a server.

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. You should see disk names like `wd0`, `wd1`, `sd0` or `sd1`.

7. *Selecting which sets to install*

   The next step is to choose which distribution sets you wish to install. Options are provided for full, minimal, and custom installations. If you choose sets on your own, **base**, **etc**, and a kernel must be selected.

8. *Partitioning the disk*

   - Which portion of the disk to use.

     You will be asked if you want to use the entire disk or only part of the disk. If you decide to use the entire disk for NetBSD, **sysinst** will check for the presence of other operating systems and you will be asked to confirm that you want to overwrite these.

9. *Editing the NetBSD disklabel*

   The partition table of the NetBSD part of a disk is called a *disklabel*. If your disk already has a disklabel written to it, you can choose **Use existing partition sizes**. Otherwise, select **Set sizes of NetBSD partitions**.

   After you have chosen your partitions and their sizes (or if you opted to use the existing partitions), you will be presented with the layout of the NetBSD disklabel and given one more 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.

10. *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, select **yes**.

    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.

11. *Getting the distribution sets*

    The NetBSD distribution consists of a number of *sets* that come in the form of gzipped tarfiles. At this point, 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'.

12. *Installation from CD-ROM*

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

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

13. *Installation using ftp*

    To be able to install using ftp, you first need to configure your network setup if you haven't already done so. **sysinst** will do this for you, asking you if you want to use DHCP. If you do not use DHCP, you can enter network configuration details yourself. If you do not have DNS set up for the machine that you are installing on, you can just press `RETURN` in answer to this question, 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, you will need to specify an IP address instead of a hostname for the ftp server.

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

14. *Installation using NFS*

    To be able to install using NFS, you first need to configure your network setup if you haven't already done so. **sysinst** will do this for you, asking you if you want to use DHCP. If you do not use DHCP, you can enter network configuration details yourself. If you do not have DNS set up for the machine that you are installing on, you can just press `RETURN` in answer to this question, 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, you will need to specify an IP address instead of a hostname for the NFS server.

15. *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.

16. *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.

17. *Extracting the distribution sets*

    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 on machines with slow graphics consoles or serial consoles. Alternatively, you can choose to see 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, the 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 from coming up without access restrictions.

18. *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
    ```

19. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 5.0. 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**. To boot from your first disk, type:

```
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. The most important steps are 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).

   When you have finished editing `/etc/rc.conf`, type **exit** at the prompt to leave the single-user shell and continue with the multi-user boot.

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

   ```
   ifconfig_hme0="inet 192.0.2.123 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. 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.

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 installed the X Window System, you may want to read the chapter about X in the NetBSD Guide:
   http://netbsd.org/docs/guide/en/chap-x.html

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, pkgsrc. pkgsrc automatically handles any changes necessary to make the software run on NetBSD. This includes the retrieval and installation of any other packages on which the software may depend.

   - More information on the package system is available at
     http://www.NetBSD.org/docs/software/packages.html
   - A list of available packages suitable for browsing is at
     ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc/README.html
   - Precompiled binaries can be found at
     ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/
     usually in the sparc64/5.0/All subdir. You can install them with the following commands under sh(1):

     ```
     # PKG_PATH=ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/sparc64/5.0/All
     # export PKG_PATH
     # pkg_add -v tcsh
     # pkg_add -v bash
     # pkg_add -v perl
     # pkg_add -v apache
     # pkg_add -v kde
     # pkg_add -v firefox
     ...
     ```

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

     ```
     # setenv PKG_PATH ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/sparc64/5.0/A
     ```

     Note: Some mirror sites don't mirror the /pub/pkgsrc directory. If you would like to use such mirrors, you could also try the /pub/NetBSD/packages/current-packages/NetBSD/sparc64/5.0/All directory, which may have the same contents.

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

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

     ```
     Warning: package 'foo' was built for a different version
     of the OS:
     NetBSD/i386 M.N (pkg) vs. NetBSD/i386 5.0 (this host),
     ```

This warning is harmless if the formal major release numbers are the same between the pkg and your host. Please refer to the NetBSD release glossary and graphs at
http://www.NetBSD.org/releases/release-map.html
for more information about NetBSD's release numbering scheme.

- The framework for compiling packages can be obtained by retrieving the file
  ftp://ftp.NetBSD.org/pub/NetBSD/packages/pkgsrc.tar.gz
  It is typically extracted into `/usr/pkgsrc` (though other locations work fine) with the commands:

```
# cd /usr
# tar -zxpf pkgsrc.tar.gz
```

  After extracting, see the `doc/pkgsrc.txt` file in the extraction directory (e.g., `/usr/pkgsrc/doc/pkgsrc.txt`) 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 pkgsrc 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 easiest way to upgrade to NetBSD 5.0 is with binaries, and that is the method documented here.

To do the upgrade, you must boot the install kernel using one of the methods described above. You must also have at least the **base** and **kern** binary distribution sets available. 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, boot blocks, 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 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. Also, file systems are checked before unpacking the sets. Fetching the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for help.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 5.0 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 5.0.

The pthread libraries from previous versions of NetBSD require that the `sysctl`(3) node `kern.no_sa_support` be set to `0`. This affects the following environments:

- Running a 5.0 kernel with an older userland.
- Running an older userland inside a chroot'ed environment on a 5.0 system.
- Running older statically linked pthread applications.

The 5.0 kernel defaults to `0` for `kern.no_sa_support`, which covers the first case. However, please note that a full installation of 5.0 (either from scratch or through an upgrade) will set `kern.no_sa_support` to 1 during the boot process. This means that for the last two cases, you will have to manually set `kern.no_sa_support` to `0`, using either the `sysctl`(8) command or through `sysctl.conf`(5).

Note that **`sysinst`** 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 3.x releases

See the section below on upgrading from NetBSD 4.x as well.

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 were removed in 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, please see the "Components removed from NetBSD" at
  http://www.NetBSD.org/releases/formal-4/NetBSD-4.0.html#removals

- 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 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 affecting an upgrade from NetBSD 4.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 '_httpd' and '_timedc' and the groups '_httpd' and '_timedc' need to be created.
- Unprivileged use of the `mount`(8) command now requires the *`nosuid`* and *`nodev`* options to be explicitly specified. Previously, these options were automatically enforced even if they were not explicitly specified.
- A number of things have been removed from the NetBSD 5.0 release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Using online NetBSD documentation

Documentation is available if you installed 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. See **http://www.NetBSD.org/mailinglists/** for a web interface.

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.

Bugs also can be submitted and queried with the web interface at
http://www.NetBSD.org/support/send-pr.html

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **http://www.NetBSD.org/mailinglists/**

If you're interested in doing a serious amount of work on a specific port, you probably should contact the 'owner' of that port (listed below).

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

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

  for their 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.
- The many organizations that provide NetBSD mirror sites.
- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.
- We list the individuals and organizations that have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it at **http://www.NetBSD.org/donations/** (If you're not on that list and should be, tell us! We probably were not able to get in touch with you, to verify that you wanted to be listed.)
- Finally, we thank all of the people who've put sweat and tears into developing NetBSD since its inception in January, 1993. (Obviously, there are a lot more people who deserve thanks here. If you're one of them, and would like to be mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

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

The portmasters (and their ports):

Erik Berls	cyber@NetBSD.org	**cobalt**
Manuel Bouyer	bouyer@NetBSD.org	**xen**
Simon Burge	simonb@NetBSD.org	**evbmips**
Simon Burge	simonb@NetBSD.org	**pmax**
Simon Burge	simonb@NetBSD.org	**sbmips**
Julian Coleman	jdc@NetBSD.org	**atari**
Marcus Comstedt	marcus@NetBSD.org	**dreamcast**
Andrew Doran	ad@NetBSD.org	**amd64**
Andrew Doran	ad@NetBSD.org	**i386**
Matthias Drochner	drochner@NetBSD.org	**cesfic**
Gavan Fantom	gavan@NetBSD.org	**iyonix**
Jaime A Fournier	ober@NetBSD.org	**zaurus**
Matt Fredette	fredette@NetBSD.org	**sun2**
Ichiro FUKUHARA	ichiro@NetBSD.org	**hpcarm**
Chris Gilbert	chris@NetBSD.org	**cats**
Ben Harris	bjh21@NetBSD.org	**acorn26**
Ross Harvey	ross@NetBSD.org	**alpha**
Nick Hudson	skrll@NetBSD.org	**hp700**
Martin Husemann	martin@NetBSD.org	**sparc64**
IWAMOTO Toshihiro	toshii@NetBSD.org	**hpcarm**
Darrin Jewell	dbj@NetBSD.org	**next68k**
Søren Jørvang	soren@NetBSD.org	**sgimips**
Wayne Knowles	wdk@NetBSD.org	**mipsco**
Takayoshi Kochi	kochi@NetBSD.org	**ia64**
Paul Kranenburg	pk@NetBSD.org	**sparc**
Michael Lorenz	macallan@NetBSD.org	**macppc**
Anders Magnusson	ragge@NetBSD.org	**vax**
Cherry G. Mathew	cherry@NetBSD.org	**ia64**
NISHIMURA Takeshi	nsmrtks@NetBSD.org	**x68k**
Tohru Nishimura	nisimura@NetBSD.org	**luna68k**
Tohru Nishimura	nisimura@NetBSD.org	**sandpoint**
Andrey Petrov	petrov@NetBSD.org	**sparc64**
Scott Reynolds	scottr@NetBSD.org	**mac68k**
Tim Rightnour	garbled@NetBSD.org	**ofppc**
Tim Rightnour	garbled@NetBSD.org	**prep**
Tim Rightnour	garbled@NetBSD.org	**rs6000**
Noriyuki Soda	soda@NetBSD.org	**arc**
Ignatios Souvatzis	is@NetBSD.org	**amiga**
Jonathan Stone	jonathan@NetBSD.org	**pmax**
Shin Takemura	takemura@NetBSD.org	**hpcmips**
Matt Thomas	matt@NetBSD.org	**alpha**
Matt Thomas	matt@NetBSD.org	**netwinder**
Jason Thorpe	thorpej@NetBSD.org	**algor**
Jason Thorpe	thorpej@NetBSD.org	**evbarm**
Jason Thorpe	thorpej@NetBSD.org	**shark**
Izumi Tsutsui	tsutsui@NetBSD.org	**ews4800mips**
Izumi Tsutsui	tsutsui@NetBSD.org	**hp300**
Izumi Tsutsui	tsutsui@NetBSD.org	**news68k**

Valeriy E. Ushakov	uwe@NetBSD.org	**landisk**
Nathan Williams	nathanw@NetBSD.org	**sun3**
Steve Woodford	scw@NetBSD.org	**evbppc**
Steve Woodford	scw@NetBSD.org	**mvme68k**
Steve Woodford	scw@NetBSD.org	**mvmeppc**
Reinoud Zandijk	reinoud@NetBSD.org	**acorn32**

The NetBSD 5.0 Release Engineering team:

Manuel Bouyer	bouyer@NetBSD.org
David Brownlee	abs@NetBSD.org
James Chacon	jmc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Håvard Eidnes	he@NetBSD.org
Liam J. Foy	liamjfoy@NetBSD.org
John Heasley	heas@NetBSD.org
Geert Hendrickx	ghen@NetBSD.org
Soren Jacobsen	snj@NetBSD.org
Phil Nelson	phil@NetBSD.org
Jeff Rizzo	riz@NetBSD.org

NetBSD Developers:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@NetBSD.org
Jukka Andberg	jandberg@NetBSD.org
Julian Assange	proff@NetBSD.org
Lennart Augustsson	augustss@NetBSD.org
Christoph Badura	bad@NetBSD.org
Bang Jun-Young	junyoung@NetBSD.org
Dieter Baron	dillo@NetBSD.org
Robert V. Baron	rvb@NetBSD.org
Alan Barrett	apb@NetBSD.org
Grant Beattie	grant@NetBSD.org
Jason Beegan	jtb@NetBSD.org
Erik Berls	cyber@NetBSD.org
Hiroyuki Bessho	bsh@NetBSD.org
John Birrell	jb@NetBSD.org
Mason Loring Bliss	mason@NetBSD.org
Charles Blundell	cb@NetBSD.org
Rafal Boni	rafal@NetBSD.org
Stephen Borrill	sborrill@NetBSD.org
Sean Boudreau	seanb@NetBSD.org
Manuel Bouyer	bouyer@NetBSD.org
John Brezak	brezak@NetBSD.org
Allen Briggs	briggs@NetBSD.org
Mark Brinicombe	mark@NetBSD.org
Aaron Brown	abrown@NetBSD.org
Andrew Brown	atatat@NetBSD.org

David Brownlee	abs@NetBSD.org
Frederick Bruckman	fredb@NetBSD.org
Jon Buller	jonb@NetBSD.org
Simon Burge	simonb@NetBSD.org
Robert Byrnes	byrnes@NetBSD.org
Pavel Cahyna	pavel@NetBSD.org
D'Arcy J.M. Cain	darcy@NetBSD.org
Daniel Carosone	dan@NetBSD.org
Dave Carrel	carrel@NetBSD.org
James Chacon	jmc@NetBSD.org
Mihai Chelaru	kefren@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Ben Collver	ben@NetBSD.org
Marcus Comstedt	marcus@NetBSD.org
Jeremy Cooper	jeremy@NetBSD.org
Chuck Cranor	chuck@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Aidan Cully	aidan@NetBSD.org
Garrett D'Amore	gdamore@NetBSD.org
Johan Danielsson	joda@NetBSD.org
John Darrow	jdarrow@NetBSD.org
Jed Davis	jld@NetBSD.org
Matt DeBergalis	deberg@NetBSD.org
Arnaud Degroote	degroote@NetBSD.org
Rob Deker	deker@NetBSD.org
Chris G. Demetriou	cgd@NetBSD.org
Tracy Di Marco White	gendalia@NetBSD.org
Jaromír Dolecek	jdolecek@NetBSD.org
Andy Doran	ad@NetBSD.org
Roland Dowdeswell	elric@NetBSD.org
Emmanuel Dreyfus	manu@NetBSD.org
Matthias Drochner	drochner@NetBSD.org
Jun Ebihara	jun@NetBSD.org
Håvard Eidnes	he@NetBSD.org
Jaime A Fournier	ober@NetBSD.org
Stoned Elipot	seb@NetBSD.org
Michael van Elst	mlelstv@NetBSD.org
Enami Tsugutomo	enami@NetBSD.org
Bernd Ernesti	veego@NetBSD.org
Erik Fair	fair@NetBSD.org
Gavan Fantom	gavan@NetBSD.org
Hauke Fath	hauke@NetBSD.org
Hubert Feyrer	hubertf@NetBSD.org
Jason R. Fink	jrf@NetBSD.org
Matt J. Fleming	mjf@NetBSD.org
Marty Fouts	marty@NetBSD.org
Liam J. Foy	liamjfoy@NetBSD.org
Matt Fredette	fredette@NetBSD.org

Thorsten Frueauf	frueauf@NetBSD.org
Castor Fu	castor@NetBSD.org
Ichiro Fukuhara	ichiro@NetBSD.org
Quentin Garnier	cube@NetBSD.org
Thomas Gerner	thomas@NetBSD.org
Simon J. Gerraty	sjg@NetBSD.org
Justin Gibbs	gibbs@NetBSD.org
Chris Gilbert	chris@NetBSD.org
Eric Gillespie	epg@NetBSD.org
Brian Ginsbach	ginsbach@NetBSD.org
Paul Goyette	pgoyette@NetBSD.org
Michael Graff	explorer@NetBSD.org
Brian C. Grayson	bgrayson@NetBSD.org
Matthew Green	mrg@NetBSD.org
Andreas Gustafsson	gson@NetBSD.org
Ulrich Habel	rhaen@NetBSD.org
Jun-ichiro itojun Hagino	itojun@NetBSD.org
HAMAJIMA Katsuomi	hamajima@NetBSD.org
Adam Hamsik	haad@NetBSD.org
Juergen Hannken-Illjes	hannken@NetBSD.org
Charles M. Hannum	mycroft@NetBSD.org
Ben Harris	bjh21@NetBSD.org
Ross Harvey	ross@NetBSD.org
Eric Haszlakiewicz	erh@NetBSD.org
John Hawkinson	jhawk@NetBSD.org
HAYAKAWA Koichi	haya@NetBSD.org
John Heasley	heas@NetBSD.org
Geert Hendrickx	ghen@NetBSD.org
René Hexel	rh@NetBSD.org
Iain Hibbert	plunky@NetBSD.org
Kouichirou Hiratsuka	hira@NetBSD.org
Michael L. Hitch	mhitch@NetBSD.org
Ádám Hóka	ahoka@NetBSD.org
Jachym Holecek	freza@NetBSD.org
David A. Holland	dholland@NetBSD.org
Christian E. Hopps	chopps@NetBSD.org
Ken Hornstein	kenh@NetBSD.org
Marc Horowitz	marc@NetBSD.org
Eduardo Horvath	eeh@NetBSD.org
Nick Hudson	skrll@NetBSD.org
Shell Hung	shell@NetBSD.org
Martin Husemann	martin@NetBSD.org
Dean Huxley	dean@NetBSD.org
Love Hörnquist Åstrand	lha@NetBSD.org
Roland Illig	rillig@NetBSD.org
Bernardo Innocenti	bernie@NetBSD.org
Tetsuya Isaki	isaki@NetBSD.org
ITOH Yasufumi	itohy@NetBSD.org
IWAMOTO Toshihiro	toshii@NetBSD.org

Matthew Jacob	mjacob@NetBSD.org
Soren Jacobsen	snj@NetBSD.org
Lonhyn T. Jasinskyj	lonhyn@NetBSD.org
Darrin Jewell	dbj@NetBSD.org
Nicolas Joly	njoly@NetBSD.org
Chris Jones	cjones@NetBSD.org
Søren Jørvang	soren@NetBSD.org
Takahiro Kambe	taca@NetBSD.org
Masanori Kanaoka	kanaoka@NetBSD.org
Antti Kantee	pooka@NetBSD.org
Frank Kardel	kardel@NetBSD.org
Mattias Karlsson	keihan@NetBSD.org
KAWAMOTO Yosihisa	kawamoto@NetBSD.org
Mario Kemper	magick@NetBSD.org
Min Sik Kim	minskim@NetBSD.org
Thomas Klausner	wiz@NetBSD.org
Klaus Klein	kleink@NetBSD.org
John Klos	jklos@NetBSD.org
Wayne Knowles	wdk@NetBSD.org
Takayoshi Kochi	kochi@NetBSD.org
John Kohl	jtk@NetBSD.org
Daniel de Kok	daniel@NetBSD.org
Jonathan A. Kollasch	jakllsch@NetBSD.org
Paul Kranenburg	pk@NetBSD.org
Lubomir Kundrak	lkundrak@NetBSD.org
Jochen Kunz	jkunz@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Kentaro A. Kurahone	kurahone@NetBSD.org
Arnaud Lacombe	alc@NetBSD.org
Kevin Lahey	kml@NetBSD.org
David Laight	dsl@NetBSD.org
Johnny C. Lam	jlam@NetBSD.org
Martin J. Laubach	mjl@NetBSD.org
Greg Lehey	grog@NetBSD.org
Ted Lemon	mellon@NetBSD.org
Christian Limpach	cl@NetBSD.org
Frank van der Linden	fvdl@NetBSD.org
Joel Lindholm	joel@NetBSD.org
Tonnerre Lombard	tonnerre@NetBSD.org
Mike Long	mikel@NetBSD.org
Michael Lorenz	macallan@NetBSD.org
Warner Losh	imp@NetBSD.org
Tomasz Luchowski	zuntum@NetBSD.org
Federico Lupi	federico@NetBSD.org
Brett Lymn	blymn@NetBSD.org
Paul Mackerras	paulus@NetBSD.org
MAEKAWA Masahide	gehenna@NetBSD.org
Anders Magnusson	ragge@NetBSD.org
Cherry G. Mathew	cherry@NetBSD.org

David Maxwell	david@NetBSD.org
Gregory McGarry	gmcgarry@NetBSD.org
Dan McMahill	dmcmahill@NetBSD.org
Jared D. McNeill	jmcneill@NetBSD.org
Neil J. McRae	neil@NetBSD.org
Julio M. Merino Vidal	jmmv@NetBSD.org
Perry Metzger	perry@NetBSD.org
Luke Mewburn	lukem@NetBSD.org
Jean-Yves Migeon	jym@NetBSD.org
Brook Milligan	brook@NetBSD.org
Minoura Makoto	minoura@NetBSD.org
Simas Mockevicius	symka@NetBSD.org
der Mouse	mouse@NetBSD.org
Joseph Myers	jsm@NetBSD.org
Ken Nakata	kenn@NetBSD.org
Takeshi Nakayama	nakayama@NetBSD.org
Phil Nelson	phil@NetBSD.org
John Nemeth	jnemeth@NetBSD.org
Bob Nestor	rnestor@NetBSD.org
NISHIMURA Takeshi	nsmrtks@NetBSD.org
Tohru Nishimura	nisimura@NetBSD.org
NONAKA Kimihiro	nonaka@NetBSD.org
Takehiko NOZAKI	tnozaki@NetBSD.org
Tobias Nygren	tnn@NetBSD.org
OBATA Akio	obache@NetBSD.org
Jesse Off	joff@NetBSD.org
Tatoku Ogaito	tacha@NetBSD.org
OKANO Takayoshi	kano@NetBSD.org
Masaru Oki	oki@NetBSD.org
Atsushi Onoe	onoe@NetBSD.org
Greg Oster	oster@NetBSD.org
Rui Paulo	rpaulo@NetBSD.org
Jonathan Perkin	sketch@NetBSD.org
Andrey Petrov	petrov@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Peter Postma	peter@NetBSD.org
Dante Profeta	dante@NetBSD.org
Chris Provenzano	proven@NetBSD.org
Niels Provos	provos@NetBSD.org
Mindaugas Rasiukevicius	rmind@NetBSD.org
Michael Rauch	mrauch@NetBSD.org
Marc Recht	recht@NetBSD.org
Darren Reed	darrenr@NetBSD.org
Jeremy C. Reed	reed@NetBSD.org
Antoine Reilles	tonio@NetBSD.org
Tyler R. Retzlaff	rtr@NetBSD.org

Scott Reynolds	scottr@NetBSD.org
Michael Richardson	mcr@NetBSD.org
Tim Rightnour	garbled@NetBSD.org
Alan Ritter	rittera@NetBSD.org
Jeff Rizzo	riz@NetBSD.org
Hans Rosenfeld	hans@NetBSD.org
Gordon Ross	gwr@NetBSD.org
Steve Rumble	rumble@NetBSD.org
Ilpo Ruotsalainen	lonewolf@NetBSD.org
Heiko W. Rupp	hwr@NetBSD.org
Blair J. Sadewitz	bjs@NetBSD.org
David Sainty	dsainty@NetBSD.org
SAITOH Masanobu	msaitoh@NetBSD.org
Kazuki Sakamoto	sakamoto@NetBSD.org
Curt Sampson	cjs@NetBSD.org
Wilfredo Sanchez	wsanchez@NetBSD.org
Ty Sarna	tsarna@NetBSD.org
SATO Kazumi	sato@NetBSD.org
Jan Schaumann	jschauma@NetBSD.org
Matthias Scheler	tron@NetBSD.org
Silke Scheler	silke@NetBSD.org
Karl Schilke (rAT)	rat@NetBSD.org
Amitai Schlair	schmonz@NetBSD.org
Konrad Schroder	perseant@NetBSD.org
Georg Schwarz	schwarz@NetBSD.org
Lubomir Sedlacik	salo@NetBSD.org
Christopher SEKIYA	sekiya@NetBSD.org
Reed Shadgett	dent@NetBSD.org
John Shannon	shannonjr@NetBSD.org
Tim Shepard	shep@NetBSD.org
Takeshi Shibagaki	shiba@NetBSD.org
Naoto Shimazaki	igy@NetBSD.org
Takao Shinohara	shin@NetBSD.org
Takuya SHIOZAKI	tshiozak@NetBSD.org
Daniel Sieger	dsieger@NetBSD.org
Chuck Silvers	chs@NetBSD.org
Thor Lancelot Simon	tls@NetBSD.org
Jeff Smith	jeffs@NetBSD.org
Noriyuki Soda	soda@NetBSD.org
Wolfgang Solfrank	ws@NetBSD.org
SOMEYA Yoshihiko	someya@NetBSD.org
Bill Sommerfeld	sommerfeld@NetBSD.org
Jörg Sonnenberger	joerg@NetBSD.org
Ignatios Souvatzis	is@NetBSD.org
T K Spindler	dogcow@NetBSD.org
Bill Squier	groo@NetBSD.org
Jonathan Stone	jonathan@NetBSD.org
Bill Studenmund	wrstuden@NetBSD.org
Kevin Sullivan	sullivan@NetBSD.org

SUNAGAWA Keiki	kei@NetBSD.org
Kimmo Suominen	kim@NetBSD.org
Robert Swindells	rjs@NetBSD.org
Shin Takemura	takemura@NetBSD.org
TAMURA Kent	kent@NetBSD.org
Shin'ichiro TAYA	taya@NetBSD.org
Ian Lance Taylor	ian@NetBSD.org
Matt Thomas	matt@NetBSD.org
Jason Thorpe	thorpej@NetBSD.org
Christoph Toshok	toshok@NetBSD.org
Greg Troxel	gdt@NetBSD.org
Tsubai Masanari	tsubai@NetBSD.org
Izumi Tsutsui	tsutsui@NetBSD.org
UCHIYAMA Yasushi	uch@NetBSD.org
Masao Uebayashi	uebayasi@NetBSD.org
Shuichiro URATA	ur@NetBSD.org
Valeriy E. Ushakov	uwe@NetBSD.org
Todd Vierling	tv@NetBSD.org
Aymeric Vincent	aymeric@NetBSD.org
Paul Vixie	vixie@NetBSD.org
Mike M. Volokhov	mishka@NetBSD.org
Krister Walfridsson	kristerw@NetBSD.org
Lex Wennmacher	wennmach@NetBSD.org
Leo Weppelman	leo@NetBSD.org
Assar Westerlund	assar@NetBSD.org
Todd Whitesel	toddpw@NetBSD.org
Frank Wille	phx@NetBSD.org
Nathan Williams	nathanw@NetBSD.org
Rob Windsor	windsor@NetBSD.org
Dan Winship	danw@NetBSD.org
Jim Wise	jwise@NetBSD.org
Michael Wolfson	mbw@NetBSD.org
Colin Wood	ender@NetBSD.org
Steve Woodford	scw@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Yuji Yamano	yyamano@NetBSD.org
David Young	dyoung@NetBSD.org
Reinoud Zandijk	reinoud@NetBSD.org
S.P.Zeidler	spz@NetBSD.org
Maria Zevenhoven	maria7@NetBSD.org
Christos Zoulas	christos@NetBSD.org

Other contributors:

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

Legal Mumbo-Jumbo

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

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

NetBSD is a registered trademark of The NetBSD Foundation, Inc.
This product includes software developed by the University of California, Berkeley and its contributors.
This product includes software developed by the NetBSD Foundation.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See http://www.netbsd.org/ for information about NetBSD.
This product contains software developed by Ignatios Souvatzis for the NetBSD project.
This product contains software written by Ignatios Souvatzis and Michael L. Hitch for the NetBSD project.
This product includes software developed by Intel Corporation and its contributors.
This product contains software written by Michael L. Hitch for the NetBSD project.
This product includes software developed by Microsoft.
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 Adrian Portelli for the NetBSD project.
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 Bruce M. Simpson.
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 Daan Vreeken.
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 K. Kobayashi.
This product includes software developed by K. Kobayashi and H. Shimokawa.
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 Florian Stoehr.
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 Garrett D'Amore.
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 Henrik Vestergaard Draboel.
This product includes software developed by Herb Peyerl.
This product includes software developed by Hidetoshi Shimokawa.
This product includes software developed by Hubert Feyrer for the NetBSD Project.
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 Jeremy C. Reed for the NetBSD project.
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 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 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 Tinguely and Jim Lowe
This product includes software developed by Markus Wild.
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 Pritchard.
This product includes software developed by Mike Pritchard and contributors.
This product includes software developed by Minoura Makoto.
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 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 Steven M. Bellovin.
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 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 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 Yen Yen Lim and North Dakota State University.
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 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 Urbana-Champaign Independent Media Center.
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 by Kyma Systems.
This product includes software developed for the NetBSD Project by Kyma Systems LLC.
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