NAME

INSTALL — Installation procedure for NetBSD/sparc.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 5.1 on the *sparc* 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 5.1 on a machine of the sparc architecture.

- Fetch files needed to install NetBSD. Fetch a CD image, `sparccd-5.1.iso` or the floppy disk images, `sparc/install/floppy/disk1.gz` and `sparc/install/floppy/disk2`. You need either the pair of floppies or the CD to boot your system. Alternatively, you may netboot the installation kernel. This process is covered below, in detail.
- The actual binary distribution is in the `sparc/binary/sets/` directory. When you boot the install floppies or CD-ROM, the installation program can fetch these files for you (using e.g. ftp) if you have a network connection. There are several other methods to get the binary sets onto your machine.

  You will at a minimum need the following sets: `kern-GENERIC.tgz`, `base.tgz` and `etc.tgz`. In a typical workstation installation you will probably want all the installation sets.
- Make sure your sparc's CD-ROM drive is bootable. Burn the CD. Otherwise, write the floppy images directly to a pair of floppies (after uncompressing disk1.gz).

  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 OpenBoot PROM "**`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 one of the following commands (depending on your model): "**`b sd(,30,)`**", "**`boot sd(,30,)`**", or "**`boot cdrom`**".

  The command to boot from floppy is either "**`boot fd(,,1)`**" or "**`boot floppy`**". The installer will prompt you to insert the second floppy when it is ready for it.

- 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 5.1 release contains complete binary releases for many different system architectures. (A few ports are not fully supported at this time and are thus not part of the binary distribution. Please see the NetBSD web site at **http://www.NetBSD.org/** for information on them.)

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

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

Dedication

NetBSD 5.1 is dedicated to the memory of Martti Kuparinen, who was the victim of a traffic accident in June 2010.

Martti's technical contributions are too many to list here in full. He created and maintained numerous packages in pkgsrc, updated two packet filter solutions distributed with NetBSD and improved several hardware drivers. Beyond that he was always helpful and friendly. His example encouraged users to contribute to the project and share their work with the community. Some of these users later became NetBSD developers themselves thanks to Martti's efforts.

Changes Between the NetBSD 5.0 and 5.1 Releases

The NetBSD 5.1 release is the first feature update of the NetBSD 5.0 release branch. It represents a selected subset of fixes deemed critical for security or stability reasons, as well as new features and enhancements.

Please note that all fixes in security/critical updates (i.e., NetBSD 5.0.1, 5.0.2, etc.) are cumulative, so the latest update contains all such fixes since the corresponding minor release. These fixes also appear in minor releases (i.e., NetBSD 5.1, 5.2, etc.).

The complete list of changes can be found in the CHANGES-5.1:

http://ftp.NetBSD.org/pub/NetBSD/NetBSD-5.1/CHANGES-5.1

file in the top level directory of the NetBSD 5.1 release tree. An abbreviated list is as follows:

Security Advisory Fixes

- NetBSD-SA2009-004 (NetBSD OpenPAM `passwd`(1) changing weakness):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-004.txt.asc
- NetBSD-SA2009-005 (Plaintext Recovery Attack Against SSH):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-005.txt.asc
- NetBSD-SA2009-006 (Buffer overflows in ntp):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-006.txt.asc
- NetBSD-SA2009-007 (Buffer overflows in `hack`(6)):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-007.txt.asc
- NetBSD-SA2009-008 (OpenSSL ASN1 parsing denial of service and CMS signature verification weakness):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-008.txt.asc

- NetBSD-SA2009-009 (OpenSSL DTLS Memory Exhaustion and DSA signature verification vulnerabilities):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-009.txt.asc
- NetBSD-SA2009-010 (ISC dhclient subnet-mask flag stack overflow):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-010.txt.asc
- NetBSD-SA2009-011 (ISC DHCP server Denial of Service vulnerability):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-011.txt.asc
- NetBSD-SA2009-012 (SHA2 implementation potential buffer overflow):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-012.txt.asc
- NetBSD-SA2009-013 (BIND named dynamic update Denial of Service vulnerability):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2009-013.txt.asc
- NetBSD-SA2010-002 (OpenSSL TLS renegotiation man in the middle vulnerability):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-002.txt.asc
- NetBSD-SA2010-003 (azalia(4)/hdaudio(4) negative mixer index panic):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-003.txt.asc
- NetBSD-SA2010-004 (amd64 per-page No-execute (NX) bit disabled):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-004.txt.asc
- NetBSD-SA2010-005 (NTP server Denial of Service vulnerability):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-005.txt.asc
- NetBSD-SA2010-006 (Buffer length checking errors in CODA):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-006.txt.asc
- NetBSD-SA2010-007 (Integer overflow in libbz2 decompression code):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-007.txt.asc
- NetBSD-SA2010-008 (sftp(1)/ftp(1)/glob(3) related resource exhaustion):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-008.txt.asc
- NetBSD-SA2010-010 (Buffer Length Handling Errors in netsmb):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-010.txt.asc
- NetBSD-SA2010-011 (OpenSSL Double Free Arbitrary Code Execution):
  http://ftp.NetBSD.org/pub/NetBSD/security/advisories/NetBSD-SA2010-011.txt.asc

Advisories prior to NetBSD-SA2009-004 do not affect NetBSD 5.0: *.:*
http://www.NetBSD.org/support/security/patches-5.0.html

Other Security Fixes

- openssl: Fix CVE-2009-4355 and CVE-2010-0740.
- Fix crash in openssl: handshake_dgst[] may be used without being allocated, causing NULL pointer dereference.
- Update BIND server and tools to 9.5.2-P2, fixing CVE-2009-0025, CVE-2009-4022, and CVE-2010-0097.
- `ntpd`(8): Fix CVE-2009-3563.
- expat: Fix SA36425 and CVE-2009-3560.

- fts(3): Avoid possible integer overflow on really deep dirs, and subsequent collateral damage. Received from OpenBSD via US-CERT as VU #590371.
- Fix a couple issues with POSIX message queues:
  - An invalid signal number passed to mq_notify() could crash the kernel on delivery -- add a boundary check.
  - A user could set mq_maxmsg (the maximal number of messages in a queue) to a huge value on mq_open(O_CREAT) and later use up all kernel memory by mq_send() -- add a sysctl'able limit which defaults to 16*mq_def_maxmsg.
- arc4random(3): Keep arc4_i and arc4_j synchronised after a rekeying. This prevents accidentally ending up in a short ARC4 cycle.
- freetype: Fix CVE-2009-0946.
- ftpd(8): Fix a remote crash. PR 43023.
- openldap: Fix CVE-2009-3767.
- Fix an NX regression observed on amd64 kernels, where per-page execution right was disabled (therefore leading to the inability of the kernel to detect fraudulent use of memory mappings marked as not being executable).

Kernel

- Fix random "filesystem full" messages on large FFS file systems.
- Fix a regression in the 4.4BSD scheduler, improving interactive performance under load.
- Remove a race where physio_done() may use memory already freed. PR 39536.
- Fix a crash observed when trying to load a corrupted ELF kernel module.
- Fix PR 41566, where writes on the controlling tty were not being awoken from blocks.
- Various fixes for POSIX message queues.
- Fix a couple deadlocks in the VFS subsystem. PR 41374.
- Fix an issue on some architectures where a softint could fire on the wrong CPU.
- Fixes for POSIX advisory locks.
- A number of other stability fixes.
- Fix a case where setpriority(2) returned EACCES instead of EPERM. PR 41489.
- lockf(3) passes its arguments through to fcntl(2) but is supposed to support -ve lengths (lock area before current offset). Nothing in libc or the kernel allowed for this, so some random part of the file would get locked. PR 41620.
- Fix ktrace of data from iovec based system calls. PR 41819.
- Fix stack size enforcement.

File Systems

- fsck_ext2fs(8): Ignore the "-P" option as intended, to make this work with, e.g., "fsck_flags=-pP" in /etc/rc.conf. PR 41490.
- UFS quotas: Add missing mutex_destroy() before pool_cache_put(). Prevents a "Mutex error: lockdebug_alloc: already initialized" panic.

- tunefs(8): Allow tunefs to clear any type of WAPBL log, not only in-filesystem ones.
- fsck_ffs(8): Do some basic checks of the WAPBL journal, and fail if the kernel would refuse to mount the filesystem read/write. Add code to clear the bogus journal when not run in preen (-p) mode.
- wapbl(4): If the WAPBL journal can't be read (ffs_wapbl_replay_start() fails), mount the filesystem anyway if MNT_FORCE is present. This allows to still boot single-user a system with a corrupted WAPBL on /, and so get a chance to run fsck to fix it.
- smbfs:
  - Fix some panics while trying to umount a smbfs share.
  - Fix detection of SMB capabilities according to the CIFS spec:
    - SMB_CAP_LARGE_FILES advertises support for 64-bit file offsets.
    - SMB_CAP_LARGE_READX and SMB_CAP_LARGE_WRITEX advertise support for large reads and writes (larger than 64KB). PR 42175.
  - Add support for file sizes greater than 4GB.
  - Prevent malicious local program from causing a kernel crash.
- A number of fixes for ext2fs. PR 28712.
- procfs: Fix memory leak. PR 42053.
- Fix cp(1) from NTFS. PR 38531.
- cd9660: Change cd9660_mount, in MNT_UPDATE case, to check dev_t's for equality instead of just vnode pointers. Fixes erroneous "Invalid argument" errors from mount(8) with -u against cd9660 root in the presence of mfs or tmpfs /dev prepared after initial mountroot.
- A number of fixes to ffs snapshots.
- sysctlfs: Fix a crash while trying to read nodes on amd64. PR 41494.
- Various improvements to UDF.
- xattr: Fix system crash which could be triggered by a malicious command.
- coda: Fix incomplete ioctl parameter verification.
- ffs: Fix a race condition which could lead to data corruption.
- wapbl(4): Fix replay problems which could corrupt the fs. PR 43336.

Networking

- ath(4): Remove the binary HAL and update to the open source HAL.
- Add the age(4) driver for Attansic L1.
- Add the ale(4) driver for Atheros AR8121/AR8113/AR8114.
- brgphy(4): Add support for BCM5462, BCM54K2 and BCM5722.
- arp(4): Don't require the gateway address to have room for both an interface name and address. Fixes a regression in 'arp -s ...' on interfaces such as xennet0 with unusually long names. PR 41878.
- Make tcp msl (max segment life) tunable via sysctl net.inet.tcp.msl.
- Fixed a number of bge(4) bugs.

- `ifpci`(4): Fix endianness issues when accessing the B-channel fifos.
- `wm`(4):
  - Numerous fixes for various chips.
  - Add SIOCSIFADDR support for setting the AF_LINK address, necessary for agr to be able to set the mac addresses of each port to the agr address (i.e., so it can receive all intended traffic at the hardware level).
  - Enable hardware VLAN support.
- `re`(4):
  - Detect RTL8169CP, RTL8168D/8111D, and RTL8103E variants.
  - Add hardware checksum support for newer 8168/8111/8102 chips.
  - Fix RX hardware checksum for DESCV2 chips. PR 40605.
- `le`(4): Don't immediately switch UTP/AUI ports on lost carrier.
- `agr`(4): Add vlan support and hardware offload capabilities. Add support for disabling the LACP protocol by setting LINK1 on the agr interface.
- `bnx`(4):
  - Protect against spurious "bnx0: Double mbuf allocation failure!" panics.
  - Add support for BCM5709 and BCM5716.
- Follow exactly the recommendation of draft-ietf-tcpm-tcpsecure-11.txt: Don't check against the last ack received, but the expected sequence number. This makes RST handling independent of delayed ACK.
- `vr`(4): Add suspend/resume support.
- `carp`(4): Improve logging. PR 38260.
- Give 100BASE-TX full duplex higher priority than 100BASE-T4.
- Fix wpa with `ral`(4).
- `vlan`(4): Inherit the parent's TCP segmentation offload capability.
- dhcpcd:
  - Update to 5.1.3.
  - Add an rc.d script.
  - Only start dhcpcd per interface if not running the full dhcpcd daemon. Only stop dhcpcd per interface if it's running for the interface. PR 40320.
- Fix a panic when trying to disable IPFilter before enabling it. PR 41364.
- `ping`(8): Deal with source route and record route specially giving a meaningful error message when remote side doesn't support record route. PR 41111.
- `ifconfig`(8): Don't require a "vlan" argument with "-vlanif". "ifconfig vlan0 -vlanif" now works as one would expect.
- `tap`(4): Fix a potential leak on device close. `bpf`(4): Prevent malicious bytecode from crashing the kernel with a divide-by-zero trap.

- UDPv6: Prevent local crash by malicious user program.

Miscellaneous Drivers

- ehci(4): Add a workaround for ATI SB600 and SB700 revisions A12 and A13 to avoid a USB subsystem hang when the system has multiple USB devices connected to it or one device is re-connected often.
- uftdi(4): Add support for multiple channel cards, specifically quad channel FT4232H.
- ums(4): Add quirks to make MS Wireless Laser Mouse 6000 work. PR 41737. Add support for USB HID devices that report absolute coordinates instead of relative movement data, e.g. touchpanels. Add support for the Microsoft Natural Ergonomic Desktop 7000 mouse.
- apm(4): Fix suspend/resume.

Audio

- Add hdaudio(4), a standards-compliant driver for Intel High Definition Audio. It will replace azalia(4) eventually.
- Add gcscaudio(4), an AMD Geode CS5536 audio driver.
- Add support for playback- or capture-only devices. PR 42050.
- pad(4): Catch up to audio(4) device_t/softc split to prevent pad from corrupting its child device's softc.
- audio(4): OSS audio allows mixer operations on the dsp device. NetBSD would previously return EINVAL in these circumstances. This can break audio in apps running under Linux emulation. Select whether to call mixer_ioctl() or audio_ioctl() based on whether the command smells like a mixer ioctl or not.

Security

- pam(3): Restore the good old UNIX behavior of root password changing: only root may change the root password.
- racoon(8): Fix a bug where racoon used a wrong IPsec-SA handle that was for another peer if it received an ISAKMP message for IPsec-SA that has the same message-id as the message-id that was received before.
- ipsec(4): Add a missing splx() call. PR 41701.
- opencrypto(9):
  - Extend the API to allow the destination buffer size to be specified when it is not the same size as the input buffer.
  - Add user-space access to compression features.
  - Add software gzip support (CRYPTO_GZIP_COMP).
  - Add the fast version of crc32 from zlib to libkern.
  - Fix PRs 41069 and 41070.

Storage

- Add support for RAIDframe parity maps. Drastically reduces the amount of time spent checking parity after an unclean shutdown by keeping better track of which regions might have had outstanding writes. Enabled by default; can be disabled on a per-set basis, or tuned, with the new raidctl(8) commands.
- Add sdmmc framework.

- Add `sdhc`(4), a driver for SD controllers following the SD Host Controller Standard Simplified Specification.
- Add `wb`(4), a driver for Winbond W83L518D SD/MMC readers.
- `siisata`(4): Sync with HEAD, fixing a number of bugs. PR 41579.
- `mfi`(4): Fix command list corruption seen on heavy I/O load. Add support for MFI gen2 devices.
- `twa`(4): Disable completely bogus DIAGNOSTIC check.
- `wdc`(4): use 8bits access to legacy IDE registers through the SATA interface (except data registers). Stops errors such as the following when probing SATA drives through controllers that offer the legacy pciide interface: viaide1 channel 0: reset failed for drive 0
- `piixide`(4): Add Intel 3400 support.

Platform specific

- x86 (amd64 and i386)
  - Add a workaround for a bug with some Opteron revisions where locked operations sometimes do not serve as memory barriers, allowing memory references to bleed outside of critical sections.
  - `ichlpcib`(4): Fix watchdog code:
    - The timer bound constants are in tick, so convert period to tick before checking it against the bounds.
    - For ICH5 or older, fix code that would have always written a 0 period to the register.
  - Add CPU topology detection support for AMD processors.
  - `asus`(4):
    - Add experimental cpu fan/voltage switching support (sysctl hw.asus0.∗).
    - Add fan sensor.
  - Add hw.wake.∗ sysctl subtree for toggling which devices are allowed to restore the system from sleep. By default, the following devices are enabled for wake: sleep/power buttons, lid switch, pc kbd controller.
  - `agp`(4): Add support for Intel G35, G45, and Q45.
  - Cut down on the number of lines used in acpi autoconf messages.
  - Extend CPU probe of Intel processors to handle extended-models: new Intel 45nm processors, Core i7, Atom, and the 45nm Xeon MP. Properly decode several new Intel cache descriptors, as listed in the most recent (March 2009) edition of Intel's Application Note 485. PRs 41289 and 41290.
  - `cpuctl`(8): Add newer VIA C7 core and VIA Nano. When printing an unknown VIA CPU, default to 'Unknown IDT/VIA' instead of 'C3'.
  - Fix NetBSD under qemu with ACPI enabled. PR 38729.
  - `acpi`(4): If the firmware describes duplicate keyboard controller nodes, don't panic when the driver fails to map registers. PR 39671.
  - Fix a bug where mapping the very end of iomem accidentally returns an address in the ISA hole. Fixes ohci on VirtualPC 7 for Mac, which places OHCI at base address 0xfffff000 size 0x1000.

- int 15h/AH=86h (WAIT) doesn't work properly on all hardware and emulators, so for the countdown use the more coarsely grained sleep implementation based on int 1ah/AH=00h (GET SYSTEM TIME).
- Use the TSC and current multiplier to calculate bus clock on VIA C7 Esther.
- Add support for VIA C7 temperature sensors (options VIA_C7TEMP) and enable in i386 GENERIC kernel.
- Fix a regression in the boot loader where pressing a letter not bound to a menu item would select a numeric item.
- Enable `cgd`(4) support on all amd64 and i386 XEN kernels.

• amd64
  - Handle protection faults properly, returning SIGSEGV instead of SIGBUS.
  - Ensure FP state is reset, if FP is used in a signal handler. PR 39299.
  - Build kernel modules with -mno-red-zone like the kernel is built.
  - On amd64, add a third free list distinct from the default free list, holding RAM between 16Mb and 4Gb. This helps preventing `bus_dma`(9) memory allocation failures for 32bit DMA on large-memory machines.

• i386
  -
  - The FPU Tag word is a 16bit register, in FPU (387) mode it defines 2-bit tags for each FPU data register, in MMX mode it defines 1-bit tags for each data register. The single bit tags are stored in the lower 8 bits and thus in the first byte of the save frame.
  - Fix a local user crash.

• ARM: Work-around a possible process exit corner case which can leave stale data in the cache after a context-switch. PR 41058.

• sgimips
  - haltwo(4): Set delta value for the mixer's master channel.
  - Add a driver for the Indy's front panel buttons. Power button presses are reported to sysmon, volume control buttons are reported to PMF.
  - Fixes for newport video.

• alpha
  - A number of multiprocessor fixes. PRs 41106 and 42174.
  - Add support for booting off a couple of common RAID adapters found on several models of alpha systems: mlx [Mylex DAC060] and iop [I2O]. PR 25829.
  - Change the kernel text to 0xfffffc0000430000 (which is where Tru64 has its kernel).
  - The tsc(4) bus initialization was using a single statically allocated extent storage for each tsp(4), which caused a LOCKDEBUG kernel to fail because the extent storage contained a mutex which panics when the second mutex_init() is attempted. Put the extent storage into the tsp_config structure so each tsp(4) gets its own. PR 38358.

- amiga
  - Switch amiga to the common m68k pmap.
  - Fix a problem with the timecounter running backwards everytime the hardware counter wraps and the clock-interrupt is not yet serviced by hardclock().
  - Improve precision of small kernel delays.
  - Fix keyboard handshaking problems on Amiga 1200.
- atari
  - Rewrite the `binpatch`(8) utility to add support for ELF binaries, old src/usr.sbin/mdsetimage sources which support misc executable formats without LGPL'ed libbfd.
  - Add a workaround for annoying "WARNING: negative runtime; monotonic clock has gone backwards" message.
  - `fd`(4): Correctly detect the default density.
- hp300: Make install.md probe `cd`(4) devices properly.
- hpcmips: Fix pcic kthread creation timing. PRs 41791 and 41164.
- m68k: Ensure functions like `mmap`(2), `mremap`(2), `shmat`(2) or `sbrk`(2) return -1 in case of an error. A side benefit of this is to fix a segfault caused by jemalloc when `mmap`(2) failed.
- macppc: pbms(4): Avoid an immediate crash during attach, and fix the aspect ratio of the trackpad on the geyser2 model.
- pmax: Make `ksyms`(4) actually work.
- sh3: Fix logic error in copyinstr() when deciding whether to return EFAULT or ENAMETOOLONG.
- sparc64:
  - Fix long double support in 32bit libc. PR 41406.
  - When preparing the initial trap frame for a new forked lwp, explicitly clear condition code. Otherwise we might catch a signal before we ever return to userland. PR 41302.
  - Fix a disk I/O regression under heavy load.
  - Build the sunleo X driver.
  - Add `lom`(4), a driver for LOMlite lights out management hardware monitor and watchdog timer.
  - sab(4): Allow the RSC to be the console on an E250, by checking for RSC-specific properties, and by not changing the port baud rate.
- vax
  - binutils: Allocate relocation section using bfd_zalloc() to ensure no garbage relocations when not all the entries are used. PR 39182.
  - Keep track of the previous ICR value and hardclock_ticks to ensure the 32 bit counter doesn't go backwards. Also, the ICR runs from -10000 to -1, so adjust the value when reading it. Now mfpr works quite nicely on a 4000/90.
- sparc
  - Enable `ddb`(4) in GENERIC-like kernels.

- Fix a number of issues with floppies.
- Add apc(4), a driver for the Aurora Personality Chip (APC) found on SPARCstation-4/5, and emulated by qemu to idle the simulator when the CPU is idle. Only the CPU idle part implemented at this time.

- xen
  - Now a XEN3_DOM0 kernel properly updates the CMOS time.
  - Implement DIOCGDISKINFO for xbd disk driver.
  - xbdback: implement and publish "feature-flush-cache". xbd: if feature-flush-cache is present, use it for DIOCCACHESYNC. Should improve WAPBL reliability of Xen guests on a NetBSD dom0.
  - xennet: Write a "feature-rx-notify" to the xenstore, which is used by recent linux dom0 kernels. This reduces packet loss when using a NetBSD domU on such linux dom0.
  - xennetback: Announce feature-rx-copy and feature-rx-flip. Add support for request-rx-copy. PR 40650.
  - Add i368PAE support to Xen3 dom0.
  - Add Xen3 PCI pass-through support.
  - Fix stalled xbdback detach that would stall the whole xenbus thread, preventing new domUs from being created.
  - Make it possible to use netbsd-5 domUs running on a Xen2 hypervisor.

Miscellaneous

- `sh`(1): Make the cd builtin accept and ignore -P. PR 42557.
- `fdisk`(8): Fix issues with large disks.
- `savecore`(8): Instead of exiting with an obscure error message if -N /kernelname isn't specified, blithely assume the kernel will consume around 20 megs.
- `newfs`(8): Issue a better error message if attempting to create a file system on a block device. Inspired by PR 41127.
- `newfs_msdos`(8): Make fs size detection get proper size rather than disk size.
- `termcap`(3): Only add the ZZ capability for termcap entries that are larger than 1023 bytes.
- `dkctl`(8): List the partition types addwedge understands. PR 37252. Make dkctl conform to its man page and print the device name on addwedge when the addition was successful (as well as indicating success).
- If the current locale doesn't define the 'thousands' grouping info then use sane defaults (',' every 3 digits). PR 40714.
- `pthread`(3):
  - Make nanosleep cancelable again.
  - Improve the algorithm used in pthread creation so that it does a better job of reusing dead threads.
  - Make sure thread id is set correctly in case a threaded program forks from a thread other than the main thread. Fixes issues with, e.g., ruby.

- newsyslog(8): Reset ziptype on each line. Failure to do this caused any log file to be compressed if it was listed after a line using Z or J flag.
- nvi(1):
  - Fix a problem where the pattern / didn't match a dollar sign. PR 41781.
  - Make :ESC, #+ and #- work again.
  - Rename "expandtabs" to "expandtab" to match documentation.
  - Implement the "et" abbreviation for expandtab.
  - Fix regexp on LP64 systems. PR 41924.
  - Fix format string bug: filenames may contain % characters.
  - Fix ˜ on big-endian architecturs.
- pstat(8): Distinguish between UFS1 and UFS2 inodes by reading the ufsmount structure, the previous heuristic of comparing the size fields of inode and dinode failed.
- btpin(1): Add a -P flag to attempt immediate pairing.
- ksh(1): Support 0xnn for hexadecimal constants, as well as 16#nn. PR 40512.
- Add support for fr_*.UTF-8 locale. Setting LANG to fr_*.UTF-8 won't get the message catalog right (they're encoded in iso-8859-1), but other locale functions should work properly.
- Add Solaris-like dlinfo() interface to the ELF dynamic linker. Implement RTLD_DI_LINKMAP which returns a pointer to the linkmap chain at the given object.
- kill(1): Make sure that numerical signals and pids are in range for their types. PR 42143.
- pcap(3): pcap_lookupnet(): reset ifr before SIOCGIFNETMASK. Without it we get back a bogus netmask. PR 41367.
- man(1): Accept a pathname to a man file, e.g., "man ./man.1"
- setlocale(3): Handle nonexistent locales properly. PR 42124.
- amldb(8): Check validity of parent to avoid a potential segfault.
- sort(1): Fix a ton of bugs including but not limited to PR 18614 PR 27257 PR 25551 PR 221 82 PR 31095 PR 30504 PR 36816 PR 37860 PR 39308 PR 42094.
- user(8): Fix -p flag: rm_eo is the first character *after* the match, so no need for a +1. Blowfish hashes are only 53 chars long, not 54.
- Update libevent to 1.4.11-stable.
- hunt(6): Fix possible remote DoS of a running hunt game, and prevent a possible theoretical attack involving >= 1 billion ammo.
- systat(1): Fix problems with SIGWINCH. PR 42161.
- gzip(1):
  - Fix support for multi-section bzip2 files, as created by pbzip2.
  - Add "pack" uncompression support.
  - Avoid an overflow in suffix handling.

- audio{ctl,play,record}: Add extended WAVE header support, attempt to play a bunch more WAV files.
- Update Postfix to 2.6.5.
- `df`(1): Block numbers are measured in f_frsize units. Make -P option use this instead of f_bsize. Also account for reserved blocks like normal non-P output. PR 41541.
- Update libfetch to 2.30.
- `window`(1): Fix a SEGV with certain terminal types. PR 41581.
- Update pkg_install to 20100204:
  - audit-packages.conf(5) has been superseded by pkg_install.conf(5). The default configuration is the same.
  - Support for pkg_view(1) has been retired.
  - The functionality of audit-packages(1) and download-vulnerability-list(1) has moved into `pkg_admin`(1). Wrapper scripts that handle the common use cases are provided.
- `racoonctl`(8): Adjust ADMINPORTDIR to match that of racoon (`/var/run`). PR 41376.
- `schedctl`(8): Skip LSIDL and LSZOMB threads when retrieving info.
- `postinstall`(8) now knows about `/etc/dhcpcd.conf`.
- `drvctl`(8): Allow querying for root devices in the tree by specifying an empty device name. Ensure that l_devname is NUL-terminated and fail otherwise. If drvctl -l is used without argument, print the root device nodes.
- `paste`(1): Null-terminate the delimiter list string after processing escapes (which can shorten it) because the code that issues delimiters depends on it being null-terminated. This caused e.g. paste -d ' ' a b to print a '0' at the beginning of each line. PR 41159.
- Don't attempt to read or write ~/.lesshst if it's not a regular file or a symlink to a regular file. Previously, symlinking to `/dev/null` would cause less to trash `/dev/null` if run with sufficient privileges. PR 42237.
- Make rtsold_flags default to -a. PR 39657.
- libevent: Add -fno-strict-aliasing to work around problems with GCC 4 and strict-aliasing.
- Renamed a number of internal getline() functions to get_line() so as to compile under -current.
- Install the Xvidtune app-defaults file. PR 41577.
- Fixes to Linux compat:
  - In `sendmsg`(2), copy the msghdr structure before trying to use it.
  - In linux_sys_sched_getaffinity(), do not leak memory on error.
- Various METALOG fixes, including sorting entries. PRs 24457 and 41155.
- Update DRM code.
- `cdce`(4): Add support for the Openmoko Freerunner. PR 40049.
- `pms`(4): Add elantech multi-touch support.
- Big X.Org update.

- ugen(4): ugen_get_cdesc() can return NULL. When this happens, return EINVAL error instead of crashing.
- Fix a race between nfsd and local rm. PR 41147.
- puc(4): Add support for Oxford Semiconductor OXmPCI952 2 port UARTs. Add support for the IBM 481033H SCC.
- sysinst(8): Try ATA/SCSI identify commands and when successful, use the model information along with the disk size when creating the 'Available disks' menu. PR 41925.
- Fix PR kern/41659: add missing splx() in FAST_IPSEC code.
- Add hungarian keyboard layout.
- Add Swiss German keyboard layout.
- Introduce MKSUBPIXEL, which allows enabling subpixel rendering code in FreeType.
- Support drives over 1TB in size in sysinst. Error if drive is over 2TB in in size (and thus over the disklabel limit).
- Update and add some TNF ssh keys to /etc/ssh/ssh_known_hosts.
- Prevent makefs(8) from creating invalid ISO format on rockridge support which causes fatal errors in ARC BIOS firmware on MIPS Magnum R4000. PR 42410.
- sysinst(8): When creating /etc/fstab: for the first swap partition use type "sw,dp" instead of "sw", so dump device gets configured correctly if swap is not on the second partition. PR 42148.
- sys/atomic.h: Make atomics usable from C++.
- wc(1): Add support for "-L" option (longest line) as present in the GNU and FreeBSD versions of "wc".
- libm: Add f{min,max,dim}{,l,f}.
- jemalloc: Fix race condition on reallocation of huge category. PR 42876.
- sys/null.h: Enclose (void *)0 in an extra set of parenthese to make the result usable in arbitrary expressions. PR 41890.

Known Problems

Using block device nodes (e.g., wd0a) 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 (e.g., rwd0a), 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.

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 5.1 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 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 5.1 Release Contents

The root directory of the NetBSD 5.1 release is organized as follows:

`.../NetBSD-5.1/`

`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-5.1`	Changes between the 5.0 and 5.1 releases.
`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 5.1 has a binary distribution.

The source distribution sets can be found in subdirectories of the `source` subdirectory of the distribution tree. They contain the complete sources to the system. The source distribution sets are as follows:

gnusrc This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.
79 MB gzipped, 450 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, 32 MB uncompressed

src This set contains all of the base NetBSD 5.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
59 MB gzipped, 350 MB uncompressed

syssrc This set contains the sources to the NetBSD 5.1 kernel for all architectures as well as the `config`(1) utility.
34 MB gzipped, 197 MB uncompressed

xsrc This set contains the sources to the X Window System.
127 MB gzipped, 694 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:

`MD5` MD5 digests in the format produced by the command:
cksum -a MD5 *file*.

`SHA512` SHA512 digests in the format produced by the command:
cksum -a SHA512 *file*.

The SHA512 digest is safer, but MD5 checksums are provided so that a wider range of operating systems can check the integrity of the release files.

NetBSD/sparc subdirectory structure

The sparc-specific portion of the NetBSD 5.1 release is found in the `sparc` subdirectory of the distribution: `.../NetBSD-5.1/sparc/`. 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-GENERIC_SCSI3.gz`
A kernel with SCSI ID 3 explicitly mapped to `sd0`.

`netbsd-KRUPS.gz`
A kernel for JavaStation-NC (Krups) machines.

`netbsd-MRCOFFEE.gz`
A kernel for JavaStation-1 (Mr. Coffee) machines using Open Boot Prom firmware.

`netbsd-MRCOFFEE_OFW.gz`
A kernel for JavaStation-1 (Mr. Coffee) machines using OpenFirmware.

`netbsd-GENERIC_SUN4U.gz`
A kernel which supports sparc64 hardware in 32-bit mode only. See the sparc64 INSTALL notes for help installing this kernel on your sparc64 hardware.

`sets/` sparc binary distribution sets; see below.

`installation/`

- `bootfs/` Netbootable RAM disk kernel and the file system in the RAM disk; see below.
- `floppy/` sparc boot and installation floppies; see below.
- `miniroot/` sparc miniroot file system image; see below.
- `netboot/` Netboot bootloader and root filesystem with install tools; see below.
- `tape/` sparc boot and installation tapes; see below.

Binary distribution sets

The NetBSD sparc binary distribution sets contain the binaries which comprise the NetBSD 5.1 release for sparc. The binary distribution sets can be found in the `sparc/binary/sets` subdirectory of the NetBSD 5.1 distribution tree, and are as follows:

base The NetBSD 5.1 sparc **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.
27 MB gzipped, 81 MB uncompressed

comp Things needed for compiling programs. This set includes the system include files (`/usr/include`) and the various system libraries (except the shared libraries, which are included as part of the **base** set). This set also includes the manual pages for all of the utilities it contains, as well as the system call and library manual pages.
39 MB gzipped, 141 MB uncompressed

etc This distribution set contains the system configuration files that reside in `/etc` and in several other places. This set *must* be installed if you are installing the system from scratch, but should *not* be used if you are upgrading.
1 MB gzipped, 1 MB uncompressed

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

kern-GENERIC
This set contains a NetBSD/sparc 5.1 `GENERIC` kernel named `/netbsd`. You *must* install this distribution set.
2 MB gzipped, 4 MB uncompressed

kern-GENERIC.MP
This set contains a NetBSD/sparc 5.1 `GENERIC.MP` kernel, which will use multiple processors (if present), named `/netbsd`.
2 MB gzipped, 4 MB uncompressed

kern-GENERIC_SCSI3
This set contains a NetBSD/sparc 5.1 `GENERIC_SCSI3` kernel, which will set SCSI ID 3 to `sd0`, named `/netbsd`.
2 MB gzipped, 4 MB uncompressed

kern-GENERIC_SUN4U
This set contains a NetBSD/sparc 5.1 `GENERIC_SUN4U` kernel, which supports only UltraSPARC systems in 32 bit mode, named `/netbsd`.
4 MB gzipped, 9 MB uncompressed

kern-KRUPS
This set contains a NetBSD/sparc 5.1 `GENERIC-KRUPS` kernel, which supports the JavaStation-NC, named `/netbsd`.
2 MB gzipped, 3 MB uncompressed

kern-MRCOFFEE
This set contains a NetBSD/sparc 5.1 `MRCOFFE` kernel, which which supports the OpenBOOT PROM 2 JavaStation-1, named `/netbsd`.
2 MB gzipped, 3 MB uncompressed

kern-MRCOFFEE_OFW
This set contains a NetBSD/sparc 5.1 `MRCOFFE_OFW` kernel, which which supports the Open-Firmware JavaStation-1, named `/netbsd`.
2 MB gzipped, 3 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, 10 MB uncompressed

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

xbase The basic files needed for a complete X client environment. This does not include the X servers.
7 MB gzipped, 20 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
13 MB gzipped, 47 MB uncompressed

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

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

xserver The X server. This includes Xsun, Xsun24, XsunMono and Xvfb servers with man pages.
13 MB gzipped, 35 MB uncompressed

The sparc 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: The distribution cannot be extracted using the SunOS **`tar`** program

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

NetBSD/sparc System Requirements and Supported Devices

The minimum configuration requires 4 MB of RAM and 100 MB of disk space. To install the entire system requires considerably more space, and more RAM to achieve reasonable performance. NetBSD with 4 MB of RAM feels like Solaris with 4 MB of RAM — slow. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.

Note: The sparc port does not have functional SMP support in this release. We expect this to be fixed in the next release, but in the meantime, NetBSD 4.0.1 has working SMP support.

Supported machines

- sun4
  - 4/100
  - 4/200
  - 4/300
  - 4/400
- sun4c
  - SPARCstation/server 1
  - SPARCstation/server 1+
  - SPARCstation/server 2
  - IPC
  - ELC
  - IPX
  - SLC
- sun4c clones
  - Aries Parrot II
  - Opus/5000
  - Opus/5250
  - FORCE Sparc CPU1e/4
  - FORCE Sparc CPU2e/4

  (and possibly more)

  Note: NetBSD doesn't support the VME interface on the FORCE models
- sun4m
  - 4/600
  - SPARC Classic
  - SPARC LX
  - SPARCstation 4
  - SPARCstation/server 5
  - SPARCstation/server 10
  - SPARCstation/server 20
  - SPARCstation Voyager
  - JavaStation-1 (Mr. Coffee)
  - JavaStation-NC (Krups)

  Note: All known SuperSPARC, hyperSPARC, microSPARC, and TurboSPARC CPUs are supported

- sun4m clones
  - Axil 245
  - Axil 311
  - Axil 320
  - Cycle C C Cycle 5-IP
  - DTKstation/Classic+
  - RDI PowerLite-85
  - RDI PowerLite-110
  - RDI BriteLite
  - Ross sparc 20
  - Transtec SS5/170
  - Tatung microCOMPstation 5
  - Twinhead TWINstation 5G
  - TWS SuperCOMPstation-20S
  - Tadpole SPARCbook 3 series

  (and possibly more)

Machines not supported by NetBSD/sparc

- sun4d
  - SPARCcenter 1000
  - SPARCcenter 2000
- sun4u and other UltraSPARC based machines in 64-bit mode. They are supported in 32-bit mode with the NetBSD/sparc `GENERIC_SUN4U` kernel.

  UltraSPARC based machines in 32-bit or 64-bit mode are supported by the NetBSD/sparc64 port. Refer to

  http://www.NetBSD.org/ports/sparc64/index.html
- Solbourne machines

Supported devices

- sun4m mbus CPU modules; Sun 4/600, SPARCstation/server 10, SPARCstation/server 20, and clones
  - *cpu0*; All single-CPU systems are supported
  - *cpuN*; All hardware compatible dual-, tri- and quad-CPU SuperSPARC-I, SuperSPARC-II, and hyperSPARC systems are supported with the SMP kernel (`GENERIC.MP`)

    http://www.NetBSD.org/ports/sparc/faq.html#smp-cpus
- sun4c and sun4m video; Sbus card, on-board, or optional
  - *bwtwo*; 1 bpp framebuffer.
    On-board and Sbus card (MG1 and MG2)
  - *cgthree*; 8 bpp framebuffer.
    On-board and Sbus card
  - *cgsix*; 8 bpp accelerated framebuffer.
    On-board and Sbus card (GX, GX+, TGX, and TGX+)
  - *cgfourteen*; SX 24 bpp accelerated framebuffer.
    On-board SPARCstation 10SX and SPARCstation 20, requires optional 4 or 8 MB VSIMM,
    limited to *cgthree* emulation

- *tcx*; S24 accelerated color framebuffer.
  On-board SPARCstation 4 (8 bpp), JavaStation-1 (24 bpp), and optional on SPARCstation 5 (24 bpp),
  limited to *cgthree* emulation
- *pnozz*; Weitek p9100 24 bpp accelerated framebuffer.
  On-board Tadpole SPARCbook 3GS and 3GX,
  limited to *cgthree* emulation
- *zx*; ZX and TZX 24 bpp accelerated framebuffer.
  Sbus card (leo)
  limited to text console (no X)
- *igsfb*; 24 bpp IGS C1682 accelerated framebuffer.
  On-board JavaStation-NC
  limited to text console (no X)

• sun4 video; VME card, on-board, or P4 (supported only on 4/100 and 4/300 series machines)

- *bwtwo*; 1 bpp framebuffer.
  On-board and P4
- *cgtwo*; 8 bpp framebuffer.
  VME card
- *cgfour*; 8 bpp framebuffer with overlay.
  P4
- *cgsix*; LEGO 8 bpp accelerated framebuffer.
  P4
- *cgeight*; TC 24 bpp framebuffer with overlay.
  P4 (not well tested)

• Network interfaces

- *be*; 100 Mbps half-duplex "Big Mac" ethernet.
  Sbus card
- *en*; Efficient Networks, Inc. ATM interface.
  Sbus card
- *hme*; 100 Mbps full-duplex "Happy Meal" ethernet.
  On-board (JavaStation-NC) and Sbus card
  This hardware is not supported on the sun4c platform
- *ie*; 10 Mbps Intel 82586 ethernet.
  On-board (4/100 and 4/200, may not be working properly on 4/400) and VME card
- *le*; 10 Mbps AMD 9790 Lance ethernet.
  On-board (4/300, sun4c, and sun4m) and Sbus card
- *qec*; One 100 Mbps be interface and four 10 Mbps Mace ethernet interfaces.
  Sbus card
- *qfe*; four 100 Mbps full-duplex "Happy Meal" ethernet interfaces.
  Sbus card
- misc: various ethernet interfaces are found on multi-function Sbus cards (e.g. along with SCSI), the ethernet interfaces work even if the other functions do not

- SCSI interfaces
  - *esp*; 5 MB/s and 10 MB/s (Fast) NCR 53C9x SCSI controller, and 20 MB/s (Fast+Wide) FAS336 controller (known as 'fas' on Solaris).
    On-board (sun4c, sun4m, 4/300, Ultra 1e, and Ultra 2) and Sbus cards
  - *isp*; 10 MB/s (Fast) and 20 MB/s (Fast+Wide) Qlogic ISP 10x0 family.
    Sbus card
  - *si*; 5 MB/s NCR 5380 SCSI controller.
    VME card (Interrupt driven and polled DMA)
  - *sw*; 5 MB/s NCR 5380 "SCSI Weird" controller.
    Onboard 4/110 (polled DMA only)
  - misc: SCSI functionality is sometimes found on multi-function Sbus cards (e.g. along with ethernet), the SCSI *esp* and *isp* portions of those cards should work, even if the other functions do not.

    Note: Non-Sun SCSI CD-ROMs may have trouble booting any OS, NetBSD or otherwise. Your CD-ROM must support 512 byte blocks for the OpenBoot PROM to load a file. Most newer CD-ROM drives support this automatically, and many older drives allow you to set it using a jumper. See
    http://www.ebsinc.com/solaris/suncdrom.html
    for information on bootable CD-ROMs
- VME disk interfaces
  - *xd*; Xylogics 7053 VME/SMD disk controller
  - *xy*; Xylogics 450/451 VME disk controller

  Note: VME/IPI disks are not supported
- Floppy drives
  - *fd*; floppy disk drive.
    On-board sun4c and sun4m
  - *sd*; SCSI floppy drive, acts as a SCSI drive to kernel.
    found with some SPARCbook 3 machines
- PCMCIA/Cardbus devices
  - *nell*; PCMCIA interface
    On-board (SPARCstation Voyager) and Sbus card.
    All devices listed in the Machine-independent PCMCIA drivers page should work, but very few have been tested with NetBSD/sparc.
    http://www.NetBSD.org/support/hardware/pcmcia.html

  Note: See the FAQ for more information.
  http://www.NetBSD.org/ports/sparc/faq.html#nell
- Serial interfaces
  - *zs*; Zilog 8530 serial ports, which can be used as the system console
    On-board sun4, sun4c, and sun4m
  - *com*; Zilog 85C30 serial ports (ns16550a)
    On-board Tadpole SPARCbook 3GS and 3GX (internal 14.4 kbps modem, use `/dev/ttyC0`) JavaStation-1, and JavaStation-NC

- Input devices
  - Sun keyboard and mouse
  - SPARCbook 3 keyboard and joystick
  - Third-party keyboards, mice, and trackballs
  - PS/2 keyboards and mice on JavaStation models
- Audio interfaces
  - *audioamd*; AMD 7930-based 8000 Hz, 8 bit, mono, mu-law.
    On-board sun4c and SPARC Classic
  - *audiocs*; CS4231-based 44 KHz, 16 bit, stereo, PCM.
    On-board SPARCStation 5, JavaStation-1, JavaStation-NC, and optional on SPARCStation 4.
- Misc cards
  - *mtty / mbpp*; Magma Serial/Parallel ports.
    Sbus card
  - *xbox*;
    Sun Sbus expander device
- Parallel ports
  - *bpp*; Bi-directional parallel port.
    On-board and Sbus card

Unsupported devices

- DBRIe-based audio and ISDN.
  On-board some sun4m machines (such as SPARCstation 10, SPARCstation 20, SPARC LX, SPARCstation Voyager, and Tadpole SPARCbook 3)
- *sc*; Sun SCSI.
  VME card
- *cgtwelve*; I.e. GS, 24 bpp framebuffer.
  Sbus card
- PCMCIA interface
  On-board Tadpole SPARCbook 3GS and 3GX

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.

CD-ROM / DVD Find out where the distribution set files are on the CD-ROM or DVD. Likely locations are `binary/sets` and `sparc/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 `sparc/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 `204.152.190.15` and the IPv6 address is `2001:4f8:3:7:230:48ff:fec6:9aaa:21` (as of May, 2010).

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-5.1
# cd sparc/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

Configuring your PROM

Before you start, you should configure your PROM. There are three categories of PROM: "sunmon" (sometimes called Restricted Prompt, sun monitor, or old command mode), OpenBoot PROM 1, and OpenBoot PROM 2. The sun4 machines only have sunmon. Some of the early sun4c models default to sunmon but have OpenBoot PROM 1. The later sun4c models and all sun4m models have OpenBoot PROM 2.

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 a PROM prompt. If you are using a Tadpole SPARCbook, you press the `Pause` and `a` keys. 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 sparc system is `ff:ff:ff:ff:ff:ff`, 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.

Next, you should set your system to always use the OpenBoot PROM (sometimes called "new command mode") if it defaults to sunmon. The NetBSD kernel relies on some of the functionality provided by the OpenBoot PROM. If your machine gives you a '>' prompt instead of an 'ok' prompt, type:

```
> n
ok setenv sunmon-compat? false
```

ok

Next, if you are using any security features of OpenBoot PROM, you should turn them off — NetBSD can't deal well with this.

ok **setenv security-mode none**

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

The OpenBoot PROM 1 machines (SPARCstation/server 1, SPARCstation/server 1+, IPC, and SLC) have an odd SCSI quirk you should be aware of. There are three SCSI addressing schemes used by your system: SCSI target ID (set by physical jumpers on the device), PROM 'unit' number (set by OpenBoot PROM 1, based on its SCSI target ID), and the name you reference within an operating system (set by the kernel, based on the PROM 'unit' number).

Sun shipped these systems with the internal drives set to SCSI target IDs 3 and 1. The default value of the OpenBoot PROM variable '`sd-targets`' is '`31204567.`' This variable maps how the OpenBoot PROM 1 assigns 'unit' numbers based on the SCSI target ID. Thus the device at SCSI target ID 3 is considered 'unit' 0, and the SCSI device at target ID 0 is 'unit' 3. When you type "**boot scsi(0,0,0)**", the OpenBoot PROM will boot from 'unit' 0 (which is SCSI target ID 3, the internal hard drive). The SunOS kernel is hard-wired to map `sd0` to SCSI target 3, and `sd3` to SCSI target 0.

PROM Unit#	*SCSI Target*	*SunOS name*
0	3	sd0
1	1	sd1
2	2	sd2
3	0	sd3
4	4	st0
5	5	st1
6	6	cdrom

The NetBSD/sparc `GENERIC` kernel does not wire things down as does SunOS. It names the disks in the order that the SCSI targets are probed (01234567). If you only have one disk, it is always `sd0` regardless of its SCSI target ID or its PROM 'unit' number, and there are no problems. If you have two disks, one at SCSI ID 2 ('unit' 2) and SCSI ID 3 ('unit' 0), then they are recognized as `sd0` and `sd1` respectively. This can be a problem if you are not aware of it, particularly when creating an fstab.

There are two approaches to fixing this problem: changing the mapping that OpenBoot PROM 1 does, and changing the NetBSD kernel configuration. To get OpenBoot PROM 1 to number the SCSI 'unit' numbers the same as the SCSI target IDs, you need to run this command:

ok **setenv sd-targets 01234567**

This may, however, cause problems if you were to later attempt to use SunOS on this machine or if you reset the OpenBoot PROM variables.

The other approach is to use a NetBSD kernel that matches the PROM's odd target mapping by treating 'unit' 0 (i.e. SCSI target ID 3) as `sd0` and 'unit' 3 (i.e. SCSI target ID 0) as `sd3`. The `GENERIC_SCSI3` kernel performs this target mapping, but the `GENERIC` and `INSTALL` kernels do not.

Note: This is also a concern when you start building your own customised kernels.

The machines with OpenBoot PROM 2 (SPARCstation/server 2, ELC, IPX, and all sun4m models) have a similar SCSI target mapping in the form of a **devalias** entry. That is, the device alias **disk** is shorthand for the disk at SCSI ID 3 on the internal SCSI controller. Normally, the **disk** device alias is what the PROM uses as the default boot device, i.e. in the absence of a *device* argument to the **boot** command.

Note that there are also pre-configured device alias entries for **disk0, disk1, disk2** and **disk3**, which are in fact a one-to-one mapping to the SCSI targets **0** to **3** (all on the internal SCSI controller) .

Again, it may be advantageous to use a fixed "SCSI target" to "NetBSD disk unit" mapping in your kernel configuration file (such as is done in the `GENERIC_SCSI3` kernel) to ensure that your disks remain showing up at the same NetBSD device unit numbers even if you add disks to your system at a later time.

Determining how to access your SCSI disk from the PROM

sunmon and OpenBoot PROM 1 use an archaic `sd(`*c*`,`*u*`,`*p*`)` syntax to address SCSI devices. OpenBoot PROM 2 uses a more intuitive syntax using device aliases.

To calculate the parameters for sunmon and OpenBoot PROM 1:

c specifies the SCSI controller number (first is 0, second is 1, ...)

u the hexadecimal number obtained from evaluating the expression *(8 ∗ TARGET) + LUN*

p the partition number from which to boot (`0` = `a`, `1` = `b`, etc.)

Therefore, to boot from the swap partition on the internal hard drive (first SCSI bus, target 0, lun 0, partition 1), one would use:

ok **boot sd(0,0,1)**

To boot from a CD-ROM (first SCSI bus, target 6, lun 0, partition dynamically determined), one would use:

ok **boot sd(0,30,)**

And, to boot from a kernel named `netbsd-GENERIC` on the fourth partition (`d`, often the `/usr` partition) on an external hard drive (first SCSI bus, target 2, lun 0, partition 3), one would use:

ok **boot sd(0,10,3)netbsd-GENERIC**

Now, for OpenBoot PROM 2, SCSI devices are specified by an OpenBOOT **devalias** which provides simple mnemonics for the full path to the device. Type **devalias** in OpenBoot PROM 2 to get a list of all of the available aliases. Just the alias and partition are necessary when booting.

Therefore, to boot from the swap partition on the internal hard drive (OpenBoot PROM 2 assumes the internal hard drive is at target 3), one would use:

ok **boot disk:b**

To boot from a CD-ROM (OpenBoot PROM 2 assumes the CD-ROM is 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 hard drive (target 2, partition 3), one would use:

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

The full device path specifier for OpenBoot PROM 2 depends on how OpenBoot PROM 2 recognizes your SCSI controller. Typically, one would use something like: `/sbus/esp/sd@`*t*`,`*p* where t is the SCSI target and p is the partition number.

Determining how to boot from an SBUS card

Some SBUS cards have firmware that lets you use them as a boot device. These cards do not automatically create a **devalias** entry, so you must traverse the device tree to figure out what the OpenBoot PROM calls your card. You will be using OpenBoot PROM 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 HME card in a SPARC Classic.

```
ok cd /
ok ls
ffd3b790 TI,TMS390S10@0,f8fffffc
ffd2d254 virtual-memory@0,0
ffd2d198 memory@0,0
ffd2b65c obio
ffd2b310 iommu@0,10000000
ffd2b2a0 openprom
ffd24af0 aliases
ffd24abc options
ffd24a88 packages
```

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

```
ok cd iommu
ok ls
ffd2b454 sbus@0,10001000
ok cd sbus
ok ls
ffd467e8 cgthree@3,0
ffd42a1c SUNW,hme@1,8c00000
ffd4297c le@0,c00000
ffd40d28 ledma@4,8400010
ffd40c9c SUNW,bpp@4,c800000
ffd40c00 audio@4,1300000
ffd3dc68 espdma@4,8400000
ok cd SUNW,hme
ok ls
ok pwd
/iommu@0,10000000/sbus@0,10001000/SUNW,hme@1,8c00000
```

OK, now we know the path to the HME 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
reset          seek           load           open           close
watch-net      selftest       obp-selftest   write          read
preamble-32    enable-link-pulse             disable-link-pulse
force-speeds   reset-transceiver             use-bit-bang-mode
use-frame-mode                dump-phys      transfer-speed=10
transfer-speed=100            mii-write      mii-read
create-rev-id
```

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

```
ok .attributes
hm-rev                   00000022
version                  1.18
model                    SUNW,501-2919
device_type              network
intr                     00000037  00000000
interrupts               00000004
address-bits             00000030
max-frame-size           00004000
reg                      00000001  08c00000  00000108
```

```
                                        00000001  08c02000  00002000
                                        00000001  08c04000  00002000
                                        00000001  08c06000  00002000
                                        00000001  08c07000  00000020
     name                               SUNW,hme
     ok
```

So, when it's time to type in a boot command, use the shortened version of the **pwd** command. In this example, you'd type:

ok **boot /iommu/sbus/SUNW,hme**

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/sparc 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: fd0[a-h] le0 le1 hme0 sd0[a-h] halt
root device (default sd0a): hme0
dump device: hme0
file system (default generic): nfs
root on hme0
```

Deciding on partition sizes

If you're installing NetBSD/sparc for the first time it's a good idea to look at the partition sizes of disk you intend installing NetBSD/sparc on.

A minimal installation of NetBSD requires about 100 MB. A full installation requires considerably more. A good initial size for the swap partition is twice the amount of physical memory in your machine (unlike SunOS 4.x, there are no restrictions on the size of the swap partition that would render part of your memory unusable).

Note that there are limitations on the size of the root partition for various models.

On sun4 machines, the NetBSD sparc boot loader can only boot from RAID partitions that start at the beginning of the disk.

On sun4 and early PROM version sun4c machines, the PROM can only boot from the first 1Gb of the disk.

On later PROM version sun4c and early PROM version sun4m machines, the PROM can only boot from the first 2Gb of the disk.

On later PROM version sun4m machines, the PROM can only boot from the first 4Gb of the disk.

Configuration of network interfaces

Some network devices (i.e. the built-in *le* interface on sun4m machines) allow a choice between operating on a UTP or a AUI port once the NetBSD kernel is running. The *le* driver supports automatic detection of the port which is actually connected to the wire. Additionally, some of the Fast ethernet devices (such as *be*, *hme*, *qec*, and *qfe*) support selection of various speeds and options. The default is to attempt to automatically detect the speed.

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` or `AUI` to select the AUI connector, or `10baseT` or `UTP` to select the UTP connector. Fast ethernet interfaces default to `auto`, which usually does not detect properly and runs at '10BaseT' speed. The options are `10baseT`, `10baseTX`, and `auto`. The *hme* and *qfe* interfaces also allow `10baseT-FDX` and `100baseT-FDX`.

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 your system. The easiest way is to use the `miniroot.fs` image copied to your local disk's swap partition or a scratch drive. If your Sparc is hooked up in a network, you may configure another UNIX-like machine as a netboot server for your Sparc. Alternatively, there is a pair of floppy images that will boot your system and run the installer.

For more information on the commands and variables available in the OpenBoot PROM (present in all sun4c and sun4m machines), go to **http://docs.sun.com** and search for "openboot reference" (without quotes).

Installing NetBSD by placing a bootable file system on a partitioned hard drive

`installation/miniroot/miniroot.fs.gz` is a compressed, self-contained NetBSD file system holding all utilities necessary to install NetBSD onto a disk attached to your system. It is distributed as a compressed plain file you will transfer to a raw disk partition. You will then boot this installer using the appropriate PROM command. The simplest case is where you place the `miniroot.fs` file system on the swap partition of your disk. Alternatively, you can place it on any other unused partition on any disk (such as a Zip disk). Be forewarned that you will not be able to install NetBSD onto the partition that contains the `miniroot.fs` as this partition is needed during the install process.

Loading the file system onto a raw partition is straightforward. First, download `miniroot.fs` to your system. Next, reboot in single-user mode (i.e. **`boot -s`**) to ensure that your system will not be swapping. Finally, place the `miniroot.fs` on your partition of choice. On NetBSD or SunOS the commands are:

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

Here, `/dev/rsd0b` is assumed to be your swap partition. If you decide to place `miniroot.fs` on a non-swap partition, it will overwrite all of the contents of that partition, but you will not need to reboot into single-user mode to write it.

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

```
# halt
```

Then boot the installer by typing the appropriate command at the PROM prompt (this example is for the swap partition):

sunmon

> **`b sd(,,1)netbsd -s`**

OpenBoot PROM 1

ok **`boot sd(,,1)netbsd -s`**

OpenBoot PROM 2

ok **`boot disk:b netbsd -s`**

The monitor boot command will load the NetBSD kernel contained in the file system image. After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the installation scripts.** below.

Installing NetBSD by using a bootable CD-ROM

If you wish to burn a CD-R containing a bootable NetBSD installation, then you can either burn the prepared disk image in `installation/cdrom/sparccd-5.1.iso` or create your own. If you wish to create your own, you should follow the directions on the NetBSD Bootable CD-ROM HOW-TO at:

http://www.NetBSD.org/docs/bootcd.html#sparcimage

Once you have downloaded `sparccd-5.1.iso` or created your own disk image, then you need to burn it to a CD-R.

The CD-ROM attached to your Sparc must support 512 byte sectors to be bootable. All Sun brand CD-ROMs will work, as well as many other manufacturers. See the Sun CD-ROM FAQ: **http://saturn.tlug.org/suncdfaq/**

Sun sets their CD-ROM drives to SCSI ID 6. We recommend you do the same.

Boot the installer by typing the appropriate command at the PROM prompt:

sunmon

> **b sd(,30,)**

OpenBoot PROM 1

ok **boot sd(,30,)**

OpenBoot PROM 2

ok **boot cdrom**

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

Installing NetBSD by using the bootable floppies

The NetBSD install floppies only work on the sun4c and sun4m machines. Making the install floppies is fairly straightforward.

NetBSD and SunOS use the same commands. First, get two 1.44 MB floppy disks and format them either using the `fdformat`(8) command or a PC.

\# **fdformat**

Be sure that the 'fdformat' command completes successfully before proceeding; on NetBSD success is a string of all '`V 's`', and on SunOS success is a string of '`. 's`'.

Next, transfer the two floppy images (`installation/floppy/disk1.gz` and `installation/floppy/disk2`) to the disks you just formatted. You can do this step from NetBSD, SunOS, or a Windows machine using **`rawrite.exe`** (available in the NetBSD/i386 distribution). Insert the first floppy and run the following commands:

```
# gunzip disk1.gz
# dd if=disk1 of=/dev/rfd0a bs=36k
# eject floppy
```

Insert the second floppy and run the following commands:

```
# dd if=disk2 of=/dev/rfd0a bs=36k
# eject floppy
```

Note that disk2 may be a symlink to .tgz file; do not uncompress it, just follow the instructions above, verbatim. After writing the two floppies, be sure to label them so you can later identify them.

Insert the floppy made from `disk1` into your Sparc. From the OpenBoot prompt, boot the floppy with the following command:

OpenBoot PROM 1

ok **`boot fd(,,1)`**

OpenBoot PROM 1 (alternate)

ok **`boot /fd`**

OpenBoot PROM 2

ok **`boot floppy`**

After the initial probe messages you'll be asked to insert the floppy labeled "NetBSD disk2". If the floppy is not automatically ejected, you can either use a straightened paperclip to eject the disk or do the following:

Press the `STOP` key (sometimes called the `L1` key, found on the left side of your keyboard) and the `a` key, which will halt your system and give you a PROM prompt.

ok **`eject`**
ok **`go`**

Now, insert the floppy labeled `disk2`. After the installation tools have been loaded, you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the installation scripts.** below.

Installing NetBSD by using a netboot setup

1. Introduction

   To netboot a sparc, you must configure one or more servers to provide information and files to your sparc (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 sparc systems except JavaStations with OpenFirmware PROM use a combination of RARP and BOOTP for the discovery stage. In the past, these systems used RARP and BOOTPARAMS. JavaStations booting from OpenFirmware use DHCP. TFTP is used in the bootstrap phase to download the bootstrap program, `boot.net`, which has been linked to a file name appropriate to the client's architecture and IP address as described in the TFTP section below. JavaStations with OpenFirmware need a special bootstrap program, `bootjs.net`, which is a version of boot.net in ELF format. It is not linked to the IP address. 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.

                #
                # Only needed for JavaStations with OpenFirmware!
                #
                filename                "bootjs.net";
```

```
                        #
                        # 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 everything
                        # 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
        }
```

Note that for JavaStations booting from OpenFirmware you do not need the

```
allow bootp;
```

statement, since this machines use real DHCP.

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)

   Note: no **rarpd** is needed to boot OpenFirmware based JavaStations.

   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 sparc machine to /tftpboot. Get boot.net from the installation/netboot directory of the distribution.

```
# cp boot.net /tftpboot
```

Now, you need to link boot.net to the filename that your sparc will look for. It will look for a filename composed of the machine's IP address (in hexadecimal) followed by the machine's architecture, separated by a period. For example, a sun4c machine which has been assigned IP address 192.168.1.10, will make a TFTP request for C0A8010A.SUN4C.

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 boot.net C0A8010A.SUN4C
```

For OpenFirmware based JavaStations you use the file bootjs.net and do not create a symlink for the IP address. This machines do full DHCP autoconfiguration, so the bootstrap filename is provided by the DHCP server.

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/netboot/rootfs.tgz`.

   ```
   # cp *tgz /export/client/root
   # cd /export/client/root
   # tar -xpzf rootfs.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/exports
     # /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 Sparc from the server by entering the appropriate **boot** command at the monitor prompt. Depending on the PROM version in your machine, this command takes one of the following forms:

sunmon

> **b le()netbsd**

OpenBoot PROM 1

ok **b le()netbsd**

OpenBoot PROM 2

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 installation scripts.** below.

Installing NetBSD by using a bootable tape

Sun sets their tape drives to SCSI ID 4. We recommend you do the same.

Get the tape images (`installation/tape/tapefile1.gz` and `installation/tape/tapefile2`) and transfer them to a tape. Make sure you use the 'no rewind scsi tape' device. Run the following commands:

```
# gunzip tapefile1.gz
# mt -f /dev/nrst0 rew
# dd if=tapefile1 of=/dev/nrst0 bs=4k conv=osync
# dd if=tapefile2 of=/dev/nrst0 bs=4k conv=osync
```

Now you can transfer the NetBSD 5.1. distribution sets from (`binary/sets`) to the tape by using a series of additional

```
# dd if=<aset>.tgz of=/dev/nrst0 bs=4k conv=osync
```

commands. See also the section **Getting the NetBSD System on to Useful Media.** Be sure to mark the location of these files on the tape; you'll need them during the installation procedure.

Insert the tape into your Sparc's tapestation. Boot the installer by typing the appropriate command at the PROM prompt:

sunmon
> **b st()**

OpenBoot PROM 1
ok **boot st()**

OpenBoot PROM 2
ok **boot tape**

After the initial probe messages you'll be asked to confirm the tape device and tape file number containing the installation tools. Then, proceed to the section **Running the installation scripts.** below.

Running the installation scripts

The first thing that the installation scripts will inquire about is the type of console you're using on your machine. If you're using a Sun frame buffer display, choose the terminal type **sun** (this is presented as the default) .

Note: if your frame buffer device is a cgsix Sbus board, you should change the terminal type to **sun-cgsix** to avoid triggering a bug in the board's firmware code that can garble your screen.

If you're using a terminal device connected to a serial port, choose a terminal type appropriate for that device, e.g. **vt100** or **vt200**.

Next, you will be presented with a choice of two installation methods: a new full-screen **sysinst** program, or the traditional script-based **miniroot** program. The **sysinst** program is easier to use, but the **miniroot** program is more flexible. Both of these installation methods will follow the same set of procedures and will fully install NetBSD on your Sparc.

You will frequently be asked for confirmation before the system proceeds with each phase of the installation process.

Occasionally, you will have to provide a piece of information such as the name of the disk you want to install on, partitioning information, or IP addresses and domain names you want to assign. If your system has more than one disk, you may want to look at the output of the `dmesg`(8) command to see how your disks have been identified by the kernel.

Note: If you repartition a hard drive, be certain that every partition starts on a cylinder boundary for compatibility with the Sparc ROM.

The installation script goes through the following phases:
- determination of the disk on which to install NetBSD
- checking/creation of the partition information on the disk
- creating and mounting the NetBSD file systems
- setup of network configuration
- extraction of the distribution tar files
- installation of bootstrap programs

Now try a reboot. Initially we'd suggest you **boot sd()netbsd -bs**, then try multiuser after that. If you boot single-user the NetBSD incantation to make the root file system writable is

mount -u /dev/sd0a /

Your PROM might have been setup to instruct the boot program to load a file called `vmunix`. On OpenBoot PROM systems you can change it to load `netbsd` instead using the following commands:

OpenBoot PROM 1

ok **setenv boot-from sd(0,0,0)netbsd**

OpenBoot PROM 2

ok **setenv boot-file netbsd**
ok **setenv boot-device disk:a**

Congratulations, you have successfully installed NetBSD 5.1.

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_le0="inet 192.0.2.123 netmask 255.255.255.0"
   ```

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

   ```
   ifconfig_le0="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 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 `sparc/5.1/All` subdir. You can install them with the following commands under `sh`(1):

     ```
     # PKG_PATH=ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/sparc/5.1/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/sparc/5.1/All
     ```

     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/sparc/5.1/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 5.1 (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 5.1 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 5.1 system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in `/dev`. If you've changed the contents of `/dev` by hand, you will need to be careful about this, but if not, you can just cd into `/dev`, and run the command:

sh MAKEDEV all

Finally, you will want to delete old binaries that were part of the version of NetBSD that you upgraded from and have since been removed from the NetBSD distribution.

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 5.1.

If your port uses X.Org and you see messages from the X server indicating that no devices were found, you may need to run **X -configure** and update your existing xorg.conf to use the BusID line from the newly-generated config file.

Dual-head support for PC systems has become broken for many configurations with the update to xorg-server 1.6.x, which has removed the userland PCI configuration mechanism, and needs to rely upon the OS. We hope to correct this for future releases. Workaround: The only workaround is non-trivial and requires programming several PCI BAR registers as they previously were in NetBSD 5.0.

If you are updating to NetBSD 5.1 without the aid of sysinst or postinstall and your port uses X.Org, be sure to remove `/usr/X11R7/lib/X11/xkb/symbols/pc` before extracting the xbase set. In the version of X.Org shipped with 5.0, this was a directory, but in more recent X.Org versions it is a file.

On ports using X.Org, libpixman and libXfont had their major versions bumped. This can be a source of trouble if using binary packages built on 5.0.x with a clean install of 5.1 (upgrades from 5.0.x will include both old and new versions of libpixman and libXfont). Compatibility packages have been prepared and are available from:

ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/sparc/5.0/emulators/compat50-5.1.tgz

(or a convenient mirror). This URL can be passed directly to the `pkg_add`(1) command.

pkg_install now depends on the pkgdb cache for automatic conflict detection. It is recommended to rebuild the cache with

pkg_admin rebuild

`audit-packages.conf`(5) has been superseded by `pkg_install.conf`(5). The default configuration is the same.

Support for `pkg_view`(1) has been retired.

The functionality of `audit-packages`(1) and `download-vulnerability-list`(1) has moved into `pkg_admin`(1). However, wrapper scripts that handle the common use cases are provided.

Issues when running older binaries on NetBSD 5.1

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.x kernel with an older userland.
- Running an older userland inside a chroot'ed environment on a 5.x system.
- Running older statically linked pthread applications.

The 5.x kernel defaults to `0` for `kern.no_sa_support`, which covers the first case. However, please note that a full installation of 5.x (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 5.1 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 5.1 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