NAME

INSTALL — Installation procedure for NetBSD/mvme68k.

CONTENTS

About this Document....2
What is NetBSD?....2
Changes Between The NetBSD 5.0 and 5.0.1 Releases....3
  Security Advisory Fixes....3
  Kernel....3
  Networking....4
  Drivers....4
  Platform specific....4
  Userland....4
  Miscellaneous....5
Changes Between The NetBSD 4.0 and 5.0 Releases....5
  General kernel....5
  Networking....6
  File systems....7
  Security....7
  Drivers....8
  Platforms....12
  Userland....17
  Components removed from NetBSD....21
  Known Problems....21
Features to be removed in a later release....22
The NetBSD Foundation....22
Sources of NetBSD....22
NetBSD 5.0.1 Release Contents....22
  NetBSD/mvme68k subdirectory structure....23
  Binary distribution sets....24
NetBSD/mvme68k System Requirements and Supported Devices....26
  Supported VME147 hardware....27
  Supported VME162/VME172 hardware....27
  Supported VME167/VME177 hardware....27
Getting the NetBSD System on to Useful Media....28
  Creating boot/install tapes....28
  Boot/Install from NFS server....29
  Install/Upgrade from CD-ROM....29
  Install/Upgrade via FTP....30
Preparing your System for NetBSD installation....30
Installing the NetBSD System....32
  Installing from tape....32
  Installing from NFS....36
  Booting the miniroot....38
  Miniroot install program:....39
  Running the sysinst installation program....40
    Introduction....40
    General....40
    Quick install....40
    Booting NetBSD....41

Network configuration.....41
Installation drive selection and parameters.....41
Selecting which sets to install.....41
Partitioning the disk.....41
Preparing your hard disk.....42
Getting the distribution sets.....42
Installation from CD-ROM.....42
Installation using ftp.....42
Installation using NFS.....43
Installation from an unmounted file system.....43
Installation from a local directory.....43
Extracting the distribution sets.....43
Finalizing your installation.....43
Post installation steps.....44
Upgrading a previously-installed NetBSD System.....46
Upgrading using the miniroot 46
Manual upgrade 47
Compatibility Issues With Previous NetBSD Releases.....47
Issues affecting an upgrade from NetBSD 3.x releases 48
Issues affecting an upgrade from NetBSD 4.x releases 49
Using online NetBSD documentation.....49
Administrivia.....50
Thanks go to.....50
We are.....51
Legal Mumbo-Jumbo.....60
The End.....64

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 5.0.1 on the *mvme68k* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on 57 different system architectures (ports) across 15 distinct CPU families, and is being ported to more. The NetBSD 5.0.1 release contains complete binary releases for many different system architectures. (A few ports are not fully supported at this time and are thus not part of the binary distribution. Please see the NetBSD web site at **http://www.NetBSD.org/** for information on them.)

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

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

Changes Between The NetBSD 5.0 and 5.0.1 Releases

The NetBSD 5.0.1 release is the first security/critical update of the NetBSD 5.0 release branch. This represents a selected subset of fixes deemed critical in nature for stability or security reasons.

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 will also appear in future minor releases (i.e., NetBSD 5.1, 5.2, etc.), together with other less-critical fixes and feature enhancements.

The complete list of changes can be found in the CHANGES-5.0.1:
http://ftp.NetBSD.org/pub/NetBSD/NetBSD-5.0.1/CHANGES-5.0.1
file in the top level directory of the NetBSD 5.0.1 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

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

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. Fixes PR kern/39536.
- Fix a crash observed when trying to load a corrupted ELF kernel module.
- Fix PR kern/41566, where writes on the controlling tty were not being awoken from blocks.
- Various fixes for POSIX message queues.
- Fix a possible deadlock in the VFS subsystem.
- Fixes for POSIX advisory locks.
- A number of other stability fixes.

Networking

- Follow exactly the recommendation of draft-ietf-tcpm-tcpsecure-11.txt: Don't check gainst the last ack received, but the expected sequence number. This makes RST handling independent of delayed ACK.
- Fix a panic when trying to disable IPFilter before enabling it. Fixes PR kern/41364.

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.
- `wm`(4):
  - On i82563, FreeBSD's em driver says that the ready bit in the MDIC register may be incorrectly set. Insert delay(200) like the em driver. Fixes PR kern/41014.
  - Add workaround for 82543GC. We need to force speed and duplex on the MAC equal to what the PHY speed and duplex configuration is. Fixes PR kern/36430.
  - Fix many problems and panic on TBI's cards (PR kern/32009).

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.
- amd64: Handle protection faults properly, returning SIGSEGV instead of SIGBUS.
- hp300: Make install.md probe `cd`(4) devices properly.
- 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. Fixes PR port-sparc64/41406.
  - 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. Fixes PR port-sparc64/41302.
- vax: binutils: Allocate relocation section using bfd_zalloc() to ensure no garbage relocations when not all the entries are used. Fixes PR port-vax/39182.

Userland

- Update pkg_install to 20090724.

  Note: 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.

- Update libfetch to 2.23.
- `racoonctl`(8): Adjust ADMINPORTDIR to match that of racoon (/var/run). Fixes PR bin/41376.
- `schedctl`(8): Skip LSIDL and LSZOMB threads when retrieving info.
- `postinstall`(8) now knows about /etc/dhcpcd.conf.

Miscellaneous

- The X.Org s3 driver now works.
- Install the Xvidtune app-defaults file.
- Fixes to Linux compat:
  - In sendmsg(2), do 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. Addresses PR toolchain/24457 and PR bin/41155.

Changes Between The NetBSD 4.0 and 5.0 Releases

The NetBSD 5.0 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many user-land enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize the massive development that went into the NetBSD 5.0 release. The complete list of changes can be found in the CHANGES:

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

and CHANGES-5.0:

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

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

Some highlights include:

General kernel

- Introduced new synchronization primitives and made almost all core kernel subsystems use fine-grained locking, resulting in much greater performance on MP systems. See `mutex`(9), `rwlock`(9), and `condvar`(9).

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

Networking

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

File systems

- Added `wapbl`(4), a preview of metadata journaling for FFS. Contributed by Wasabi Systems.
- Added read-only support for Apple HFS+ file systems. See `mount_hfs`(8).
- Added read-only support for Silicon Graphics EFS file systems. See `mount_efs`(8).
- Added write support for UDF file systems. See `mount_udf`(8).
- Added `mount_psshfs`(8), to support `puffs`(3) sshfs.
- Added `mount_9p`(8), to support 9P file services with `puffs`(3).
- Added `rump_nfs`(8), a userspace NFS client.
- Added `mount_sysctlfs`(8), to support browsing, querying, and modifying the `sysctl`(3) hierarchy.
- Added `refuse`(3) for FUSE compatibility functionality, layered on top of `puffs`(3).
- Added the user kernel file system library, `ukfs`(3), for standalone file system access.
- Added `p2k`(3), a `puffs`(3) to kernel vfs adaption library.
- `puffs`(4): Added support for NFS exporting puffs file servers.
- Added 32 bit UID/GID support for ext2fs.
- Added a `newfs_ext2fs`(8) utility to create Ext2 file systems.
- /proc is now mounted by default. Extended the Linux emulation of /proc to include /proc/stat, /proc/loadavg, and /proc/<pid>/statm. Linux compatibility mode is now the default. Use -o nolinux if Linux compatibility is not desired.
- Write speed to FAT file systems was significantly increased and the 255 head limit was removed.
- Imported FreeBSD's unionfs.
- `fsck_ext2fs`(8), `fsck_ffs`(8), `fsck_lfs`(8): Disable userid to username lookups by default and add -U flag to perform them.

- scan_ffs(8): Added -b option to search a partition for valid alternate superblocks.
- fsck_ffs(8): Added -x/-X options, which allow running fsck_ffs -n on a snapshot of a live filesystem.

Security

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

Drivers

- Audio:
  - Added pad(4), a pseudo-audio device driver for feeding back raw PCM data to userland.
  - Added sgsmix(4), a driver for the SGS 7433 mixer found in some G3 Macs.
  - umidi(4): Added support for Roland UA25, UA4FX, and SonicCell devices.
  - dbri(4): Added support for audio input.
  - auvia(4) now works on big endian machines.
  - azalia(4): AD1984 support was greatly improved. Added support for Realtek ALC662-GR and ALC269 codecs, which are found in EeePCs. Added support for ALC268.
  - auich(4): Fixed a clock accounting problem that prevented detecting sample rates correctly.
- Hardware Monitoring:
  - Imported envsys2, which brings, among other things, the ability to send events to powerd(8). See envsys(4) and sysmon_envsys(9).
  - Added support for hardware monitoring on Sun Ultra Enterprise 450.
  - Added aiboost(4), a driver for the ASUS AI Booster ACPI Hardware monitor.
  - Added smsc(4), a driver for the hardware monitoring portion of the SMSC LPC47B397.
  - Added aps(4), a driver for the IBM Thinkpad Active Protection System.
  - Added coretemp(4), a driver for Intel Core (and newer) on-die thermal sensors.
  - Added finsio(4), a driver for various Fintek Super I/O chips.
  - Added amdtemp(4), a driver for AMD CPU on-die thermal sensors.
  - Added dbcool(4), a driver for Analog Devices dbCool chips including ADT7460, ADT7463, ADT7467, and ADM1030.

  - Added alipm(4), a driver for the Acer Labs M7101 Power Management Controller.
  - Added admtemp(4), a driver for the Analog Devices ADM1021 temperature sensor.
  - viaenv(4): Added support for VIA VT8231.
  - nsclpcsio(4): Added support for the VLM logical device.
  - lm(4): Added an i2c attachment for the LM78 family of temp sensor and fan controllers.
- Networking:
  - Added jme(4), a driver for the JMicron Technologies JME250 Gigabit Ethernet and JME260 Fast Ethernet controllers.
  - Added u3g(4), a driver for many multi-port 3G datacards.
  - Added iwn(4), a driver for the Intel Wireless LAN 4965AGN adapter.
  - Added zyd(4), a driver for ZyDAS ZD1211/ZD1211B USB IEEE 802.11b/g wireless network devices.
  - Added uhmodem(4), a driver for 3G wireless modems including Huawei E220 and E620, E-mobile D01HW and D02HW, and NTT DoCoMo a2502.
  - Added lii(4), a driver for the Atheros L2 Fast Ethernet controller.
  - Added btuart(4), a driver for Bluetooth HCI UART (H4).
  - Firmware images are now shipped for ipw(4), iwi(4), wpi(4), and iwn(4). For ipw(4) and iwi(4), the Intel EULA has to be accepted via sysctl(8).
  - fxp(4): Fixed some TX timeout and RX pool corruption problems. Added a workaround for a hardware ip4csum-tx bug.
  - bge(4): Added support for BCM5786 and BCM5906(M). Fixed fiber card support.
  - nfe(4): Added support for NVIDIA MCP67/73 Ethernet controllers. Fixed wakeup issues on some newer chips. Fixed a problem with receiving jumbo frames.
  - btbc(4): Added support for AnyCom BlueCard devices.
  - rum(4): Added support for MELCO WLI-U2-SG54HP, PLANEX GW-US54Mini2, COREGA CG-WLUSB2GL and K.K. CG-WLUSB2GPX, and ABOCOM WUG2700.
  - makphy(4): Added support for Marvell 88E1116 Gigabit PHY.
  - gem(4): Added support for Sun PCI SX fiber cards and Sun SBus SX fiber cards.
  - wm(4): Added support for the Intel PRO/1000 PT Quad Port Server Adapter. Fixed a bug on receiving a jumbo frame which lead to a panic in sbcompress(). Added support for more ICH9 devices. Fixed an EEPROM-trashing bug on ICH8 and ICH9 chipsets.
  - udav(4): Added support for Shantou ADM8515.
  - brgphy(4): Added support for BCM5708C.
  - re(4): Made hardware vlan(4) insertion/extraction work properly. Added support for the Realtek 8102E/8102EL PCIe 10/100 Ethernet adapters, as well as the 8111C chips that are found on many Intel-based motherboards.
  - sk(4), msk(4): Fixed a lock panic on receiving jumbo packets.

- msk(4): Fixed a Yukon EC Ultra cold power up issue.
- bnx(4): Added support for SerDes controllers.
- vge(4): Added ifconfig down and ALTQ support.

- Storage:
  - Added arcmsr(4), a driver for Areca Technology Corporation SATA RAID controllers.
  - Added siisata(4), a driver for Silicon Image SteelVine SATA-II controllers (SiI3124, SiI3132, and SiI3531).
  - isp(4): Major update, including 4Gb (24XX) card support and new firmware sets.
  - piixide(4): Added support for ICH10.
  - ahcisata(4): Added support for ATAPI devices.
  - svwsata(4): Added support for ServerWorks HT-1000 SATA controller.
  - njata(4): Added support for Workbit CF32A CF adapter.
  - viaide(4): Added support for VIA CX700, CX700M2, NVIDIA MCP67, and MCP73/77 controllers.
  - mfi(4): Added support for LSI SAS1078 and Dell PERC 6 controllers.
  - mpt(4): Fixed performance problems for old revisions of the Symbios 53c1030.
  - cac(4): Added initial bio(4) support; only volume status is handled at this time.
  - siop(4): Added support for the non-PCI NCR 53c720/770 in big-endian mode.
  - twa(4): Added support for 3ware 9650 and 9690, based on contributions from Wasabi Systems.
  - ciss(4): Added bio(4) support.
  - ataraid(4): Added NVIDIA MediaShield, JMicron RAID, and Intel MatrixRAID support. Added support for status reports through bio(4).
  - ixpide(4): Added support for ATI SB700/SB800 controllers.
  - aac(4): Added support for raw I/O mode and >2TB.
  - umass(4): Added support for Sony GPS GPS-CS1. Fixed a panic on device removal.
  - aic(4): Worked around an rbus resource allocation problem so that aic PCMCIA cards work again.
  - Implemented DIOCACHESYNC for RAIDframe, vnd(4), and cgd(4).
  - rccide: Added support for the ServerWorks HT-1000 IDE controller.
  - wd33c93: Enabled Fast SCSI transfers for WD33C93B parts with appropriate input clocks.
- USB:
  - Added uvideo(4), a driver for USB Video Class capture devices, from Patrick Mahoney's Google Summer of Code 2008 project.
  - Added uslsa(4), a driver for CP210x USB-RS232 devices.
  - Added uchcom(4), a driver for WinChipHead CH341/340 and HL-340 USB-Serial adapters.
  - Added uberry(4), a driver to allow RIM BlackBerries to charge from the USB port.

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

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

Platforms

- x86 (i386, amd64):
  - Removed GENERIC.MP and made MULTIPROCESSOR mandatory.
  - Enabled BUFQ_PRIOCSCAN in GENERIC for better responsiveness under heavy disk I/O.
  - Added fatboot, a sub-512 byte bootloader that can be written into the first sector of a FAT16 file system.
  - Added support for enhanced speedstep on Intel Atom CPUs.
  - Added a High Precision Event Timer driver.
  - Merged amd64 and i386 pmap. Large pages are always used if available.
  - Added support for the Intel 82G965 chipset.
  - Added boot.cfg(5) to configure the bootloader.
  - The boot menu now allows disabling ACPI and/or SMP.
  - Added support for ACPI S3 (suspend-to-RAM) on MP systems.
  - Added mbr(8) variants that directly access serial ports.
  - It is now possible on some systems to get to a ddb(4) session on a VGA console if the system crashed while X11 was running.
  - Added detection of errata for AMD Family 10h steppings A and 2.

  - Added checks for erratum 261 on AMD Family 10h stepping 3 processors.
  - `boot`(8): Added support for the multiboot protocol. This allows booting Xen without GRUB.
  - The kernel now recognizes CD-ROMs as booted devices.
  - Added an Intel On Demand Clock Modulation driver.
- i386:
  - Removed support for 80386 level CPUs.
  - Added support for Microsoft Xbox.
  - Added `cmos`(4), a driver for CMOS RAM.
  - Added support for VIA Esther.
  - i386 bootblocks are now able to boot amd64 kernels and support for a.out kernels has been removed.
  - Removed pccons.
  - Added support for sparse kernel core dumps.
  - Removed the MATH_EMULATE option.
  - Added the i386-specific COMPAT_30_PTHREAD option, which restores binary compatibility with netbsd-3 libpthread.
- amd64:
  - Added support for PCI_BUS_FIXUP and PCI_ADDR_FIXUP kernel options.
- acorn32:
  - Added support for Kinetic cards.
  - Fixed up boot32 to work on A7000+.
- algor:
  - Fixed a `delay`(9) issue and now the P5064 kernel works in gxemul.
- alpha:
  - Make dynamic executables work under `compat_osf1`(8) again.
- amiga:
  - Added a `wdc`(4) frontend for the buddha and catweazle Z2 hardware.
- arc:
  - Added support for booting from UFS2 partitions.
  - Added `sysinst`(8) support.
- arm:
  - Added armv6 kernel support.
  - Added arm32 support for kernel crashdumps.
  - Added VFP support.
  - Added AT91 support.

  - Added FA526 support.
  - Added the missing kernel bits that kept `eeprom`(8) from actually changing firmware settings.
- atari:
  - Fixed the sysinst miniroot image and retired the old miniroot.
  - Fixed a hang on loading md_root from floppy during installation.
  - Many pmap bugs were fixed and it was synced with amiga's pmap.
  - Fixed a lost interrupt problem on Falcon `wdc`(4).
  - Renamed the BOOT kernel to SMALL030 and removed BOOTX.
  - Added support for installation from GEMDOS partitions.
  - Fixed an ncr5380 SCSI freezing problem that occurred during boot.
- bebox:
  - Converted to generic powerpc PCI and ISA frameworks.
  - Added support for `genfb`(4).
- cobalt:
  - Added `sysinst`(8) support.
  - Added support for booting from UFS2 partitions.
  - Added support for booting from Ext2fs partitions.
  - Added netboot support for 21041 on Qube 2700.
  - Startup and shutdown messages, as well as a banner and kernel name are now printed on the LCD panel.
  - Added kernel and bootloader support for optional Z85C30 serial on Qube 2700.
  - Fixed a botched `spl`(9) bug which could cause a network freeze on traffic between two network interfaces.
- evbarm:
  - Added support for the Cortina Systems SL3516 evaluation board.
  - Added support for NPE Ethernet found on IXP425 boards.
  - Added a PXA2x0 RTC driver.
- evbmips:
  - Added support for Infineon ADM5120.
- evbppc:
  - Added support for Xilinx Virtex II-Pro/4-FX.
  - Converted PM/PPC to generic powerpc clock, PCI, and interrupt frameworks.
  - Fixed sysinst to refer to the proper WALNUT and OPENBLOCKS266 kernels.
- hp700:

  - Added support for booting from CD-ROM images.
  - Fixed kernel profiling.
  - Kernels are now compiled with -O2.
  - Added `gdb`(1) support.
  - Added bootinfo support.
  - Added support for NCR 53C720.
- hpcmips:
  - Fixed a silent hangup problem on TX3922 machines.
- ibmnws:
  - Converted to generic powerpc clock, PCI, and interrupt frameworks.
- landisk:
  - Added `sysinst`(8) support.
  - Fixed MD `bus_dmamap_load`(9) so that NFS write works with `re`(4).
- mac68k:
  - Added support for more NuBus video cards, including VillageTronic MacPicasso 340.
  - Added support for Creative Systems Inc Hurdler CPI NuBus parallel printer card.
  - Switched `sn`(4) to use the MI SONIC driver.
- macppc:
  - Switched to generic powerpc interrupt, clock, PCI, `bus_dma`(9), `bus_space`(9), SMP, and IPI frameworks.
  - `genfb`(4) is now the default framebuffer.
  - Switched to the new ADB subsystem.
  - Added support for bus speed control found in some Intrepid-based laptops like the 800MHz iBook G4.
  - Converted to shared ofw_autoconf.
  - Converted to shared powerpc major numbers.
  - Added an experimental Apple UniNorth AGP driver.
- mvmeppc:
  - Converted to generic powerpc clock/PCI/interrupt frameworks.
- m68k:
  - Fixed a panic in FPE code caused by 040/060 specific FDADD instructions on 020/030 machines.
  - `gcc`(1): Fix jump table addressing in the M68k codegen.
  - Fixed floating point handling regressions.
- ofppc:

  - Completely rewrote the ofppc port from scratch.
  - Added support for the bplan/Genesi Pegasos II.
  - Added support for the IBM 7044-270.
  - Added support for the IBM 7046-B50.
  - Added install support and booting from RAID.
  - Fixed ofwboot to work on an IBM CHRP-based RS/6000.
  - Added support for SMP.
- pmppc:
  - Removed pmppc as a port and moved it under evbppc.
- powerpc:
  - Added a generic interrupt handler framework.
  - Added generic shared timecounter-based clock routines.
  - Added shared PReP memory maps.
  - Added generic powerpc OpenFirmware code.
  - Added a generic powerpc IPI framework.
  - Added new generic powerpc SMP support framework.
  - Added tuned assembler from IBM for `memcpy`(3), `memmove`(3), and `memcmp`(3).
  - Added booted_kernel and booted_device sysctl.
  - Added the missing kernel bits that kept `eeprom`(8) from actually changing firmware settings.
- prep:
  - Switched to generic powerpc PCI, ISA, interrupt, clock, IPI, and SMP frameworks.
  - Added support for the MTX604 RAVEN based machine. Other RAVEN based machines are also likely to work now.
  - Added support for floppy controller/drive.
- sandpoint:
  - Switched to generic powerpc interrupt, clock, PCI, and ISA frameworks.
  - Added a DHCP/NFS bootloader.
  - Converted to shared powerpc major numbers.
- sgimips:
  - Added support for the SGI O2 PS/2 controller `macekbc`(4), onboard display adapter `crmfb`(4), and audio driver `mavb`(4).
  - Added support for Set Engineering's GIO Fast Ethernet board: `tl`(4).
  - Added support for the SGI Light/Entry/Starter LG1/LG2 framebuffers found in Indigo systems: `light`(4).

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

Userland

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

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

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

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

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

Components removed from NetBSD

In NetBSD 5.0, the following software components were removed from the system. Some were not useful anymore, or their utility did not justify the maintenance overhead. Others were not working properly and there was a lack of interest in fixing them.

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

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

Known Problems

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

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

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

Statically linked binaries using pthreads are currently broken.

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

Features to be removed in a later release

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

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

  NetBSD 5.0.1 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 5.0.1 Release Contents

The root directory of the NetBSD 5.0.1 release is organized as follows:

`.../NetBSD-5.0.1/`

`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.prev` Changes in previous NetBSD releases.

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

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

`source/` Source distribution sets; see below.

In addition to the files and directories listed above, there is one directory per architecture, for each of the architectures for which NetBSD 5.0.1 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.
82 MB gzipped, 422 MB uncompressed

sharesrc This set contains the "share" sources, which include the sources for the man pages not associated with any particular program; the sources for the typesettable document set; the dictionaries; and more.
7 MB gzipped, 29 MB uncompressed

src This set contains all of the base NetBSD 5.0.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
58 MB gzipped, 284 MB uncompressed

syssrc This set contains the sources to the NetBSD 5.0.1 kernel for all architectures as well as the `config`(1) utility.
36 MB gzipped, 180 MB uncompressed

xsrc This set contains the sources to the X Window System.
133 MB gzipped, 699 MB uncompressed

All the above source sets are located in the `source/sets` subdirectory of the distribution tree.

The source sets are distributed as compressed tar files. Except for the **pkgsrc** set, which is traditionally unpacked into `/usr/pkgsrc`, all sets may be unpacked into `/usr/src` with the command:

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

In each of the source distribution set directories, there are files which contain the checksums of the files in the directory:

`BSDSUM` Historic BSD checksums for the various files in that directory, in the format produced by the command:
cksum -o 1 *file*.

`CKSUM` POSIX checksums for the various files in that directory, in the format produced by the command:
cksum *file*.

`MD5` MD5 digests for the various files in that directory, in the format produced by the command:
cksum -a MD5 *file*.

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

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

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

NetBSD/mvme68k subdirectory structure

The mvme68k-specific portion of the NetBSD 5.0.1 release is found in the `mvme68k` subdirectory of the distribution: `.../NetBSD-5.0.1/mvme68k/`. 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-VME147.gz`
A kernel for MVME147 boards.
`netbsd-VME162.gz`
A kernel for MVME162 boards.
`netbsd-VME167.gz`
A kernel for MVME167 boards.
`netbsd-VME172.gz`
A kernel for MVME172 boards.
`netbsd-VME177.gz`
A kernel for MVME177 boards.
`sets/` mvme68k binary distribution sets; see below.

`installation/`
`miniroot/` mvme68k miniroot file system image; see below.
`netboot/` Two programs needed to boot mvme68k kernels over the network; see below.
`tapeimage/`
Tape boot programs, and a RAMDISK kernel; see below.

Binary distribution sets

The NetBSD mvme68k binary distribution sets contain the binaries which comprise the NetBSD 5.0.1 release for mvme68k. The binary distribution sets can be found in the `mvme68k/binary/sets` subdirectory of the NetBSD 5.0.1 distribution tree, and are as follows:

base The NetBSD 5.0.1 mvme68k **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.
25 MB gzipped, 75 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.
35 MB gzipped, 130 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/mvme68k 5.0.1 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
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.
6 MB gzipped, 19 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
11 MB gzipped, 43 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.
4 MB gzipped, 10 MB uncompressed

The mvme68k 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.

The following are included in the `mvme68k/installation` directory:

`miniroot/`

Note: The **`sysinst`** method of installation is the preferred method for installing NetBSD/mvme68k. *The Traditional method of installation is deprecated and will be removed in a future release.* Some other important differences between the two installation methods are:

- The Installer and Mkfs utilities are not used if the installation is done using the **`sysinst`** method via booting an Installation Kernel.
- Mkfs creates a 4.3BSD "old" format file system. This is the only file system format understood by the Installer. The **`sysinst`** process creates "new" file systems which can't be processed by Mkfs or the Installer.
- The Installer (and probably Mkfs) is known to have problems with BSD file systems that are larger than 1 GB or extend beyond the 1 GB physical limit. Keep this in mind if doing a Traditional method installation with these utilities.
- The Booter is known to have problems booting from a NetBSD file system if the inode of the kernel file is too large. For large disks it is advisable to have a small root partition and one or more usr-type partitions for other files. This ensures the inode of the kernel in the root file system will be small.
- The Booter is designed to work with 4.3BSD "old" file systems, but there are enough similarities between the "old" and "new" file system formats in the first few hundred blocks that the Booter can usually boot a Kernel from a 4.3BSD new file system. If you choose to use "new" file system formats, it is advisable to use a small root partition and one or more usr-type partitions for other files. This is the default layout used by **`sysinst`**.

`miniroot.gz` A copy of the miniroot file system. This file system image is copied into the swap partition of the disk which you intend to boot NetBSD from. Normally, this step will be performed manually from the ramdisk boot environment.

`netboot/`

`sboot` An MVME147 bootstrap program in Motorola S-Record format. This is required when you wish to boot an MVME147 over the network since the ROM has no built-in network support. Instructions for getting this program into memory are discussed later.

`netboot` A standalone 2nd stage bootstrap program loaded over the network via TFTP. This is responsible for fetching and starting the NetBSD mvme68k kernel from an NFS server.

`tapeimage/`

`stboot` This file contains a boot sector for Motorola MVME boards. It must be the first file written to a tape in order to make the tape bootable.

`bootst` This file contains a bootstrap program which knows how to load the NetBSD mvme68k ramdisk image from tape. This must be the second file written to the tape.

`netbsd-RAMDISK.gz` This is the NetBSD mvme68k ramdisk image. It contains a `GENERIC` kernel and a built in RAMDISK with just enough tools to partition a disk, dump the miniroot kernel to it and make the disk bootable. This must be the third file written to the tape.

Note: Each directory in the mvme68k binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/mvme68k System Requirements and Supported Devices

NetBSD/mvme68k 5.0.1 runs on Motorola MVME147 , MVME162 , MVME167 , MVME172 , and MVME177 Single Board Computers.

The minimal configuration requires 8 MB of RAM and ~200 MB of disk space. To install the entire system requires much more disk space. To run X (clients only) or compile the system, more RAM is recommended. Good performance requires 16 MB of RAM, or 32 MB when running the X Window System.

Note that you can install NetBSD 5.0.1 on a system with only 4 MB of onboard RAM, but you will need to use a VMEbus RAM card with at least another 4 MB to augment the onboard memory in order to actually install the system.

A swap partition of 2-3∗RAM is recommended.

Note that the NetBSD/mvme68k installation procedure uses a *miniroot* image which is placed into the swap area of the disk. The swap partition must be large enough to hold this miniroot image (> 7.5 MB).

Supported VME147 hardware

- Serial ports (RS232)
  - built-in console
  - ttyZ1
  - ttyZ2
  - ttyZ3
- Parallel port
  - On-board centronics style printer port
- Network interfaces:
  - On-board Lance Ethernet (le)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board wd33c93 SCSI bus interface chip (async only)
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

Supported VME162/VME172 hardware

- Serial ports (RS232)
  - built-in console
  - ttyC1
  - ttyC2
  - ttyC3
- Network interfaces:
  - On-board Intel Ethernet (ie)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board ncr53c710 SCSI I/O processor chip
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

Supported VME167/VME177 hardware

- Serial ports (RS232)
  - built-in console
  - ttyZ1
  - ttyZ2
  - ttyZ3
- Parallel port
  - On-board centronics style printer port
- Network interfaces:
  - On-board Intel Ethernet (ie)
- SCSI
  - Most SCSI disks, tapes, CD-ROMs, etc
  - On-board ncr53c710 SCSI I/O Processor chip
- Miscellaneous:
  - Battery-backed real-time clock
  - VMEbus RAM cards
  - Any VMEbus interface cards with Machine Independent drivers

If it's not on the above lists, there is no support for it in this release.

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- Tape
- NFS
- CD-ROM
- FTP

Note that installing on a 'bare' machine requires either a bootable tape drive or an ethernet to a compatible NFS server. MVME147 may also need to be booted over an RS232 connection.

The procedure for transferring the distribution sets onto installation media depends on the type of media. Instructions for each type of media are given below.

In order to create installation media, you will need all the files and subdirectories in these two directories:

`.../NetBSD-5.0.1/mvme68k/installation`

`.../NetBSD-5.0.1/mvme68k/binary`

Creating boot/install tapes

Installing from tape is the simplest method of all. This method uses two tapes, one containing a bootable ramdisk and miniroot, the other containing the installation sets.

The boot tape is created as follows:

```
# cd .../NetBSD-5.0.1/mvme68k/installation
# set T = /dev/nrst0
# mt -f $T rewind
# dd if=tapeimage/stboot of=$T obs=8k conv=osync
# dd if=tapeimage/bootst of=$T obs=8k conv=osync
# gzip -dc tapeimage/netbsd-RAMDISK.gz | dd of=$T obs=8k conv=osync
# gzip -dc miniroot/miniroot.fs.gz  | dd of=$T obs=8k conv=osync
# mt -f $T rewind
```

The installation set tape is created as follows:

```
# cd .../NetBSD-5.0.1/mvme68k/binary/sets
# set T = /dev/nrst0
# mt -f $T rewind
# for f in base etc comp games man misc text; do
    gzip -d < $f.tgz | dd of=$T bs=8k conv=osync
# done
# mt -f $T rewind
```

If the tape does not work as expected, you may need to explicitly set the EOF mark at the end of each tape segment. Consult the tape-related manual pages on the system where the tapes are created for more details.

Boot/Install from NFS server

If your machine has a disk and network connection, but no tape drive, it may be convenient for you to install NetBSD over the network. This involves temporarily booting your machine over NFS, just long enough so you can initialize its disk. This method requires that you have access to an NFS server on your network so you can configure it to support diskless boot for your machine. Configuring the NFS server is normally a task for a system administrator, and is not trivial.

If you are using a NetBSD system as the boot-server, have a look at the `diskless`(8) manual page for guidelines on how to proceed with this. If the server runs another operating system, consult the documentation that came with it (i.e. `add_client`(8) on SunOS).

Booting an MVME147 from ethernet is not possible without first downloading a small bootstrap program (sboot) via RS232. See the section entitled **Installing from NFS** for details on how to accomplish this.

sboot expects to be able to download a second stage bootstrap program via TFTP after having acquired its IP address through RARP It will look for a filename derived from the machine's IP address expressed in hexadecimal, with an extension of '`.147`'. For example, an MVME147 with IP address 130.115.144.11 will make an TFTP request for `8273900B.147`. Normally, this file is just a symbolic link to the NetBSD/mvme68k `netboot` program, which should be located in a place where the TFTP daemon can find it (remember, many TFTP daemons run in a chroot'ed environment). The netboot program may be found in

the install directory of this distribution.

The MVME162 , MVME167 , MVME172 , and MVME177 boot ROMs have code builtin to boot over ethernet from a TFTP server. You should configure it to download the same `netboot` program as is used for MVME147.

The netboot program will query a bootparamd server to find the NFS server address and path name for its root, and then load a kernel from that location. The server should have a copy of the netbsd-RAMDISK kernel in the root area for your client (no other files are needed in the client root, although it might be a convenient place to put the uncompressed miniroot image) and `/etc/bootparams` on the server should have an entry for your client and its root directory. Note that you should rename the `netbsd-RAMDISK` kernel to just `netbsd` in the client's root directory before trying to netboot the client.

The client will need access to the miniroot image, which can be provided using NFS or remote shell. If using NFS, miniroot.fs.gz should be expanded on the server, because doing so from the RAMDISK shell is not so easy. The unzipped miniroot takes about 7.5 MB of space.

If you will be installing NetBSD on several clients, it may be useful to know that you can use a single NFS root for all the clients as long as they only use the netbsd-RAMDISK kernel. There will be no conflict between clients because the RAM-disk kernel will not use the NFS root. No swap file is needed; the RAM-disk kernel does not use that either.

Install/Upgrade from CD-ROM

This method requires that you boot from another device (i.e. tape or network, as described above). You may need to make a boot tape on another machine using the files provided on the CD-ROM. Once you have booted netbsd-RAMDISK (the RAMDISK kernel) and loaded the miniroot, you can load any of the distribution sets directly from the CD-ROM. The **`install`** program in the miniroot automates the work required to mount the CD-ROM and extract the files.

Install/Upgrade via FTP

This method requires that you boot from another device (i.e. tape or network, as described above). You may need to make a boot tape on another machine using the files in .../install (which you get via FTP). Once you have booted netbsd-RAMDISK (the RAM-disk kernel) and loaded the miniroot, you can load any of the distribution sets over the net using FTP. The **`install`** program in the miniroot automates the work required to configure the network interface and transfer the files.

This method, of course, requires network access to an FTP server. This might be a local system, or it might even be ftp.NetBSD.org itself. If you wish to use ftp.NetBSD.org as your FTP file server, you may want to keep the following information handy:

```
No IP Address: ftp.NetBSD.org
Login: anonymous
Password: <your e-mail address>
Server path: /pub/NetBSD/NetBSD-5.0.1/mvme68k/binary
```

Note: If you're not using a nameserver during installation, you might find 204.152.184.75 handy; it's the IP address of ftp.NetBSD.org as of October, 2000.

Preparing your System for NetBSD installation

mvme68k machines usually need little or no preparation before installing NetBSD, other than the usual, well advised precaution of *backing up all data* on any attached storage devices.

The exception to the above is that MVME162 , MVME167 , MVME172 and MVME177 boards require a jumper to be removed or DIP switch changed before NetBSD can be installed. On MVME162-LX and MVME172-LX pins 1-2 of jumper J11 *must* be removed. On MVME162-P2/P4 and MVME172-P2/P4 switch S4, position 8 *must* be set to OFF. On MVME167 and MVME177 pins 1-2 of jumper J1 *must* be removed.

Once you've made any necessary jumper changes, the following instructions should make your machine "NetBSD Ready".

Power-up your MVME147 board. You should have the *bug* prompt:

```
COLD Start

Onboard  RAM start = $00000000,  stop = $007FFFFF

147-Bug>
```

Or, if you have an MVME162/172 or MVME167/177 board (the following boot message is from MVME167; the others are similar):

```
MVME167 Debugger/Diagnostics Release Version 2.3 - 02/25/94
COLD Start

Local Memory Found =02000000 (&33554432)

MPU Clock Speed =33Mhz

167-Bug>
```

Make sure the RAM size looks ok (if you've got an 8 MB MVME147 or a 32 MB MVME167 you should have the same value as we do). Also make sure the clock is ticking:

```
1xx-Bug>time
Sunday 12/21/31 16:25:14
1xx-Bug>time
Sunday 12/21/31 16:25:15
1xx-Bug>
```

Note that NetBSD bases its year at 1968, and adds the year offset in the system's real-time clock to get the current year. So the 31 here equates to 1999. You may have to adjust your clock using the **set** command to comply with NetBSD 's requirements. Don't worry if the 'Day of the week' is not correct, as NetBSD doesn't use it. Motorola has acknowledged a year 2000 bug in some versions of the MVME147 whereby the day of the week doesn't get set correctly by the 147Bug PROM.
NetBSD does not affect

Also make sure that your board's ethernet address is initialised to the correct value. You'll find the address on a label on the inside of the MVME147's front panel, and on the VMEbus P2 connector of the other board types. On the MVME147, enter the last five digits of the address using the **lsad** command. On the MVME162/172 and MVME167/177, you should use the **cnfg** command.

The NetBSD kernel reads the first two long words of the onboard NVRAM to determine the starting and ending address of any VMEbus RAM that should be used by the system. You should verify that this area is set properly for your configuration.

If you have no VMEbus RAM boards, the values should be set to zero (0).

For an MVME162, MVME167, MVME172 or MVME177 board, at the 1xx-Bug> prompt:

```
1xx-Bug>mm fffc0000 ;1
fffc0000: xxxxxxxx?0
fffc0004: xxxxxxxx?0
fffc0008: xxxxxxxx?.
1xx-Bug>
```

For an MVME147 board, at the 147Bug prompt:

```
147Bug>mm fffe0764 ;1
fffe0764: xxxxxxxx?0
fffe0768: xxxxxxxx?0
fffe076c: xxxxxxxx?.
```

If you do have VMEbus RAM available and want NetBSD to use it, the first long word should be set to the starting address of this RAM and the second long word should be set to the ending address.

If you have more than one VMEbus RAM board installed, the starting and ending addresses must be contiguous from one board to the next. Also note that, for various reasons beyond the scope of this document, VMEbus RAM should be configured in A32 address space.

To install successfully to a local SCSI disk, you need to ensure that the system is aware of what targets are connected to the SCSI bus. This can be done by issuing the following command:

```
1xx-Bug> iot;t
```

At this point, Bug will scan for any attached SCSI devices. After a short delay, a list of SCSI devices will be displayed. 147Bug will ask if LUNs should be assigned from SCSI ids, to which you should answer Y. You should also answer Y when asked if the information is to be saved to NVRAM. 16xBug does not prompt for this information.

The following installation instructions will assume that your target SCSI disk drive appears at SCSI-ID 0. If you have a tape drive, the instructions assume is is configured for SCSI-ID 5. When the RAMDISK root boots, NetBSD will refer to these devices as `sd0` and `rst0` respectively. You may wish to note these down; you'll be using them a lot. :-)

Installing the NetBSD System

Installing NetBSD is a relatively complex process, but if you have this document in hand it should not be too difficult.

There are several ways to install NetBSD onto your disk. If your MVME147 machine has a tape drive the easiest way is **Installing from tape** (details below). All other machines can be installed easily over the network from a suitable NFS server. See **Installing from NFS** for details. Otherwise, if you have another mvme68k machine running NetBSD you can initialize the disk on that machine and then move the disk.

Installing from tape

Create the NetBSD/mvme68k 5.0.1 boot tape as described in the section entitled **Preparing a boot tape**. Then, with the tape in the drive, type the following at the Bug prompt:

```
147-Bug> bo 5
```

```
16x-Bug> bo 0,50
```

As mentioned earlier, this assumes your tape is jumpered for SCSI-ID 5.

As the tape loads (which may take 20 to 30 seconds), you will see a series of status messages. It may be useful if you can capture these messages to a file, or a scrollable xterm window. In particular, you should make a note of the lines which describe the geometry of the SCSI disks detected by NetBSD. They are of the form:

```
sd0 at scsibus0 targ 0 lun 0: <Maker, Disk, Foo> SCSI1 0/direct fixed
sd0: 800 MB, 800 cyl, 16 head, 128 sec, 512 bytes/sect x 1638400 sectors
```

The information of most interest is the number of sectors; for the fictitious disk above, it's 1638400. You will need this number when you come to create a disklabel for that drive.

Here is an example of an MVME147 system booting from tape:

```
RAM address from VMEbus = $00000000

Booting from: VME147, Controller 5, Device 0
Loading: Operating System

Volume: NBSD

IPL loaded at:  $003F0000
>> BSD MVME147 tapeboot [$Revision: 1.20.2.1 $]
578616+422344+55540+[46032+51284]=0x11a6e4
Start @ 0x8000 ...
Copyright (c) 1996, 1997, 1998, 1999, 2000
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5 (RAMDISK) #1: Sun Oct 29 16:19:04 GMT 2000
    steve@fatbob:/usr/src/sys/arch/mvme68k/compile/RAMDISK
Motorola MVME-147S: 25MHz MC68030 CPU+MMU, MC68882 FPU
real mem  = 7237632
avail mem = 6381568
using 88 buffers containing 360448 bytes of memory
mainbus0 (root)
pcc0 at mainbus0: Peripheral Channel Controller, rev 0, vecbase 0x40
clock0 at pcc0 offset 0x0 ipl 5: Mostek MK48T02, 2048 bytes of NVRAM
  .
  .
```

Note: The exact text of the messages will vary depending on which mvme68k variant you're using.

Finally, you will see the following "welcome" message:

```
        Welcome to the NetBSD/mvme68k RAMDISK root!

This environment is designed to do only four things:
  1:  Partition your disk (use the command:  edlabel /dev/rsd0c)
  2:  Copy a miniroot image into the swap partition  (/dev/rsd0b)
  3:  Make that partition bootable (using 'installboot')
  4:  Reboot (using the swap partition, i.e. /dev/sd0b).

Copying the miniroot can be done several ways, allowing the source
of the miniroot image to be on any of these:
    boot tape,  NFS server, TFTP server, rsh server

The easiest is loading from tape, which is done as follows:
        mt -f /dev/nrst0 rewind
        mt -f /dev/nrst0 fsf 3
        dd if=/dev/nrst0 of=/dev/rsd0b bs=8k conv=sync
(For help with other methods, please see the install notes.)

To reboot using the swap partition after running installboot, first
```

```
use
halt,
then at the Bug monitor prompt use a command like:
               1x7Bug> bo 0,,b:

To view this message again, type:  cat /.welcome
#
```

You must now create a disklabel on the disk you wish to use for the root file system (/). This will usually be sd0. The disklabel is used by NetBSD to identify the starting block and size of each partition on the disk.

Partitions are named sd0a, sd0b, sd0c, etc, up to sd0h. The mvme68k port of NetBSD makes some assumptions about the first three partitions on a boot disk:

sd0a The root file system (/).
sd0b The swap partition.
sd0c The whole disk. Also known as the *raw partition*.

The *raw partition* is special; NetBSD is able to use it even if the disk has no label. You should never create a file system on the Raw Partition, even on a non-boot disk.

It is good practice to put /usr on a different partition than / (root, AKA sd0a). So, the first available partition for /usr is sd0d. Refer to the section entitled **NetBSD System Requirements and Supported Devices** for information on the recommended sizes of the / (root), /usr and swap partitions.

You are not required to define any partitions beyond sd0d, but if you have a large disk drive, you might want to create several other partitions for file systems such as /home or /usr/src. Note that at this time you are only required to partition the root/boot disk; you will get the opportunity to partition any other disks in your system from the main **miniroot** installation program.

To create the disklabel and partitions, use the edlabel program, passing it the name of the Raw Partition of your root/boot disk.

```
# edlabel /dev/rsd0c
edlabel menu:
print - display the current disk label
modify - prompt for changes to the label
write - write the new label to disk
quit - terminate program
edlabel>
```

The program shows what commands it recognizes; **print**, **modify**, **write**, and **quit**. It will accept the first letter of a command if you don't feel like typing each one in full.

To start creating the basic partitions, you should enter **m** (modify) at the edlabel prompt, then enter the letter corresponding to the first partition, **a**.

```
edlabel> m

modify subcommands:
@: modify disk parameters
a-h: modify partition
s: standardize geometry
q: quit this subcommand
edlabel/modify> a
a (root) 0 (0/00/00) 0 (0/00/00) unused
start as <blkno> or <cyls/trks/sects>: 0
```

```
length as <nblks> or <cyls/trks/sects>: 65536
type: 4.2BSD
edlabel/modify>
```

When you enter the start and length of a partition, you can use either blocks or cylinder/track/sector notation. If this is the first time you've partitioned a disk for NetBSD, it's probably easiest to use block notation. The above example creates partition 'a', starting at block zero and with a size of 65536 blocks. Note that the usual size of a block is 512 bytes, so this creates a 32 MB partition.

The type of the partition should be 4.2BSD, otherwise you won't be able to create a file system on it.

Next, create a swap partition (b). Note that the minimum size of this swap partition should be 8 MB, otherwise you won't be able to use a miniroot to complete the NetBSD installation!

```
edlabel/modify> b
b (swap) 0 (0/00/00) 0 (0/00/00) unused
start as <blkno> or <cyls/trks/sects>: 65536
length as <nblks> or <cyls/trks/sects>: 32768
type: swap
edlabel/modify>
```

Here, we specify a value for start such that the swap partition follows immediately after partition 'a', i.e. 65536. The length of the swap partition should be a multiple of the amount of RAM you have in your system. Here, we've chosen 32768, or 16 MB. The next available block on the drive is thus 65536 + 32768. We will use this to create partition 'd' for our /usr file system. (Note that for a busy system, or a system with more than 8 MB of RAM, you'll be better off with a 32 or 64 MB swap partition.)

```
edlabel/modify> d
d (user) 0 (0/00/00) 0 (0/00/00) unused
start as <blkno> or <cyls/trks/sects>: 98304
length as <nblks> or <cyls/trks/sects>: 1540096
type: 4.2BSD
edlabel/modify> q
edlabel>
```

As you can see, we've chosen to assign the remainder of the disk to /usr. Since there are 1638400 sectors on the example disk (did you remember to note down the number of sectors on your disk during boot?), and partition **d** starts at sector 98304, a simple bit of arithmetic (1638400 - 98304) gives **d** a size of 1540096.

Note that the above partition sizes are just guidelines. If your disk is large enough, you should resize the partitions appropriately and perhaps also create a /var partition as well.

You now need to write this new disklabel, together with the partition details you've just entered, to disk. You might also try the '**p**' command to view the partitions. Once written, you can quit back to the shell using '**q**'.

```
edlabel> p

        type_num: 4
        sub_type: 0
       type_name: SCSI disk
       pack_name: fictitious
    bytes/sector: 512
   sectors/track: 128
 tracks/cylinder: 16
       cylinders: 800
sectors/cylinder: 2048
partition      start         (c/t/s)      nblks         (c/t/s)  type
```

```
 a (root)          0       (0/00/00)      65536      (32/00/00)  4.2BSD
 b (swap)      65536      (32/00/00)      32768      (48/00/00)  swap
 c (disk)          0       (0/00/00)    1638400     (800/00/00)  unused
 d (user)      98304      (48/00/00)    1540096     (752/00/00)  4.2BSD

edlabel> w
edlabel> q
#
```

Now that your disk's partitioned, you need to get the proper installation miniroot image onto it. The miniroot image is designed to be copied into the swap partition of your disk. This is a safe place which won't be overwritten by the installation procedure. From the shell prompt, use the following commands to copy the miniroot image from tape to swap (b).

```
# mt -f /dev/nrst0 rewind
# mt -f /dev/nrst0 fsf 3
# dd if=/dev/nrst0 of=/dev/rsd0b bs=8k conv=osync
```

The disk and the miniroot must now be made bootable using the `installboot`(8) command. To do this, issue the following commands:

```
# mount /dev/sd0b /mnt
# installboot /mnt/usr/mdec/bootsd /bootxx /dev/rsd0b
# umount /dev/sd0b
```

You can now shutdown the system.

```
# halt
signal 15
syncing disks... done
unmounting / (root_device)...
halted

147-Bug> reset
Reset Local SCSI Bus [Y,N] N? y
Automatic reset of known SCSI Buses on RESET [Y,N] Y?
Cold/Warm Reset flag [C,W] = C?
Execute Soft Reset [Y,N] N? y
```

Resetting the other types of MVME boards are very similar. You should now reboot from that just installed miniroot. See the section entitled **Booting the miniroot** for details.

Installing from NFS

Before you can install from NFS, you must have already configured your NFS server to support your machine as a bootable client. Instructions for configuring the server are found in the section entitled **Getting the NetBSD System onto Useful Media** above.

To get started on the MVME147, you need to download `sboot` into RAM (you will find **`sboot`** in the `install` directory of the mvme68k distribution). You can either do that through the console line or through a 2nd serial connection. For example, an MVME147 connected to a sun4/110 and accessed via `tip`(1) can be loaded as follows:

```
lo 0
~Ccat sboot
go 4000
```

Which will look like this:

```
147-Bug> lo 0
~CLocal command? cat sboot
```

away for 11 seconds

```
!

147-Bug> g 4000
Effective address: 00004000

sboot: serial line bootstrap program (end = 6018)

>>>
```

Now, if you want to do it through serial line 1, then connect serial line one to a machine. At the `147-Bug>` prompt type **`tm 1`** You should then login to the machine it is connected to. Then press `CONTROL-A` to escape to Bug. Do **`lo 1;x=cat sboot`** ... then when that is done you can reconnect **`tm 1`** and logout. Then do **`go 4000`** and you've got the `>>>` prompt of **`sboot`**.

Once you've got the `>prompt, you can boot the RAMDISK` kernel from the server:

```
>>> b

le0: ethernet address: 8:0:3e:20:cb:87
My ip address is: 192.168.1.4
Server ip address is: 192.168.1.1
4800
Download was a success!
```

See below for the next step in booting MVME147.

The MVME162, MVME167, MVME172 and MVME177 boards are able to download netboot directly using TFTP. To enable this, you must first configure the networking parameters on the board as described in the section entitled "Preparing your System for NetBSD Installation. On a properly configured MVME162/172 or MVME167/177, all you need to type is:

```
1xx-Bug> nbo
```

For all board types, the boot messages are very similar:

```
Start @ 0x8000 ...
>> BSD MVME147 netboot (via sboot) [$Revision: 1.20.2.1 $]
device: le0 attached to 08:00:3e:20:cb:87
boot: client IP address: 192.168.1.4
boot: client name: soapy
root addr=192.168.1.1 path=/export/soapy
578616+422344+55540+[46032+51284]=0x11a6e4
Start @ 0x8000 ...
Copyright (c) 1996, 1997, 1998, 1999, 2000
    The NetBSD Foundation, Inc.  All rights reserved.
Copyright (c) 1982, 1986, 1989, 1991, 1993
    The Regents of the University of California.  All rights reserved.

NetBSD 1.5 (RAMDISK) #1: Sun Oct 29 16:19:04 GMT 2000
    steve@fatbob:/usr/src/sys/arch/mvme68k/compile/RAMDISK
Motorola MVME-147S: 25MHz MC68030 CPU+MMU, MC68882 FPU
real mem  = 7237632
```

```
avail mem = 6381568
using 88 buffers containing 360448 bytes of memory
mainbus0 (root)
pcc0 at mainbus0: Peripheral Channel Controller, rev 0, vecbase 0x40
clock0 at pcc0 offset 0x0 ipl 5: Mostek MK48T02, 2048 bytes of NVRAM
  .
  .
```

After the boot program loads the `RAMDISK` kernel, you should see the welcome screen as shown in the "tape boot" section above.

You now need to create a disklabel with partition information on the SCSI disk on which you intend to create your root file system (`/`). Follow the instructions in the previous section entitled **Installing from tape** to do this. (But stop short of the part which describes how to copy the miniroot from tape.)

You must now configure the network interface before you can access the NFS server containing the miniroot image. For example the command:

```
# ifconfig le0 inet 192.168.1.4 up
```

will bring up the MVME147 network interface `le0` with that address. The command:

```
# ifconfig ie0 inet 192.168.1.4 up
```

will bring up the MVME162/172 or MVME167/177 network interface `ie0` with that address. The next step is to copy the miniroot from your server. This can be done using either NFS or remote shell. (In the examples that follow, the server has IP address 192.168.1.1) You may then need to add a default route if the server is on a different subnet:

```
# route add default 192.168.1.2 1
```

You can look at the route table using:

```
# route show
```

Now mount the NFS file system containing the miniroot image:

```
# mount -r 192.168.1.1:/export/soapy /mnt
```

The procedure is simpler if you have space for an expanded (not compressed) copy of the miniroot image. In that case:

```
# dd if=/mnt/miniroot of=/dev/rsd0b bs=8k
```

Otherwise, you will need to use **gzcat** to expand the miniroot image while copying.

```
# gzcat miniroot.fs.gz | dd of=/dev/rsd0b obs=8k conv=osync
```

You must now make the disk bootable. Refer to the previous section on installing from tape, where it describes how to run **installboot**. This is immediately following the part which explains how to copy the miniroot from tape.

Booting the miniroot

Assuming the miniroot is installed on partition 'b' of the disk with SCSI-ID 0, then the boot command is:

```
1xx-Bug> bo 0,,b:
```

The command line parameters above are:

0 controller (usually zero)

,, bug argument separators

b: tell the bootstrap code to boot from partition b

You should see a bunch of boot messages, followed by messages from the miniroot kernel just as you did when the RAMDISK kernel booted.

You will then be prompted to enter the root device. Since the miniroot was booted from the swap partition, you should enter sd0b. You will then be asked for the swap device and file system type. Just press RETURN twice to accept the defaults. When asked to enter a terminal type, either accept the default, or use whatever the TERM environment variable is set to in the shell of your host system:

```
boot device: sd0
root device (default sd0a): sd0b
dump device (default sd0b): (return)
file system (default generic): (return)
root on sd0b dumps on sd0b
mountroot: trying ffs...
root file system type: ffs
init: copying out path '/sbin/init' 11
erase ^H, werase ^W, kill ^U, intr ^C
Terminal type? [vt100] return
```

Congratulations! The system should now be running the miniroot installation program.

Miniroot install program:

The miniroot's install program is very simple to use. It will guide you through the entire process, and is well automated.

All recent releases of NetBSD provide the option of using **sysinst** instead of the old installation and upgrade shell scripts. At this time, **sysinst** is not capable of installing distribution sets from tape. In such a case, you should fallback to the shell script installation or upgrade procedure. In all other cases, **sysinst** should be used. Note that as soon as **sysinst** can deal with tapes, the shell script tools will be removed.

The miniroot's install program (both **sysinst** and the shell script version) will:

- Allow you to place disklabels on additional disks. The disk we are installing on should already have been partitioned using the RAMDISK kernel.

  Note that partition sizes and offsets are expressed in sectors. When you fill out the disklabel, you will need to specify partition types and file system parameters. If you're unsure what these values should be, use the following defaults:

  ```
  fstype: 4.2BSD
  fsize: 1024
  bsize: 4096
  cpg: 16
  ```

  If the partition will be a swap partition, use the following:

  ```
  fstype: swap
  fsize: 0 (or blank)
  bsize: 0 (or blank)
  cpg: 0 (or blank)
  ```

Note that partition 'c' is special; it covers then entire disk and should not be assigned to a file system.

The number of partitions is currently fixed at 8.

- Create file systems on target partitions.
- Allow you to set up your system's network configuration. Remember to specify host names without the domain name appended to the end. For example use `foo` instead of `foo.bar.org`. If, during the process of configuring the network interfaces, you make a mistake, you will be able to re-configure that interface by simply selecting it for configuration again.
- Mount target file systems. You will be given the opportunity to manually edit the resulting `/etc/fstab`.
- Extract binary sets from the media of your choice.
- Copy configuration information gathered during the installation process to your root file system (`/`).
- Make device nodes in your root file system under `/dev`.
- Copy a new kernel onto your root partition (`/`).
- Install a new boot block.
- Check your file systems for integrity.

First-time installation on a system through a method other than the installation program is possible, but strongly discouraged.

Running the sysinst installation program

1. *Introduction*

   Using **sysinst**, installing NetBSD is a relatively easy process. Still, you should read this document and have it in hand when doing the installation process. This document tries to be a good guide to the installation, and as such, covers many details for the sake of completeness. Do not let this discourage you; the install program is not hard to use.

2. *General*

   The following is a walk-through of the steps you will take while getting NetBSD installed on your hard disk. **sysinst** is a menu driven installation system that allows for some freedom in doing the installation. Sometimes, questions will be asked and in many cases the default answer will be displayed in brackets ("[ ]") after the question. If you wish to stop the installation, you may press `CONTROL-C` at any time, but if you do, you'll have to begin the installation process again from scratch by running the `/sysinst` program from the command prompt. It is not necessary to reboot.

3. *Quick install*

   First, let's describe a quick install. The other sections of this document go into the installation procedure in more detail, but you may find that you do not need this. If you want detailed instructions, skip to the next section. This section describes a basic installation, using a CD-ROM install as an example.

   - What you need.
     - The distribution sets (in this example, they are on CD).
     - One 1.44 MB 3.5" floppy.
     - A CD-ROM drive (SCSI), a hard disk and a minimum of 4 MB of memory installed.

  - The hard disk should have at least 100 + *n* megabytes of space free, where *n* is the number of megabytes of main memory in your system. If you wish to install the X Window System as well, you will need at least 215 MB more.

- The Quick Installation
  - Insert the first boot floppy you just created and boot the computer. After language selection, the main menu will be displayed.

```
.*****************************************.
* NetBSD-5.0.1 Install System             *
*                                         *
*>a: Install NetBSD to hard disk          *
* b: Upgrade NetBSD on a hard disk        *
* c: Re-install sets or install additional sets *
* d: Reboot the computer                  *
* e: Utility menu                         *
* x: Exit Install System                  *
.*****************************************.
```

  - If you wish, you can configure some network settings immediately by choosing the **Utility menu** and then **Configure network**. It isn't actually required at this point, but it may be more convenient. Go back to the main menu.
  - Choose **install**.
  - You will be guided through some steps regarding the setup of your disk, and the selection of distributed components to install. When in doubt, refer to the rest of this document for details.
  - After your disk has been prepared, choose **CD-ROM** as the medium. The default values for the path and device should be ok.
  - After all the files have been unpacked, go back to the main menu and select **reboot**, after you have removed the bootfloppy from the drive.
  - NetBSD will now boot. If you haven't already done so in **`sysinst`**, you should log in as `root` and set a password for that account. You are also advised to edit `/etc/rc.conf` to match your needs.
  - Your installation is now complete.

4. *Booting NetBSD*

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

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

   It will take a while to load the kernel from the floppy, probably around a minute or so, then, the kernel boot messages will be displayed. This may take a little while also, as NetBSD will be probing your system to discover which hardware devices are installed. The most important thing to know is that `wd0` is NetBSD's name for your first IDE disk, `wd1` the second, etc. `sd0` is your first SCSI disk, `sd1` the second, etc.

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

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

5. *Network configuration*

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

6. *Installation drive selection and parameters*

   To start the installation, select **Install NetBSD to hard disk** from the main menu.

   The first thing is to identify the disk on which you want to install NetBSD. **`sysinst`** will report a list of disks it finds and ask you for your selection. You should see disk names like `sd0` or `sd1`.

7. *Selecting which sets to install*

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

8. *Partitioning the disk*

   - Which portion of the disk to use.

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

9. *Editing the NetBSD disklabel*

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

   After you have chosen your partitions and their sizes (or if you opted to use the existing partitions), you will be presented with the layout of the NetBSD disklabel and given one more chance to change it. For each partition, you can set the type, offset and size, block and fragment size, and the mount point. The type that NetBSD uses for normal file storage is called **4.2BSD**. A swap partition has a special type called **swap**. Some partitions in the disklabel have a fixed purpose.

`a`	Root partition (`/`)
`b`	Swap partition.
`c`	The NetBSD portion of the disk.
`d`–`h`	Available for other use. Traditionally, `d` is the partition mounted on `/usr`, but this is historical practice and not a fixed value.

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

10. *Preparing your hard disk*

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

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

11. *Getting the distribution sets*

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

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

12. *Installation from CD-ROM*

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

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

13. *Installation using ftp*

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

    You will also be asked to specify the host that you want to transfer the sets from, the directory on that host, the account name and password used to log into that host using ftp, and optionally a proxy server to use. If you did not set up DNS, you will need to specify an IP address instead of a hostname for the ftp server.

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

14. *Installation using NFS*

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

    You will also be asked to specify the host that you want to transfer the sets from and the directory on that host that the files are in. This directory should be mountable by the machine you are installing on, i.e., correctly exported to your machine.

    If you did not set up DNS, you will need to specify an IP address instead of a hostname for the NFS server.

15. *Installation from an unmounted file system*

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

16. *Installation from a local directory*

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

17. *Extracting the distribution sets*

    Before extraction begins, you can elect to watch the files being extracted; the name of each file that is extracted will be shown. This can slow down the installation process considerably on machines with slow graphics consoles or serial consoles. Alternatively, you can choose to see a progress bar. This is the preferred option as it shows progress without significantly slowing down the installation process.

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

18. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 5.0.1. You can now reboot the machine and boot NetBSD from hard disk.

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state. 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 **vt220** (or whatever is appropriate for your terminal type) and press `RETURN`. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

   At this point, you need to configure at least one file in the `/etc` directory. You will need to mount your root file system read/write with:

   ```
   # /sbin/mount -u -w /
   ```

   Change to the `/etc` directory and take a look at the `/etc/rc.conf` file. Modify it to your tastes, making sure that you set `rc_configured=YES` so that your changes will be enabled and a multi-user boot can proceed. Default values for the various programs can be found in `/etc/defaults/rc.conf`, where some in-line documentation may be found. More complete documentation can be found in `rc.conf`(5).

   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. 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.

3. Adding accounts

   Use the useradd(8) command to add accounts to your system. Do not edit /etc/passwd directly! See vipw(8) and pwd_mkdb(8) if you want to edit the password database.

4. The X Window System

   If you 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

5. Installing third party packages

   If you wish to install any of the software freely available for UNIX-like systems you are strongly advised to first check the NetBSD package system, 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 mvme68k/5.0.1/All subdir. You can install them with the following commands under sh(1):

     ```
     # PKG_PATH=ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/mvme68k/5.0.1/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/mvme68k/5.0.1
```

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/mvme68k/5.0.1/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 5.0.1 (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.

6. Misc

   - Edit /etc/mail/aliases to forward root mail to the right place. Don't forget to run newaliases(1) afterwards.
   - The /etc/postfix/main.cf file will almost definitely need to be adjusted. If you prefer a different MTA, then install it using 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

It is possible to easily upgrade your existing NetBSD/mvme68k system using the upgrade program in the miniroot or by manually performing the same steps as the miniroot upgrade program.

Upgrading using the miniroot

If you wish to upgrade your system by this method, simply select the **upgrade** option once the miniroot has booted. The upgrade program with then guide you through the procedure. The upgrade program will:

1. Enable the network based on your system's current network configuration.
2. Mount your existing file systems.
3. Extract binary sets from the media of your choice.
4. Fixup your system's existing `/etc/fstab`, changing the occurrences of **ufs** to **ffs** and let you edit the resulting file.
5. Make new device nodes in your root file system under `/dev`.
6. Don't forget to extract the **kern** set from the distribution.

   Note: The existing kernel *will not* be backed up; doing so would be pointless, since older kernels may not be capable of running NetBSD 5.0.1 executables.
7. Install a new boot block.
8. Check your file systems for integrity.
9. You'll have to reboot your system manually

Manual upgrade

While using the miniroot's upgrade program is the preferred method of upgrading your system, it is possible to upgrade your system manually. To do this, follow the following procedure:

1. Place *at least* the **base** binary set in a file system accessible to the target machine. A local file system is preferred, since the NFS subsystem in the NetBSD 5.0.1 kernel may be incompatible with your old binaries.
2. Back up your pre-existing kernel and copy the 5.0.1 kernel into your root partition (`/`).
3. Reboot with the 5.0.1 kernel into single-user mode.
4. Check all file systems:

   ```
   # /sbin/fsck -pf
   ```
5. Mount all local file systems:

   ```
   # /sbin/mount -a -t nonfs
   ```
6. If you keep `/usr` or `/usr/share` on an NFS server, you will want to mount those file systems as well. To do this, you will need to enable the network:

   ```
   # sh /etc/rc.d/network start
   ```
7. Make sure you are in the root file system (`/`) and extract the **base** binary set:

   ```
   # cd /
   # pax -zrvpe -f /path/to/base.tgz
   ```
8. Install a new boot block:

   ```
   # cd /usr/mdec
   # cp bootsd /.bootsd
   # ./installboot /.bootsd bootxx  root-disk
   ```

E.g.: *root-disk* could be `/dev/rsd0a`.

9. Sync the file systems:

```
# sync
```

10. At this point you may extract any other binary sets you may have placed on local file systems, or you may wish to extract additional sets at a later time. To extract these sets, use the following commands:

```
# cd /
# pax -zrvpe -f path_to_set
```

Note: You *should not* extract the **etc** set if upgrading. Instead, you should extract that set into another area and carefully merge the changes by hand.

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 5.0.1.

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.

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

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

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

Note that **sysinst** will automatically invoke

```
postinstall fix
```

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

Issues affecting an upgrade from NetBSD 3.x releases

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

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

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

Issues fixed by *postinstall*:

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

The following issues need to be resolved manually:

- The users '_proxy', '_rwhod', and '_sdpd' and the groups '_proxy', '_rwhod' and '_sdpd' need to be created and the user 'uucp' needs to be updated.
- A number of things were removed in the NetBSD 4.0 release, including: the evbsh5 port, the Fortran 77 compiler (g77), NETCCITT, NETNS, Sendmail, Sushi, UUCP, and Vinum. If you were using any of these, please see the "Components removed from NetBSD" at
  http://www.NetBSD.org/releases/formal-4/NetBSD-4.0.html#removals
- The replacement of Sendmail by Postfix can be handled automatically by *postinstall* but it is not done by default. If you want to transition to Postfix, the command

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

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

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

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

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

Issues affecting an upgrade from NetBSD 4.x releases

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

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

Issues fixed by *postinstall*:

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

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

The following issues need to be resolved manually:

- The users '_httpd' and '_timedc' and the groups '_httpd' and '_timedc' need to be created.
- Unprivileged use of the mount(8) command now requires the *nosuid* and *nodev* options to be explicitly specified. Previously, these options were automatically enforced even if they were not explicitly specified.
- A number of things have been removed from the NetBSD 5.0.1 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 5.0.1 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 David Jones.
This product includes software developed by Steve C. Woodford.

The End