NAME

INSTALL — Installation procedure for NetBSD/hp300.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 6.1 on the *hp300* 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 6.1 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 6.0 and 6.1 Releases

The NetBSD 6.1 release provides numerous significant functional enhancements, including support for new devices, integration of hundreds of bug fixes, 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.

The complete list of changes can be found in the CHANGES-6.1:
http://ftp.NetBSD.org/pub/NetBSD/NetBSD-6.1/CHANGES-6.1
file in the top level directory of the NetBSD 6.1 release tree. An abbreviated list is as follows:

Security Advisory Fixes

- NetBSD-SA2013-001: *,:*
  http://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2013-001.txt.asc
  Kernel panic triggered from userland
- NetBSD-SA2013-002: *,:*
  http://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2013-002.txt.asc
  kqueue related kernel panic triggered from userland
- NetBSD-SA2013-003: *,:*
  http://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2013-003.txt.asc
  RNG Bug May Result in Weak Cryptographic Keys

- NetBSD-SA2013-004: *,:*
  http://ftp.netbsd.org/pub/NetBSD/security/advisories/NetBSD-SA2013-004.txt.asc
  Vulnerabilities in grep

Security Fixes

- expat: Fix CVE-2012-1147, CVE-2012-1148 and CVE-2012-0876.
- BIND: Address CVE-2012-5688: Named could die on specific queries with dns64 enabled.
- Fixed a crash which could happen with sysctl security.curtain=1 set. (PR#47598): **http://gnats.NetBSD.org/47598**
- Address CVE-2012-5667: Multiple integer overflows in GNU grep.
- Several fixes to entropy handling in the kernel.
- Prevent ktrace from being able to crash the kernel.

General kernel

- posix_spawn(): Fix processes with attributes.
- Resolve races between vget() and vrele() resulting in vget() returning dead vnodes.
- Prevent crash when unsupported fd's are used with kevent.
- Fix a bug where kmem_alloc() could be called from interrupt context.
- WAPBL: coalesce writes to the journal to speed up wapbl_flush() on raid5 by a factor of 3 to 4.

Networking

- Fix "atomic fragments" for IPv6.
- ipf: Fix alignment issues in ipmon. (PR#47101): **http://gnats.NetBSD.org/47101**
- npf: handle delayed checksums in the network stack. (PR#47235): **http://gnats.NetBSD.org/47235**
- Fixed some locking issues in `ipf`(4)
- `npf`(3): Many fixes and updates. Config syntax has changed somewhat, please check your config files.
- Fixed sendto(2) issue with IPv6 UDP datagrams. (PR#47408): **http://gnats.NetBSD.org/47408**

File systems

- smbfs: Make smbfs actually work on big-endian ports.
- smbfs: various fixes. (PR#25070): **http://gnats.NetBSD.org/25070**
- Allow mounting ext2fs and msdosfs in 32-bit compat mode.
- v7fs: fix off-by-one error.

Drivers

- `ciss`(4): don't try to handle sensors if there aren't any.
- `isp`(4): Fix FC-AL devices on QLogic ISP2100. (PR#47302): **http://gnats.NetBSD.org/47302**
- `mfi`(4): Add support for LSI Thunderbolt (SAS2208) controllers.
- `mfi`(4): Add support for running the MegaCLI Linux binary to manage the controller.

- `puc`(4): Additional device support.
- `bge`(4): Add support for Apple's Thunderbolt to Gigabit Ethernet adapter. (PR#46961): **http://gnats.NetBSD.org/46961**
- `mvgbe`(4): fix a number of issues
- `nfe`(4): Avoid bounce buffering and improve performance.
- `wm`(4): Fix handling of 82578D and 82579V chips. Various other fixes.
- `fss`(4): Don't crash when running multiple dump -X. (PR#47514): **http://gnats.NetBSD.org/47514**
- `fss`(4): Fixed panic. (PR#47020): **http://gnats.NetBSD.org/47020**

Platforms

- alpha:
  - Fixed longjmp/setjmp on alpha for statically linked programs such as those in /rescue.
- amiga:
  - New `ed`(4) device driver for Hydra and ASDG Zorro2 bus network cards.
- arm:
  - Fix gcc bugid 51408 for arm. (PR#46972): **http://gnats.NetBSD.org/46972**
  - Fixed a bit check in the Kirkwood IRQ code.
  - Fixed incorrect counts for 'vmstat -e' on Kirkwood.
  - Make bigendian ARM get the MBR partition and offset correct. (PR#47081): **http://gnats.NetBSD.org/47081**
  - Fixed bug where IRQ masks of GPIO were set incorrectly on Marvell SoCs.
  - Fixed static linking on ARM platforms. (PR#47139): **http://gnats.NetBSD.org/47139**
  - Properly block IRQs >= 32. (PR#45371): **http://gnats.NetBSD.org/45371**
  - Greatly improved Raspberry Pi support - USB and onboard Ethernet now work.
  - Enabled DCache Streaming Switch and Write Allocate for Sheeva CPUs.
  - Added support for the watchdog timer in some Marvell SoCs.
  - Fix for non-MBR disks on ARM. (PR#47463): **http://gnats.NetBSD.org/47463**
- mips:
  - MIPS ports have switched to gdb 7.3.1. GDB6 is removed.
- vax:
  - Reverted MAXPARTITION bump to 16, lowering it to 12 instead, to fix bootblocks and installboot for VAX 780 and other machines booting via VMB.EXE.
- x68k:
  - Update bootloader: cdboot, netboot support and bugfixes.
  - Add floppy format support.

- x86 (i386, amd64):
  - Work around a possible gcc bug generating bad assembler code. (PR#45673): **http://gnats.NetBSD.org/45673**
  - Disable C1E on AMD K8 CPUs, to prevent freeze during boot.
  - Fix running BSDi and NetBSD <= 1.0 binaries. (PR#41341): **http://gnats.NetBSD.org/41341**
  - Add support in ld.elf_so for relocation type R_386_TLS_TPOFF32. (PR#47061): **http://gnats.NetBSD.org/47061**
  - Fix some ACPI panics. (PR#47016): **http://gnats.NetBSD.org/47016**
  - The amd64 port now supports dtrace. (Not compiled in by default)
- xen:
  - Prevent a memory corruption issue that locks up a Xen DomU, and can potentially cause file system corruption. (PR#47056,: **http://gnats.NetBSD.org/47056**
    PR#47057: **http://gnats.NetBSD.org/47057**)
  - Fix: Xen Dom0 NetBSD kernel could crash by adding duplicate xenwatches.

Userland fixes

- Library Updates
  - `pthread`(3): libpthread minor number was bumped for the addition of pthread_cond_setclock(). Threaded binaries built on NetBSD 6.1 will not function on NetBSD 6.0.
  - `pthread`(3): libpthread can be dlopen()ed again.
  - `terminfo`(3): Various fixes to libterminfo.
  - libc: fix memory corruption in rpc. (PR#13082): **http://gnats.NetBSD.org/13082**
  - libutil: Fixed a bug in `parsedate`(3) where it would treat all input as if it were in the local time zone.
- Update to tzdata2013c.
- cdb: don't refuse to open databases without entries or keys.
- Address graphics corruption in recent Cairo, manifested most commonly by certain rendered text sections appearing as solid rectangular blocks of color.
- Added `flock`(1), inspired by the Linux program of the same name.
- Updated Postfix to version 2.8.13.
- `wsconsctl`(8): add support for touchscreen calibration. (PR#45872): **http://gnats.NetBSD.org/45872**
- openssl: Fixed a multiple-free bug.

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 6.1 Release Contents

The root directory of the NetBSD 6.1 release is organized as follows:

`.../NetBSD-6.1/`

`CHANGES`	Changes between the 5.0 and 6.0 releases.
`CHANGES-6.0`	Changes between the initial 6.0 branch and final release of 6.0.
`CHANGES-6.1`	Changes between the final release of 6.0 and final release of 6.1.
`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 6.1 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.

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.

src This set contains all of the base NetBSD 6.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 6.1 kernel for all architectures as well as the `config`(1) utility.

xsrc This set contains the sources to the X Window System.

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

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

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

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

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

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

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

NetBSD/hp300 subdirectory structure

The hp300-specific portion of the NetBSD 6.1 release is found in the `hp300` subdirectory of the distribution: `.../NetBSD-6.1/hp300/`. 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-RAMDISK.gz`
    A gzipped NetBSD kernel containing code for all of the hardware supported in this release with an embedded ramdisk-based installer. This is the same kernel that is present on the miniroot filesystem, but uses a newer more user-friendly installation program.
  - `netbsd-RAMDISK.symbols.gz`
    Symbols for `netbsd-RAMDISK.gz`.
- `sets/` hp300 binary distribution sets; see below.

`installation/`
- `miniroot/` hp300 miniroot file system image; see below.
- `misc/` Miscellaneous hp300 installation utilities; see installation section below.
  - `HP-IB.geometry` A file containing geometry for some HB-IB disk drives.
  - `SYS_INST.gz` A gzipped copy of the SYS_INST miniroot installation program. This is only necessary if you can't use the RAMDISK based installer.
  - `SYS_UBOOT.gz` A gzipped copy of the universal boot block. Supports Network, tape and disk booting. This is useful if you are installing a diskless NetBSD/hp300 system.

Binary distribution sets

The NetBSD hp300 binary distribution sets contain the binaries which comprise the NetBSD 6.1 release for hp300. The binary distribution sets can be found in the `hp300/binary/sets` subdirectory of the NetBSD 6.1 distribution tree, and are as follows:

base The NetBSD 6.1 hp300 **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.

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.

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.

games This set includes the games and their manual pages.

kern-GENERIC
This set contains a NetBSD/hp300 6.1 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.

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.

misc This set includes the system dictionaries, the typesettable document set, and other files from `/usr/share`.

modules This set includes kernel modules to add functionality to a running system.

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

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

xbase The basic files needed for a complete X client environment. This does not include the X servers.

xcomp The extra libraries and include files needed to compile X source code.

xfont Fonts needed by the X server and by X clients.

xetc Configuration files for X which could be locally modified.

xserver The X server.

The hp300 binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

The instructions given for extracting the source sets work equally well for the binary sets, but it is worth noting that if you use that method, the filenames stored in the sets are relative and therefore the files are extracted *below the current directory*. Therefore, if you want to extract the binaries into your system, i.e. replace the system binaries with them, you have to run the **`tar -xzpf`** command from the root directory (`/`) of your system.

Note: Each directory in the hp300 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/hp300 System Requirements and Supported Devices

NetBSD/hp300 6.1 will run on most HP 9000/300- and 400-series machines. The smallest amount of RAM that has been tested is 4 MB. If you wish to run X, more RAM is recommended.

Supported hardware

- CPUs
  - 318 (16 MHz 68020, with built-in monochrome framebuffer, no expansion)
  - 319 (16 MHz 68020, with built-in 6 bit color framebuffer, no expansion)
  - 320 (16 MHz 68020, 16 KB L2 cache, up to 7.5 MB RAM)
    Requires Human Interface board
  - 330 (16 MHz 68020, 4 MB on motherboard, up to 8 MB RAM)
    Requires Human Interface board
  - 340 (16 MHz 68030, up to 16 MB RAM)

- 345 (50 MHz 68030 with 32 KB L2 cache, up to 128 MB RAM, built-in SCSI)
- 350 (25 MHz 68020, 32 KB L2 cache, up to 48 MB RAM)
  Requires Human Interface board
- 360 (25 MHz 68030, with 4 MB RAM built-in, up to 48 MB RAM)
  Requires System Interface board
- 362 (25 MHz 68030, up to 16 MB RAM, built-in SCSI)
  Framebuffer is unsupported, you must use a serial console
- 370 (33 MHz 68030, 64 KB L2 cache, up to 48 MB RAM)
  Requires System Interface board
- 375 (50 MHz 68030 with 32 KB L2 cache, up to 128 MB RAM, built-in SCSI)
- 380 (25 MHz 68040, up to 128 MB RAM, built-in SCSI)
- 382 (25 MHz 68040, up to 32 MB RAM, built-in SCSI)
  Framebuffer is unsupported, you must use a serial console
- 385 (33 MHz 68040, up to 128 MB RAM, built-in SCSI)
- 400s, 400t, 400dl (50 MHz 68030 with 32 KB L2 cache, built-in SCSI, up to 128 MB RAM)
- 425s, 425t, 425dl (25 MHz 68040 with built-in SCSI, up to 128 MB RAM)
- 425e (25 MHz 68040, built-in SCSI)
  The BootROM does not support serial console and NetBSD/hp300 *does not support the 425e framebuffer. When the* NetBSD *kernel begins to boot it will default to using the serial port for console*
- 433s, 433t, 433dl (33 MHz 68040 with built-in SCSI, up to 128 MB RAM)

• HP-IB devices
- *rd*; CS80 disks: 2200, 2203, 7912, 7914, 7933, 7936, 7937, 7945, 7946, 7957, 7958, and 7959
- *rd*; CS80 floppy disks: 9122, 9134 (possibly others)
  Requires use of HP-UX LIF utilities via HP-UX emulation
- *ct*; CS80 Low-density 16 track cartridge (67 MB): 7912, 7914, 7946, and 9144
  Cannot read or write to 32 track tapes
- *ct*; CS80 High-density 32 track cartridge (134 MB): 9145
  Cannot write to 16 track tapes (read only)
- *mt*; CS80 Half-inch tape: 7974A, 7978A/B, 7979A, 7980A, and 7980XC.

  Note: You should connect HP-IB tape drives and printers to the slow HP-IB interface and hard drives to the fast HP-IB interface (if present).

• SCSI devices
- *sd*; SCSI hard drives
- *cd*; SCSI CD-ROM drives
- *sd*; SCSI Magneto-optical drives
- *st*; SCSI tape drives: HP 35450A (DDS-1 DAT), Exabyte EXB-8200 (8mm), Archive (QIC-24), Archive Viper (QIC-60), Archive Viper (QIC-150), Archive Python 25501 (DAT), and Archive Python 28849 (DAT)
- *ch*; SCSI autochangers

• Serial interfaces
- *com*; Built-in single serial port on System Interface board and Human Interface board
- *com*; Built-in Apollo 4-port on Series 400 workstations
- *dcm*; 98638 8-port (DIO-II board, appears to kernel as two 98642 devices)
- *dcm*; 98642 4-port (DIO-I board)
- *com*; 98626, 98644 built-in or add-on (DIO-I board) single serial port

Note: See the FAQ for more detailed specs and information on configuring:
http://www.NetBSD.org/ports/hp300/faq.html#serialinfo

- Network interfaces
  - *le*; 98643 built-in and add-on (DIO-I board) ethernet cards.
- Graphics Devices
  - 98544 monochrome Topcat (1024x768, 1 bit, DIO-II)
  - 98545A color Topcat (1024x768, 4 bits, DIO-II)
  - 98547 color Topcat (1024x768, 6 bits)
  - 98548A monochrome Catseye (1024x768, 1 bit)
  - 98549A color Catseye (1024x768, 6 bits)
  - 98550A Catseye Hi-Res Color (a.k.a. CH) (1280x1024 @ 60 Hz, 8 bits, DIO-II)
  - 98700 and 98710 Gatorbox (1280x1024 @ 60 Hz, 4 or 8 bits)
    Requires 98287 (DIO-I board) to connect to workstation
  - 98720 and 98721 color Renaissance SRX (1280x1024 @ 60 Hz, 8, 12 or 24 bits)
    Requires 98724 (DIO-I board) or 98725 (DIO-II board) to connect to workstation
  - 98730 and 98731 DaVinci TurboSRX (1280x1024 @ 60 Hz, 8, 16, or 24 bits)
    Requires 98726A (DIO-II board) to connect to workstation
  - A1096A monochrome Hyperion (1280x1024, 1 bit)
  - A1416A Kathmandu (a.k.a. Color VRX) (1280x1024 @ 60 hz, 8 bits, DIO-II)
- HP-HIL devices
  - Keyboards
  - Two and three button mice
  - 46094 Quadrature Port (supports normal serial mice)
  - Graphics tablets
  - Dial boxes
  - ID modules
- Miscellaneous boards/interfaces
  - 98265A/98652 SCSI interface (DIO-I board, daughtercard for 98562, or built-in)
  - 98561 Human Interface board (DIO-I board with serial port, HP-HIL, and HP-IB)
  - 98562 System Interface board (DIO-II board with serial port, HP-HIL, HP-IB, DMA, and ethernet, has optional fast HP-IB or SCSI daughterboard)
  - 98620 DMA card (DIO-I board, for use with 98561)
  - 98624 HP-IB interface (DIO-I board or built-in)
  - 98625A and 98625B 'fast' HP-IB interface (DIO-I board or daughtercard for 98562)

Each serial interface has its own quirks, and some of them use non-standard pins. The FAQ describes how to configure and connect serial consoles to hp300 systems.

http://www.NetBSD.org/ports/hp300/faq.html#serialconsole

When you try booting from a system with a framebuffer that is not supported by NetBSD/hp300, the screen will turn black, and it will try using the serial port for the console.

Unsupported hardware

- CPUs
  - 310 (10 MHz 68010, with built-in monochrome framebuffer, rs232 (25 pin), hil, and slow hpib)
  - 332 (16? MHz 68030, with no floating point unit)
- Graphics Devices
  - SGC support (for 425e built-in EVRX framebuffer)
  - SGC support (for some rare configurations of 4XXt and 4XXdl workstations)

  - 98702 TurboVRX DIO-II graphics device
  - 98705 Tigershark PersonalVRX DIO-II graphics device
  - internal video on models 362/382

- Miscellaneous boards/interfaces
  - domain keyboard and mouse, on Series 400 workstations. Needs wscons
  - parallel port, on 345, 362, 375, 380, 382, 382, and Series 400 workstations
  - EISA support, on Series 400 workstations
  - Token Ring cards (a.k.a. ATR)
  - VME bus adapter, on Series 300 workstations
  - 98628A single port dcl (DIO-I board)
  - 98625C fast HP-IB option on models 345 and 375
  - internal SCSI floppy drive on 362, 382, and 425e

Getting the NetBSD System on to Useful Media

You should wait to decide where to put the NetBSD distribution sets until you have figured out how you are going to boot your system. Refer back to this section after you have done so.

Installation is supported from several media types, including:

- CD-ROM / DVD
- FTP
- Remote NFS partition
- Tape
- Existing NetBSD partitions, if performing an upgrade

The steps necessary to prepare the distribution sets for installation depend upon which installation medium you choose. The steps for the various media are outlined below.

CD-ROM / DVD Find out where the distribution set files are on the CD-ROM or DVD. Likely locations are `binary/sets` and `hp300/binary/sets`.

Proceed to the instructions on installation.

FTP The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. If you don't have DHCP available on your network, you will need to know the numeric IP address of that site, and, if it's not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

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

NFS Place the NetBSD distribution sets you wish to install into a directory on an NFS server, and make that directory mountable by the machine on which you are installing or upgrading NetBSD. This will probably require modifying the `/etc/exports` file on the NFS server and resetting its mount daemon (mountd). (Both of these actions will probably require superuser privileges on the server.)

You need to know the numeric IP address of the NFS server, and, if you don't have DHCP available on your network and the server is not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

Once the NFS server is set up properly and you have the information mentioned above, you can proceed to the next step in the installation or upgrade process. If you're installing NetBSD from scratch, go to the section on preparing your hard disk, below. If you're upgrading an existing installation, go directly to the section on upgrading.

Tape To install NetBSD from a tape, you need to make a tape that contains the distribution set files, in 'tar' format.

If you're making the tape on a UNIX-like system, the easiest way to do so is probably something like:

```
# tar -cf tape_device dist_directories
```

where *tape_device* is the name of the tape device that describes the tape drive you're using; possibly /dev/rst0, or something similar, but it will vary from system to system. (If you can't figure it out, ask your system administrator.) In the above example, *dist_directories* are the distribution sets' directories, for the distribution sets you wish to place on the tape. For instance, to put the **kern-GENERIC, base,** and **etc** distributions on tape (in order to do the absolute minimum installation to a new disk), you would do the following:

```
# cd .../NetBSD-6.1
# cd hp300/binary
# tar -cf tape_device kern-GENERIC base etc
```

Note: You still need to fill in *tape_device* in the example.

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

Preparing your System for NetBSD installation

There are two installation tools available. The traditional miniroot installer is script-based and may be netbooted or may be dumped to a disk and run locally. The ramdisk kernel with the **sysinst** installation utility is more flexible, but can only be netbooted and has not been extensively tested.

There are several possible installation configurations described in this document. Other configurations are possible, but less common. If you are unable to install based on the information in this document, post a message to port-hp300@NetBSD.org asking for help. The configurations described in this document are as follows:

- hp300 netboots SYS_UBOOT from a NetBSD server running rbootd(8), or other server platforms (Linux, FreeBSD, SunOS, Solaris, HP-UX etc.) with YAMAMORI Takenori's sun-rbootd package and then runs the miniroot installation tools or a purely diskless installation from a server on the same subnet (you must have root access). For more information, refer to the NetBSD Diskless HOW-TO at
  http://www.NetBSD.org/docs/network/netboot/
  and
  http://www.NetBSD.org/docs/network/netboot/rbootd/
- hp300 loads SYS_UBOOT from a local disk, tape, or floppy and then runs the miniroot installation tools or a purely diskless installation from a server as described above. You will need HP-UX on your hp300 if it does not have a SCSI interface or a floppy drive.

- hp300 loads `SYS_INST` from a local disk. You partition the drive and then run the miniroot installation tools from that drive. You will need HP-UX on your hp300 if it does not have a SCSI interface.

The preferred method is to use another NetBSD server to netboot the hp300 client. This procedure will not work on the handful of models which are incapable of netbooting. In particular, the 320, 350, 330, 318, and 319 might not have a recent enough BootROM. The BootROM revision is printed when your workstation is first powered on (or rebooted). Revision B or later will definitely work. BootROMs with numeric revisions such as 1.1 (on a 400s) will netboot without any problems. You can netboot from any built-in or add-on ethernet board on a model with a supported BootROM.

If you have access to a NetBSD/hp300 system, it is *much* easier to simply upgrade than to install from scratch. Skip down to the section on **Upgrading a previously-installed NetBSD System**

Formatting your hard drives

NetBSD/hp300 does not have the capability to perform a low-level format of hard drives. SCSI disks can be formatted on any platform with SCSI support and then used on NetBSD/hp300. HP-IB disks can only be formatted by the HP-UX `mediainit(1)` command. You may need to first create the device nodes for your disk, as HP-UX was never very good about populating `/dev/rdsk` properly.

```
# mknod /dev/dsk/IDs0 b 0 0xSCID00
# mknod /dev/rdsk/IDs0 c 4 0xSCID00
# mediainit -v /dev/rdsk/IDs0
```

ID is the HP-IB address (a.k.a. slave) of the disk in hexadecimal. This is usually between `00` and `07`, but possibly up to `1F` (31 decimal) .

SC is the Select Code of the disk controller. This is usually `07` for slow (i.e. built-in) HP-IB or `0E` (14 decimal) for SCSI or fast HP-IB.

Designing your disk's partition table

This step can sometimes be a real pain, especially when using `SYS_INST`. It's best to calculate it ahead of time.

If you are installing to an HP-IB disk, you will need information about your disk's geometry, based on 512-byte sectors. The file `installation/misc/HP-IB.geometry` in the distribution has geometry information for several HP-IB disks, but may be incomplete. Geometry may be calculated from an HP-UX `/etc/disktab` entry, but note that HP-UX geometry is based on 1024 byte sectors, while NetBSD's is based on 512 byte sectors. You should have all partitions start on cylinder boundaries.

If you are installing to a SCSI disk, you don't need to worry about the details of the geometry. Just create a disklabel based on the total number of sectors available on the disk.

A quick note about partitions: Since the target disk will become the boot disk for your new NetBSD/hp300 installation, you will need to treat the '`a`' and '`c`' partitions in a special manner. Due to the size of the NetBSD/hp300 boot program (it spills into the area after the disklabel), it is necessary to offset the beginning of the '`a`' partition. For HP-IB disks, it is best to offset it by one cylinder from the beginning of the disk. For SCSI disks, just offset it by 100 KB (200 sectors). Later, the '`c`' partition will be marked with the type '`boot`' and may not be used for a file system. (For those unfamiliar with historic BSD partition conventions, the '`c`' partition is defined as the 'entire disk', or the 'raw partition'.)

Note: You will need at least a 6 MB swap partition (although 2-3*RAM is recommended) if you are unable to netboot the installer, as the miniroot is temporarily placed in this partition.

Here is an example disklabel from a 7959B HP-IB hard drive:

```
# /dev/rrd0a:
type: HP-IB
disk: rd7959B
```

```
label:
flags:
bytes/sector: 512
sectors/track: 42
tracks/cylinder: 9
sectors/cylinder: 378
cylinders: 1572
total sectors: 594216
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset    fstype   [fsize bsize   cpg]
  a:    37800      378    4.2BSD     1024  8192    16   #
  b:    66150    38178      swap     1024  8192    16   #
  c:   594216        0      boot                        # (Cyl.    0 - 1571)
  d:   489888   104328    4.2BSD     1024  8192    16   #
```

Installing the bootstrap program locally

For earlier models incapable of netbooting, you need to install the bootstrap program on a bootable local device, such as a hard disk, floppy disk, or tape drive. If you will be booting the miniroot over the network, then you will be installing `installation/misc/SYS_UBOOT`.

If you do not have access to a netboot server to serve the miniroot installer, you can use a primitive bootstrap program `installation/misc/SYS_INST` to load the miniroot from a locally attached device (such as a disk, tape or CD-R). This is not recommended, as `SYS_INST` is difficult to use, buggy, and provides no error checking when partitioning your disk.

If your system has SCSI, this is easy. Just take a scratch SCSI disk (hard disk, zip disk, or CD-R) and use any computer to dump the bootstrap program to it. For example, to dump it to the `sd1` disk on a non-i386 platform:

dd if=SYS_UBOOT of=/dev/sd1c

If your system has a floppy drive, you can write the bootstrap program to it using any computer with a floppy drive. You will need to dump it using a utility like **rawrite** or dd(1). Make sure to read back from the floppy to verify that the file has been written correctly.

If your system does not have SCSI or a floppy drive, you will need a bootable operating system on your hp300 so you can write files to the HP-IB device. You should probably write the bootstrap program to the disk you will be installing NetBSD onto.

Using HP-UX to write to an HP-IB disk:

dd if=SYS_UBOOT of=/dev/rdsk/IDs0

ID is the HP-IB address (a.k.a. slave) of the disk in hexadecimal. This is usually between `00` and `07`, but possibly up to `1F` (31 decimal) .

Using HP-UX to write to an HP-IB tape:

```
# dd if=SYS_UBOOT of=/dev/rmt/0mnb obs=20b conv=osync
```

Installing the miniroot file system locally

This step is only necessary if you are not loading the miniroot installer from a netboot server. Follow the same procedure for the bootstrap program, except use the uncompressed miniroot file system (`installation/miniroot/miniroot.fs.gz`) instead of the bootstrap program. The only quirk is that you should place it at the offset of the swap partition you calculated above in the disklabel. In the example disklabel above, the offset is 38178 sectors of 512 bytes. Therefore, the `dd`(1) command would be something like:

```
# gunzip miniroot.fs.gz
# dd if=miniroot.fs of=/dev/rdsk/IDs0 seek=38178b
```

Note the '`b`' after the offset, which specifies blocks of 512 bytes.

By dumping the miniroot to disk where the swap partition will be, you're saving a step later where `SYS_INST` tries to download the miniroot over NFS. Just make sure that when you enter the partition table into `SYS_INST` you use the same block offset for the swap partition as you dumped the miniroot.

Configuring the netboot server

This step will configure your netboot server to provide `SYS_UBOOT` and the miniroot installer to your hp300.

1. Introduction

   To netboot a hp300, you must configure one or more servers to provide information and files to your hp300 (the 'client'). If you are using NetBSD (any architecture) on your netboot server(s), the information provided here should be sufficient to configure everything. Additionally, you may wish to look at the `diskless`(8) manual page and the manual pages for each daemon you'll be configuring. If the server(s) are another operating system, you should consult the NetBSD Diskless HOW-TO, which will walk you through the steps necessary to configure the netboot services on a variety of platforms.

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

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

   Briefly, the netboot process involves discovery, bootstrap, kernel and file system stages. In the first stage, the client discovers information about where to find the bootstrap program. Next, it downloads and executes the bootstrap program. The bootstrap program goes through another discovery phase to determine where the kernel is located. The bootstrap program tries to mount the NFS share containing the kernel. Once the kernel is loaded, it starts executing. For RAM disk kernels, it mounts the RAM disk file system and begins executing the installer from the RAM disk. For normal (non-RAM disk) kernels, the kernel tries to mount the NFS share that had the kernel and starts executing the installation tools or `init`(8). All supported hp300 systems use HP's proprietary RMP (the `rbootd`(8) daemon) for the first discovery stage and bootstrap download stages. The bootstrap program uses DHCP for its discovery stage. NFS is used in both the kernel and file system stages to download the kernel, and to access files on the file server.

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

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

2. `rbootd`(8)

   Get `SYS_UBOOT` from the `installation/misc` directory of the distribution.

   ```
   # mkdir -p /usr/mdec/rbootd
   # cp SYS_UBOOT /usr/mdec/rbootd
   # chmod -R a+rX /usr/mdec/rbootd
   ```

   Create `/etc/rbootd.conf` with the following line:

   ```
   CC:CC:CC:CC:CC:CC SYS_UBOOT
   ```

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

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

   If your netboot server is not running NetBSD but other OSs (like Linux, Solaris etc.), you have to use YAMAMORI Takenori's `sun-rbootd` package instead of native `rbootd`(8). Please refer the "Setting up the rbootd server" section in the NetBSD Diskless HOW-TO
   http://www.NetBSD.org/docs/network/netboot/rbootd/
   for details.

3. `dhcpd`(8)

   The bootstrap program uses DHCP to discover the location of the kernel. Put the following lines in your `/etc/dhcpd.conf` (see `dhcpd.conf`(5) and `dhcp-options`(5) for more information):

   ```
   ddns-update-style none;
                   # Do not use any dynamic DNS features
                   #
   allow bootp;    # Allow bootp requests, thus the dhcp server
                   # will act as a bootp server.
                   #
   authoritative;  # master DHCP server for this subnet
                   #
   subnet 192.168.1.0 netmask 255.255.255.0 {
                   # Which network interface to listen on.
                   # The zeros indicate the range of addresses
                   # that are allowed to connect.
   }
   group {
                   # Set of parameters common to all clients
                   # in this "group".
                   #
           option broadcast-address        192.168.1.255;
           option domain-name              "test.net";
           option domain-name-servers      dns.test.net;
           option routers                  router.test.net;
           option subnet-mask              255.255.255.0;
                   #
                   # An individual client.
                   #
           host client.test.net {
   ```

```
                hardware ethernet       CC:CC:CC:CC:CC:CC;
                fixed-address           192.168.1.10;
                #
                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name        "client";

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

                #
                # If your DHCP server is not your NFS server, supply the
                # address of the NFS server. Since we assume you run everything
                # on one server, this is not needed.
                #
                # next-server server.test.net;
        }
 #you may paste another "host" entry here for additional
 #clients on this network
}
```

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

touch /var/db/dhcpd.leases

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

/etc/rc.d/dhcpd restart

4. `nfsd`(8), `mountd`(8), and `rpcbind`(8)

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

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

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

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

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

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

5. NetBSD kernel and installation tools

   Now, if you place a kernel named `netbsd` in `/export/client/root` your client should boot the kernel. If you are netbooting the installer, you can use either the traditional miniroot-based installer `installation/miniroot/miniroot.fs.gz` or the experimental RAM disk-based installer `binary/kernel/netbsd-RAMDISK.gz`.

To use the miniroot-based installer, mount the miniroot file system on your netboot server. This procedure does *not* work on any operating system other than NetBSD. You'll also need to either set up a new NFS share point or an FTP server for the distribution files, as they won't fit inside the miniroot file system.

```
# gunzip miniroot.fs.gz
# vnconfig -c /dev/vnd0c /path/to/miniroot.fs
# mount -o ro /dev/vnd0c /export/client/root
# ls /export/client/root
 .profile     dist/        install.md   mnt/         sbin/        usr/
 bin/         etc/         install.sub  mnt2/        tmp/         var/
 dev/         install*     kern/        netbsd*      upgrade*
```

If there are no files present in your exported directory, then something is wrong.

To use the RAM disk-based installer, uncompress and rename the kernel. Also, copy the distribution files to the client's root directory.

```
# cp *tgz /export/client/root
# gunzip netbsd-RAMDISK.gz
# mv netbsd-RAMDISK /export/client/root/netbsd
```

If you are running your hp300 diskless, simply use `binary/kernel/netbsd-GENERIC.gz`.

6. Client file system

   You can skip this step if you do not plan to run your client diskless after installation. Otherwise, you need to extract and set up the client's installation of NetBSD. The Diskless HOW-TO describes how to provide better security and save space on the NFS server over the procedure listed here. See
   http://www.NetBSD.org/docs/network/netboot/nfs.html
   for details.

   - Extracting distribution sets

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

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

   - Set up swap

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

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

   - Create device nodes

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

     This procedure only works on NetBSD hosts.

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

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

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

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

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

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

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

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

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

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

7. Setting up the server daemons

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

   ```
   rbootd=YES       rbootd_flags=""
   dhcpd=YES        dhcpd_flags="-q"
   nfs_server=YES         # enable server daemons
   mountd=YES
   rpcbind=YES      rpcbind_flags="-l"   # -l logs libwrap
   ```

Put Series 400 systems in HP-UX Compatible Boot Mode

Series 400 systems can be configured to boot either HP-UX or DomainOS. To boot NetBSD/hp300 you must have your system configured in '`HP-UX Compatible Boot Mode`'. If, when you power on your machine, it does *not* present a menu like the following, then you need to change your configuration.

```
Copyright 1990,
Hewlett-Packard Company.
All Rights Reserved.

BOOTROM  Series 400  Rev. 1.1
MD12 REV 1.2 1990/08/07.14:27:08
[...]
```

1. Attach a Domain keyboard or an HIL keyboard.
   The BootROM knows how to use either, even if NetBSD doesn't yet.

2. Put your system into 'service mode'.
   For a 4XXs, there's a toggle switch on the back of the machine (near the top). For a 4XXt or 4XXdl, press the green button on the front, behind the silly door. For a 425e, there's a toggle switch on the back of the machine (in the middle). The second green LED should light up.

3. Reset the machine.
   Press the reset button. For a 4XXs, there's a small plunger on the back of the machine (near the top). For a 4XXt or 4XXdl, there's a white button on the front, behind the silly door. For a 425e, there's a

button on the back of the machine.

4. Press RETURN to get the Domain boot prompt (>).
   You can type H to get a list of available commands.

5. Type the following sequence of commands to convert to 'HP−UX Compatible Boot Mode'.

   > **CF**
   Type [key] RETURN ? **2**
   Type [key] RETURN ? **2**
   Type T or P RETURN ? **P**
   Type [key] RETURN ? **E**

6. Be sure to turn 'service mode' *off* when you're done. It may prevent you from selecting which device to boot from.

See the FAQ for additional help.
http://www.NetBSD.org/ports/hp300/faq.html#domain

Searching for a bootable system

All the early hp300 Boot ROMs are very primitive and only allow a few simple operations. You can only interact with it after it is first powered on. If you reboot the machine, it will ignore anything you type and start loading the same OS you previously booted.

At any time after it recognizes the keyboard, while it is doing its self test or searching for a bootable system, you can hit reset to return it to a cold-boot configuration. On HIL keyboards, this is control−shift−break, where break is the key in the upper left (where escape is on sane keyboards). There is no equivalent over serial terminal, you'll need to power-cycle your machine.

After it beeps (i.e. recognizes the HIL keyboard), press RETURN twice to get the list of bootable devices.

```
SEARCHING FOR A SYSTEM (RETURN To Pause)
```

The newer HP Boot ROM, present on Series 400 machines and some of the later 300s (345, 375, 380, 382, 385) is capable of a little bit more. To select which device to boot from, press RETURN once after it beeps twice (i.e. recognizes the HIL keyboard) to get the list of bootable devices.

```
RESET To Power−Up, SPACE clears input   Select System, type RETURN        ?
```

The FAQ lists additional things you can do with the BootROM and describes the order the BootROM looks for bootable devices.
http://www.NetBSD.org/ports/hp300/faq.html

A normal power-on sequence (from a 400s) looks something like this:

```
Copyright 1990,
Hewlett−Packard Company.
All Rights Reserved.

BOOTROM  Series 400  Rev. 1.1
MD12 REV 1.2 1990/08/07.14:27:08
MC68030 Processor
MC68882 Coprocessor
Configuration EEPROM
Utility Chip at 41
HP−HIL.Keyboard
 RESET To Power−Up
 Loading Memory
Self−Test Mode
```

```
 RESET To Power-Up, SPACE clears input
Select System, type RETURN
HP-IB
DMA-C0
 Self-Test Mode
RAM 33554158 Bytes
HP98644 (RS-232) at 9
HP PARALLEL at 12
HP98265 (SCSI S 32) at 14
HP98643 (LAN) at 21, AUI, 080009115DB3
Bit Mapped Video at 133 (Console)
 System Search Mode
                                        :RODIME RO3000T, 1406, 0
                                          1Z SYS_UBOOT
                                        :LAN080009115DB3, 2100, 0
                                          2Z SYS_UBOOT
                                        :HP7959, 702, 0, 0
                                          1H SYSHPUX
                                          1D SYSDEBUG
                                          1B SYSBCKUP
                                        :HP9122, 0700, 0, 0
                                          3Z SYS_INST
```

You should see your bootstrap program somewhere in this list. If it's not here, then your hp300 can't boot it and there's a problem somewhere. To boot from a particular device, type in the two character name for it and press RETURN. In this example, you'd type `2Z` to boot from the network.

Selecting ethernet port on Series 400

Series 400 machines have two ethernet media types built into the motherboard. You may only use one at a time. When your Series 400 workstation goes through the self-test when powered on or rebooted, it will say one of the following:

```
HP98643 (LAN) at 21, AUI
HP98643 (LAN) at 21, Thin
```

If the wrong type of network is selected, you will need to change the ethernet port. You will need to open the case (4XXt, 4XXdl, 4XXe) or remove the motherboard (4XXs) to access the jumper. Be sure to use static-prevention measures, as you could easily fry your motherboard from carelessness. If you are uncomfortable with this, ask a friend who is aware of these issues. There is a block of 8 jumpers at the rear of the motherboard, labeled AUI/Thin. You will need to put the jumpers in the position necessary for your type of ethernet.

Running SYS_INST

This step is necessary *only* if you cannot netboot.

Chose `SYS_INST` from the list of bootable devices that the BootROM found. `SYS_INST` will load and prompt you for a command.

A quick note about disk numbers: While in the `SYS_INST` program, you may use different unit numbers for the disks than when the NetBSD kernel is running. The unit number for a disk while in `SYS_INST` is calculated with the following formula:

```
unit = (controller * 8) + slaveID
```

Controllers are numbered 0, 1, ... starting with the lowest select code. SCSI controllers and HP-IB controllers are counted separately. Therefore, if you had a system with an internal HP-IB interface at select code 7, a fast HP-IB interface at select code 14, and a SCSI interface at select code 16, unit numbers might be something like the following:

Location	**Unit**	
HP-IB at 7, slaveID 2	2	(disk: rd2)
HP-IB at 14, slaveID 5	13	(disk: rd13)
SCSI at 16, slaveID 0	0	(disk: sd0)

You will need to place a disklabel on the disk.

sys_inst> **disklabel**

Note: It may be worth selecting the **zap** option initially to ensure that the disklabel area is clear. This may be especially important if an HP-UX boot block had been previously installed on the disk.

Select the **edit** option, and answer the questions about your disk. There may be several questions which you may not be sure of the answers to. Listed below are guidelines for SCSI and HP-IB disks:

Bad sectoring?	NO
Ecc?	NO
Interleave?	1
Trackskew?	0
Cylinderskew?	0
Headswitch?	0
Track-to-track?	0
Drivedata 0-4?	0 (for all Drivedata values)

Next, you will be asked to fill out the partition map. You must provide responses for all 8 partitions. Remember, you must have the sector offset for the 'b' partition match the location you dumped the miniroot file system image. Set the size and offset of any unused partition to 0. Note that sizes and offsets are expressed in 'n sectors', assuming 512 byte sectors. Care should be taken to ensure that partitions begin and end on cylinder boundaries (i.e. size and offset is an even multiple of the number of sectors per cylinder). While this is not technically necessary, it is generally encouraged.

Note: When setting the partition type of the 'b' partition, make sure to specify it as an `ffs` partition so that the miniroot can be mounted (even if this will be a swap partition). You will be given a chance to clean this up later in the installation process.

Once you have edited the label, select the **show** option to verify that it is correct. If so, select **write** and **done**. Otherwise, you may re-edit the label.

In an earlier step, we already copied the miniroot image to the target disk.

Boot from the miniroot file system.

sys_inst> **boot**

Enter the disk from which to boot. The kernel in the miniroot file system will be booted into single-user mode.

Choosing a kernel location

Once the bootstrap program `SYS_UBOOT` has started, it will pause and let you chose a kernel location, name, and options:

```
>> NetBSD/hp300 Primary Boot, Revision 1.13
>> (gregm@mcgarry, Mon Apr 15 08:46:32 NZST 2002)
>> HP 9000/425e SPU
>> Enter "reset" to reset system.
Boot: [[[le0a:]netbsd][-a][-c][-d][-s][-v][-q]] :-
```

If your kernel is on a different device than `SYS_UBOOT` then you will need to type in where to find it. This is the case, for example, if your model is incapable of netbooting and you started `SYS_UBOOT` from a floppy, and the miniroot installer is on a netboot server. In this case, you'd type in '`le0`' at the prompt.

If you've installed the miniroot on your disk, you can always boot from that by using partition 'b' when prompted by `SYS_UBOOT`. For example, to boot the miniroot from an HP-IB disk on controller 0 at slave ID 2, you'd type:

Boot: [[[rd0a:]netbsd][-a][-c][-d][-s][-v][-q]] :- **rd2b:netbsd**

Installing the NetBSD System

The miniroot's install program is very simple to use. It will guide you through the entire process, and is well automated. If you need to restart the installer, hit `Control-C` which will return you to a shell prompt. From there, just start it over:

./install

The experimental RAM disk-based installer is not described here, but is very self-explanatory.

The miniroot's install program will:

1. Allow you to place disklabels on additional disks. 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:

   ```
   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)
   ```

   You will also need to specify the number of partitions. The number of partitions is determined by the 'index' of the last partition letter, where `a = 1`, `b = 2`, etc. Therefore, if the last filled partition is partition 'g', there are 7 partitions. Any partitions with size of 0 may be removed from the list.

   Anything after a '#' is a comment.

   The following is an example disklabel partition map:

   ```
   7 partitions:
   #      size   offset    fstype   [fsize bsize   cpg]
   a:    30912      448    4.2BSD     1024  8192    16   # (Cyl.    1 - 69)
   b:   130816    31360      swap                         # (Cyl.   70 - 361)
   c:  1296512        0      boot                         # (Cyl.    0 - 2893)
   e:    81984   162176    4.2BSD     1024  8192    16   # (Cyl.  362 - 544)
   f:   102592   244160    4.2BSD     1024  4096    16   # (Cyl.  545 - 773)
   g:   949760   346752    4.2BSD     1024  8192    16   # (Cyl.  774 - 2893)
   ```

2. Create file systems on target partitions.

3. 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.

4. Mount target file systems. You will be given the opportunity to manually edit the resulting `/etc/fstab`.

5. Extract binary sets from the media of your choice.
6. Copy configuration information gathered during the installation process to your root file system (/).
7. Make device nodes in your root file system under `/dev`.
8. Copy a new kernel onto your root partition (/).
9. Install a new boot block.
10. 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.

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 **hp300h** for a local console, or whatever is appropriate for your serial console. 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
   Also, you may want to read through the NetBSD/hp300 FAQ entry on X11.
   http://www.NetBSD.org/ports/hp300/faq.html#x11

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/pkgsrc/current/pkgsrc/README.html
   - Precompiled binaries can be found at
     ftp://ftp.NetBSD.org/pub/pkgsrc/packages/NetBSD/
     usually in the hp300/6.1/All subdir. If you installed pkgin(1) in the **sysinst** post-installation configuration menu, you can use it to automatically install binary packages over the network. Assuming that /usr/pkg/etc/pkgin/repositories.conf is correctly configured, you can install them with the following commands:

     ```
     # pkgin install tcsh
     # pkgin install bash
     # pkgin install perl
     # pkgin install apache
     # pkgin install kde
     # pkgin install firefox
     ...
     ```

     Note: Some mirror sites don't mirror the /pub/pkgsrc directory.

     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.

- If you did not install it from the **sysinst** post-installation configuration menu, the pkgsrc(7) framework for compiling packages can be obtained by retrieving the file

  ftp://ftp.NetBSD.org/pub/pkgsrc/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/hp300 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. While you can boot the miniroot using the same methods described above for a fresh install of NetBSD/hp300 there are easier and less intrusive options since your disk is already labeled and bootable. The easiest is to dump the miniroot to your swap partition and boot from that.

1. Download the files you'll need to upgrade

   In particular, make sure you have on your locally mounted file systems base.tgz and miniroot.fs.gz

2. Boot your hp300 into 'single-user mode':

   Follow the instructions in the section above on **Chosing a kernel location** and type -s at the prompt.

3. Extract and install a new boot block:

   Make sure you install the bootstrap program distributed with this version of NetBSD/hp300.

   ```
   # tar -xpvzf base.tgz ./usr/mdec
   # disklabel -B -b ./usr/mdec/uboot.lif  root-disk
   ```

   E.g.: *root-disk* could be sd0 or rd0. We'll assume rd0 for now.

4. Install the miniroot file system:

   First make sure that your 'b' partition has enough room for the uncompressed miniroot (otherwise it might overwrite another partition or the end of the disk).

```
# gunzip miniroot.fs.gz
# dd if=miniroot.fs of=/dev/rd0b
```

5. Boot the miniroot:

   Follow the instructions in the section above on **Chosing a kernel location** and type `rd0b:netbsd` at the prompt.

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 6.1 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 6.1 kernel may be incompatible with your old binaries.
2. Back up your pre-existing kernel and copy the 6.1 kernel into your root partition (/) .
3. Extract and install a new boot block:

   Make sure you install the bootstrap program distributed with this version of NetBSD/hp300.

   ```
   # tar -xpvzf base.tgz ./usr/mdec
   # disklabel -B -b ./usr/mdec/uboot.lif  root-disk
   ```
   E.g.: *root-disk* could be `sd0` or `rd0`.
4. Reboot with the 6.1 kernel into single-user mode.
5. Check all file systems:

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

   ```
   # /sbin/mount -a -t nonfs
   ```
7. 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
```

8. Make sure you are in the root file system (/) and extract the **base** binary set:

```
# cd /
# pax -zrvpe -f /path/to/base.tgz
```

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 6.1.

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 5.x releases

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).

The implementation of SHA2-HMAC in KAME_IPSEC as used in NetBSD-5 and before did not comply to current standards. FAST_IPSEC does, with the result that old and new systems cannot communicate over IPSEC, if one of the affected authentication algorithms (hmac_sha256, hmac_sha384, hmac_sha512) is used.

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 6.1 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:

Alan Barrett	apb@NetBSD.org
Alistair Crooks	agc@NetBSD.org
Matthew Green	mrg@NetBSD.org
Chuck Silvers	chs@NetBSD.org
Matt Thomas	matt@NetBSD.org
YAMAMOTO Takashi	yamt@NetBSD.org
Christos Zoulas	christos@NetBSD.org

The portmasters (and their ports):

Reinoud Zandijk	reinoud	**acorn32**
Matt Thomas	matt	**alpha**
Ignatios Souvatzis	is	**amiga**
Ignatios Souvatzis	is	**amigappc**
Noriyuki Soda	soda	**arc**
Julian Coleman	jdc	**atari**
Matthias Drochner	drochner	**cesfic**
Erik Berls	cyber	**cobalt**
Antti Kantee	pooka	**emips**
Simon Burge	simonb	**evbmips**
Steve Woodford	scw	**evbppc**
Izumi Tsutsui	tsutsui	**ews4800mips**
Izumi Tsutsui	tsutsui	**hp300**
Nick Hudson	skrll	**hp700**
Valeriy E. Ushakov	uwe	**hpcsh**
Matt Thomas	matt	**ibmnws**
Gavan Fantom	gavan	**iyonix**
Valeriy E. Ushakov	uwe	**landisk**
Izumi Tsutsui	tsutsui	**luna68k**
Scott Reynolds	scottr	**mac68k**
Michael Lorenz	macallan	**macppc**
Steve Woodford	scw	**mvme68k**
Steve Woodford	scw	**mvmeppc**
Matt Thomas	matt	**netwinder**
Izumi Tsutsui	tsutsui	**news68k**
Tim Rightnour	garbled	**ofppc**
Simon Burge	simonb	**pmax**
Tim Rightnour	garbled	**prep**
Tim Rightnour	garbled	**rs6000**
Tohru Nishimura	nisimura	**sandpoint**
Simon Burge	simonb	**sbmips**
Søren Jørvang	soren	**sgimips**
SAITOH Masanobu	msaitoh	**sh3**
Martin Husemann	martin	**sparc64**
Anders Magnusson	ragge	**vax**
NISHIMURA Takeshi	nsmrtks	**x68k**

Manuel Bouyer	`bouyer`	**xen**

The NetBSD 6.1 Release Engineering team:

Stephen Borrill	`sborrill@NetBSD.org`
Manuel Bouyer	`bouyer@NetBSD.org`
David Brownlee	`abs@NetBSD.org`
James Chacon	`jmc@NetBSD.org`
Julian Coleman	`jdc@NetBSD.org`
Alistair G. Crooks	`agc@NetBSD.org`
Håvard Eidnes	`he@NetBSD.org`
John Heasley	`heas@NetBSD.org`
Martin Husemann	`martin@NetBSD.org`
Soren Jacobsen	`snj@NetBSD.org`
Phil Nelson	`phil@NetBSD.org`
Jeremy C. Reed	`reed@NetBSD.org`
Jeff Rizzo	`riz@NetBSD.org`
SAITOH Masanobu	`msaitoh@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`
Zafer Aydogan	`zafer@NetBSD.org`
Christoph Badura	`bad@NetBSD.org`
Marc Balmer	`mbalmer@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`
Erik Berls	`cyber@NetBSD.org`
Hiroyuki Bessho	`bsh@NetBSD.org`
John Birrell	`jb@NetBSD.org`
Rafal Boni	`rafal@NetBSD.org`
Stephen Borrill	`sborrill@NetBSD.org`
Sean Boudreau	`seanb@NetBSD.org`
Manuel Bouyer	`bouyer@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`
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
Taylor R. Campbell	riastradh@NetBSD.org
Daniel Carosone	dan@NetBSD.org
Dave Carrel	carrel@NetBSD.org
James Chacon	jmc@NetBSD.org
Mihai Chelaru	kefren@NetBSD.org
Aleksey Cheusov	cheusov@NetBSD.org
Bill Coldwell	billc@NetBSD.org
Julian Coleman	jdc@NetBSD.org
Marcus Comstedt	marcus@NetBSD.org
Jeremy Cooper	jeremy@NetBSD.org
Thomas Cort	tcort@NetBSD.org
Chuck Cranor	chuck@NetBSD.org
Alistair Crooks	agc@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
Steven Drake	sbd@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
Makoto Fujiwara	mef@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
Oliver V. Gould	ver@NetBSD.org
Paul Goyette	pgoyette@NetBSD.org
Michael Graff	explorer@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
Yorick Hardy	yhardy@NetBSD.org
Ben Harris	bjh21@NetBSD.org
Eric Haszlakiewicz	erh@NetBSD.org
John Hawkinson	jhawk@NetBSD.org
Emile Heitor	imil@NetBSD.org
John Heasley	heas@NetBSD.org
Lars Heidieker	para@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
Daniel Horecki	morr@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
Darran Hunt	darran@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
Søren Jørvang	soren@NetBSD.org
Takahiro Kambe	taca@NetBSD.org
Antti Kantee	pooka@NetBSD.org
Frank Kardel	kardel@NetBSD.org
KAWAMOTO Yosihisa	kawamoto@NetBSD.org
Min Sik Kim	minskim@NetBSD.org
KIYOHARA Takashi	kiyohara@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
Jonathan A. Kollasch	jakllsch@NetBSD.org
Radoslaw Kujawa	rkujawa@NetBSD.org
Jochen Kunz	jkunz@NetBSD.org
Martti Kuparinen	martti@NetBSD.org
Arnaud Lacombe	alc@NetBSD.org
Kevin Lahey	kml@NetBSD.org
David Laight	dsl@NetBSD.org
Johnny C. Lam	jlam@NetBSD.org
Guillaume Lasmayous	gls@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
MAEKAWA Masahide	gehenna@NetBSD.org
Anders Magnusson	ragge@NetBSD.org
John Marino	marino@NetBSD.org
Roy Marples	roy@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
Constantine A. Murenin	cnst@NetBSD.org
Joseph Myers	jsm@NetBSD.org
Tuomo Mäkinen	tjam@NetBSD.org
Zoltán Arnold NAGY	zoltan@NetBSD.org
Ken Nakata	kenn@NetBSD.org
Takeshi Nakayama	nakayama@NetBSD.org
Alexander Nasonov	alnsn@NetBSD.org
Phil Nelson	phil@NetBSD.org
John Nemeth	jnemeth@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
Ryo ONODERA	ryoon@NetBSD.org
Atsushi Onoe	onoe@NetBSD.org
Greg Oster	oster@NetBSD.org
Jonathan Perkin	sketch@NetBSD.org
Fredrik Pettai	pettai@NetBSD.org
Herb Peyerl	hpeyerl@NetBSD.org
Matthias Pfaller	matthias@NetBSD.org
Chris Pinnock	cjep@NetBSD.org
Adrian Portelli	adrianp@NetBSD.org
Chris Provenzano	proven@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
Jens Rehsack	sno@NetBSD.org
Antoine Reilles	tonio@NetBSD.org
Tyler R. Retzlaff	rtr@NetBSD.org
Scott Reynolds	scottr@NetBSD.org
Tim Rightnour	garbled@NetBSD.org
Jeff Rizzo	riz@NetBSD.org
Hans Rosenfeld	hans@NetBSD.org

Steve Rumble	rumble@NetBSD.org
Rumko	rumko@NetBSD.org
Jukka Ruohonen	jruoho@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
Naoto Shimazaki	igy@NetBSD.org
Ryo Shimizu	ryo@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
Jörg Sonnenberger	joerg@NetBSD.org
Ignatios Souvatzis	is@NetBSD.org
T K Spindler	dogcow@NetBSD.org
Matthew Sporleder	mspo@NetBSD.org
Bill Squier	groo@NetBSD.org
Adrian Steinmann	ast@NetBSD.org
Bill Studenmund	wrstuden@NetBSD.org
Kevin Sullivan	sullivan@NetBSD.org
Kimmo Suominen	kim@NetBSD.org
Grégoire Sutre	gsutre@NetBSD.org
Sergey Svishchev	shattered@NetBSD.org
Robert Swindells	rjs@NetBSD.org
Shin Takemura	takemura@NetBSD.org
TAMURA Kent	kent@NetBSD.org
Shin'ichiro TAYA	taya@NetBSD.org
Hasso Tepper	hasso@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
Mark Weinem	weinem@NetBSD.org
Lex Wennmacher	wennmach@NetBSD.org
Leo Weppelman	leo@NetBSD.org
Assar Westerlund	assar@NetBSD.org
Frank Wille	phx@NetBSD.org
Nathan Williams	nathanw@NetBSD.org
Rob Windsor	windsor@NetBSD.org
Jim Wise	jwise@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
Arnaud Ysmal	stacktic@NetBSD.org
Reinoud Zandijk	reinoud@NetBSD.org
S.P.Zeidler	spz@NetBSD.org
Tim Zingelman	tez@NetBSD.org
Christos Zoulas	christos@NetBSD.org

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:

This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Internet Research Institute, Inc.
This product includes software developed by Leo Weppelman and Waldi Ravens.
This product includes software developed by Mika Kortelainen
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
This product includes software developed by Adam Glass and Charles M. Hannum.
This product includes software developed by Adam Glass.
This product includes software developed by Advanced Risc Machines Ltd.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Projet.
This product includes software developed by Alex Zepeda.
This product includes software developed by Amancio Hasty and Roger Hardiman
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 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 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 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 Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by Dean Huxley.
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 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 Ian W. Dall.
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 Jason L. Wright

This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/
This product includes software developed by Joachim Koenig-Baltes.
This product includes software developed by Jochen Pohl for The NetBSD Project.
This product includes software developed by Joerg Wunsch
This product includes software developed by John Birrell.
This product includes software developed by John P. Wittkoski.
This product includes software developed by John Polstra.
This product includes software developed by Jonathan R. Stone for the NetBSD Project.
This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project.
This product includes software developed by Jonathan Stone.
This product includes software developed by 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 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 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 Matthew Fredette.
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 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 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 Stephan Thesing.
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 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 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 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.

The End