NAME

INSTALL – Installation procedure for NetBSD/sparc.

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.5.2 on the *sparc* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

Quick install notes for the impatient

This section contains some brief notes describing what you need to install NetBSD 1.5.2 on a machine of the sparc architecture.

- Fetch the CD image, `sparc_bootable.iso` or the floppy disk images, `install/floppy/disk1.gz` and `install/floppy/disk2`. You need either the pair of floppies or the CD to boot your system. Alternatively, you may netboot the installation kernel, which requires several local netboot services. The details are not covered here, as setting up a netboot server is hardly "quick".
- The actual binary distribution is in the `binary/sets/` directory. When you boot the install floppies or CD-ROM, the installation program can fetch these files for you (using e.g. ftp), if you have a network connection. There are several other methods to get the binary sets onto the machine you install.

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

  The disk(s) you just prepared will be used to boot the installation kernel, which contains all the tools required to install NetBSD.
- You will need to get to the OpenBoot PROM "**`ok`**" prompt. After your system first powers on, and displays some initial information, press the STOP–A keys. At the "**`ok`**" prompt, type the command to boot your system into NetBSD. The command to boot from CD is one of the following commands (depending on your model): "**`b sd(,30,)`**", "**`boot sd(,30,)`**", or "**`boot cdrom`**".

  The command to boot from floppy is either "**`boot fd(,,1)`**" or "**`boot floppy`**". The installer will prompt you to insert the second floppy when it is ready for it.
- For third-party programs which are not part of the base NetBSD distribution, you will want to explore the **`pkgsrc`** system with its more than 2100 program packages.

What is NetBSD?

The NetBSD Operating System is a fully functional Open Source UNIX-like operating system derived from the University of California, Berkeley Networking Release 2 (Net/2), 4.4BSD-Lite, and 4.4BSD-Lite2 sources. NetBSD runs on thirty-one different system architectures featuring twelve distinct families of CPUs, and is being ported to more. The NetBSD 1.5.2 release contains complete binary releases for fifteen different machine types. (The sixteen remaining are not fully supported at this time and are thus not part of the binary distribution. For information on them, please see the NetBSD web site at **http://www.netbsd.org/**.)

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.

Upgrade path to NetBSD 1.5.2

If you are not installing your system "from scratch" but instead are going to upgrade an existing system already running NetBSD you need to know which versions you can upgrade with NetBSD 1.5.2.

NetBSD 1.5.2 is an upgrade of NetBSD 1.5, 1.5.1, and earlier major and patch releases of NetBSD.

The intermediate development versions of code available on the main trunk in our CVS repository (also known as "NetBSD-current") from *after* the point where the release cycle for 1.5 was started are designated by version identifiers such as 1.5A, 1.5B, etc. These identifiers do not designate releases, but indicate major changes in internal kernel APIs. Note that the kernel from NetBSD 1.5.2 can *not* be used to upgrade a system running one of those intermediate development versions. Trying to use the NetBSD 1.5.2 kernel on such a system *will* probably result in problems.

Please also note that it is not possible to do a direct "version" comparison between any of the intermediate development versions mentioned above and 1.5.2 to determine if a given feature is present or absent in 1.5.2. The development of 1.5 and the subsequent "point" releases is done on a separate branch in the CVS repository. The branch was created when the release cycle for 1.5 was started, and during the release cycle of 1.5 and its patch releases, selected fixes and enhancements have been imported from the main development trunk. So, there are features in 1.5.2 which were not in, e.g. 1.5B, and vice versa.

Major Changes Between 1.5.1 and 1.5.2

The main purpose of NetBSD 1.5.2 is to fix some problems discovered shortly after the release of NetBSD 1.5.1. Some highlights are:

- `telnetd`(8) has been fixed to deal with a security issue (SA2001-012).
- A weakness in the OpenSSL libcrypto's pseudo random number generator has been fixed (SA2001-013).
- Floating point emulation on i386 (to run on FPU-less CPUs) which was broken late in 1.5.1's cycle has been restored to working order.
- Bugs in the compiler have been worked around to produce a working miniroot image on sparc64.
- The script-based installers on amiga, atari and mvme68k have been updated to deal with new behaviour from `mount`(8).
- `dump`(8) has been fixed to deal with a local security issue (SA2001-014).
- Several system calls have had defensive length checks applied on their arguments (SA2001-015).
- The DHCP software has been upgraded to ISC Version 3, Release Candidate 10.

Additionally, a few more bugs have been fixed. See the CHANGES-1.5.2 file for the complete list of changes.

Major Changes Between 1.5 and 1.5.1

The complete list of changes between NetBSD 1.5 and 1.5.1 can be found in the file CHANGES-1.5.1 in the top directory of the source tree. The following are highlights only:

- A driver for the Aironet/Cisco wireless PCMCIA cards has been added; see `an`(4).
- NFS client performance has been improved, typically by 40% for writes but possibly up to 100% in certain setups.
- The `siop`(4) driver has improved in performance and robustness.
- Support for cloning pseudo-interfaces has been added. See `ifconfig`(8).
- Support for 802.1Q virtual LANs has been added. See `vlan`(4).
- The `isp`(4) driver has been upgraded to (among other things) work on MacPPC.
- BIND has been upgraded to version 8.2.3 (SA2001-001).
- Support for booting from RAIDframe RAID1 mirrors on i386 added.
- The `lfs`(4) file system has again been substantially updated, but is still experimental.
- Ultra/66 support has been added for capable VIA chipsets, and Ultra/100 support has been added for the HPT370, Promise and Intel ICH2 controllers in the `pciide`(4) driver. Support for Intel 82801BAM controllers has also been added, and handling of Ali controllers has been improved.
- OpenSSH has been updated to deal with a security issue (SA2001-003).
- Sendmail has been upgraded to version 8.11.3.
- The `ex`(4) driver has added support for 3Com 3c555, 3c556 and 3c556B MiniPCI Ethernet cards.
- A driver for the on-board audio hardware found on many Apple PowerMacs has been added; see `awacs`(4).
- The `sip`(4) driver has been fixed to properly support the dp83815, as found in current Netgear FA311 10/100 cards.
- `ftpd`(8) has been updated to deal with two security issues (SA2000-018 and SA2001-005).
- `ntpd`(8) has been updated to deal with a security issue (SA2001-004).
- `telnetd`(8) has been updated to deal with a security issue (SA2000-017).
- A vulnerability on i386 related to USER_LDT has been fixed (SA2001-002).
- The Linux emulation has been enhanced to prepare for the support of using the Linux version of VMware.
- IP checksumming speed has been improved on i386 compared to NetBSD 1.5 by about 10%.
- Support for the Socket Communications LP-E Type II PCMCIA NE2000 clone card has been added to `ne`(4).
- The DHCP software has been upgraded to ISC version 3, Beta 2, patchlevel 23, to fix core dumps in `dhclient`(8), among other things. Please note that the new `dhcpd`(8) forces you to configure a "ddns-update-style" of either "ad-hoc", "interim" or "none".

- Various fixes and enhancements to INET6 and IPSEC code; among them improved interaction between IPF/Nat and IPSEC.
- The Heimdal `kerberos`(8) implementation has been upgraded to version 0.3e.
- Support for Accton EN2242 and other AmdTek AN985 cards added to the `tlp`(4) driver.
- Several country-specific keyboard mappings have been added for USB keyboards.
- A driver for Yamaha YMF724/740/744/745-based sound cards has been added, see `yds`(4).
- The maximum number of BSD disklabel partitions on the i386 port has been increased from 8 to 16.
- Drivers for the AC'97 based audio sound chips ESS Technology Maestro 1, 2, and 2E (see `esm`(4)), NeoMagic 256 (see `neo`(4)), and Cirrus Logic CrystalClear PCI Audio CS4281 (see `clct`(4)) have been added.

In addition, many bugs have been fixed—more than 95 problems reported through our problem tracking system have been fixed, and some other non-reported problems have also been found and fixed. See the CHANGES-1.5.1 file for the complete list.

The NetBSD Packages Collection (pkgsrc) which is used to maintain, build, track dependencies, and maintain NetBSD-specific fixes to third-party programs, has received a major overhaul for NetBSD 1.5.1. Some highlights are:

- Many new packages were added to the collection, which now includes about 2100 packages. Many of them are also available as pre-compiled binaries on ftp.netbsd.org and its mirrors. Many packages have been modified and enhanced to compile and function properly on big-endian (m68k, sparc), and LP64 architectures (alpha, sparc64).
- KDE2 and KOffice together consitutes a fully integrated office environment with no license problems, available for i386, alpha and many other architectures.
- Mozilla 0.9, KDE2's Konqueror, and Links 0.95 are just a few examples of the web browsers available.
- A support package for running VMware on NetBSD/i386 was added, it's called suse_vmware. The official VMware code, a valid license, and Wasabi Systems' compatibility package are all needed to run VMware.
- Internal changes of the pkgsrc system include strong checksums to prevent malicious distribution files, as well as restructuring of the package structure in pkgsrc for faster extraction of pkgsrc tar files and upgrades via SUP and CVS.

Please note that at the moment, **`sysinst`** will not assist you in installing pre-built third-party binary packages or the pkgsrc system itself, so you will have to manually install packages using **`pkg_install`** or fetch and extract the pkgsrc.tgz tar file to get started.

Lastly, it should be noted that the X11 binaries shipped in NetBSD 1.5.2 is still based on XFree86 version 3.3.6. Several newer graphics cards are inadequately supported by that code base, but on the other hand support for several older graphics cards is not available in newer XFree86 code. NetBSD is in the process of moving to XFree86 version 4, and is currently maintaining both the XFree86 3.3.6 and the XFree86 version 4 code in the **xsrc** source set, and you may at compile time pick which sources to build and install. To ease installation, testing and use of the XFree86 version 4 code, a binary snapshot based on XFree86 version 4.0.3 will be made available for at least the i386 architecture shortly after the release of NetBSD 1.5.2.

Changes Between The NetBSD 1.4 and 1.5 Releases

The NetBSD 1.5 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many userland enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize over one year of development that went into the NetBSD 1.5.2 release. Some highlights include:

Kernel

- Ports to new platforms including: arc, cobalt, hpcmips, news68k, sgimips, and sparc64.
- Improved performance and stability of the UVM virtual memory subsystem.
- Implementation of generic kernel locking code, as well as a restructure and re-tuning of the scheduler, to be used by the future symmetric multi-processing (SMP) implementation.
- Improved compatibility support for Linux, OSF1, and SVR4 programs.
- New compatibility support for Win32 programs.
- Support for dynamically loaded ELF kernel modules.
- Kernel process tracing using `ktruss`(1).
- Deletion of swap devices using `swapctl`(8).
- Easier hot-pluggability of keyboards and mice using a new wscons device—wsmux.
- Improved PCMCIA and CardBus support, including support for detaching of devices and cards, resulting in better support for notebooks and PDA devices.
- Numerous hardware improvements, including areas such as: audio, UDMA/66 support for ATA drives, USB, and wireless networking.

Networking

- Addition of IP version 6 (IPv6) and IPsec to the networking stack, from the KAME project. This includes addition of kernel code for IPv6/IPsec, IPv4/v6 dual-stack user applications and supporting libraries. Due to this, the shlib major version for `pcap`(3) is incremented and you may need to recompile userland tools. The KAME IPv6 part includes results from the unified-IPv6 effort.

File system

- Significant Fast file system (FFS) performance enhancements via integration of Kirk McKusick's soft updates and trickle sync code.
- Support for the Windows NT 'NTFS' file system (read-only at this stage).
- Support for revision 1 of the Linux 'ext2fs' file system.
- Enhanced stability and usability of LFS (the BSD log-structured file system).
- Various RAIDframe enhancements including: auto-detection of RAID components and auto-configuration of RAID sets, and the ability to configure the root file system (`/`) on a RAID set.
- Support for Microsoft Joliet extensions to the ISO9660 CD file system.
- Improved file system vnode locking mechanisms, thus resolving a source of several panics in the past.
- Support for NFS and RPC over IPv6.
- The server part of NFS locking (implemented by `rpc.lockd`(8)) now works.

Security

- Strong cryptographic libraries and applications integrated, including the AES cipher Rijndael, the OpenSSL library, more complete Kerberos IV and Kerberos V support, and an SSH server and client.
- sysctl(3) interfaces to various elements of process and system information, allowing programs such as ps(1), dmesg(1) and the like to operate without recompilation after kernel upgrades, and remove the necessity to run setgid kmem (thus improving system security).
- Disable various services by default, and set the default options for disabled daemons to a higher level of logging.
- Several code audits were performed. One audit replaced string routines that were used without bounds checking, and another one identified and disabled places where format strings were used in unsafe ways, allowing arbitrary data to be entered by (possibly) malicious users to overwrite application code, and leading from Denial of Service attacks to compromised systems.
- sshd(8) and ssh(1) now require rnd(4) kernel random number devices.

System administration and user tools

- Conversion of the rc(8) system startup and shutdown scripts to an 'rc.d' mechanism, with separate control scripts for each service, and appropriate dependency ordering provided by rcorder(8).
- postfix(1) provided as alternative mail transport agent to sendmail(8).
- User management tools useradd(8), usermod(8), userdel(8), groupadd(8), groupmod(8), and groupdel(8) added to the system.
- Incorporation of a login class capability database (/etc/login.conf) from BSD/OS.
- Improved support for usernames longer than eight characters in programs such as at(1) and w(1).
- Many enhancements to ftpd(8) providing features found in larger and less secure FTP daemons, such as user classes, connection limits, improved support for virtual hosting, transfer statistics, transfer rate throttling, and support for various IETF ftpext working group extensions.
- The ftp(1) client has been improved even further, including transfer rate throttling, improved URL support, command line uploads. See the man page for details.

Miscellaneous

- Updates to the NetBSD source code style code (located in /usr/share/misc/style) to use ANSI C only (instead of K&R) and reflect current (best) practice, and begin migrating the NetBSD source code to follow it.
- Implementation of many SUSv2 features to the curses(3) library, including support for color.
- Updates of most third party packages that are shipped in the base system, including file(1), ipfilter(4), ppp(4), and sendmail(8) to the latest stable release.
- Many new packages in the *pkgsrc* system, including standard desktops like KDE and GNOME as well as latest Tcl/Tk and perl and many of the components of the Java Enterprise platform. The package framework itself now has full wildcard dependency support.

Kernel interfaces have continued to be refined, and more subsystems and device drivers are shared among the different ports. You can look for this trend to continue.

NetBSD 1.5.2 is the sixth release for the sparc.

Some (but not all!) notable sparc-specific improvements include:

- Migration to the ELF binary format from a.out.
- Support for the JavaStation 1.
- HyperSPARCs now run reliably.
- Tadpole console and mouse now work.
- Support for the sun4/400 class of machines.

NetBSD 1.5.2 on sparc is, as usual, also fully backward compatible with old NetBSD/sparc binaries, so you don't need to recompile all your local programs provided you set the appropriate binary compatibility options in your kernel configuration.

The Future of NetBSD

The NetBSD Foundation has been incorporated as a non-profit organization. Its purpose is to encourage, foster and promote the free exchange of computer software, namely the NetBSD Operating System. The foundation will allow for many things to be handled more smoothly than could be done with our previous informal organization. In particular, it provides the framework to deal with other parties that wish to become involved in the NetBSD Project.

The NetBSD Foundation will help improve the quality of NetBSD by:

- providing better organization to keep track of development efforts, including co-ordination with groups working in related fields.
- providing a framework to receive donations of goods and services and to own the resources necessary to run the NetBSD Project.
- providing a better position from which to undertake promotional activities.
- periodically organizing workshops for developers and other interested people to discuss ongoing work.

We intend to begin narrowing the time delay between releases. Our ambition is to provide a full release every six to eight months.

We hope to support even *more* hardware in the future, and we have a rather large number of other ideas about what can be done to improve NetBSD.

We intend to continue our current practice of making the NetBSD-current development source available on a daily basis.

We intend to integrate free, positive changes from whatever sources submit them, providing that they are well thought-out and increase the usability of the system.

Above all, we hope to create a stable and accessible system, and to be responsive to the needs and desires of NetBSD users, because it is for and because of them that NetBSD exists.

Sources of NetBSD

Refer to **http://www.netbsd.org/Sites/net.html**.

NetBSD 1.5.2 Release Contents

The root directory of the NetBSD 1.5.2 release is organized as follows:

`.../NetBSD-1.5.2/`

`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 1.5.2 distribution.
`README.files`	README describing the distribution's contents.
`TODO`	NetBSD 's todo list (also somewhat incomplete and out of date).
`patches/`	Post-release source code patches.
`source/`	Source distribution sets; see below.

In addition to the files and directories listed above, there is one directory per architecture, for each of the architectures for which NetBSD 1.5.2 has a binary distribution. There are also `README.export-control` files sprinkled liberally throughout the distribution tree, which point out that there are some portions of the distribution that may be subject to export regulations of the United States, e.g. code under `src/crypto` and `src/sys/crypto`. It is your responsibility to determine whether or not it is legal for you to export these portions and to act accordingly.

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.
22.3 MB gzipped, 98.8 MB uncompressed

pkgsrc This set contains the "pkgsrc" sources, which contain the infrastructure to build third-party packages.
7.4 MB gzipped, 73.0 MB uncompressed

sharesrc This set contains the "share" sources, which include the sources for the man pages not associated with any particular program; the sources for the typesettable document set; the dictionaries; and more.
3.3 MB gzipped, 13.2 MB uncompressed

src This set contains all of the base NetBSD 1.5.2 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
24.8 MB gzipped, 123.1 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.5.2 kernel for all architectures; `config`(8); and `dbsym`(8).
18.0 MB gzipped, 90.9 MB uncompressed

xsrc This set contains the sources to the X Window System.
78.1 MB gzipped, 393.6 MB uncompressed

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

The source sets are distributed as compressed tar files. They may be unpacked into `/usr/src` with the command:

```
# ( cd / ; tar -zxpf - ) < set_name.tgz
```

The `sets/Split/` subdirectory contains split versions of the source sets for those users who need to load the source sets from floppy or otherwise need a split distribution. The split sets are named `set_name.`*xx* where `set_name` is the distribution set name, and *xx* is the sequence number of the file, starting with ‘‘aa’’ for the first file in the distribution set, then ‘‘ab’’ for the next, and so on. All of these files except the last one of each set should be exactly 240,640 bytes long. (The last file is just long enough to contain the remainder of the data for that distribution set.)

The split distributions may be reassembled and extracted with **cat** as follows:

```
# cat set_name.?? | ( cd / ; tar -zxpf - )
```

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

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

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

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

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

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

NetBSD/sparc subdirectory structure

The sparc-specific portion of the NetBSD 1.5.2 release is found in the `sparc` subdirectory of the distribution:
`.../NetBSD-1.5.2/sparc/`

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

`installation/`
- `floppy/` sparc boot and installation floppies; see below.
- `miniroot/` sparc miniroot file system image; see below.
- `misc/` Statically-linked versions of **gzip** (GNU gzip) and **gtar** (GNU tar).
- `netboot/` Server boot-file image for diskless machines.

Binary distribution sets

The NetBSD sparc binary distribution sets contain the binaries which comprise the NetBSD 1.5.2 release for the sparc. There are eight binary distribution sets. The binary distribution sets can be found in the `sparc/binary/sets` subdirectory of the NetBSD 1.5.2 distribution tree, and are as follows:

base The NetBSD 1.5.2 sparc **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional. It includes shared library support, and excludes everything described below.
17.5 MB gzipped, 43.7 MB uncompressed

comp Things needed for compiling programs. This set includes the system include files (`/usr/include`) and the various system libraries (except the shared libraries, which are included as part of the **base** set). This set also includes the manual pages for all of the utilities it contains, as well as the system call and library manual pages.
14.1 MB gzipped, 51.9 MB uncompressed

etc This distribution set contains the system configuration files that reside in `/etc` and in several other places. This set *must* be installed if you are installing the system from scratch, but should *not* be used if you are upgrading.
0.1 MB gzipped, 0.6 MB uncompressed

games This set includes the games and their manual pages.
3.0 MB gzipped, 7.2 MB uncompressed

kern This set contains a NetBSD/sparc 1.5.2 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
1.2 MB gzipped, 2.8 MB uncompressed

man This set includes all of the manual pages for the binaries and other software contained in the **base** set. Note that it does not include any of the manual pages that are included in the other sets.
5.0 MB gzipped, 19.9 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
2.6 MB gzipped, 10.1 MB uncompressed

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.
1.3 MB gzipped, 4.4 MB uncompressed

NetBSD maintains its own set of sources for the X Window System in order to assure tight integration and compatibility. These sources are based on XFree86, and tightly track XFree86 releases. The binaries shipped with NetBSD 1.5.2 are based on XFree86 version 3.3.6. NetBSD is in the process of moving to XFree86 version 4, and the X source set actually contains source for both XFree86 3.3.6 and XFree86 4, and the ability to decide at compile-time which one to build and install. The X Window System binary sets distributed with NetBSD are:

xbase The basic files needed for a complete X client environment. This does not include the X servers.
2.8 MB gzipped, 8.2 MB uncompressed

xcomp The extra libraries and include files needed to compile X source code.
1.8 MB gzipped, 8.3 MB uncompressed

xcontrib Programs that were contributed to X.
0.2 MB gzipped, 0.6 MB uncompressed

xfont Fonts needed by X.
6.2 MB gzipped, 7.5 MB uncompressed

xserver The Xsun, Xsun24, XsunMono and Xprt servers with man pages.
3.3 MB gzipped, 8.5 MB uncompressed

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

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

Note: The distribution cannot be extracted using the SunOS **`tar`** program so statically-linked SunOS versions of **`gzip`** (GNU gzip) and **`gtar`** (GNU tar) are provided in `sparc/installation/misc`.

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

NetBSD/sparc System Requirements and Supported Devices

Supported machines

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

  (and possibly more)

  Note: NetBSD doesn't support the VME interface on the FORCE models
- sun4m
  - 4/600
  - SPARC Classic
  - SPARC LX
  - SPARCstation 4

- SPARCstation/server 5
- SPARCstation/server 10
- SPARCstation/server 20
- Voyager
- JavaStation 1

Note: All known MBus modules are supported

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

  (and possibly more)

Machines not supported by NetBSD/sparc

- sun4d
  - SPARCcenter 1000
  - SPARCcenter 2000
- sun4u and other UltraSPARC based machines

  These are supported by the NetBSD sparc64 port. Refer to
  http://www.netbsd.org/Ports/sparc64/index.html
- Solbourne machines
- Other JavaStation models

The minimal configuration requires 4 MB of RAM and ~60 MB of disk space. To install the entire system requires much more disk space, and to run X or compile the system, more RAM is recommended, as NetBSD with 4 MB of RAM feels like Solaris with 4 MB of RAM - slow. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.

Supported devices

- sun4c and sun4m video; Sbus card, on-board, or optional
  - *bwtwo*; 1 bpp framebuffer. On-board and Sbus card (MG1 and MG2)
  - *cgthree*; 8 bpp framebuffer. On-board and Sbus card
  - *cgsix*; 8 bpp accelerated framebuffer. On-board and Sbus card (GX, GX+, TGX, and TGX+)
  - *cgfourteen*; SX 24 bpp accelerated framebuffer. On-board SPARCstation 10SX and SPARCstation 20, requires optional 4 or 8 MB VSIMM, limited to *cgthree* emulation
  - *tcx*; S24 accelerated color framebuffer. On-board SPARCstation 4 (8 bpp) and optional on SPARCstation 5 (24 bpp), limited to *cgthree* emulation
  - *pnozz*; Weitek p9100 24 bpp accelerated framebuffer. On-board Tadpole SPARCbook 3GS and 3GX, limited to *cgthree* emulation

- sun4 video; VME card, on-board, or P4 (supported only on 4/100 and 4/300 series machines)
  - *bwtwo*; 1 bpp framebuffer. On-board and P4
  - *cgtwo*; 8 bpp framebuffer. VME card
  - *cgfour*; 8 bpp framebuffer with overlay. P4
  - *cgsix*; LEGO 8 bpp accelerated framebuffer. P4
  - *cgeight*; 24 bpp framebuffer with overlay. P4 (not well tested)
- Network interfaces
  - *be*; 100 Mbps half-duplex ‘‘Big Mac’’ ethernet. Sbus card
  - *en*; Efficient Networks, Inc. ATM interface. Sbus card
  - *hme*; 100 Mbps full-duplex ‘‘Happy Meal’’ ethernet. Sbus card (not well tested)
  - *ie*; 10 Mbps Intel 82586 ethernet. On-board (4/100 and 4/200, may not be working properly on 4/400) and VME card
  - *le*; 10 Mbps AMD 9790 Lance ethernet. On-board (4/300, sun4c, and sun4m) and Sbus card
  - *qec*; One 100 Mbps be interface and four 10 Mbps Mace ethernet interfaces. Sbus card
  - *qfe*; quad 100 Mbps full-duplex ‘‘Happy Meal’’ ethernet (four ethernet interfaces). Sbus card (not well tested)
  - misc: various ethernet interfaces are found on multi-function Sbus cards (e.g. along with SCSI), the ethernet interfaces work even if the other functions do not
- SCSI interfaces
  - *esp*; 5 MB/s and 10 MB/s (Fast) NCR 53C9x SCSI controller. On-board (sun4c, sun4m, and 4/300) and Sbus card
  - *isp*; 10 MB/s (Fast) and 20 MB/s (Fast+Wide) Qlogic ISP 10x0 family. Sbus card
  - *si*; 5 MB/s NCR 5380 SCSI controller. VME card (Interrupt driven and polled DMA)
  - *sw*; 5 MB/s NCR 5380 ‘‘SCSI Weird’’ controller. Onboard 4/110 (polled DMA only)
  - misc: SCSI functionality is sometimes found on multi-function Sbus cards (e.g. along with ethernet), the SCSI *esp* and *isp* portions of those cards should work, even if the other functions do not.

    Note: See **http://saturn.tlug.org/suncdfaq/** for information on bootable CD-ROMs
- VME disk interfaces
  - *xd*; Xylogics 7053 VME/SMD disk controller
  - *xy*; Xylogics 450/451 VME disk controller

  Note: VME/IPI disks are not supported
- Floppy drives
  - *fd*; floppy disk drive. On-board sun4c and sun4m
  - *sd*; SCSI floppy drive, acts as a SCSI drive to kernel. found with some SPARCBook 3 machines
- Serial interfaces
  - On-board serial ports based on Zilog 8530 and Zilog 85C30 (can be used as console if needed)
- Input devices
  - Sun keyboard and mouse
  - SPARCBook 3 keyboard and joystick
  - Third-party keyboards, mice, and trackballs
- Audio

- *audioamd*; AMD 7930-based 8000 Hz, 8 bit, mono, mu-law. On-board sun4c and SPARC Classic
- *audiocs*; CS4231-based 44 KHz, 16 bit, stereo, PCM. On-board SPARCStation 5 and optional on SPARCStation 4 (does not support audio input)

- Misc cards
  - *mtty / mbpp*; Magma Serial/Parallel ports. Sbus card
  - *xbox*; Sun Sbus expander device

Unsupported devices

- Use of more than one processor in sun4m machines (it boots, but only uses one processor)
- DBRIe-based audio and ISDN. On-board some sun4m machines (such as SS10, SS20, SPARC LX, and Tadpole SPARCbook)
- *sc*; Sun SCSI. VME card
- *bpp*; Bi-directional parallel port. On-board and Sbus card
- *cgtwelve*; I.e. GS, 24 bpp framebuffer. Sbus card
- *leo*; I.e. ZX and TZX, 24 bpp accelerated framebuffer. Sbus card
- *nell*; Sbus-to-PCMCIA bridge

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- CD-ROM
- MS-DOS floppy
- FTP
- Remote NFS partition
- Tape
- Existing NetBSD partitions, if performing an upgrade

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

CD-ROM Find out where the distribution set files are on the CD-ROM.

Proceed to the instruction on installation.

MS-DOS floppy Count the number of `set_name.xx` files that make up the distribution sets you want to install or upgrade. You will need one sixth that number of 1.44 MB floppies.

Format all of the floppies with MS-DOS. Do *not* make any of them bootable MS-DOS floppies, i.e. don't use `format /s` to format them. (If the floppies are bootable, then the MS-DOS system files that make them bootable will take up some space, and you won't be able to fit the distribution set parts on the disks.) If you're using floppies that are formatted for MS-DOS by their manufacturers, they probably aren't bootable, and you can use them out of the box.

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

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

FTP The preparations for this installation/upgrade method are easy; all you need to do is make sure that there's an FTP site from which you can retrieve the NetBSD distribution when you're about to install or upgrade. You need to know the numeric IP address of that site, and, if it's not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself. If you don't have access to a functioning nameserver during installation, the IP address of **ftp.netbsd.org** is `204.152.184.75` (as of April, 2001).

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

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

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 of 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 the server is not on a network directly connected to the machine on which you're installing or upgrading NetBSD, you need to know the numeric IP address of the router closest to the NetBSD machine. Finally, you need to know the numeric IP address of the NetBSD machine itself.

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

Note: This method of installation is recommended only for those already familiar with using BSD network configuration and management commands. If you aren't, this documentation should help, but is not intended to be all-encompassing.

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

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

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

where *`tape_device`* is the name of the tape device that describes the tape drive you're using; possibly `/dev/rst0`, or something similar, but it will vary from system to system. (If you can't figure it out, ask your system administrator.) In the above example, *`dist_directories`* are the distribution sets' directories, for the distribution sets you wish to place on the tape. For instance, to put the **misc, 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-1.5.2
# cd sparc/binary
# tar -cf tape_device misc etc kern
```

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.

Upgrade If you are upgrading NetBSD, you also have the option of installing NetBSD by putting the new distribution sets somewhere in your existing file system, and using them from there. To do that, you must do the following:

Place the distribution sets you wish to upgrade somewhere in your current file system tree. Please note that the `/dev` on the floppy used for upgrades only knows about `wd0`, `wd1`, `sd0`, `sd1`, and `sd2`. If you have more than two IDE drives or more than three SCSI drives, you should take care not to place the sets on the high-numbered drives.

At a bare minimum, you must upgrade the **base** and **kern** binary distributions, and so must put the **base** and **kern** sets somewhere in your file system. If you wish, you can do the other sets, as well, but you should *not* upgrade the **etc** distribution; it contains contains system configuration files that you should review and update by hand.

Once you have done this, you can proceed to the next step in the upgrade process, actually upgrading your system.

Preparing your System for NetBSD installation

Configuring your PROM

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

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

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

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

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

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

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

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

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

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

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

```
ok setenv sd-targets 01234567
```

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

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

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

The machines with OpenBoot PROM 2 (SPARCstation/server 2, ELC, IPX, and all sun4m models) have s similar SCSI target mapping in the form of a **`devalias`** entry. That is, the device alias **`disk`** is shorthand for the disk at SCSI ID 3 on the internal SCSI controller. Normally, the **`disk`** device alias is what the PROM uses as the default boot device, i.e. in the absence of a *`device`* argument to the **`boot`** command. Note that there are also pre-configured device alias entries for **`disk0, disk1, disk2`** and **`disk3`**, which are in fact a one-to-one mapping to the SCSI targets **`0`** to **`3`** (all on the internal SCSI controller).

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

Determining how to access your SCSI disk from the PROM

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

To calculate the parameters for sunmon and OpenBoot PROM 1:

- *c* specifies the SCSI controller number (first is 0, second is 1, ...)
- *u* the hexadecimal number obtained from evaluating the expression *(8 ∗ TARGET) + LUN*
- *p* the partition number from which to boot ('`0`' = '`a`', '`1`' = '`b`', etc.)

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

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

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

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

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

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

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

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

```
ok boot disk:b
```

To boot from a CD-ROM (OpenBoot PROM 2 assumes the CD-ROM is at target 6), one would use:

```
ok boot cdrom
```

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

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

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

Deciding on partition sizes

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

Asumming a classic partition scheme with `/` (root) and `/usr` filesystems, a comfortable size for the NetBSD root filesystem partition is about 20 MB; a good initial size for the swap partition is twice the amount of physical memory in your machine (though, unlike SunOS 4.x, there are no restrictions on the size of the swap partition that would render part of your memory unusable). A full binary installation takes about 60 MB in `/usr`.

Most sparc systems have trouble booting if the root partition extends beyond the first 2 GB of your disk.

Configuration of network interfaces

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

If automatic detection is not available or not working properly in your environment, you may have to specify the type connection using the **media** parameter of `ifconfig`(8). During installation, you'll get the opportunity to specify the appropriate medium. Use `10base5` or `AUI` to select the AUI connector, or `10baseT` or `UTP` to select the UTP connector. Fast ethernet interfaces default to `auto`, which usually does not detect properly and runs at '10BaseT' speed. The options are `10baseT`, `10baseTX`, and `auto`. The *hme* and *qfe* interfaces also allow `10baseT-FDX` and `100baseT-FDX`.

Installing the NetBSD System

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

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

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

Installing NetBSD by placing a bootable filesystem on a partitioned hard drive

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

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

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

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

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

```
# halt
```

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

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

OpenBoot PROM 1	`ok` **`boot sd(,,1)netbsd -s`**
OpenBoot PROM 2	`ok` **`boot disk:b netbsd -s`**

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

Installing NetBSD by using a bootable CD-ROM

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

http://www.netbsd.org/Documentation/bootcd.html#sparcimage

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

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

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

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

sunmon	`>` **`b sd(,30,)`**
OpenBoot PROM 1	`ok` **`boot sd(,30,)`**
OpenBoot PROM 2	`ok` **`boot cdrom`**

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

Installing NetBSD by using the bootable floppies

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

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

```
# fdformat
```

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

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

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

Insert the second floppy and run the following commands:

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

Be sure to label your floppies so you can later identify them.

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

OpenBoot PROM 1	`ok` **`boot fd(,,1)`**
OpenBoot PROM 1 (alternate)	`ok` **`boot /fd`**
OpenBoot PROM 2	`ok` **`boot floppy`**

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

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

```
ok eject
ok go
```

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

Installing NetBSD by using a netboot setup

First, you must setup a netboot server to provide the services and files your client needs. If you are using a NetBSD system as the netboot server, have a look at the `diskless`(8) manual page for a more detailed guide on how to proceed with this. If the server runs another operating system, you should consult the NetBSD Diskless HOW-TO, which will walk you through the steps necessary to configure a netboot server on a variety of platforms.

http://www.netbsd.org/Documentation/network/netboot/

This section will briefly describe what happens during a Sparc netboot, and the next section will give a brief set of directions on how to set up your server. Your Sparc initially broadcasts a RARP request, and expects a server to reply with your client's IP address. Next, it downloads a second stage bootstrap program via TFTP from the server that responded to the RARP request. It will look for a filename composed of the machine's IP address followed by the machine's architecture, separated by a period. For example, a sun4c machine which has been assigned IP address 130.115.144.11, will make an TFTP request for `8273900B.SUN4C`. This file is a symbolic link to the second-stage boot program, which should be located in a place where the TFTP daemon can find it (remember, many TFTP daemons run in a chroot'ed environment).

The boot program to use is `installation/netboot/boot.net` Alternatively, you can find the bootstrap program in `/usr/mdec/boot.net` in the NetBSD/sparc distribution.

Note: The only difference between `/usr/mdec/boot.net` and `/usr/mdec/boot` is in the a.out header, which has been magically transformed in a way that makes it usable on all versions of Sparc PROMs.

After the boot program has been loaded into memory and given control by the PROM, it starts locating the machine's remote root directory through the BOOTPARAM protocol. First a BOOTPARAM WHOAMI request is broadcast on the local net. The answer to this request should contain the client's name. This name is used in next step, a BOOTPARAM GETFILE request - sent to the server that responded to the WHOAMI request - requesting the name and address of the machine that will serve the client's root directory, as well as the path of the client's root on that server.

Finally, this information is used to issue a REMOTE MOUNT request to the client's root filesystem server, asking for an NFS file handle corresponding to the root filesystem. Once the NFS mount is complete, the boot program starts reading from the remote root filesystem in search of the kernel which is then read into memory.

Configuring your netboot server

Follow this section to configure a NetBSD machine as your netboot server. We will use 'CC:CC:CC:CC:CC:CC' as the MAC address (ethernet hardware address) of your netboot client machine that you wish to install NetBSD on. We use '192.168.1.10' as the IP address of your client, and 'client.test.net' as the name. The server name is 'server.test.net', and the path on the server to the NFS exported directory is /export/client/root although you may choose to use a different path. Create an /etc/ethers file with the following line:

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

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

```
192.168.1.10 client
```

If **rarpd** is currently running, restart it (kill -HUP), otherwise start **rarpd**:

```
# /usr/sbin/rarpd -a
```

Next, create /etc/bootparams with the following line:

```
client root=server:/export/client/root
```

If **rpc.bootparamd** is currently running, restart it (**kill -HUP**), otherwise start **rpc.bootparamd**:

```
# /usr/sbin/rpc.bootparamd
```

Now, make sure the bootloader in /tftpboot is linked to, or a copy of, boot.net and is named appropriately for your client (in this case, it would be C0A8010A.SUN4C).

```
# cp boot.net /tftpboot
# ln -s /tftpboot/boot.net /tftpboot/C0A8010A.SUN4C
```

Now, you should edit /etc/inetd.conf and make sure that the line starting with tftp is uncommented. You will need to **kill -HUP** the **inetd** process to enable tftp if this line was previously commented out. The next step is to set up NFS exports. 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 NFS services, you only need to **kill -HUP** the **mountd** process. Otherwise, you need to run:

```
# /usr/sbin/mountd
# /usr/sbin/nfsd -tun 4
```

Now, you need to place the files your netboot client will need. As noted above in the section **Preparing your System for NetBSD Installation** you have several options when choosing a location to store the distribution filesets. However, the easiest way is to put the distribution files into the exported directory for your client on the server.

```
# mv *.tgz /export/client/root
```

Your client will need a kernel to boot (use binary/kernel/netbsd.GENERIC.gz).

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

Your client will need installation tools (use `installation/netboot/rootfs.tgz`).

```
# gunzip < rootfs.tgz | (cd /export/client/root && tar -xpf -)
```

Now, netboot your Sparc from the server by entering the appropriate **`boot command at the monitor prompt.`** Depending on the PROM version in your machine, this command takes one of the following forms:

sunmon	> **`b le()netbsd`**
OpenBoot PROM 1	ok **`b le()netbsd`**
OpenBoot PROM 2	ok **`b net netbsd`**

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

Installing NetBSD by using a bootable tape

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

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

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

Now you can transfer the NetBSD 1.5.2. distribution sets from (`binary/sets`) to the tape by using a series of additional

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

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

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

sunmon	> **`b st()`**
OpenBoot PROM 1	ok **`boot st()`**
OpenBoot PROM 2	ok **`boot tape`**

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

Running the installation scripts

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

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

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

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

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

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

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

The installation script goes through the following phases:

- determination of the disk on which to install NetBSD
- checking/creation of the partition information on the disk
- creating and mounting the NetBSD filesystems
- setup of network configuration
- extraction of the distribution tar files
- installation of bootstrap programs

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

```
# mount -u /dev/sd0a /
```

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

OpenBoot PROM 1	`ok` **setenv boot-from sd(0,0,0)netbsd**
OpenBoot PROM 2	`ok` **setenv boot-file netbsd**
	`ok` **setenv boot-device disk:a**

Congratulations, you have successfully installed NetBSD 1.5.2.

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state, with the most important ones described below.

1. Configuring `/etc/rc.conf`

   If you or the installation software haven't done any configuration of `/etc/rc.conf` (**sysinst** usually will), the system will drop you into single user mode on first reboot with the message

   ```
   /etc/rc.conf is not configured. Multiuser boot aborted.
   ```

   and with the root file system (`/`) mounted read-write. When the system asks you to choose a shell, simply press RETURN to get to a `/bin/sh` prompt. If you are asked for a terminal type, respond with **sun** for a local console, or whatever is appropriate for your serial console (some systems display garbage with a **sun** terminal type, you may need to use **sun-ss5**) and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

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

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

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

If your `/usr` directory is on a separate partition and you do not know how to use **ed**, you will have to mount your `/usr` partition to gain access to **ex** or **vi**. Do the following:

```
# mount /usr
# export TERM=sun
```

If you have `/var` on a separate partition, you need to repeat that step for it. After that, you can edit `/etc/rc.conf` with `vi`(1). When you have finished, type **exit** at the prompt to leave the single-user shell and continue with the multi-user boot.

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

```
ifconfig_le0="inet 123.45.67.89 netmask 255.255.255.0"
ifconfig_de0="inet 123.45.67.89 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.

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. If you're using the machine in a networked environment, you should create an account for yourself (see below) and protect it and the ''root'' account with good passwords. Unless you have connected an unusual terminal device as the console you can just press RETURN when it prompts for `Terminal type? [...]`.

3. Adding accounts

   Use the `useradd`(8) command to add accounts to your system, *do not* edit `/etc/passwd` directly. See `useradd`(8) for more information on how to add a new user to the system.

4. The X Window System

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

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

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. This automatically handles any changes necessary to make the software run on NetBSD, retrieval and installation of any other packages on which the software may depend, and simplifies installation (and deinstallation), both from source and precompiled binaries.

- More information on the package system is at
  http://www.netbsd.org/Documentation/software/packages.html

- A browsable listing of available packages is at
  ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/README.html

- Precompiled binaries can be found at
  ftp://ftp.netbsd.org/pub/NetBSD/packages,
  usually in the `1.5.2/sparc/All` subdir. You can install them with the following commands:

  ```
  # export PKG_PATH=ftp://ftp.netbsd.org/pub/NetBSD/packages/1.5.2/sparc/All
  # pkg_add -v tcsh
  # pkg_add -v cvs
  # pkg_add -v apache
  # pkg_add -v perl
  ...
  ```

  The above commands will install the tcsh shell, the CVS source code management system, the Apache web server and the perl programming language as well as all the packages they depend on.

- Package sources for compiling packages on your own can be obtained by retrieving the file
  ftp://ftp.netbsd.org/pub/NetBSD/NetBSD-current/tar_files/pkgsrc.tar.gz
  They are typically extracted into `/usr/pkgsrc` (though other locations work fine), with the commands:

  ```
  # mkdir /usr/pkgsrc
  # ( cd /usr/pkgsrc ; tar -zxpf - ) < pkgsrc.tar.gz
  ```

  After extracting, then see the `README` file in the extraction directory (e.g. `/usr/pkgsrc/README`) 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/mail/sendmail.cf` file will almost definitely need to be adjusted; files aiding in this can be found in `/usr/share/sendmail`. See the README file there for more information.
   - Edit `/etc/rc.local` to run any local daemons you use.
   - Many of the `/etc` files are documented in section 5 of the manual; so just invoking

     ```
     # man 5 filename
     ```

     is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

The upgrade to NetBSD 1.5.2 is a binary upgrade; it can be quite difficult to advance to a later version by recompiling from source due primarily to interdependencies in the various components.

To do the upgrade, you must boot from the installer kernel using one of the methods described above. To do the upgrade, you must have the boot floppy available. You must also have at least the **base** and **kern** binary distribution sets available, so that you can upgrade with them, using one of the upgrade methods described above. Finally, you must have sufficient disk space available to install the new binaries. Since the old binaries are being overwritten in place, you only need space for the new binaries, which weren't previously on the system. If you have a few megabytes free on each of your root (`/`) and `/usr` partitions, you should have enough space.

Since upgrading involves replacing the boot blocks on your NetBSD partition, the kernel, and most of the system binaries, it has the potential to cause data loss. You are strongly advised to *back up any important data on your disk*, whether on the NetBSD partition or on another operating system's partition, before beginning the upgrade process.

The upgrade procedure using the **sysinst** tool is similar to an installation, but without the hard disk partitioning. The original `/etc` directory is renamed to `/etc.old`, and no attempt is made to merge any of the previous configuration into the new system except that the previous `/etc/fstab` file is copied into the new configuration. Getting the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for how to do this. Also, some sanity checks are done, i.e. file systems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 1.5.2 system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in `/dev`. If you've changed the contents of `/dev` by hand, you will need to be careful about this, but if not, you can just cd into `/dev`, and run the command:

```
# sh MAKEDEV all
```

You must also deal with certain changes in the formats of some of the configuration files. The most notable change is that the options given to many of the file systems in `/etc/fstab` have changed, and some of the file systems have changed names. To find out what the new options are, it's suggested that you read the manual page for the file system's mount commands, for example `mount_nfs`(8) for NFS.

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

NetBSD/sparc has switched its executable format from the old a.out format to ELF, the now more commonly used and supported format. Your old binaries will continue to work just fine. The installation procedure will try to take the necessary steps to accomplish this. The most important step is to move the old a.out shared libraries in `/usr/lib` and `/usr/X11R6/lib` (if X was installed) to `/emul/aout`, where they will be automatically found if an older a.out binary is executed. Sysinst will use an existing `/emul` and / or `/emul/aout` directory if available, and will create it (as a symbolic link to `/usr/aout`) if necessary.

If you already had a `/emul` directory, or a symbolic link by that name, sysinst should rename it and tell you about it.

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 1.5.2.

General issues

- `/etc/rc` modified to use `/etc/rc.d/*`

  Prior to NetBSD 1.5, `/etc/rc` was a traditional BSD style monolithic file; each discrete program or substem from `/etc/rc` and `/etc/netstart` has been moved into separate scripts in `/etc/rc.d/`.

  At system startup, `/etc/rc` uses `rcorder`(8) to build a dependency list of the files in `/etc/rc.d` and then executes each script in turn with an argument of 'start'. Many `rc.d` scripts won't start unless the appropriate `rc.conf`(5) entry in `/etc/rc.conf` is set to 'YES.'

  At system shutdown, `/etc/rc.shutdown` uses `rcorder`(8) to build a dependency list of the files in `/etc/rc.d` that have a "KEYWORD: shutdown" line, reverses the resulting list, and then executes each script in turn with an argument of 'stop'. The following scripts support a specific shutdown method: `cron`, `inetd`, `local`, and `xdm`.

Local and third-party scripts may be installed into `/etc/rc.d` as necessary. Refer to the other scripts in that directory and `rc`(8) for more information on implementing `rc.d` scripts.

Issues affecting an upgrade from NetBSD 1.4 or prior

- `named`(8) leaks version information

  Previous releases of NetBSD disabled a feature of `named`(8) where the version number of the server could be determined by remote clients. This feature has not been disabled in NetBSD 1.5, because there is a `named.conf`(5) option to change the version string:

  ```
  option {
          version "newstring";
  };
  ```

- `sysctl`(8) pathname changed

  `sysctl`(8) is moved from `/usr/sbin/sysctl` to `/sbin/sysctl`. If you have hardcoded references to the full pathname (in shell scripts, for example) please be sure to update those.

- `sendmail`(8) configuration file pathname changed

  Due to `sendmail`(8) upgrade from 8.9.x to 8.10.x, `/etc/sendmail.cf` is moved to `/etc/mail/sendmail.cf`. Also, the default `sendmail.cf`(5) refers different pathnames than before. For example, `/etc/aliases` is now located at `/etc/mail/aliases`, `/etc/sendmail.cw` is now called `/etc/mail/local-host-names`, and so forth. If you have customized `sendmail.cf`(5) and friends, you will need to move the files to the new locations. See `/usr/share/sendmail/README` for more information.

Using online NetBSD documentation

Documentation is available if you first install the manual distribution set. Traditionally, the ''man pages'' (documentation) are denoted by '`name(section)`'. Some examples of this are

- `intro`(1),
- `man`(1),
- `apropros`(1),
- `passwd`(1), and
- `passwd`(5).

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter **apropos** *subject-word*

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

Administrivia

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

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

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

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

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

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

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

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

- Also, our thanks go to:

  Mike Hibler
  Rick Macklem
  Jan-Simon Pendry
  Chris Torek

  for answering lots of questions, fixing bugs, and doing the various work they've done.

- UC Berkeley's Experimental Computing Facility provided a home for sun-lamp in the past, people to look after it, and a sense of humor. Rob Robertson, too, has added his unique sense of humor to things, and for a long time provided the primary FTP site for NetBSD.
- Vixie Enterprises for hosting the NetBSD FTP, SUP, and WWW servers.
- Redback Networks, Inc. for hosting the NetBSD mail and GNATS server.
- The Helsinki University of Technology in Finland for hosting the NetBSD CVS server.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.

- The many organisations 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.
- Dave Burgess burgess@cynjut.infonet.net has been maintaining the 386BSD/NetBSD/FreeBSD FAQ for quite some time, and deserves to be recognized for it.
- The following individuals and organizations (each in alphabetical order) have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it:

  Steve Allen
  Jason Birnschein
  Mason Loring Bliss
  Jason Brazile
  Mark Brinicombe
  David Brownlee
  Simon Burge
  Dave Burgess
  Ralph Campbell
  Brian Carlstrom
  James Chacon
  Bill Coldwell
  Charles Conn
  Tom Coulter
  Charles D. Cranor
  Christopher G. Demetriou
  Scott Ellis
  Hubert Feyrer
  Castor Fu
  Greg Gingerich
  William Gnadt
  Michael Graff
  Guenther Grau
  Ross Harvey
  Charles M. Hannum
  Michael L. Hitch
  Kenneth Alan Hornstein
  Jordan K. Hubbard
  Søren Jørvang
  Scott Kaplan
  Noah M. Keiserman
  Harald Koerfgen
  John Kohl
  Chris Legrow
  Ted Lemon
  Norman R. McBride
  Neil J. McRae
  Perry E. Metzger
  Toru Nishimura
  Herb Peyerl
  Mike Price
  Dave Rand

Michael Richardson
Heiko W. Rupp
Brad Salai
Chuck Silvers
Thor Lancelot Simon
Bill Sommerfeld
Paul Southworth
Eric and Rosemary Spahr
Ted Spradley
Kimmo Suominen
Jason R. Thorpe
Steve Wadlow
Krister Walfridsson
Jim Wise
Reinoud Zandijk
Christos Zoulas

AboveNet Communications, Inc.
Advanced System Products, Inc.
Avalon Computer Systems
Bay Area Internet Solutions
Brains Corporation, Japan
Canada Connect Corporation
Co-operative Research Centre for Enterprise Distributed Systems Technology
Demon Internet, UK
Digital Equipment Corporation
Distributed Processing Technology
Easynet, UK
Free Hardware Foundation
Innovation Development Enterprises of America
Internet Software Consortium
MS Macro System GmbH, Germany
Numerical Aerospace Simulation Facility, NASA Ames Research Center
Piermont Information Systems Inc.
Precedence Technologies Ltd
Salient Systems Inc.
VMC Harald Frank, Germany
Warped Communications, Inc.
Whitecross Database Systems Ltd.

(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 mentioned, tell us!)

We are...

(in alphabetical order)

The NetBSD core group:

Alistair Crooks	agc@netbsd.org
Jun-ichiro itojun Hagino	itojun@netbsd.org
Frank van der Linden	fvdl@netbsd.org
Luke Mewburn	lukem@netbsd.org
Christos Zoulas	christos@netbsd.org

The portmasters (and their ports):

Mark Brinicombe	mark@netbsd.org	**arm32**
Simon Burge	simonb@netbsd.org	**pmax**
Jeremy Cooper	jeremy@netbsd.org	**sun3x**
Matt Fredette	fredette@netbsd.org	**sun2**
Ross Harvey	ross@netbsd.org	**alpha**
Jun-ichiro itojun Hagino	itojun@netbsd.org	**sh3**
Ben Harris	bjh21@netbsd.org	**arm26**
Eduardo Horvath	eeh@netbsd.org	**sparc64**
Darrin Jewell	dbj@netbsd.org	**next68k**
Søren Jørvang	soren@netbsd.org	**cobalt**
Søren Jørvang	soren@netbsd.org	**sgimips**
Wayne Knowles	wdk@netbsd.org	**mipsco**
Paul Kranenburg	pk@netbsd.org	**sparc**
Anders Magnusson	ragge@netbsd.org	**vax**
Minoura Makoto	minoura@netbsd.org	**x68k**
Phil Nelson	phil@netbsd.org	**pc532**
Tohru Nishimura	nisimura@netbsd.org	**luna68k**
NONAKA Kimihiro	nonaka@netbsd.org	**prep**
Scott Reynolds	scottr@netbsd.org	**mac68k**
Kazuki Sakamoto	sakamoto@netbsd.org	**bebox**
Noriyuki Soda	soda@netbsd.org	**arc**
Wolfgang Solfrank	ws@netbsd.org	**ofppc**
Ignatios Souvatzis	is@netbsd.org	**amiga**
Jonathan Stone	jonathan@netbsd.org	**pmax**
Shin Takemura	takemura@netbsd.org	**hpcmips**
Jason Thorpe	thorpej@netbsd.org	**alpha**
Jason Thorpe	thorpej@netbsd.org	**hp300**
Tsubai Masanari	tsubai@netbsd.org	**macppc**
Tsubai Masanari	tsubai@netbsd.org	**newsmips**
Izumi Tsutsui	tsutsui@netbsd.org	**news68k**
Frank van der Linden	fvdl@netbsd.org	**i386**
Leo Weppelman	leo@netbsd.org	**atari**
Nathan Williams	nathanw@netbsd.org	**sun3**
Steve Woodford	scw@netbsd.org	**mvme68k**

The NetBSD 1.5.2 Release Engineering team:

Chris G. Demetriou	cgd@netbsd.org
Håvard Eidnes	he@netbsd.org

Ted Lemon	mellon@netbsd.org
John Hawkinson	jhawk@netbsd.org
Perry Metzger	perry@netbsd.org
Curt Sampson	cjs@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Todd Vierling	tv@netbsd.org

Developers and other contributors:

Nathan Ahlstrom	nra@NetBSD.org
Steve Allen	wormey@netbsd.org
Julian Assange	proff@netbsd.org
Lennart Augustsson	augustss@netbsd.org
Christoph Badura	bad@netbsd.org
Dieter Baron	dillo@netbsd.org
Robert V. Baron	rvb@netbsd.org
Jason Beegan	jtb@netbsd.org
Erik Berls	cyber@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Rafal Boni	rafal@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org
Allen Briggs	briggs@netbsd.org
Aaron Brown	abrown@netbsd.org
Andrew Brown	atatat@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@netbsd.org
Dave Burgess	burgess@cynjut.infonet.net
Robert Byrnes	byrnes@netbsd.org
D'Arcy J.M. Cain	darcy@netbsd.org
Dave Carrel	carrel@netbsd.org
James Chacon	jmc@netbsd.org
Bill Coldwell	billc@netbsd.org
Julian Coleman	jdc@netbsd.org
Chuck Cranor	chuck@netbsd.org
Aidan Cully	aidan@netbsd.org
Johan Danielsson	joda@netbsd.org
Matt DeBergalis	deberg@netbsd.org
Rob Deker	deker@netbsd.org
Chris G. Demetriou	cgd@netbsd.org
Jaromír Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Emmanuel Dreyfus	manu@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Håvard Eidnes	he@netbsd.org

Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Jason R. Fink	jrf@netbsd.org
Thorsten Frueauf	frueauf@netbsd.org
Castor Fu	castor@netbsd.org
Ichiro Fukuhara	ichiro@netbsd.org
Brian R. Gaeke	brg@dgate.org
Thomas Gerner	thomas@netbsd.org
Simon J. Gerraty	sjg@netbsd.org
Justin Gibbs	gibbs@netbsd.org
Adam Glass	glass@netbsd.org
Michael Graff	explorer@netbsd.org
Brad Grantham	grantham@tenon.com
Brian C. Grayson	bgrayson@netbsd.org
Matthew Green	mrg@netbsd.org
Juergen Hannken-Illjes	hannken@netbsd.org
Charles M. Hannum	mycroft@netbsd.org
Eric Haszlakiewicz	erh@netbsd.org
John Hawkinson	jhawk@netbsd.org
HAYAKAWA Koichi	haya@netbsd.org
René Hexel	rh@netbsd.org
Michael L. Hitch	mhitch@netbsd.org
Christian E. Hopps	chopps@netbsd.org
Ken Hornstein	kenh@netbsd.org
Marc Horowitz	marc@netbsd.org
Nick Hudson	skrll@netbsd.org
Martin Husemann	martin@netbsd.org
Dean Huxley	dean@netbsd.org
Bernardo Innocenti	bernie@netbsd.org
ITOH Yasufumi	itohy@netbsd.org
IWAMOTO Toshihiro	toshii@netbsd.org
Matthew Jacob	mjacob@netbsd.org
Lonhyn T. Jasinskyj	lonhyn@netbsd.org
Chris Jones	cjones@netbsd.org
Takahiro Kambe	taca@netbsd.org
Antti Kantee	pooka@netbsd.org
Masanori Kanaoka	kanaoka@netbsd.org
KAWAMOTO Yosihisa	kawamoto@netbsd.org
Mario Kemper	magick@netbsd.org
Lawrence Kesteloot	kesteloo@cs.unc.edu
Thomas Klausner	wiz@netbsd.org
Klaus Klein	kleink@netbsd.org
Wayne Knowles	wdk@netbsd.org
John Kohl	jtk@netbsd.org
Kevin Lahey	kml@netbsd.org
Johnny C. Lam	jlam@netbsd.org
Martin J. Laubach	mjl@netbsd.org

Ted Lemon	mellon@netbsd.org
Joel Lindholm	joel@netbsd.org
Mike Long	mikel@netbsd.org
Warner Losh	imp@netbsd.org
Tomasz Luchowski	zuntum@netbsd.org
Federico Lupi	federico@netbsd.org
Brett Lymn	blymn@netbsd.org
Paul Mackerras	paulus@netbsd.org
MAEKAWA Masahide	gehenna@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Neil J. McRae	neil@netbsd.org
Perry Metzger	perry@netbsd.org
der Mouse	mouse@netbsd.org
Joseph Myers	jsm@netbsd.org
Ken Nakata	kenn@netbsd.org
Bob Nestor	rnestor@netbsd.org
NISHIMURA Takeshi	nsmrtks@netbsd.org
NONAKA Kimihiro	nonaka@netbsd.org
Jesse Off	joff@netbsd.org
Tatoku Ogaito	tacha@netbsd.org
Masaru Oki	oki@netbsd.org
Atsushi Onoe	onoe@netbsd.org
Greg Oster	oster@netbsd.org
Herb Peyerl	hpeyerl@netbsd.org
Matthias Pfaller	matthias@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Michael Rauch	mrauch@netbsd.org
Waldi Ravens	waldi@moacs.indiv.nl.net
Darren Reed	darrenr@netbsd.org
Michael Richardson	mcr@netbsd.org
Tim Rightnour	garbled@netbsd.org
Gordon Ross	gwr@netbsd.org
Heiko W. Rupp	hwr@netbsd.org
SAITOH Masanobu	msaitoh@netbsd.org
Curt Sampson	cjs@netbsd.org
Wilfredo Sanchez	wsanchez@netbsd.org
Ty Sarna	tsarna@netbsd.org
SATO Kazumi	sato@netbsd.org
Matthias Scheler	tron@netbsd.org
Karl Schilke (rAT)	rat@netbsd.org
Konrad Schroder	perseant@netbsd.org
Reed Shadgett	dent@netbsd.org
Tim Shepard	shep@netbsd.org
Takao Shinohara	shin@netbsd.org
Takuya SHIOZAKI	tshiozak@netbsd.org
Chuck Silvers	chs@netbsd.org

Thor Lancelot Simon	tls@netbsd.org
Jeff Smith	jeffs@netbsd.org
Bill Sommerfeld	sommerfeld@netbsd.org
Bill Squier	groo@netbsd.org
Bill Studenmund	wrstuden@netbsd.org
Kevin Sullivan	sullivan@netbsd.org
SUNAGAWA Keiki	kei@netbsd.org
Kimmo Suominen	kim@netbsd.org
TAMURA Kent	kent@netbsd.org
Shin'ichiro TAYA	taya@netbsd.org
Matt Thomas	matt@netbsd.org
Christoph Toshok	toshok@netbsd.org
UCHIYAMA Yasushi	uch@netbsd.org
Shuichiro URATA	ur@netbsd.org
Todd Vierling	tv@netbsd.org
Aymeric Vincent	aymeric@netbsd.org
Paul Vixie	vixie@netbsd.org
Krister Walfridsson	kristerw@netbsd.org
Lex Wennmacher	wennmach@netbsd.org
Assar Westerlund	assar@netbsd.org
Todd Whitesel	toddpw@netbsd.org
Rob Windsor	windsor@netbsd.org
Dan Winship	danw@netbsd.org
Jim Wise	jwise@netbsd.org
Michael Wolfson	mbw@netbsd.org
Colin Wood	ender@netbsd.org
YAMAMOTO Takashi	yamt@netbsd.org
Yuji Yamano	yyamano@netbsd.org
Reinoud Zandijk	reinoud@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 the University of California, Berkeley and its contributors. This product includes software developed by The NetBSD Foundation, Inc. This product includes software developed by the NetBSD Foundation, Inc. and its contributors. This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory. This product includes software developed by Adam Glass and Charles Hannum. 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 Alistair G. Crooks. 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 Charles D. Cranor and Washington University. This product includes software developed by Charles D. Cranor. This product includes software developed by Charles 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 Hannum. This product includes software developed by Charles M. Hannum. This product includes software developed by Chris Provenzano. 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 David Jones and Gordon Ross. This product includes software developed by Dean Huxley. This product includes software developed by Eric S. Hvozda. This product includes software developed by Ezra Story. This product includes software developed by Gardner Buchanan. This product includes software developed by Gordon Ross. This product includes software developed by Gordon W. Ross and Leo Weppelman. This product includes software developed by Gordon W. Ross. This product includes software developed by Hauke Fath. This product includes software developed by HAYAKAWA Koichi. This product includes software developed by Hellmuth Michaelis and Joerg Wunsch. This product includes software developed by Herb Peyerl. This product includes software developed by Holger Veit and Brian Moore for use with "386BSD" and similar operating systems. This product includes software developed by Hubert Feyrer for the NetBSD Project. This product includes software developed by Iain Hibbert. This product includes software developed by Ian W.

Dall. This product includes software developed by Ignatios Souvatzis for the NetBSD Project. 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 John Polstra. This product includes software developed by Jonathan Stone and Jason R. Thorpe for the NetBSD Project. This product includes software developed by Jonathan Stone 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 Kenneth Stailey. This product includes software developed by Leo Weppelman. This product includes software developed by Lloyd Parkes. This product includes software developed by Manuel Bouyer. This product includes software developed by Marc Horowitz. 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 Martin Husemann and Wolfgang Solfrank. This product includes software developed by Mats O Jansson and Charles D. Cranor. This product includes software developed by Mats O Jansson. This product includes software developed by Matthias Pfaller. This product includes software developed by Michael L. Hitch. This product includes software developed by Niels Provos. This product includes software developed by Paul Kranenburg. This product includes software developed by Paul Mackerras. This product includes software developed by Peter Galbavy. This product includes software developed by Philip A. Nelson. This product includes software developed by Rodney W. Grimes. This product includes software developed by Roland C. Dowdeswell. This product includes software developed by Rolf Grossmann. This product includes software developed by Scott Bartram. This product includes software developed by SigmaSoft, Th. Lockert. 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 Theo de Raadt and John Brezak. This product includes software developed by Theo de Raadt. This product includes software developed by Tohru Nishimura for the NetBSD Project. This product includes software developed by TooLs GmbH. This product includes software developed by Winning Strategies, Inc. This product includes software developed by Zembu Labs, Inc. This product includes software developed by the Center for Software Science at the University of Utah. This product includes software developed by the Computer Systems Laboratory at the University of Utah. This product includes software developed by the University of Calgary Department of Computer Science and its contributors. This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman. This product includes software developed for the FreeBSD project. This product includes software developed for the Internet Software Consortium by Ted Lemon. This product includes software developed for the NetBSD Project by Frank van der Linden. This product includes software developed for the NetBSD Project by Jason R. Thorpe. This product includes software developed for the NetBSD Project by John M. Vinopal. This product includes software developed for the NetBSD Project by Matthias Drochner. This product includes software developed for the NetBSD Project by Matthieu Herrb. This product includes software developed for the NetBSD Project by Perry E. Metzger. This product includes software developed for the NetBSD Project by Piermont Information Systems Inc. This product includes software developed for the NetBSD Project by Ted Lemon. This product includes software developed by LAN Media Corporation and its contributors. This product includes software developed by Michael Graff for the NetBSD Project. This product includes software developed by Niklas Hallqvist, C Stone and Job de Haas. This product includes software developed by Eric Young (eay@mincom.oz.au). 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 University of Oregon. This product includes software developed by the University of Southern California and/or Information Sciences Institute. This product includes software developed by Internet Initiative Japan Inc. This product includes software developed by Reinoud Zandijk.

This product includes software developed by the Alice Group.

This product includes software developed by David Miller.

The End