NAME

INSTALL – Installation procedure for NetBSD/sparc64.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.6.1 on the *sparc64* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

- `.ps` PostScript.
- `.html` Standard Internet HTML.
- `.more` The enhanced text format used on UNIX-like systems by the `more`(1) and `less`(1) pager utility programs. This is the format in which the on-line *man* pages are generally presented.
- `.txt` Plain old ASCII.

You are reading the *PostScript* version.

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 fifty three different system architectures (ports), featuring seventeen machine architectures across eleven distinct CPU families, and is being ported to more. The NetBSD 1.6.1 release contains complete binary releases for thirty eight different system architectures. (The fifteen 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 1.6

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 1.6.

NetBSD 1.6 is an upgrade of NetBSD 1.5.3 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 1.6 was started are designated by version identifiers such as 1.6A, 1.6B, etc. These identifiers do not designate releases, but indicate major changes in internal kernel APIs. Note that the kernel from NetBSD 1.6 can *not* be used to upgrade a system running one of those intermediate development versions. Trying to use the NetBSD 1.6 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 1.6 to determine if a given feature is present or absent in 1.6. The development of 1.6 and the subsequent "point" releases is done on a separate branch in the CVS repository. The branch was created when the release cycle for 1.6 was started, and during the release cycle of 1.6 and its patch releases selected fixes and enhancements have been imported from the main development trunk.

Changes Between The NetBSD 1.5 and 1.6 Releases

The NetBSD 1.6 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, new and updated kernel subsystems, and many user-land enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

It is impossible to completely summarize over eighteen months of development that went into the NetBSD 1.6.1 release. Some highlights include:

Kernel

- Ports to new platforms including: algor, dreamcast, evbarm, hpcarm, hpcsh, newsmips, sandpoint, sgimips, sun2, and walnut.
- Unified Buffer Cache (UBC) removes size restriction of the file system's buffer cache to use all available RAM (if not otherwise used!) and improves overall system performance.
- Round-robin page colouring implemented for various ports for better cache utilisation, more deterministic run-time behaviour, and faster program execution.
- A rewritten SCSI middle layer to provide a cleaner interface between the different kernel layers, including a kernel thread to handle error recovery outside of the interrupt context. See `scsipi`(9).
- A new pipe implementation with significantly higher performance due to lower overheads, which uses the UVM Page Loan facility.
- New boot loader flags **-v** (*bootverbose*) and **-q** (*bootquiet*), to be used by kernel code to optionally print information during boot.
- An in-kernel boot time device configuration manager `userconf`(4), activated with the **-c** boot loader flag.
- A work-in-progress snapshot of ACPI support, based on the 20010831 snapshot of the Intel ACPICA reference implementation.
- USB 2.0 support, in the form of a preliminary driver for the `ehci`(4) host controller.
- Basic kernel support for IrDA in the form of the `irframe`(4) IrDA frame level driver. Serial dongles and the `oboe`(4) driver are currently supported.
- Kernel configuration files can be embedded into the kernel for later retrieval. Refer to `INCLUDE_CONFIG_FILE` in `options`(4) for more information.
- Many more kernel tunable variables added to `sysctl`(8).
- Linux binary emulation has been greatly improved, and now supports Linux kernel version 2.4.18.

Networking

- Hardware assisted IPv4 TCP and UDP checksumming and caching of the IPv6 TCP pseudo header. Support for checksum offloading on the DP83820 Gigabit Ethernet, 3Com 3c90xB, 3Com 3c90xC, and Alteon Tigon/Tigon2 Gigabit Ethernet cards.
- Zero-Copy for TCP and UDP transmit path achieved through page loaning code for **sosend**().
- In-kernel ISDN support, from the ISDN4BSD project.
- 802.1Q VLAN (virtual LAN) support. See `vlan`(4).
- IPFilter now supports IPv6 filtering.

- ndbootd(8) added; used to netboot NetBSD/sun2 machines.
- racoon(8) added; IKE key management daemon for IPsec key negotiation, from the KAME project.
- WEP encryption supported in ifconfig(8) and awi(4) driver.
- wi(4) and wiconfig(8) now support scanning for access points, and defaults to BSS instead of ad-hoc mode.
- Bridging support; currently only for ethernet. See bridge(4).
- In-kernel PPP over Ethernet (PPPoE) - RFC 2516, with much lower overhead than user-land PPPoE clients. See pppoe(4).
- ifwatchd(8) added; invokes up-script and down-script when a network interface goes up and down. Used by pppoe(4).

File system

- Enhanced stability of LFS version 2, the BSD log-structured file system.
- dump(8), dumpfs(8), fsck_ffs(8), fsirand(8), newfs(8), and tunefs(8) support a **-F** option to manipulate file system images in regular files.
- makefs(8) added; creates file system images from a directory tree. (Currently ffs only.)
- Enhanced **ffs_dirpref**() by Grigoriy Orlov, which noticeably improves performance on FFS file systems when creating directories, and subsequently manipulating them.
- Fixes for free block tracking and directory block allocation in FFS softdeps.
- Correctly support FFS file systems with a large number of cylinder groups.
- Fix the endian independant FFS (FFS_EI) support.
- newfs(8) calculates default block size from the file system size, and uses the largest possible cylinders/group (cpg) value if **-c** isn't given.
- dpti(4) driver added; an implementation of the DPT/Adaptec SCSI/I2O RAID management interface. Allows the use of the Linux versions of **dptmgr**, **raidutil**, **dptelog**, (etc).
- Support for Windows 2000 'NTFS' (NTFS5).
- Tagged queueing support for SCSI drivers based on the ncr53c9x controller.

Security

- Addition of a chroot(8) hierarchy for services including named(8), ntpd(8), and sshd(8).
- Additional passwd(5) ciphers: MD5, and DES with more encryption rounds. See passwd.conf(5).
- Several more code audits were performed.
- /etc/security performs many more checks and is far more flexible in how it monitors changes. See security.conf(5).

System administration and user tools

- sushi(8) added; a menu based system administration tool.
- pgrep(1) and pkill(1) added; find or signal processes by name or other attributes.
- System upgrades are made easier through the etcupdate(8) script which helps updating the /etc config files interactively, and the /etc/postinstall script which is provided to check for or fix configuration changes that have occurred in NetBSD.

- stat(1) added; a user interface to the information returned by the stat(2) system call.
- BSD sort(1) replaces GNU sort(1).
- The "stop" operation for rc.d(8) scripts waits until the service terminates before returning. This improves the reliability of "restart" operations as well.
- Swap devices can be removed at system shutdown by enabling swapoff in rc.conf(5).
- An optional watchdog timer which will terminate rc.shutdown(8) after the number of seconds provided in rcshutdown_timeout from rc.conf(5).

Miscellaneous

- Support for multibyte LC_CTYPE locales has been integrated from the Citrus project. Many Chinese, Japanese, Korean, and other encodings are now available.
- Full support for cross-compilation of the base system, even as a non-root user! src/build.sh is available for doing arbitrary cross-builds; see src/BUILDING for more information. At least 38 ports for the NetBSD 1.6.1 release were cross-built on a NetBSD/i386 system using this mechanism.
- Migrated the following CPU platforms to ELF: arm, and m68k (including amiga, hp300, mac68k, mvme68k, sun2, and x68k).
- Updates of most third party packages that are shipped in the base system to the following latest stable releases:
  - amd 6.0.6
  - BIND 8.3.3
  - binutils 2.11.2
  - bzip2 1.0.2
  - cvs 1.11
  - dhcp 3.0.1rc9
  - file 3.38
  - gcc 2.95.3
  - groff 1.16.1
  - Heimdal 0.4e
  - IPfilter 3.4.27
  - kerberos4 1.1
  - ksh from pdksh 5.2.14p2
  - less 374
  - nvi 1.79
  - OpenSSH 3.4
  - OpenSSL 0.9.6g
  - Postfix 1.1.11
  - ppp 2.4.0
  - routed 2.24
  - sendmail 8.11.6
  - tcpdump 3.7.1
- Many new packages in the *pkgsrc* system, including the latest open source desktop KDE3, OpenOffice, perl, Apache and many more. At the time of writing, there are over 3000 third party packages available in pkgsrc.
- Added AGP GART driver agp(4) for faster access to graphics boards.

- `init`(8) will create an mfs (memory based file system) `/dev` if `/dev/console` is missing.
- `vmstat`(8) displays kernel hash statistics with **-H** and **-h** *hash*.
- `wscons`(4) supports blanking of VGA consoles.

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.

sparc64 specific

NetBSD 1.6.1 is the second major release for the sparc64!

Some (but not all!) notable sparc64-specific improvements include:

- Support for the FAS336 SCSI controller, which is on-board on the Ultra 1e and Ultra 2, and on appropriate Sbus cards.
- Support for PCI based audio (*audiocs*) .
- Ability to be run in 32-bit mode on the NetBSD/sparc port (instead of NetBSD/sparc64) by using the former's `GENERIC_SUN4U` kernel.

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 1.6.1 Release Contents

The root directory of the NetBSD 1.6.1 release is organized as follows:

`.../NetBSD-1.6.1/`

`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 1.6.1 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 1.6.1 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.
55 MB gzipped, 247 MB uncompressed

pkgsrc This set contains the "pkgsrc" sources, which contain the infrastructure to build third-party packages.
12 MB gzipped, 94 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.
4 MB gzipped, 16 MB uncompressed

src This set contains all of the base NetBSD 1.6.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
27 MB gzipped, 136 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.6.1 kernel for all architectures; `config`(8); and `dbsym`(8).
22 MB gzipped, 114 MB uncompressed

xsrc This set contains the sources to the X Window System.
78 MB gzipped, 394 MB uncompressed

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

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

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

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/sparc64 subdirectory structure

The sparc64-specific portion of the NetBSD 1.6.1 release is found in the sparc64 subdirectory of the distribution: .../NetBSD-1.6.1/sparc64/

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/ sparc64 binary distribution sets; see below.

installation/
- miniroot/
  sparc64 miniroot file system image; see below.

`misc/` Miscellaneous sparc64 installation utilities; see installation section, below.

`netboot/`
Server boot-file image for diskless machines.

Binary distribution sets

The NetBSD sparc64 binary distribution sets contain the binaries which comprise the NetBSD 1.6.1 release for the sparc64. There are eight binary distribution sets. The binary distribution sets can be found in the `sparc64/binary/sets` subdirectory of the NetBSD 1.6.1 distribution tree, and are as follows:

base The NetBSD 1.6.1 sparc64 **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional. It includes shared library support, and excludes everything described below.
21 MB gzipped, 60 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.
17 MB gzipped, 73 MB uncompressed

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

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

kern-GENERIC
This set contains a NetBSD/sparc64 1.6.1 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
3 MB gzipped, 5 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.
7 MB gzipped, 27 MB uncompressed

misc This set includes the (rather large) system dictionaries, the typesettable document set, and other files from `/usr/share`.
3 MB gzipped, 8 MB uncompressed

text This set includes NetBSD's text processing tools, including `groff`(1), all related programs, and their manual pages.
2 MB gzipped, 6 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. They are currently equivalent to XFree86 3.3.6. Binary sets for the X Window System are distributed with NetBSD. The sets are:

xbase The basic files needed for a complete X client environment. This does not include the X servers.
3 MB gzipped, 9 MB uncompressed

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

xcontrib Programs that were contributed to X.
1 MB gzipped, 1 MB uncompressed

xfont Fonts needed by X.
6 MB gzipped, 7 MB uncompressed

xmisc Miscellaneous X programs.
1 MB gzipped, 1 MB uncompressed

xserver The Xsun, Xsun24, XsunMono and Xprt servers with man pages. The 64-bit X server current has problems with pixmaps.
4 MB gzipped, 9 MB uncompressed

The sparc64 binary distribution sets are distributed as gzipped tar files named with the extension **.tgz**, e.g. `base.tgz`.

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

Note: Each directory in the sparc64 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/sparc64 System Requirements and Supported Devices

Supported machines

- Sbus-based UltraSPARC systems (the Ultra 1, Ultra 1 Creator/Ultra1 Creator3D, Ultra 2/Ultra 2 Creator/Ultra 2 Creator 3D)
- PCI-based UltraSPARC systems (known to work on the Ultra 5, Ultra 10, Ultra 30 and Blade 100, Netra X1, and may work on many other systems)

Unsupported machines

- Chassis-based Enterprise Systems (Ex000, Ex500, E10000)

The minimal configuration requires 32 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 32 MB of RAM feels like Solaris with 32 MB of RAM - slow. Note that until you have at least 64 MB of RAM, getting more RAM is more important than getting a faster CPU.

Supported devices

- Ethernet devices
  - Sbus LANCE ethernet (*le*)
  - Sbus HME ethernet (*hme*)
  - Sbus BigMac ethernet (*be*)
  - PCI HME ethernet (*hme*)
  - PCI Tulip (*tlp*)
  - PCI ERI (*gem*)

- Video devices
  - Sbus cgsix
- SCSI host controllers
  - ncr53c9x based controllers (*esp*)
  - ncr53c8xx based controllers (*siop* or *esiop*)
  - ISP10x0 based controllers (*isp*)
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.
- PCI IDE host controllers
  - CMD Tech PCI064[3689] IDE Controllers (*pciide*)

    Note: Access to hard disks on the secondary PCIIDE channel currently do not work, but CD-ROM does.
- Serial devices
  - Sbus z8530 serial (*zs*)
  - PCI 16550-based su and su-pnp serial (*com*)
- Parallel devices
  - Sbus parallel ports (*bpp*)
  - PCI/Ebus parallel ports (*lpt*)
- Audio devices
  - PCI based system audio (*audiocs*)

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

Unsupported devices

- Floppy drives
- Serial devices
  - PCI based systems (*sab*)
- Video devices
  - Creator3D framebuffers (*ffb*)
  - General PCI VGA support

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 June, 2002).

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-1.6.1
# cd sparc64/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

If you're installing NetBSD/sparc64 for the first time it's a good idea to look at the partition sizes of disk you intend installing NetBSD on.

Assuming a classic partition scheme with `/` (root) and `/usr` file systems, a comfortable size for the NetBSD `/` partition is about 64 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 including X11R6 takes nearly 200 MB in `/usr`.

You cannot use the security modes of the SPARC OpenBoot PROM.

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

Some network devices (i.e. certain SBus cards) allow a choice between operating on a UTP or a AUI port. The `le` driver supports automatic detection of the port which is actually connected to the wire. If automatic detection is not available or not working properly in your environment, you may have to specify the type connection using the *media* parameter of `ifconfig`(8). During installation, you'll get the opportunity to specify the appropriate medium. Use `10base5/AUI` to select the AUI connector, or `10baseT/UTP` to select the UTP connector.

Installing the NetBSD System

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

There are several ways to install NetBSD onto a disk. The easiest way in terms of preliminary setup is to install from CDROM. If you don't have access to a CDROM or CDROM burner, you can use a miniroot image that can be booted off your local disk's swap partition. Alternatively, if your UltraSPARC is hooked up in a network you can find a server and arrange for a diskless setup which is a convenient way to install on a machine whose disk does not currently hold a usable operating system (see the section **Installing NetBSD by using a diskless setup** below). If you have problems with these, it is possible to install NetBSD from Solaris (see the section 'Installing NetBSD using Solaris' below).

Installing NetBSD from CDROM

Installing from CDROM, whether it has the full distribution or just a kernel and **sysinst** is the least painful way to install NetBSD. Simply insert the CDROM in the drive, power up the computer, and when you get to the

```
ok
```

prompt type:

```
ok boot cdrom
```

If the machine starts booting automatically, hit the **L1** or **Stop** key and the **A** key on the keyboard or send a break if your console is serial port to terminate the boot process.

NetBSD should then boot and you will have the opportunity to use **sysinst**.

Installing NetBSD by using the NetBSD miniroot

The miniroot is a self-contained NetBSD file system holding all utilities necessary to install NetBSD on a local disk. It is distributed as a plain file designed to be transferred to a raw disk partition from which it can be booted using the appropriate OpenBoot PROM command. Usually, the miniroot will be loaded into the swap partition of a disk. If needed, you can use any other unused partition, but remember that the partition will then not available during the installation process.

Loading the miniroot onto your raw partition is simple. On NetBSD as well as Solaris you use a command like:

```
# dd if=miniroot.fs of=/dev/rdsk/c0t0d0s1 bs=4k conv=sync
```

(Here, `/dev/rdsk/c0t0d0s1` is assumed to be your swap partition.) There's a potential problem here if `/dev/rdsk/c0t0d0s1` is actually in use as a swap partition by your currently running system. If you don't have another disk or partition to spare, you can usually get away with running this command anyway after first booting into single-user mode to ensure a quiet system.

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

```
# halt
```

Then boot the miniroot by typing the appropriate command at the OpenBoot PROM:

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

If you've loaded the miniroot onto some other disk than `sd0` adapt the boot specifier accordingly, e.g.:

```
ok boot disk1:b netbsd -s
```

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

Installing NetBSD by using a diskless setup

First, you must setup a diskless client configuration on a server. If you are using a NetBSD system as the boot-server, have a look at the `diskless`(8) manual page for guidelines on how to proceed with this. If the server runs another operating system, you'll have to consult documentation that came with it. (On SunOS systems, `add_client`(8) is a good start.)

Your UltraSPARC expects to be able to download a second stage bootstrap program via TFTP after having acquired its IP address through RevARP when instructed to boot 'over the net'. It will look for a filename composed of the machine's IP address in hexadecimal. For example, a machine which has been assigned IP address 130.115.144.11, will make an TFTP request for `8273900B`. Normally, this file is a symbolic link to an appropriate 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). You can find the boot program in `/usr/mdec/ofwboot.net` in the NetBSD/sparc64 distribution.

Note: The `/usr/mdec/ofwboot` does not know about netbooting.

After the boot program has been loaded into memory and given control by the OpenBoot 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 (if it comes in) contains 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 (if it comes in) is used to issue a REMOTE MOUNT request to the client's root file system server, asking for an NFS file handle corresponding to the root file system. If successful, the boot program starts reading from the remote root file system in search of the kernel which is then read into memory.

As noted above in the section **Preparing your System for NetBSD Installation** you have several options when choosing a location to store the installation filesets. However, the easiest way is to put the `*.tgz` files you want to install into the root directory for your client on the server.

From here there are two methods of netboot install. The first is much simpler and is intended for clients with local disk that will be used for installation. The second is intended for clients that will use a network boot normally.

If you are going to be installing onto local disk, place the `netbsd.INSTALL.gz` file, located in the `binary/kernel` directory, into the root directory for your machine. The `netbsd.INSTALL.gz` file contains a kernel with an embedded ramdisk that has the basic installer file system that is part of the normal miniroot. Now you need to boot your workstation from the server by entering the following command at the monitor prompt:

```
ok boot net netbsd.INSTALL.gz
```

And then follow the install instructions.

If you are going to be installing to NFS, be sure to read the section about preparing your system for installation. If you have access to another NetBSD machine, the `diskless(8)` manual page contains information about diskless booting. Unpack the `base.tgz` and `etc.tgz` sets on the server in the root directory for your machine. If you elect to use a separately NFS-mounted file system for `/usr` with your diskless setup, make sure the `./usr` base files in `base.tgz` end up in the correct location. One way to do this is to temporarily use a loopback mount on the server, re-routing *root*`/usr` to your server's exported NetBSD `/usr` directory.

A few configuration files need to be edited:

root`/etc/hosts`
: Add the IP addresses of both server and client.

root`/etc/myname`
: This files contains the client's hostname; use the same name as in <root>/etc/hosts.

root`/etc/fstab`
: Enter the entries for the remotely mounted file systems. For example:

```
server:/export/root/client / nfs rw 0 0
server:/export/exec/sun4u.netbsd /usr nfs rw 0 0
```

Now you must populate the `/dev` directory for your client. If your server runs SunOS 4.x, you can simply change your working directory to *root*`/dev` and run the MAKEDEV script: **`sh MAKEDEV all`**.

On Solaris systems, **`MAKEDEV`** can also be used, but there'll be error messages about unknown user and groups. These errors are inconsequential for the purpose of installing NetBSD. However, you may want to correct them if you plan to use the diskless setup regularly. In that case, you may re-run **`MAKEDEV`** on your NetBSD machine once it has booted.

Boot your workstation from the server by entering the following command at the monitor prompt:

```
ok boot net netbsd -s
```

This will boot the NetBSD kernel in single-user mode.

If you use a diskless setup with a separately NFS-mounted `/usr` file system, mount `/usr` by hand now:

```
netbsd# mount /usr
```

Note: For miniroot installs, the text editor is vi.

When using `disklabel(8)` to edit disklabels the -e switch will invoke the editor on the label. The -i switch will run an interactive session.

At this point, it's worth checking the disk label and partition sizes on the disk you want to install NetBSD onto. NetBSD understands SunOS-style disklabels, so if your disk was previously used by SunOS there will be a usable label on it. Use **`disklabel -e <disk>`** or **`disklabel -i <disk>`** (where *<disk>* is the device name assigned by the NetBSD kernel, e.g. `sd0`) to view and modify the partition sizes. See the section **Preparing your System for NetBSD Installation** above for suggestions about disk partition sizes.

If you are installing on a SCSI disk that does *not* have a SunOS or NetBSD label on it, you may still be able to use `disklabel(8)` but you'll have to create all partitions from scratch. If your disk is listed in `/etc/disktab`, you may use the entry (which in most cases only defines a '`c`' partition to describe the whole disk) to put an initial label on the disk. Then proceed with **`disklabel -e <disk>`** **`disklabel -i <disk>`** to create a partition layout that suits your needs.

Note: Because of the built-in compatibility with SunOS-style labels, *Always make sure all your partitions start and end on cylinder boundaries.*

Here follows an example of what you'll see while in the disklabel editor. Do not touch any of the parameters except for the '`label:`' entry and the actual partition size information at the bottom (the lines starting with '`a:`', '`b:`', ...).

The size and offset fields are given in sector units. Be sure to make these numbers multiples of the of the number of sectors per cylinder: the kernel might be picky about these things, but aside from this you'll have the least chance of wasting disk space. Partitions on which you intend to have a mountable file system, should be given fstype `4.2BSD`. Remember, the '`c`' partition should describe The whole disk and typically does not require editing. The '`(Cyl. x - y)`' info that appears after the hash '`#`' character is treated as a comment and need not be filled in when altering partitions.

Note: The line containing '`8 partitions:`' is best left alone, even if you define less than eight partitions. If this line displays a different number and the program complains about it (after you leave the editor), then try setting it to '`8 partitions:`'.

Sample disklabel screen

```
netbsd# disklabel sd2
# /dev/rsd2c:
type: SCSI
disk: SCSI disk
label: Hold Your Breath
flags:
bytes/sector: 512
sectors/track: 64
tracks/cylinder: 7
sectors/cylinder: 448
cylinders: 1429
rpm: 3600
interleave: 1
trackskew: 0
cylinderskew: 0
headswitch: 0           # milliseconds
track-to-track seek: 0  # milliseconds
drivedata: 0

8 partitions:
#        size   offset    fstype [fsz bsz cpg]
  a:    50176        0    4.2BSD   0   0   0   # (Cyl.    0 - 111)
  b:    64512    50176      swap               # (Cyl.  112 - 255)
  c:   640192        0   unknown               # (Cyl.    0 - 1428)
  d:   525504   114688    4.2BSD   0   0   0   # (Cyl.  256 - 1428)
```

To begin installation or the upgrade procedure, run **`sysinst`**:

```
netbsd# sysinst
```

Running sysinst

Once you have a miniroot booted, the first thing NetBSD will ask you for is a terminal. If you are running on the keyboard and mouse you want to select **sun** while if you are using a serial console from an X terminal you need to select **xterm** Selecting the wrong terminal settings may cause display corruption or **sysinst** may fail to run.

The **sysinst** program will do most of the work of transferring the system from the distribution sets onto your disk. You will frequently be asked for confirmation before **sysinst** 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 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.

The installation script goes through the following phases:

- determination of the disk to install NetBSD on
- checking of the partition information on the disk
- setting of the local timezone
- creating and mounting the NetBSD file systems
- setup of IP configuration
- extraction of the distribution tar files
- installation of boot programs

Installing NetBSD using Solaris

These instructions were kindly contributed by Murray Stokely <murray@osd.bsdi.com>

Preparing the disk

The first step is to format and label the disk that you would like to use with NetBSD. This can be accomplished with the format command in Solaris. The format command should allow you to create disk slices and write a disklabel. You will probably at least want to create a root partition and a swap partition, plus potentially a /usr or /var partitions, but of course you can layout the disk however you see fit.

```
partition> pr
Current partition table (original):
Total disk cylinders available: 8186 + 2 (reserved cylinders)

Part      Tag    Flag   Cylinders      Size              Blocks
0        root    wm      0 - 1923      1.00GB  (1924/0/0) 2097160
1        swap    wu   1924 - 2863    500.29MB  (940/0/0)  1024600
2      backup    wu      0 - 8185      4.25GB  (8186/0/0) 8922740
3 unassigned    wm      0             0        (0/0/0)          0
4         usr    wm   2864 - 8184      2.77GB  (5321/0/0) 5799890
5 unassigned    wm      0             0        (0/0/0)          0
6      backup    wm      0 - 8185      4.25GB  (8186/0/0) 8922740
7 unassigned    wm      0             0        (0/0/0)          0
```

After your disk has been labeled you need to create file systems on your slices. The Solaris newfs command will create ffs file systems that can be used by NetBSD. You should create file systems for all of your slices except for root with a command similar to below, given we have root on c1t2d0s0 and /usr on c1t2d0s4.

```
# newfs /dev/dsk/c1t2d0s0
# newfs /dev/dsk/c1t2d0s4
```

Installing NetBSD Software

You should now mount your NetBSD root and `/usr partitions under` Solaris so that you can populate the file systems with NetBSD NetBSD binaries.

```
# mount /dev/dsk/c1t2d0s0 /mnt
# mkdir /mnt/usr
# mount /dev/dsk/c1t2d0s4 /mnt/usr
```

First, you should uncompress all of the binary distribution sets for sparc64 into a temporary directory and then extract them into the file systems you just mounted.

```
# cd ~/netbsd/binary/sets
# gunzip *.tar.gz
# mkdir ~/netbsd/temp
# cd ~/netbsd/temp
# echo ~/netbsd/binary/sets/*.tar | (cd /mnt; xargs -n1 pax -rpe)
```

Now you should copy the NetBSD kernel and second stage bootloader into your new NetBSD root partition and install the bootblocks using Solaris's installboot command.

```
# cp ~/netbsd/binary/kernel/netbsd.GENERIC /mnt
# cp /mnt/netbsd.GENERIC /mnt/netbsd
# cp ~/netbsd/installation/misc/ofwboot /mnt
# installboot ~/netbsd/installation/misc/bootblk /dev/c1t2d0s0
```

Creating NetBSD Device Nodes under Solaris

Now you will need to create a minimum set of device nodes so that NetBSD can boot correctly. You should create all of the devices listed in the 'std' section of NetBSD 's `/dev/MAKEDEV`. You can use the Solaris version of mknod to create device nodes but you must be careful to use numeric group id's since the groups are numbered differently between the two systems.

```
# mknod console    c 0 0
# mknod tty        c 2 0   ; chmod 666 tty
# mknod kmem       c 3 1   ; chmod 640 kmem ; chgrp 2 kmem
# mknod mem        c 3 0   ; chmod 640 mem ; chgrp 2 mem
# mknod null       c 3 2   ; chmod 666 null
# mknod zero       c 3 12  ; chmod 666 zero
# mknod eeprom     c 3 11  ; chmod 640 eeprom ; chgrp 2 eeprom
# mknod openprom   c 70 0  ; chmod 640 openprom;chgrp 2 openprom
# mknod drum       c 7 0   ; chmod 640 drum ; chgrp 2 drum
# mknod klog       c 16 0  ; chmod 600 klog
# mknod stdin      c 24 0  ; chmod 666 stdin
# mknod stdout     c 24 1  ; chmod 666 stdout
# mknod stderr     c 24 2  ; chmod 666 stderr
# mknod fb         c 22 0  ; chmod 666 fb
# mknod mouse      c 13 0  ; chmod 666 mouse
# mknod kbd        c 29 0  ; chmod 666 kbd
```

You also must create device nodes for the disk devices that you are installing NetBSD onto, again you can use NetBSD 's `/dev/MAKEDEV` as a reference.

```
# mknod sd0a      b 7 0
# mknod sd0b      b 7 1
# mknod sd0c      b 7 2
# mknod sd0d      b 7 3
# mknod sd0e      b 7 4
# mknod sd0f      b 7 5
# mknod sd0g      b 7 6
# mknod sd0h      b 7 7
# mknod sd1a      b 7 8
# mknod sd1b      b 7 9
# mknod sd1c      b 7 10
# mknod sd1d      b 7 11
# mknod sd1e      b 7 12
# mknod sd1f      b 7 13
# mknod sd1g      b 7 14
# mknod sd1h      b 7 15
# mknod rsd0a     c 17 0
# mknod rsd0b     c 17 1
# mknod rsd0c     c 17 2
# mknod rsd0d     c 17 3
# mknod rsd0e     c 17 4
# mknod rsd0f     c 17 5
# mknod rsd0g     c 17 6
# mknod rsd0h     c 17 7
# mknod rsd1a     c 17 8
# mknod rsd1b     c 17 9
# mknod rsd1c     c 17 10
# mknod rsd1d     c 17 11
# mknod rsd1e     c 17 12
# mknod rsd1f     c 17 13
# mknod rsd1g     c 17 14
# mknod rsd1h     c 17 15
# chgrp 5 *sd[0-1][a-h]
# chmod 640 *sd[0-1][a-h]
```

For IDE disks, use 12 instead of 7 and 26 instead of 17, for `wd0a` and `rwd0a`, etc.

Configuring the NetBSD system (still under Solaris)

You will now need to configure some of the files in `/mnt/etc` to allow the system to work properly. In particular, you will need to modify `/etc/rc.conf` or else you will only get to single user mode. Read through the examples in `/etc/defaults/rc.conf` and override any settings in `/etc/rc.conf`. You will also need to create a `/etc/fstab` file to tell the system which slice to mount as `/usr`, etc. You should follow the examples in `/usr/share/examples/fstab/`. You may also want to setup your `/etc/resolv.conf` file for name services and your `/etc/hosts file` if you are confident that you will get networking setup on your first try.

Booting NetBSD for the first time

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

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

Congratulations, you have successfully installed NetBSD 1.6.1.

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state, 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-only. When the system asks you to choose a shell, simply press RETURN to get to a /bin/sh prompt. If you are asked for a terminal type, respond with **vt220** (or whatever is appropriate for your terminal type) and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

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

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

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

   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=vt220
   ```

   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_de0="inet 123.45.67.89 netmask 255.255.255.0"
   ```

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

   ```
   ifconfig_de0="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 list of available packages suitable for browsing is at
     ftp://ftp.netbsd.org/pub/NetBSD/packages/pkgsrc/README.html
   - Precompiled binaries can be found at
     ftp://ftp.netbsd.org/pub/NetBSD/packages,
     usually in the `1.6.1/sparc64/All` subdir. You can install them with the following commands:

     ```
     # PKG_PATH=ftp://ftp.netbsd.org/pub/NetBSD/packages/1.6.1/sparc64/All
     # export PKG_PATH
     # pkg_add -v tcsh
     # pkg_add -v apache
     # pkg_add -v perl
     ...
     ```

     The above commands will install the tcsh shell, 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

To upgrade to NetBSD 1.6.1 from a previous version follow the instructions in the section **Installing NetBSD**, but run **sysinst** and choose the upgrade option rather than the install option.

The upgrade option will use the existing disk partitions to install the new system in, and also preserves the files in /etc by moving them to /etc.old.

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 1.6.1.

Issues affecting an upgrade from NetBSD 1.5

The following issues can generally be resolved by extracting the **etc** set into a temporary directory and running *postinstall*:

```
mkdir /tmp/upgrade
cd /tmp/upgrade
pax -zrpe -f /path/to/etc.tgz
./etc/postinstall -s 'pwd' check
./etc/postinstall -s 'pwd' fix
```

Issues fixed by *postinstall*:

- Various files in /etc need upgrading. These include:
  - /etc/defaults/*
  - /etc/mtree/*
  - /etc/daily
  - /etc/weekly
  - /etc/monthly
  - /etc/security
  - /etc/rc.subr
  - /etc/rc
  - /etc/rc.shutdown
  - /etc/rc.d/*
- The following files are now obsolete: /etc/rc.d/NETWORK and /etc/rc.d/gated.
- The following rc.conf(5) entries are now obsolete: amd_master, ip6forwarding, defcorename, and nfsiod_flags. critical_filesystems_beforenet has been replaced by critical_filesystems_local. critical_filesystems has been re-

placed by critical_filesystems_remote.

- The users and groups 'named', 'ntpd', and 'sshd' need to be created.
- The configuration files for ssh(1) and sshd(8) were moved from /etc to /etc/ssh, including ssh_known_hosts* files and the host key files ssh_host*_key*. /etc/ssh.conf was renamed to /etc/ssh/ssh_config, and /etc/sshd.conf was renamed to /etc/ssh/sshd_config.
- The **mux** entries in wscons.conf(5) are now obsolete.

The following issues need to be resolved manually:

- postfix(8) configuration files require upgrading.

```
cd /usr/share/examples/postfix
cp post-install postfix-files postfix-script /etc/postfix
postfix check
```

- The *de* ethernet driver was replaced with the *tlp* driver. This may require the renaming of the files /etc/ifconfig.de* to /etc/ifconfig.tlp*, renaming of rc.conf(5) entries ifconfig_de* to ifconfig_tlp*, and the reconfiguration of files such as /etc/dhclient.conf and /etc/ipf.conf.

Issues affecting an upgrade from NetBSD 1.4 or prior

- /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.

- 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) was 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
Luke Mewburn
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
Rob Windsor
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:

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):

Simon Burge	`simonb@netbsd.org`	**pmax**
Simon Burge	`simonb@netbsd.org`	**sbmips**
Jeremy Cooper	`jeremy@netbsd.org`	**sun3x**
Matt Fredette	`fredette@netbsd.org`	**sun2**
Chris Gilbert	`chris@netbsd.org`	**cats**
Ross Harvey	`ross@netbsd.org`	**alpha**
Jun-ichiro itojun Hagino	`itojun@netbsd.org`	**sh3**
Ben Harris	`bjh21@netbsd.org`	**acorn26**
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**
Frank van der Linden	`fvdl@netbsd.org`	**i386**
Anders Magnusson	`ragge@netbsd.org`	**vax**

Phil Nelson	phil@netbsd.org	**pc532**
NISHIMURA Takeshi	nsmrtks@netbsd.org	**x68k**
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**
Leo Weppelman	leo@netbsd.org	**atari**
Nathan Williams	nathanw@netbsd.org	**sun3**
Steve Woodford	scw@netbsd.org	**mvme68k**
Steve Woodford	scw@netbsd.org	**mvmeppc**
Reinoud Zandijk	reinoud@netbsd.org	**acorn32**

The NetBSD 1.6.1 Release Engineering team:

Erik Berls	cyber@netbsd.org
Håvard Eidnes	he@netbsd.org
Perry Metzger	perry@netbsd.org
Luke Mewburn	lukem@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Todd Vierling	tv@netbsd.org

NetBSD Developers:

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
Bang Jun-Young	junyoung@netbsd.org
Dieter Baron	dillo@netbsd.org
Robert V. Baron	rvb@netbsd.org
Jason Beegan	jtb@netbsd.org
Erik Berls	cyber@netbsd.org
Hiroyuki Bessho	bsh@netbsd.org
John Birrell	jb@netbsd.org
Mason Loring Bliss	mason@netbsd.org
Rafal Boni	rafal@netbsd.org
Manuel Bouyer	bouyer@netbsd.org
John Brezak	brezak@netbsd.org

Allen Briggs	briggs@netbsd.org
Mark Brinicombe	mark@netbsd.org
Aaron Brown	abrown@netbsd.org
Andrew Brown	atatat@netbsd.org
David Brownlee	abs@netbsd.org
Frederick Bruckman	fredb@netbsd.org
Jon Buller	jonb@netbsd.org
Simon Burge	simonb@netbsd.org
Robert Byrnes	byrnes@netbsd.org
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
Jeremy Cooper	jeremy@netbsd.org
Chuck Cranor	chuck@netbsd.org
Alistair Crooks	agc@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
Tracy Di Marco White	gendalia@netbsd.org
Jaromír Dolecek	jdolecek@netbsd.org
Andy Doran	ad@netbsd.org
Roland Dowdeswell	elric@netbsd.org
Emmanuel Dreyfus	manu@netbsd.org
Matthias Drochner	drochner@netbsd.org
Jun Ebihara	jun@netbsd.org
Håvard Eidnes	he@netbsd.org
Stoned Elipot	seb@netbsd.org
Enami Tsugutomo	enami@netbsd.org
Bernd Ernesti	veego@netbsd.org
Erik Fair	fair@netbsd.org
Gavan Fantom	gavan@netbsd.org
Hubert Feyrer	hubertf@netbsd.org
Jason R. Fink	jrf@netbsd.org
Matt Fredette	fredette@netbsd.org
Thorsten Frueauf	frueauf@netbsd.org
Castor Fu	castor@netbsd.org
Ichiro Fukuhara	ichiro@netbsd.org
Thomas Gerner	thomas@netbsd.org
Simon J. Gerraty	sjg@netbsd.org
Justin Gibbs	gibbs@netbsd.org
Chris Gilbert	chris@netbsd.org
Adam Glass	glass@netbsd.org
Michael Graff	explorer@netbsd.org
Brian C. Grayson	bgrayson@netbsd.org
Matthew Green	mrg@netbsd.org

Andreas Gustafsson	gson@netbsd.org
Jun-ichiro itojun Hagino	itojun@netbsd.org
Juergen Hannken-Illjes	hannken@netbsd.org
Charles M. Hannum	mycroft@netbsd.org
Ben Harris	bjh21@netbsd.org
Ross Harvey	ross@netbsd.org
Eric Haszlakiewicz	erh@netbsd.org
John Hawkinson	jhawk@netbsd.org
HAYAKAWA Koichi	haya@netbsd.org
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
Eduardo Horvath	eeh@netbsd.org
Nick Hudson	skrll@netbsd.org
Shell Hung	shell@netbsd.org
Martin Husemann	martin@netbsd.org
Dean Huxley	dean@netbsd.org
Bernardo Innocenti	bernie@netbsd.org
Tetsuya Isaki	isaki@netbsd.org
ITOH Yasufumi	itohy@netbsd.org
IWAMOTO Toshihiro	toshii@netbsd.org
Matthew Jacob	mjacob@netbsd.org
Lonhyn T. Jasinskyj	lonhyn@netbsd.org
Darrin Jewell	dbj@netbsd.org
Chris Jones	cjones@netbsd.org
Søren Jørvang	soren@netbsd.org
Takahiro Kambe	taca@netbsd.org
Antti Kantee	pooka@netbsd.org
Masanori Kanaoka	kanaoka@netbsd.org
Mattias Karlsson	keihan@netbsd.org
KAWAMOTO Yosihisa	kawamoto@netbsd.org
Mario Kemper	magick@netbsd.org
Thomas Klausner	wiz@netbsd.org
Klaus Klein	kleink@netbsd.org
Wayne Knowles	wdk@netbsd.org
John Kohl	jtk@netbsd.org
Paul Kranenburg	pk@netbsd.org
Martti Kuparinen	martti@netbsd.org
Kevin Lahey	kml@netbsd.org
Johnny C. Lam	jlam@netbsd.org
Martin J. Laubach	mjl@netbsd.org
Ted Lemon	mellon@netbsd.org
Frank van der Linden	fvdl@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
Anders Magnusson	ragge@netbsd.org
MAEKAWA Masahide	gehenna@netbsd.org
David Maxwell	david@netbsd.org
Dan McMahill	dmcmahill@netbsd.org
Gregory McGarry	gmcgarry@netbsd.org
Jared D. McNeill	jmcneill@netbsd.org
Neil J. McRae	neil@netbsd.org
Perry Metzger	perry@netbsd.org
Minoura Makoto	minoura@netbsd.org
Luke Mewburn	lukem@netbsd.org
der Mouse	mouse@netbsd.org
Joseph Myers	jsm@netbsd.org
Ken Nakata	kenn@netbsd.org
Phil Nelson	phil@netbsd.org
Bob Nestor	rnestor@netbsd.org
NISHIMURA Takeshi	nsmrtks@netbsd.org
Tohru Nishimura	nisimura@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
Chris Pinnock	cjep@netbsd.org
Dante Profeta	dante@netbsd.org
Chris Provenzano	proven@netbsd.org
Michael Rauch	mrauch@netbsd.org
Darren Reed	darrenr@netbsd.org
Scott Reynolds	scottr@netbsd.org
Michael Richardson	mcr@netbsd.org
Tim Rightnour	garbled@netbsd.org
Gordon Ross	gwr@netbsd.org
Heiko W. Rupp	hwr@netbsd.org
David Sainty	dsainty@netbsd.org
SAITOH Masanobu	msaitoh@netbsd.org
Kazuki Sakamoto	sakamoto@netbsd.org
Curt Sampson	cjs@netbsd.org
Wilfredo Sanchez	wsanchez@netbsd.org
Ty Sarna	tsarna@netbsd.org
SATO Kazumi	sato@netbsd.org
Jan Schaumann	jschauma@netbsd.org
Matthias Scheler	tron@netbsd.org
Karl Schilke (rAT)	rat@netbsd.org
Amitai Schlair	schmonz@netbsd.org

Konrad Schroder	perseant@netbsd.org
Reed Shadgett	dent@netbsd.org
Tim Shepard	shep@netbsd.org
Takeshi Shibagaki	shiba@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
Noriyuki Soda	soda@netbsd.org
Wolfgang Solfrank	ws@netbsd.org
SOMEYA Yoshihiko	someya@netbsd.org
Bill Sommerfeld	sommerfeld@netbsd.org
Ignatios Souvatzis	is@netbsd.org
Bill Squier	groo@netbsd.org
Jonathan Stone	jonathan@netbsd.org
Bill Studenmund	wrstuden@netbsd.org
Kevin Sullivan	sullivan@netbsd.org
SUNAGAWA Keiki	kei@netbsd.org
Kimmo Suominen	kim@netbsd.org
Shin Takemura	takemura@netbsd.org
TAMURA Kent	kent@netbsd.org
Shin'ichiro TAYA	taya@netbsd.org
Matt Thomas	matt@netbsd.org
Jason Thorpe	thorpej@netbsd.org
Christoph Toshok	toshok@netbsd.org
Tsubai Masanari	tsubai@netbsd.org
Izumi Tsutsui	tsutsui@netbsd.org
UCHIYAMA Yasushi	uch@netbsd.org
Masao Uebayashi	uebayasi@netbsd.org
Shuichiro URATA	ur@netbsd.org
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
Leo Weppelman	leo@netbsd.org
Assar Westerlund	assar@netbsd.org
Todd Whitesel	toddpw@netbsd.org
Nathan Williams	nathanw@netbsd.org
Rob Windsor	windsor@netbsd.org
Dan Winship	danw@netbsd.org
Jim Wise	jwise@netbsd.org
Michael Wolfson	mbw@netbsd.org
Steve Woodford	scw@netbsd.org
Colin Wood	ender@netbsd.org
YAMAMOTO Takashi	yamt@netbsd.org
Yuji Yamano	yyamano@netbsd.org
Reinoud Zandijk	reinoud@netbsd.org

Maria Zevenhoven	`maria7@netbsd.org`
Christos Zoulas	`christos@netbsd.org`

Other contributors:

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

Legal Mumbo-Jumbo

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

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

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 for the NetBSD Project.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Tinguely and Jim Lowe.
This product includes software developed by Markus Wild.
This product includes software developed by 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 designed by William Allen Simpson.
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 for the NetBSD Project by Wasabi Systems, Inc.
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 at the Information Technology Division, US Naval Research Laboratory.

This product includes software developed by the Alice Group.
This product includes software developed by David Miller.

The End