NAME

INSTALL – Installation procedure for NetBSD/hp300.

CONTENTS

About this Document..2
Dedication...2
What is NetBSD?..2
Upgrade path to NetBSD 1.6.2...2
Changes Between The NetBSD 1.6.1 and 1.6.2 Releases....................3
  Kernel 3
  Networking 4
  Security 4
  System administration and user tools 4
  Miscellaneous 4
  hp300 specific 5
Changes Between The NetBSD 1.6 and 1.6.1 Releases......................5
  Kernel 5
  Networking 5
  File system 6
  Security 6
  System administration and user tools 6
  Miscellaneous 6
Changes Between The NetBSD 1.5 and 1.6 Releases........................7
  Kernel 7
  Networking 7
  File system 8
  Security 8
  System administration and user tools 9
  Miscellaneous 9
The Future of NetBSD..10
Sources of NetBSD...11
NetBSD 1.6.2 Release Contents...11
  NetBSD/hp300 subdirectory structure 12
  Binary distribution sets 13
NetBSD/hp300 System Requirements and Supported Devices.................14
  Supported hardware 15
  Unsupported hardware 17
Getting the NetBSD System on to Useful Media..........................17
Preparing your System for NetBSD installation.........................19
  Formatting your hard drives 19
  Designing your disk's partition table 20
  Installing the bootstrap program locally 21
  Installing the miniroot file system locally 22
  Configuring the netboot server 22
  Put Series 400 systems in HP-UX Compatible Boot Mode 27
  Searching for a bootable system 28
  Selecting ethernet port on Series 400 29
  Running SYS_INST 30
  Chosing a kernel location 31

Installing the NetBSD System....................................31
Post installation steps...33
Upgrading a previously-installed NetBSD System..................35
  Upgrading using the miniroot 36
  Manual upgrade 37
Compatibility Issues With Previous NetBSD Releases..............38
  Issues affecting an upgrade from NetBSD 1.5 38
  Issues affecting an upgrade from NetBSD 1.4 or prior 39
Using online NetBSD documentation...............................40
Administrivia...40
Thanks go to..41
We are..43
Legal Mumbo-Jumbo...49
The End...51

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 1.6.2 on the *hp300* platform. It is available in four different formats titled `INSTALL.ext`, where `.ext` is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

Dedication

The NetBSD Foundation would like to dedicate the NetBSD 1.6.2 release to the memory of Erik Reid, who went missing and is presumed dead in a sailing accident on 18 February 2004. Erik's contributions to NetBSD included work on support for SGI MIPS R4000, integrating XFree86 Direct Rendering Interface (DRI), and managing the build lab. His death came as a shock, and he will be greatly missed by all of us. May he rest in peace.

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 1.6.2 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 1.6.2

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 1.6.2.

NetBSD 1.6.2 is an upgrade of NetBSD 1.6.1 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 1.6 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 1.6.1 and 1.6.2 Releases

The NetBSD 1.6.2 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, patches and updates to 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 nine months of development that went into the NetBSD 1.6.2 release. Some highlights include:

Kernel

- With `pciide`(4), make Promise controllers do DMA with large disks requiring 48-bit LBA drives.
- Add error detection when running low on swap, to improve stability in low-memory situations.
- Support for more SiS controllers were added to `pciide`(4).
- Support for the new PowerBook G4 12-inch added.
- `mlx`(4) stability improved.
- Support for ICH5 added to `pciide`(4).
- A long-standing stability problem with the original Sun4c sparc systems (SS1, SS1+, and IPC) has been found and fixed. Some optimizations done for these systems as well.
- Sun3, Sun3x, and Sun2 may now boot from tape files, through addition of seek support for tape files.
- The USERCONF option has been added to the i386 kernels; see `userconf`(4) for more information.
- Hardware random number generator support for Intel 865 and 875P chipsets added.
- Fix `wdc`(4) to work with pre-ATA drives.
- Shared libraries and other files mapped executable now count as TEXT pages for vm-usage purposes. This should allow for more appropriate handling of these pages compared to other normal file buffer pages.

- General support for multi-function pcmcia cards has been fixed.
- Various fixes to linux emulation have been added.

Networking

- `rtk`(4) multicast problem fixed.
- `fxp`(4) support yet a few more chip variants.
- `tulip`(4) driver fixed so that the DEC Alpha PWS no longer panics.
- Path MTU discovery black-hole detection has been added.
- `bce`(4) driver added for Broadcom BCM4401 chipset, as seen in recent Dell laptops.
- A workaround has been added for a race condition in the networking code which could corrupt the callout data structure.
- Various networking stack fixes for IPv4, IPv6 and IPSEC.

Security

- NetBSD-SA2003-018 DNS negative cache poisoning
- NetBSD-SA2003-017 OpenSSL multiple vulnerability
- NetBSD-SA2003-016 Sendmail - another prescan() bug CAN-2003-0694
- NetBSD-SA2003-015 Remote and local vulnerabilities in XFree86 font libraries
- NetBSD-SA2003-014 Insufficient argument checking in sysctl(2)
- NetBSD-SA2003-012 Out of bounds memset(0) in sshd
- NetBSD-SA2003-011 off-by-one error in realpath(3)
- NetBSD-SA2003-010 remote panic in OSI networking code

System administration and user tools

- Possible crash in `vi`(1) triggered by an error was fixed.
- XFree86 upgraded to version 4.3.0 for those architectures which use XFree86 version 4.
- `scsictl`(8) now supports a few new commands.
- BIND has been upgraded to version 8.3.7.
- DHCP has been upgraded to version 3.0.1rc11 with various fixes.
- CVS has been upgraded to version 1.11.10.

Miscellaneous

- At least one problem causing sysinst to crash has been fixed, and a progress bar has been added as an option to monitor the progress of the extraction of the install sets.
- Package tools upgraded to version 20030918.

And of course there have also been innumerable bug fixes and other miscellaneous enhancements. You can look for this trend to continue.

hp300 specific

Changes Between The NetBSD 1.6 and 1.6.1 Releases

The NetBSD 1.6.1 release provides numerous significant functional enhancements, including support for many new devices, integration of hundreds of bug fixes, patches and updates to 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 seven months of development that went into the NetBSD 1.6.1 release. Some highlights include:

Kernel

- Hardware random number support for some Intel chipsets has been added.
- Support for additional Adaptec RAID controllers has been added to `aac`(4).
- A number of bugs in the VM system have been fixed.
- Bug fixes to `audio`(4), `dpt`(4), `eap`(4), `emuxki`(4), `iop`(4), `siop`(4) and `umass`(4).
- Some Linux compatibility bugs have been fixed.
- A number of USB bugs have been fixed.
- acorn32 and acorn26 module recognition overhauled; APDL IDE should now work.
- `pciide`(4) support has been extended to support Promise Ultra133TX2, Promise Ultra133TX2v2, HighPoint HPT372, Ultra/133 on VIA VT8233A, and the VIA VT8235.
- Many changes to the arm architecture support, mostly triggered by evbarm port infrastructure changes.
- Amiga boot handling has been modified to better handle certain machine configurations.

Networking

- IPv6 fixes to various tools.
- Bug fixes to the `tlp`(4) and `xi`(4) drivers.
- Enhancements to the `pcn`(4) driver.
- `rtk`(4) now supports Planex FNW-3603 cardbus ethernet card.
- `ex`(4) multicast handling has been fixed.
- `wi`(4) now supports Netgear MA401RA card.
- `wm`(4) now supports more chip variants.
- `aue`(4) now supports SMC 2206USB/ETH EZ Connect adapter.
- `sip`(4) now has some Tx interrupt mitigation code, and improved support for 64-bit DP83820 cards.
- The `fxp`(4) driver has been improved for better support of certain i82558 revisions, and has been fixed to recognize some more chips.
- IPFilter has been upgraded to version 3.4.29. Please note that this requires a synchronized upgrade of kernel and the ipf user programs to work properly.
- Support for Broadcom Gigabit Ethernet devices has been added by the addition of the `bge`(4) driver.

- Some IPsec bugs have been fixed (from KAME).
- Some `ftpd`(8) interoperability bugs have been fixed.
- `mopd`(8) fixes to make the program load correctly.
- A number of `pppoe`(4) bugs have been fixed.

File system

- A number of FFS and NFS bugs have been fixed.
- Several fixes to `ffs`(4) soft dependencies handling have been incorporated.
- Some NFSv3 fixes have been applied to `amd`(8).
- Some `fsck`(8) bugs have been fixed.

Security

- BIND has been updated to 8.3.4 and security patches applied to `named`(8) and the libc resolver.
- Various security patches have been applied to sendmail.
- Padding has been added to minimum-sized IP packets in several ethernet drivers to prevent unintented information leakage.
- OpenSSL has been updated to 0.9.6g and a number of security patches applied.
- A potential buffer overflow in zlib has been fixed.
- Buffer overflow bugs in `file`(1) have been fixed.
- Some Kerberos 4 security bugs have been addressed.
- A umask security problem in GNU `tar`(1) has been fixed.

System administration and user tools

- `user`(8) has seen several fixes, some of them related to MD5 and blowfish password encryption support.
- Changes to gcc for the arm architecture introduces a minor flag day: new and old object files can not be mixed. Required for proper soft-VFP support.
- MDC2 build has been made optional, as the algorithm is patended.
- The package tools have been updated to the 20030202 version.

Miscellaneous

- Bug fixes to sysinst, the NetBSD installer.
- Various cross-build fixes have been incorporated.
- Various fixes to the toolchain and build process.
- Various fixes to the rc.d subsystem.
- A large number of sparc64 fixes have been applied.
- Timezone files have been updated to tz2002d.
- Many new packages have been added to The NetBSD *packages collection,* including the latest open source desktop KDE3, OpenOffice.org, as well as a large number of bugs fixed, many addressing security issues.

And of course there have also been innumerable bug fixes and other miscellaneous enhancements. You can look for this trend to continue.

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 1.6.2 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 1.6.2 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.

This is the seventh major release of NetBSD for the HP 9000/300 series of computers.

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 1.6.2 Release Contents

The root directory of the NetBSD 1.6.2 release is organized as follows:

`.../NetBSD-1.6.2/`

`CHANGES`	Changes since earlier NetBSD releases.
`LAST_MINUTE`	Last minute changes.
`MIRRORS`	A list of sites that mirror the NetBSD 1.6.2 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 1.6.2 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.
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 1.6.2 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.
27 MB gzipped, 136 MB uncompressed

syssrc This set contains the sources to the NetBSD 1.6.2 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/hp300 subdirectory structure

The hp300-specific portion of the NetBSD 1.6.2 release is found in the `hp300` subdirectory of the distribution: `.../NetBSD-1.6.2/hp300/`

`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/` hp300 binary distribution sets; see below.

`installation/`
- `miniroot/`
  hp300 miniroot file system image; see below.

`misc/` Miscellaneous hp300 installation utilities; see installation section, below.

Binary distribution sets

The NetBSD hp300 binary distribution sets contain the binaries which comprise the NetBSD 1.6.2 release for the hp300. There are eight binary distribution sets. The binary distribution sets can be found in the `hp300/binary/sets` subdirectory of the NetBSD 1.6.2 distribution tree, and are as follows:

base The NetBSD 1.6.2 hp300 **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional. It includes shared library support, and excludes everything described below.
17 MB gzipped, 41 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 MB gzipped, 52 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, 7 MB uncompressed

kern-GENERIC
This set contains a NetBSD/hp300 1.6.2 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.
2 MB gzipped, 3 MB uncompressed

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

xcomp The extra libraries and include files needed to compile X source code.
2 MB gzipped, 8 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

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

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

The following are included in the `hp300/installation` directory:

`miniroot/`

`miniroot.fs.gz` A copy of the miniroot file system.

`misc/`

`HP-IB.geometry` A file containing geometry for some HB-IB disk drives.

`SYS_INST.gz` A gzipped copy of the SYS_INST miniroot installation program.

`SYS_UBOOT.gz` A gzipped copy of the universal boot block. Supports Network, tape and disk booting. This is useful if you are installing a diskless NetBSD/hp300 system.

`rbootd.tgz` Source code for the rbootd program included with NetBSD. It requires that the server has a Berkeley Packet Filter (bpf). You will need to compile this version of rbootd if server system does not have this utility already.

The following are included in the `hp300/binary/kernel` directory:

`netbsd-GENERIC.gz` A gzipped `GENERIC` kernel.

`netbsd.RAMDISK.gz` A gzipped `INSTALL` kernel with embedded ramdisk.

`netbsd.RAMDISK.symbols.gz` Symbols for `netbsd.RAMDISK.gz`.
This utility is used only in a Traditional method installation.

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

NetBSD/hp300 System Requirements and Supported Devices

NetBSD/hp300 1.6.2 will run on most HP 9000/300- and 400-series machines. The smallest amount of RAM that has been tested is 4 MB. If you wish to run X, more RAM is recommended.

Supported hardware

- CPUs
  - 318 (16 MHz 68020, with built-in monochrome framebuffer, no expansion)
  - 319 (16 MHz 68020, with built-in 6 bit color framebuffer, no expansion)
  - 320 (16 MHz 68020, 16 KB L2 cache, up to 7.5 MB RAM)
    Requires Human Interface board
  - 330 (16 MHz 68020, 4 MB on motherboard, up to 8 MB RAM)
    Requires Human Interface board
  - 340 (16 MHz 68030, up to 16 MB RAM)
  - 345 (50 MHz 68030 with 32 KB L2 cache, up to 128 MB RAM, built-in SCSI)
  - 350 (25 MHz 68020, 32 KB L2 cache, up to 48 MB RAM)
    Requires Human Interface board
  - 360 (25 MHz 68030, with 4 MB RAM built-in, up to 48 MB RAM)
    Requires System Interface board
  - 362 (25 MHz 68030, up to 16 MB RAM, built-in SCSI)
    Framebuffer is unsupported, you must use a serial console
  - 370 (33 MHz 68030, 64 KB L2 cache, up to 48 MB RAM)
    Requires System Interface board
  - 375 (50 MHz 68030 with 32 KB L2 cache, up to 128 MB RAM, built-in SCSI)
  - 380 (25 MHz 68040, up to 128 MB RAM, built-in SCSI)
  - 382 (25 MHz 68040, up to 32 MB RAM, built-in SCSI)
    Framebuffer is unsupported, you must use a serial console
  - 385 (33 MHz 68040, up to 128 MB RAM, built-in SCSI)
  - 400s, 400t, 400dl (50 MHz 68030 with 32 KB L2 cache, built-in SCSI, up to 128 MB RAM)
  - 425s, 425t, 425dl (25 MHz 68040 with built-in SCSI, up to 128 MB RAM)
  - 425e (25 MHz 68040, built-in SCSI)
    The BootROM does not support serial console and NetBSD/hp300 *does not support the 425e framebuffer. When the* NetBSD *kernel begins to boot it will default to using the serial port for console*
  - 433s, 433t, 433dl (33 MHz 68040 with built-in SCSI, up to 128 MB RAM)

- HP-IB devices
  - *rd*; CS80 disks: 2200, 2203, 7912, 7914, 7933, 7936, 7937, 7945, 7946, 7957, 7958, and 7959
  - *rd*; CS80 floppy disks: 9122, 9134 (possibly others)
    Requires use of HP-UX LIF utilities via HP-UX emulation
  - CS80 Low-density 16 track cartridge (67 MB): 7912, 7914, 7946, and 9144
    ;
  - CS80 High-density 32 track cartridge (134 MB): 9145
    ;
  - *mt*; CS80 Half-inch tape: 7974A, 7978A/B, 7979A, 7980A, and 7980XC.

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

- SCSI devices
  - *sd*; SCSI hard drives
    Some SCSI II drives may have problems (not very common)
  - *sd*; SCSI CD-ROM drives
    Support for data only (no audio CDs)
  - *sd*; SCSI Magneto-optical drives

- *st*; SCSI tape drives: HP 35450A (DDS-1 DAT), Exabyte EXB-8200 (8mm), Archive (QIC-24), Archive Viper (QIC-60), Archive Viper (QIC-150), Archive Python 25501 (DAT), and Archive Python 28849 (DAT)
- *ac*; SCSI autochangers

• Serial interfaces
- *dca*; Built-in single serial port on System Interface board and Human Interface board
- *apci*; Built-in Apollo 4-port on Series 400 workstations
- *dcm*; 98638 8-port (DIO-II board, appears to kernel as two 98642 devices)
- *dcm*; 98642 4-port (DIO-I board)
- *dca*; 98626, 98644 built-in or add-on (DIO-I board) single serial port
- *dcl*; 98628A single port (DIO-I board)

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

• Network interfaces
- *le*; 98643 built-in and add-on (DIO-I board) ethernet cards.

• Graphics Devices
- 98544 monochrome Topcat (1024x768, 1 bit, DIO-II)
- 98545A color Topcat (1024x768, 4 bits, DIO-II)
- 98547 color Topcat (1024x768, 6 bits)
- 98548A monochrome Catseye (1024x768, 1 bit)
- 98549A color Catseye (1024x768, 6 bits)
- 98550A Catseye Hi-Res Color (a.k.a. CH) (1280x1024 @ 60 Hz, 8 bits, DIO-II)
- 98700 and 98710 Gatorbox (1280x1024 @ 60 Hz, 4 or 8 bits)
  Requires 98287 (DIO-I board) to connect to workstation
- 98720 and 98721 color Renaissance SRX (1280x1024 @ 60 Hz, 8, 12 or 24 bits)
  Requires 98724 (DIO-I board) or 98725 (DIO-II board) to connect to workstation
- 98730 and 98731 DaVinci TurboSRX (1280x1024 @ 60 Hz, 8, 16, or 24 bits)
  Requires 98726A (DIO-II board) to connect to workstation
- A1096A monochrome Hyperion (1280x1024, 1 bit)
- A1416A Kathmandu (a.k.a. Color VRX) (1280x1024 @ 60 hz, 8 bits, DIO-II)

• HP-HIL devices
- Keyboards
- Two and three button mice
- 46094 Quadrature Port (supports normal serial mice)
- Graphics tablets
- Dial boxes
- ID modules

• Miscellaneous boards/interfaces
- 98265A SCSI interface (DIO-I board, daughtercard for 98562, or built-in)
- 98561 Human Interface board (DIO-I board with serial port, HP-HIL, and HP-IB)
- 98562 System Interface board (DIO-II board with serial port, HP-HIL, HP-IB, DMA, and ethernet, has optional fast HP-IB or SCSI daughterboard)
- 98620 DMA card (DIO-I board, for use with 98561)
- 98624 HP-IB interface (DIO-I board or built-in)
- 98625A and 98625B 'fast' HP-IB interface (DIO-I board or daughtercard for 98562)

Each serial interface has its own quirks, and some of them use non-standard pins. The FAQ describes how to configure and connect serial consoles to hp300 systems.
http://www.netbsd.org/Ports/hp300/faq.html#serialconsole

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

Unsupported hardware

- CPUs
  - 310 (10 MHz 68010, with built-in monochrome framebuffer, rs232 (25 pin), hil, and slow hpib)
  - 332 (16? MHz 68030, with no floating point unit)
- Graphics Devices
  - SGC support (for 425e built-in EVRX framebuffer)
  - SGC support (for some rare configurations of 4XXt and 4XXdl workstations)
  - 98702 TurboVRX DIO-II graphics device
  - 98705 Tigershark PersonalVRX DIO-II graphics device
  - internal video on models 362/382
- Miscellaneous boards/interfaces
  - domain keyboard and mouse, on Series 400 workstations. Needs wscons
  - parallel port, on 345, 362, 375, 380, 382, 382, and Series 400 workstations
  - EISA support, on Series 400 workstations
  - Token Ring cards (a.k.a. ATR)
  - VME bus adapter, on Series 300 workstations
  - 98625C fast HP-IB option on models 345 and 375
  - internal SCSI floppy drive on 362, 382, and 425e

Getting the NetBSD System on to Useful Media

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

Installation is supported from several media types, including:

- CD-ROM
- 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.

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-1.6.2
# cd hp300/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.

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

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

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

- hp300 netboots `SYS_UBOOT` and then runs the miniroot installation tools or a purely diskless installation from a NetBSD server on the same subnet (you must have root access). Other server platforms will work, but are described in the NetBSD Diskless HOW-TO.
  http://www.netbsd.org/Documentation/network/netboot/
- hp300 loads `SYS_UBOOT` from a local disk, tape, or floppy and then runs the miniroot installation tools or a purely diskless installation from a NetBSD server on the same subnet (you must have root access). Other server platforms will work, but are described in the NetBSD Diskless HOW-TO.
  http://www.netbsd.org/Documentation/network/netboot/
  You will need HP-UX on your hp300 if it does not have a SCSI interface or a floppy drive.
- hp300 loads `SYS_INST` from a local disk. You partition the drive and then run the miniroot installation tools from that drive. You will need HP-UX on your hp300 if it does not have a SCSI interface.

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

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

Formatting your hard drives

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

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

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

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

Designing your disk's partition table

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

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

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

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

Here is a table of recommended partition sizes for a full install:

Partition	Suggested	Needed
/ (root)	25 MB	15 MB
/usr	150 MB	100 MB
/var	20 MB	5 MB
swap	*2-3 *RAM*	6 MB

Note: You will need at least a 6 MB swap partition if you are unable to netboot the installer, as the miniroot is temporarily placed in this partition.

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

```
# /dev/rrd0a:
type: HP-IB
disk: rd7959B
label:
flags:
bytes/sector: 512
```

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

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

Installing the bootstrap program locally

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

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

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

dd if=SYS_UBOOT of=/dev/sd1c

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

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

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

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

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

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

\# **dd if=SYS_UBOOT of=/dev/rmt/0mnb obs=20b conv=sync**

Installing the miniroot file system locally

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

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

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

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

Configuring the netboot server

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

1. Introduction

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

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

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

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

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

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

2. rbootd(8)

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

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

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

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

   You will need to start the **rbootd**. If it's already running, you will need to restart it to force it to re-read its configuration file. If the server is running NetBSD 1.5 or later, you can achieve this with:

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

3. dhcpd(8)

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

   ```
   ddns-update-style none;
                   # Do not use any dynamic DNS features
                   #
   allow bootp;    # Allow bootp requests, thus the dhcp server
                   # will act as a bootp server.
                   #
   ```

```
authoritative;  # master DHCP server for this subnet
                #
subnet 192.168.1.0 netmask 255.255.255.0 {
                # Which network interface to listen on.
                # The zeros indicate the range of addresses
                # that are allowed to connect.
}
group {
                # Set of parameters common to all clients
                # in this "group".
                #
        option broadcast-address        192.168.1.255;
        option domain-name              "test.net";
        option domain-name-servers      dns.test.net;
        option routers                  router.test.net;
        option subnet-mask              255.255.255.0;
                #
                # An individual client.
                #
        host client.test.net {
                hardware ethernet       CC:CC:CC:CC:CC:CC;
                fixed-address           192.168.1.10;
                #
                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name        "client";
                #
                # The path on the NFS server.
                #
                option root-path "/export/client/root";
        }
 #you may paste another "host" entry here for additional
 #clients on this network
}
```

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

touch /var/db/dhcpd.leases

You will need to start the **dhcpd**. If it's already running, you will need to restart it to force it to re-read its configuration file. If the server is running NetBSD 1.5 or later, you can achieve this with:

/etc/rc.d/dhcpd restart

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

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

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

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

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

If your server is currently running an NFS server, you only need to restart mountd(8). Otherwise, you need to start rpcbind(8) and nfsd(8). If the server is running NetBSD 1.5 or later, you can achieve this with:

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

5. NetBSD kernel and installation tools

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

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

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

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

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

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

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

6. Client file system

You can skip this step if you do not plan to run your client diskless after installation. Otherwise, you need to extract and set up the client's installation of NetBSD. The Diskless HOW-TO describes how to provide better security and save space on the NFS server over the procedure listed here.

http://www.NetBSD.org/Documentation/network/netboot/nfs.html

- Extracting distribution sets

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

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

- Set up swap

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

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

- Create device nodes

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

This procedure only works on NetBSD hosts.

- Set up the client's fstab(5)

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

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

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

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

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

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

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

  Edit /export/client/root/etc/hosts

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

7. Setting up the server daemons

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

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

Put Series 400 systems in HP-UX Compatible Boot Mode

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

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

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

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

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

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

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

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

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

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

See the FAQ for additional help.

http://www.netbsd.org/Ports/hp300/faq.html#domain

Searching for a bootable system

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

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

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

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

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

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

The FAQ lists additional things you can do with the BootROM and describes the order the BootROM looks for bootable devices.

http://www.netbsd.org/Ports/hp300/faq.html

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

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

BOOTROM  Series 400  Rev. 1.1
MD12 REV 1.2 1990/08/07.14:27:08
MC68030 Processor
MC68882 Coprocessor
Configuration EEPROM
Utility Chip at 41
```

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

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

Selecting ethernet port on Series 400

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

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

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

Running SYS_INST

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

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

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

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

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

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

You will need to place a disklabel on the disk.

sys_inst> **disklabel**

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

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

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

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

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

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

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

Boot from the miniroot file system.

sys_inst> **boot**

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

Chosing a kernel location

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

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

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

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

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

Installing the NetBSD System

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

./install

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

The miniroot's install program will:

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

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

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

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

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

   Anything after a '#' is a comment.

   The following is an example disklabel partition map:

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

2. Create file systems on target partitions.

3. Allow you to set up your system's network configuration. Remember to specify host names without the domain name appended to the end. For example use foo instead of foo.bar.org. If, during the process of configuring the network interfaces, you make a mistake, you will be able to re-configure that interface by simply selecting it for configuration again.

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

5. Extract binary sets from the media of your choice.

6. Copy configuration information gathered during the installation process to your root file system (/).

7. Make device nodes in your root file system under /dev.
8. Copy a new kernel onto your root partition (/).
9. Install a new boot block.
10. Check your file systems for integrity.

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

Post installation steps

Once you've got the operating system running, there are a few things you need to do in order to bring the system into a properly configured state, 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 **hp300h** for a local console, or whatever is appropriate for your serial console. and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

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

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

   Change to the /etc directory and take a look at the /etc/rc.conf file. Modify it to your tastes, making sure that you set rc_configured=YES so that your changes will be enabled and a multiuser 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=hp300h
   ```

   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"
```

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. Also, you may want to read through the NetBSD/hp300 FAQ entry on X11.
   http://www.NetBSD.org/Ports/faq.html#x11

   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 1.6.2/hp300/All subdir. You can install them with the following commands:

```
# PKG_PATH=ftp://ftp.NetBSD.org/pub/NetBSD/packages/1.6.2/hp300/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

It is possible to easily upgrade your existing NetBSD/hp300 system using the upgrade program in the miniroot or by manually performing the same steps as the miniroot upgrade program.

Upgrading using the miniroot

If you wish to upgrade your system by this method, simply select the **upgrade** option once the miniroot has booted. The upgrade program with then guide you through the procedure. While you can boot the miniroot using the same methods described above for a fresh install of NetBSD/hp300 there are easier and less intrusive options since your disk is already labeled and bootable. The easiest is to dump the miniroot to your swap partition and boot from that.

1. Download the files you'll need to upgrade

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

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

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

3. Extract and install a new boot block:

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

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

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

4. Install the miniroot file system:

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

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

5. Boot the miniroot:

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

The upgrade program will:

1. Enable the network based on your system's current network configuration.

2. Mount your existing file systems.

3. Extract binary sets from the media of your choice.

4. Fixup your system's existing /etc/fstab, changing the occurrences of **ufs** to **ffs** and let you edit the resulting file.

5. Make new device nodes in your root file system under /dev.

6. Don't forget to extract the **kern** set from the distribution.

   Note: The existing kernel *will not* be backed up; doing so would be pointless, since older kernels may not be capable of running NetBSD 1.6.2executables.

7. Install a new boot block.
8. Check your file systems for integrity.
9. You'll have to reboot your system manually

Manual upgrade

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

1. Place *at least* the **base** binary set in a file system accessible to the target machine. A local file system is preferred, since the NFS subsystem in the NetBSD 1.6.2 kernel may be incompatible with your old binaries.
2. Back up your pre-existing kernel and copy the 1.6.2 kernel into your root partition (/).
3. Extract and install a new boot block:

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

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

   E.g.: *root-disk* could be sd0 or rd0.
4. Reboot with the 1.6.2 kernel into single-user mode.
5. Check all file systems:

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

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

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

   # **cd /**
   # **pax -zrvpe -f** */path/to/***base.tgz**
9. Sync the file systems:

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

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

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

Compatibility Issues With Previous NetBSD Releases

Users upgrading from previous versions of NetBSD may wish to bear the following problems and compatibility issues in mind when upgrading to NetBSD 1.6.2.

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 replaced 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 1.6.2 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 at Ludd, University of Lule}, Sweden and its contributors.

The End