NAME

INSTALL — Installation procedure for NetBSD/atari.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 10.2_STABLE on the *atari* platform. It is available in four different formats titled `INSTALL.`*`ext`*, where *`.ext`* is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

What is NetBSD?

The NetBSD Operating System is a fully functional open-source 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 many different different system architectures (ports) across a variety of distinct CPU families, and is being ported to more. The NetBSD 10.2_STABLE release contains complete binary releases for most of these system architectures, with preliminary support for the others included in source form. For more information please visit **https://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, NetBSD would not exist.

Changes Between The NetBSD 9 and 10 Releases

The NetBSD 10.2_STABLE release provides many significant changes, including support for many new devices, hundreds of bug fixes, new and updated kernel subsystems, and numerous userland enhancements. The result of these improvements is a stable operating system fit for production use that rivals most commercially available systems.

One important new feature in this release is the support for extended attributes and access control lists on FFS file systems.

For new installations the installer will by default disable these features, so the file system is compatible with older NetBSD releases (before 10), and allow other operating systems to mount this file systems at least in read-only mode.

If you want a new installed file system to support extended attributes, change the file system type from "FFSv2" to "FFSv2ea" in the partitioning menu. You can also convert file systems later, using the `fsck_ffs`(8) utility. More details are available in *this guide:*

https://wiki.netbsd.org/tutorials/acls_and_extended_attributes_on_ffs.

If you are upgrading from a version of NetBSD -current please also check the **Compatibility Issues With Previous NetBSD Releases**. It is impossible to completely summarize the massive development that went into the NetBSD 10.2_STABLE release. The complete list of changes can be found in
CHANGES:

https://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-10/latest/CHANGES

CHANGES-10.1:

https://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-10/latest/CHANGES-10.1

CHANGES-10.2:

https://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-10/latest/CHANGES-10.2

CHANGES-10.3:

https://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-10/latest/CHANGES-10.3

which are also present in the top level directory of the NetBSD 10.2 release tree.

Features to be removed in a later release

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

- `groff`(1). Man pages are now handled with `mandoc`(1), and `groff`(1) can still be found in pkgsrc as `textproc/groff`.
- `pf`(4). This packet filter is obsolete and unmaintained in NetBSD. It will be eventually removed due to possible long-standing security issues and lack of multiprocessor support. New installations should use `npf`(7).

The NetBSD Foundation

The NetBSD Foundation is a tax exempt, not-for-profit 501(c)(3) corporation that devotes itself to the traditional goals and spirit of the NetBSD Project and owns the trademark of the word "NetBSD". It supports the design, development, and adoption of NetBSD worldwide. More information on the NetBSD Foundation, its composition, aims, and work can be found at: **https://www.NetBSD.org/foundation/**

Sources of NetBSD

Refer to *mirrors:* **https://www.NetBSD.org/mirrors/**

NetBSD 10.2_STABLE Release Contents

The root directory of the NetBSD 10.2_STABLE release is organized as follows:

`.../NetBSD-10.2_STABLE/`

`CHANGES`	Changes between the 9.0 and 10.0 releases.
`CHANGES-10.0`	Changes between the initial 10.0 branch and final release of 10.0.
`CHANGES-10.1`	Changes between the 10.0 and the 10.1 release.
`CHANGES-10.2`	Changes between the 10.1 and the 10.2 release.
`CHANGES-10.3`	Changes after the release of 10.2.
`CHANGES.prev`	Changes in previous NetBSD releases.
`README.files`	README describing the distribution's contents.
`images/`	Images (ISO 9660 or USB) for installing NetBSD. Depending on your system, these may be bootable.
`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 10.2_STABLE has a binary distribution.

The source distribution sets can be found in subdirectories of the `source` subdirectory of the distribution tree. They contain the complete sources to the system. The source distribution sets are as follows:

gnusrc This set contains the "gnu" sources, including the source for the compiler, assembler, groff, and the other GNU utilities in the binary distribution sets.

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

src This set contains all of the base NetBSD 10.2_STABLE sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 10.2_STABLE kernel for all architectures as well as the `config`(1) utility.

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

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

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

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

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

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

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

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

NetBSD/atari subdirectory structure

The atari-specific portion of the NetBSD 10.2_STABLE release is found in the `atari` subdirectory of the distribution: `.../NetBSD-10.2_STABLE/atari/`. It contains the following files and directories:

`INSTALL.html`
`INSTALL.ps`
`INSTALL.txt`
`INSTALL.more` Installation notes in various file formats, including this file. The `.more` file contains underlined text using the `more`(1) conventions for indicating italic and bold display.

`binary/`
- `kernel/`
  - `netbsd-SMALL030.gz`
    A kernel for memory-tight 68030 Falcon and TT030 machines. This is the kernel supplied on the boot.fs floppy.
  - `netbsd-ATARITT.gz`
    A kernel for the Atari TT030.
  - `netbsd-FALCON.gz`
    A kernel for the Atari Falcon, including support for 68060 processor cards.
  - `netbsd-HADES.gz`
    A kernel for 68040 or 68060 Hades.
  - `netbsd-MILAN-ISAIDE.gz`
    A kernel for 68040 or 68060 MILAN using IDE in ISA mode - for use with disks not marked byteswapped by HDDRIVER.

`netbsd-MILAN-PCIIDE.gz`
A kernel for 68040 or 68060 MILAN using IDE in PCI mode - for use with disks marked byteswapped by HDDRIVER.

`sets/` atari binary distribution sets; see below.

`installation/`

`floppy/` atari boot and installation floppies; see below.

`miniroot/` atari miniroot file system image; see below.

`misc/` Miscellaneous atari installation utilities; see installation section below.

Binary distribution sets

The NetBSD atari binary distribution sets contain the binaries which comprise the NetBSD 10.2_STABLE release for atari. The binary distribution sets can be found in the `atari/binary/sets` subdirectory of the NetBSD 10.2_STABLE distribution tree, and are as follows:

base The NetBSD 10.2_STABLE atari **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional.

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

debug This distribution set contains debug information for all base system utilities. It is useful when reporting issues with binaries or during development. This set is huge, if the target disk is small, do not install it.

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

games This set includes the games and their manual pages.

kern-SMALL030, kern-ATARITT, kern-FALCON, kern-HADES, kern-MILAN-ISAIDE, kern-MILAN-PCIIDE
These sets contain a NetBSD/atari 10.2_STABLE kernel, named `/netbsd`. You *must* install the kernel that matches your hardware.

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

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

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

rescue This set includes the statically linked emergency recovery binaries installed in `/rescue`.

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

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

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

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

xdebug This distribution set contains debug information for all X11 binaries. It is useful when reporting issues with these binaries or during development. This set is huge, if the target disk is small, do not install it.

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

xetc Configuration files for X which could be locally modified.

xserver The X server.

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

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

Additional kernels to those included in the distribution sets may be found in the `atari/binary/kernel` subdirectory of the NetBSD 10.2_STABLE distribution tree. These kernels are generally named something like `netbsd-BOOT.gz` or some other suitable name. Please note that these kernels are simply gzipped and are not in tar archives.

There are six atari floppy images to be found in the `atari/installation/floppies` subdirectory of the NetBSD 10.2_STABLE distribution. Three of them are bootable TOS kernel floppies and the other three are installation floppies. They are described in more detail below. There are gzipped versions of each available, for easier downloading. (The gzipped versions have the `.gz` extension added to their names.)

Bootable Kernel floppies

These TOS disks contains the TOS support programs (like **`loadbsd.ttp`** and **`chg_pid.ttp`** — see below) and a kernel. They are setup such that you can insert them into your floppy drive, and start the programs from GEM.

For the TT030 and Falcon, the (720 KB) floppy is named `boot.fs` and the kernel supplied is `BOOT`. For the Hades, you need the `hades-boot.fs` floppy (1.44 MB), the kernel is `HADES`. For the Milan, you need the `milan-boot.fs` floppy (1.44 MB), the kernel is MILAN-PCIIDE.

Installation floppy:

This disk contains a BSD root file system setup to help you install the rest of NetBSD. This includes formatting and mounting your `/` (root) and `/usr` partitions and getting ready to extract (and possibly first fetching) the distribution sets. There is enough on this file system to allow you to make a slip connection, configure an ethernet, mount an NFS file system or ftp. You can also load distribution sets from a SCSI tape or from one of your existing GEMDOS partitions.

The floppies meant for 720 KB disks are named `miniroot.fs.1` and `miniroot.fs.2`. There is also an image for an 1.44 MB disk: `miniroot.fs`.

There are also TOS utilities in the `atari/installation/misc` subdirectory, which you will need to get NetBSD/atari up-and-running.

- The **`gzip.ttp`** program allows you to uncompress .gz images. The usage is:

  `gzip.ttp -d filename.gz`

- The **`rawwrite.ttp`** program allows you to create the installation floppy disks from the files in the `atari/floppies` directory.

- The **aptck.ttp** program reads the partition tables present on a given disk and tries to interpret then the same way the NetBSD kernel does. If you have a disk on which GEMDOS and NetBSD are to co-exist, It is a good idea to run this before you begin the NetBSD/atari installation just to check that the kernel's view of the partition tables agree with GEMDOS's view. If you have more than 3 partitions defined on a disk you will notice that the NetBSD/atari partition starts one sector after the GEMDOS partition. This is to allow space for the auxiliary root for the 4th and subsequent partitions.
- The **loadbsd.ttp** program loads the NetBSD/atari kernel from TOS (or MiNT, MultiTOS, etc.).
- The **chg_pid.ttp** program can be used to change the id of an AHDI compatible partition. This allows you to reserve a partition for use by NetBSD/atari (id: NBD), or for use as a swap (id: SWP) partition.
- The **file2swp.ttp** program can be used to transfer an installation file system image, like `sysinst.fs` to a partition with id SWP. In this way, you can run the installation from the swap partition instead of a memory disk. This is a necessity for small memory systems.

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

NetBSD/atari System Requirements and Supported Devices

NetBSD/atari 10.2_STABLE runs on the TT030, Falcon, Hades and MilanI machines. An FPU is not required. The minimum amount of RAM required is 4 MB. On the Hades, only the Tseng PCI VGA cards (ET4000/ET6000/ET6100) are supported in the 10.2_STABLE release. When an unsupported video card is present, you can use NetBSD with a serial console only.

A minimal installation requires around 100MB of disk space, plus a swap partition size of 2*RAM (and always greater than 20MB). You will probably want to compile your own kernel, as `GENERIC` is large and bulky in order to accommodate all people, `BOOT` is small and might not have all the features you want).

Supported devices

- The builtin SCSI host adapter
  - Should support most SCSI-drives.
  - Should support most tape drives.
  - Should support most CD-ROM drives.
  - Should support most ZIP/M-O drives.
- The builtin video controller
- The builtin (720 KB / 1.44 MB) floppy drive
- The serial2/modem2 ports
- The Falcon FX memory expansion
- The Atari mouse
- A 3-button mouse (see build description in the FAQ!)
- The parallel printer
- IDE interface on both Falcon and Hades (includes ATAPI support)
- The serial interface on the first 68901 UART (modem1)
- EtherNEC Ethernet on ROM cartridge slot
- VME-bus devices (TT030/Hades)

- BVME410 Ethernet
- Circad Leonardo 24-bit VME graphics adapter
- Crazy Dots VME et4000 graphics adapter
- Riebl (and possibly PAM) Ethernet cards on the VME bus.
- SMC Elite Ultra ISA Ethernet with SMC_TT VME-ISA bridge

- PCI-bus devices (Hades and Milan only)
  - ET4000/ET6000/ET6100-PCI (VGA console)
  - ESS Technology Inc. Solo-1 Soundcard
  - 3Com 3c59x Network card

- ISA-bus devices (Hades only)
  - Teles S0/16.3-ISA ISDN adapter (with I4B)

This list is incomplete by definition. We can not test all SCSI peripherals, ISA cards or PCI cards. If you have problems with such peripherals, please contact the `port-atari@NetBSD.org` mailing list.

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- GEMDOS/TOS HD partitions
- Tape

No matter what you do, however, you'll need to have three disks handy, on which you will put the install and boot floppy images.

All the images are available from the directory `installation/floppies` under the root of the NetBSD/atari tree at your favorite archive site.

If you are using NetBSD/atari to make the floppies, you should use the command dd(1) to write the raw floppy images (.fs files) to the disk. As an example, to write the first part of the miniroot file system onto a 720 KB floppy in fd0 use:

```
# dd if=miniroot.fs.1 of=/dev/rfd0b bs=9b
```

If you are using TOS to make the floppies, grab the **rawwrite** utility from the `atari/utils` directory and issue the command:

```
rawwrite boot.fs
```

This will create the boot-floppy on the floppy in drive a. The floppies should be pre-formatted on 720 KB / 1.44 MB for both the **dd** and **rawwrite** commands to work. Pre-formatting can be best done using the desktop format command. Some other utilities seem to be giving problems.

Since the 1.3 release, it is also possible to use HD-floppies. You should than use the floppy device `/dev/rfd0c` or add the **-H** flag to **rawwrite**.

The steps necessary to prepare the distribution sets for installation depend on which method of installation you choose. The various methods are explained below.

GEMDOS To install NetBSD from a GEMDOS partition, you need to get the NetBSD distribution sets you wish to install on your system on to a GEMDOS partition.

Note where you placed the files, you will need this later.

Once you have done this, you can proceed to the next step in the installation process, preparing your hard disk.

Tape To install NetBSD from a tape, you need to somehow get the NetBSD filesets you wish to install on your system on to the appropriate kind of tape. If you make the tape on a UNIX-like system, you can create it as follows:

```
# cd .../NetBSD-10.2_STABLE/atari/binary/sets
# T=<tape_device>
# mt -f $T rewind
# for f in base etc comp games man misc rescue text
    dd if=$f.tgz of=$T conv=osync bs=5k
# done
# mt -f $T rewind
```

where "<tape_device>" is the name of the (non-rewinding!) tape device that you're using (probably something like /dev/nrst0, but we make no guarantees 8-). If you can't figure it out, ask your system administrator.

Once you have done this, you can proceed to the next step in the installation process, preparing your hard disk.

Preparing your System for NetBSD installation

Preparing your hard disk for the NetBSD installation.

Note that you will be modifying your hard drives and if you mess something up here you could lose everything on all the drives that you work with. It is therefore advised that you:

- Write down your current configurations. Do this by writing down all partition info (especially their sizes).
- *Back up the partitions you are keeping.*

If NetBSD has to share the disk with another operating system, you can either take care of partitioning your harddisk before installing NetBSD or delay this until the installer requests you to do it. If you decide to do it now, please create space for at least / (root) and swap partitions. Whatever you decide, take note of the paragraphs below! You need this info to create bootable disks.

Warning: The AHDI partitioning function erases all partitions on your harddisk even if they are not changed! We know this is rather stupid, but don't say we didn't warn you.

If you want to use an AHDI partitioning scheme and you want to be able to boot directly into NetBSD, there are some constraints on the partition layout.

As you might know, every hard disk has a 'root sector' that contains information about the size of the hard disk and the partitions on the hard disk. The root sector can only contain the necessary data for four partitions. Nobody thought that this limitation would cause any problems. After all, 640 KByte should be enough. As hard disks grew, it was necessary to define more than four partitions. In order to be more or less compatible with the old format, a new type of partition entry was defined: XGM partitions.

An XGM partition is a 'look over there' sign: Another root sector can be found at the start of the XGM partition. This root sector contains the remaining real partitions. And this is the big mystery: Partitions defined in the root sector of the hard disk are called 'primary partitions', partitions defined in the root sector of an XGM partition are called 'extended partitions'.

The bootblock will only work if the first NBD partition is a primary partition. This is not a limitation of NetBSD but a limitation of TOS/AHDI: You can only boot from primary partitions.

If you are creating your partitions with HDX, you'll have to be very careful to fulfill this rule. HDX has some very strange ideas when it comes to extended partitions. Fortunately, you can edit this stuff: The "Edit partition scheme of the unit" dialog box has a button labeled "expert". This button is inactive unless you have defined more than four partitions. Click on it *after* you have defined the sizes of the partitions.

A new dialog box appears on the screen. The left side contains two blocks of partitions; the upper block always contains the first four partitions, the lower block contains the last three partitions. If you have defined less than 7 partitions, some fields of the lower block will contain the string "unused". Some of the partitions will be displayed in reverse video: These are the extended partitions.

The right side contains six possible ranges for the extended partitions. It is not possible to define your own range, you will have to use one of the schemes offered by HDX. To quote from Ghostbusters: Choose and die. The default scheme used by HDX is the first scheme: Extended partitions start with the second partition and end with the second to last partition. If you have defined 7 partitions, partitions #2 to #5 will be extended partitions, while partitions #1, #6 and #7 will be primary partitions.

You can move the extended partition range by clicking on one of the buttons on the right side of the dialog box. Try to find one where your first NetBSD partition is a primary partition. Golden rules:

- If the disk contains no GEMDOS partitions, don't use AHDI. Let NetBSD handle it alone.
- If the disk contains one GEMDOS partition, make it partition #1 and start the extended partition range at partition #3. This allows you to boot from both the GEMDOS and the NetBSD partitions.
- If the disk contains two GEMDOS partitions, use partitions #1 and #2 for GEMDOS, partition #3 for NetBSD-root. Start the extended partition range with partition #4.
- If your disks contains three or more GEMDOS partitions, you are in trouble. Try using partitions #1 and #2 as the first two GEMDOS partitions. Use partition #3 as the first NetBSD partition. Start the extended partition range with partition #4. Put the other NetBSD extended partition range.

Booting the installer on an existing system

If you already have NetBSD installed, it is easy. Just boot into single user mode on your current system, or use the **shutdown now** command to shutdown to single-user mode. Then copy the installer using **dd**:

```
dd if=sysinst.fs of=/dev/rsd0b
```

where `/dev/rsd0b` should be the device path of the swap partition your system is configured to use. Once the file is copied, reboot back to the booter and use the '-b' option to make the kernel prompt for a root device. Now enter the swap device as your root device to start the installer.

Booting the installer on new systems.

If your system has more than 6MB RAM, you can skip directly to the section about **loadbsd.ttp,** since the amount of RAM is enough to load the installer into.

For new new installations on small memory systems, things are a bit more complicated. The easiest way is to mark an AHDI partition as id 'SWP'. You can either accomplish this with HDdriver or use the **chg_pid.ttp** program supplied on the bootfloppy. To use the fifth partition of your first drive to a swap partition, do the following:

```
chg_pid -w 0 4 SWP
```

As you can see, both the drive and the partition numbers are zero based.

When a swap partition has been created, you can transfer the installer using the **file2swp.ttp** command, which is also on the bootfloppy. This program copies a (gzipped) file to the first AHDI partition with id 'SWP' on the designated disk. To copy sysinst.fs to your first SCSI disk, use:

```
file2swp -w -f sysinst.fs s0
```

Note that **file2swp** automatically detects a gzipped file, there is no need for a .gz extension.

Using loadbsd.ttp

To start the installer, you first need to load a NetBSD kernel. This can be done in a couple ways, both of which currently require GEMDOS. You need either the bootfloppy provided in the distribution or you can copy the **loadbsd.ttp** program and kernel to a boot floppy disk (1.44 MB needed) or put them on a TOS partition. Select the loadbsd program and it will ask for parameters, supply: '**-b netbsd**' (or whatever name you copied the kernel to). You can, of course, also run it from the shell command-line in MiNT:

loadbsd -b a:/netbsd

You should see the screen clear and some information about your system as the kernel configures the hardware. Then you will be prompted for a root device. If you copied the installer to a swap partition, as explained above, you have to enter the swap partition as the boot device. When you used your first SCSI disk, this will be **sd0b.** On the first IDE drive it would be **wd0b.**

If you didn't copy the installer to swap, you need to use the RAMdisk, so remove the GEMDOS kernel boot floppy from the drive if present and insert the NetBSD install floppy 1. Now enter the name of the boot device depending on the floppy type you have inserted. Use:

- **md0a** for the old installer (1MB filesystem) on 720KB floppies.
- **md1a** for the sysinst installer on 720KB floppies.
- **md2a** for the sysinst installer on 1.44MB floppies.

The kernel will load the install file system into RAMdisk. While the kernel is loading, it will show a '.' for each track loaded. After loading 80 tracks, it will ask you to insert the next floppy. At this time, insert the NetBSD install floppy 2 and press any key. The kernel continues loading another 40 tracks before it continues to boot.

Installing the NetBSD System

Installing NetBSD is a relatively complex process, but, if you have this document in hand and are careful to read and remember the information which is presented to you by the install program, it shouldn't be too much trouble.

Before you begin, you must have already prepared your hard disk as detailed in the section on preparing your system for install.

The following is a walk-through of the steps necessary to get NetBSD installed on your hard disk.

Running the sysinst installation program

1. *Introduction*

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

2. *General*

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

3. *Quick install*

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

   - What you need.
     - The distribution sets (in this example, they are on the CD or DVD).
     - A floppy disk containing a suitable boot.fs and a floppy with the sysinst.fs image. See the "prepare" section about obtaining those.
     - A minimum of 4 MB (TT030/Falcon) or 16 MB (Hades/Milan) of memory installed.
     - An optical drive.
     - A hard drive with at least 500 MB of free space for a complete base install, not including room for swap. If you wish to install the X Window System as well, you will need at least 225 MB more.
   - The Quick Installation
     - Boot the system as described above. You should be at the **sysinst** main menu.

```
.***********************************************.
* NetBSD-10.2_STABLE Install System             *
*                                               *
*>a: Install NetBSD to hard disk                *
* b: Upgrade NetBSD on a hard disk              *
* c: Re-install sets or install additional sets *
* d: Reboot the computer                        *
* e: Utility menu                               *
* f: Config menu                                *
* x: Exit Install System                        *
.***********************************************.
```

     - If you wish, you can configure some network settings immediately by choosing the **Utility menu** and then **Configure network**. It isn't actually required at this point, but it may be more convenient. Go back to the main menu.
     - Choose **Install**.
     - You will be guided through the setup of your disk.
     - You will be asked to choose which distribution sets to install. WARNING: If you want your disk to retain an AHDI compatible partitioning, make sure to use the 'existing' disk layout.
     - When prompted, choose **CD-ROM** as the install medium if booted from CD-ROM. The default values for the path and device should be OK.
     - After the installation process has completed, you will be brought back to the main menu, where you should select **Reboot**, after you have removed the bootfloppy from the drive.
     - NetBSD will now boot. If you didn't set a password for the `root` user when prompted by **sysinst**, logging in as `root` and setting a password should be your first task. You are also advised to read `afterboot`(8).

4. *Booting NetBSD*

Boot the system as described in the "Booting the installer" section above.

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

5. *Network configuration*

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

6. *Installation drive selection and parameters*

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

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

7. *Selecting which sets to install*

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

8. *Partitioning the disk*

   - Preparing a disk which will be used for GEM and NetBSD.

     You will be prompted if you want an AHDI compatible partitioning on your disk. If you are installing NetBSD on a dedicated drive, just answer 'no' and skip to the next section.

     If you answer 'yes', the **ahdilabel** program is started. You can now change the AHDI partition IDs on your root disk. Because NetBSD imposes a special ordering in disk partitions it uses for / (root) and swap. Also, because it wants to guard you against an unwanted demolition of partitions used by other systems, you have to tell it what partitions it is allowed to use. You have to mark the partition you want to use as swap SWP or (deprecated!) NBS and the other partitions as NBD. Note that all the changes you make to the IDs are reversible as long as you remember the original value. ahdilabel is capable of creating or changing an AHDI compatible partitioning on the disk, and in the partition-ID editor, the partitions are shown in the order that AHDI created them. When you leave this editor and continue with **sysinst,** your changes to the IDs do have consequences to the partition order! They will show up as follows:

`a`	the first NBD partition
`b`	the first SWP (or NBS) partition
`d (and up)`	the rest of the partitions in AHDI order

9. *Editing the NetBSD disklabel*

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

   If you want to use NetBSD on an AHDI partitioned disk, you will have to use: **`Use Existing`**.

   After you have chosen your partitions and their sizes (or if you opted to use the existing partitions), you will be presented with the layout of the NetBSD disklabel and given one more chance to change it. For each partition, you can set the type, offset and size, block and fragment size, and the mount point.

The type that NetBSD uses for normal file storage is called **4.2BSD**. A swap partition has a special type called **swap**. Some partitions in the disklabel have a fixed purpose.

`a` Root partition (/)

`b` Swap partition.

`c` The entire disk.

`d-p` Available for other use. Traditionally, `d` is the partition mounted on `/usr`, but this is historical practice and not a fixed value.

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

10. *Preparing your hard disk*

    You are now at the point of no return. Apart from the changes you made with **ahdilabel** nothing has been modified on your disk yet. If you confirm that you want to install NetBSD, the partitions now assigned to NetBSD will be actually written to. If you are sure you want to proceed, select **yes**.

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

11. *Getting the distribution sets*

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

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

12. *Installation from CD-ROM*

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

    sysinst will then check that the files are actually present in the specified location and proceed to the extraction of the sets.

13. *Installation using FTP*

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

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

sysinst will then transfer the set files from the remote site to your hard disk.

14. *Installation using NFS*

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

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

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

15. *Installation from an unmounted file system*

    In order to install from a local file system, you will need to specify the device that the file system resides on (for example `wd1e`), the type of the file system, and the directory on the specified file system where the sets are located. **sysinst** will then check if it can indeed access the sets at that location.

16. *Installation from a local directory*

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

17. *Extracting the distribution sets*

    A progress bar will be displayed while the distribution sets are being extracted.

    After all the files have been extracted, the device node files will be created. If you have already configured networking, you will be asked if you want to use this configuration for normal operation. If so, these values will be installed in the network configuration files.

18. *Configure additional items*

    The next menu will allow you to select a number of additional items to configure, including the time zone that you're in, to make sure your clock has the right offset from UTC, the root user's shell, and the initial root password.

    You can also enable installation of binary packages, which installs the `pkgin`(1) tool for managing binary packages for third-party software. This will feel familiar to users of package tools such as **apt-get** or **yum**. If you prefer to install third-party software from source, you can install the `pkgsrc`(7) tree.

    Finally, you can enable some daemons such as `sshd`(8), `ntpd`(8), or `mdnsd`(8).

19. *Finalizing your installation*

    Congratulations, you have successfully installed NetBSD 10.2_STABLE. You can now reboot the machine and boot NetBSD from hard disk.

Post installation steps

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

1. Before all else, read `postinstall`(8).

2. Configuring `/etc/rc.conf`

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

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

   and with the root file system (`/`) mounted read-only. When the system asks you to choose a shell, simply press RETURN to get to a `/bin/sh` prompt. If you are asked for a terminal type, respond with **vt220** (or whatever is appropriate for your terminal type) and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

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

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

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

   When you have finished editing `/etc/rc.conf`, type **exit** at the prompt to leave the single-user shell and continue with the multi-user boot.

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

   ```
   ifconfig_le0="inet 192.0.2.123 netmask 255.255.255.0"
   ```

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

   ```
   ifconfig_le0="inet myname.my.dom netmask 255.255.255.0"
   ```

   To enable proper hostname resolution, you will also want to add an `/etc/resolv.conf` file or (if you are feeling a little more adventurous) run `named`(8). See `resolv.conf`(5) or `named`(8) for more information.

   Instead of manually configuring networking, DHCP can be used by setting `dhcpcd=YES` in `/etc/rc.conf`.

3. Select the proper terminal devices

   You will have to edit the `/etc/ttys` file. If you don't have a Milan, you must disable the `ttyEx` devices. For the Milan, you will have to disable the `ttyex` devices. This is easy, just change 'on' to 'off' on the corresponding lines.

4. Logging in

   After reboot, you can log in as `root` at the login prompt. If you didn't set a password in **sysinst**, there is no initial password. You should create an account for yourself (see below) and protect it and the "root" account with good passwords. By default, root login from the network is disabled (even via `ssh`(1)). One way to become root over the network is to log in as a different user that belongs to group "wheel" (see `group`(5)) and use `su`(1) to become root.

5. Adding accounts

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

6. The X Window System

   If you installed the X Window System, you may want to read the chapter about X in the *NetBSD Guide:*
   https://www.NetBSD.org/docs/guide/en/chap-x.html:

7. Installing third-party packages

   If you wish to install any of the software freely available for UNIX-like systems you are strongly advised to first check the NetBSD package system, pkgsrc. pkgsrc automatically handles any changes necessary to make the software run on NetBSD. This includes the retrieval and installation of any other packages the software may depend upon.

   - More information on the package system is available at
     https://www.NetBSD.org/docs/software/packages.html

   - A list of available packages suitable for browsing is at
     https://cdn.NetBSD.org/pub/pkgsrc/current/pkgsrc/README.html

   - Precompiled binaries can be found at
     https://cdn.NetBSD.org/pub/pkgsrc/packages/NetBSD/
     usually in the `atari/10.2_STABLE/All` subdir. If you installed `pkgin`(1) in the **sysinst** post-installation configuration menu, you can use it to automatically install binary packages over the network. Assuming that `/usr/pkg/etc/pkgin/repositories.conf` is correctly configured, you can install them with the following commands:

     ```
     # pkgin install tcsh bash perl apache xfce4 firefox
     ...
     ```

     The above command will install the Tenex-csh and Bourne Again shells, the Perl programming language, Apache web server, Xfce desktop environment and the Firefox web browser as well as all the packages they depend on.

     If it was not automatically installed, `pkgin`(1) can be installed on a fresh NetBSD system with `pkg_add`(1):

     ```
     export PKG_PATH=https://cdn.netbsd.org/pub/pkgsrc/packages/NetBSD/$(uname -p)/$
     pkg_add pkgin
     ```

   - If you wish to use the `pkgsrc`(7) framework for compiling packages and did not install it from the `sysinst`(8) post-installation configuration menu, you can obtain it by retrieving the file
     https://cdn.NetBSD.org/pub/pkgsrc/stable/pkgsrc.tar.gz.
     It is typically extracted into `/usr/pkgsrc` (though other locations work fine) with the commands:

     ```
     # cd /usr
     # tar -zxpf pkgsrc.tar.gz
     ```

     After extracting, see the `doc/pkgsrc.txt` file in the extraction directory (e.g., `/usr/pkgsrc/doc/pkgsrc.txt`) for more information.

8. Misc

   - Edit `/etc/mail/aliases` to forward root mail to the right place. Don't forget to run `newaliases`(1) afterwards.

   - Edit `/etc/rc.local` to run any local daemons you use.

- Many of the `/etc` files are documented in section 5 of the manual; so just invoking

  `# man 5 filename`

  is likely to give you more information on these files.

Upgrading a previously-installed NetBSD System

The easiest way to upgrade to NetBSD 10.2_STABLE is with binaries, and that is the method documented here.

To do the upgrade, you must first boot the system as described in the **Booting the installer** section. You must also have at least the **base** and **kern** binary distribution sets available. Finally, you must have sufficient disk space available to install the new binaries. Since files already installed on the system are overwritten in place, you only need additional free space for files which weren't previously installed or to account for growth of the sets between releases.

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

The upgrade procedure is similar to an installation, but without the hard disk partitioning.

Fetching the binary sets is done in the same manner as the installation procedure; refer to the installation part of the document for help. File systems are checked before unpacking the sets.

After a new kernel has been copied to your hard disk, your machine is a complete NetBSD 10.2_STABLE system. However, that doesn't mean that you're finished with the upgrade process. You will probably want to update the set of device nodes you have in `/dev`. If you've changed the contents of `/dev` by hand, you will need to be careful about this, but if not, you can just cd into `/dev`, and run the command:

`# sh MAKEDEV all`

sysinst will attempt to merge the settings stored in your `/etc` directory with the new version of NetBSD using the `postinstall`(8) utility. However, `postinstall`(8) is only able to deal with changes that are easily automated. It is recommended that you use the `etcupdate`(8) tool to merge any remaining configuration changes.

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 10.2_STABLE.

Note that **sysinst** will automatically invoke

`postinstall fix`

and thus all issues that are fixed by **postinstall** by default will be handled.

In NetBSD 9 and earlier, filesystems listed in `/etc/fstab` would be mounted before non-legacy **zfs** filesystems. Starting from NetBSD 10 this order has been reversed.

If you have ever run a version of NetBSD -current between April 18, 2020 and September 23, 2022 (the version numbers used in the affected time range are between 9.99.56 and 9.99.106) your FFS file systems might have broken extended attributes stored.

You should follow this *guide:*

https://wiki.netbsd.org/features/UFS2ea/

before booting the updated system multi-user for the first time.

Note that you do not need to do anything special if you never did run any affected kernel, especially if you have never run NetBSD -current.

The display drivers used for modern GPUs and the whole subsystem supporting it (DRM/KMS) have been updated to a newer version. Unfortunately not all issues with this have been resolved before the NetBSD 10.0 release. You can find a list of issues in the *Open issues with new DRM/KMS:*

https://wiki.netbsd.org/releng/netbsd-10/

section of the release engineering wiki page.

A number of things have been removed from the NetBSD 10.2_STABLE release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Using online NetBSD documentation

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

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

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

If you are unsure of what man page you are looking for, enter

```
# apropos subject-word
```

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

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at `majordomo@NetBSD.org`. See **https://www.NetBSD.org/mailinglists/** for details.

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

To report bugs, use the `send-pr`(1) command shipped with NetBSD, and fill in as much information about the problem as you can. Good bug reports include lots of details.

Bugs also can be submitted and queried with the web interface at

https://www.NetBSD.org/support/send-pr.html

There are also port-specific mailing lists, to discuss aspects of each port of NetBSD. Use majordomo to find their addresses, or visit **https://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 NetBSD, and have an idea as to how you could be useful, send us mail or subscribe to: `netbsd-users@NetBSD.org`.

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

Thanks go to

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

  Keith Bostic
  Ralph Campbell
  Mike Karels
  Marshall Kirk McKusick

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

- The Internet Systems Consortium, Inc. for hosting the NetBSD FTP, CVS, AnonCVS, mail, mail archive, GNATS, SUP, Rsync and WWW servers.
- The Internet Research Institute in Japan for hosting the server which runs the CVSweb interface to the NetBSD source tree.
- The Columbia University Computer Science Department for hosting the build cluster.
- The many organizations that provide NetBSD mirror sites.
- Without CVS, this project would be impossible to manage, so our hats go off to Brian Berliner, Jeff Polk, and the various other people who've had a hand in making CVS a useful tool.
- We list the individuals and organizations that have made donations or loans of hardware and/or money, to support NetBSD development, and deserve credit for it at **https://www.NetBSD.org/donations/** (If you're not on that list and should be, tell us! We probably were not able to get in touch with you, to verify that you wanted to be listed.)
- Finally, we thank all of the people who've put sweat and tears into developing NetBSD since its inception in January, 1993. (Obviously, there are a lot more people who deserve thanks here. If you're one of them, and would like to be mentioned, tell us!)

Legal Mumbo-Jumbo

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

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

NetBSD is a registered trademark of The NetBSD Foundation, Inc.
This product includes software developed by the University of California, Berkeley and its contributors.
This product includes software developed by the NetBSD Foundation.
This product includes software developed by The NetBSD Foundation, Inc. and its contributors.
This product includes software developed for the NetBSD Project. See https://www.NetBSD.org/ for information about NetBSD.
This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)
This product includes cryptographic software written by Eric Young (eay@mincom.oz.au)
This product includes software designed by William Allen Simpson.
This product includes software developed at Ludd, University of Luleå.
This product includes software developed at Ludd, University of Luleå, Sweden and its contributors.
This product includes software developed at the Information Technology Division, US Naval Research Laboratory.
This product includes software developed by Aaron Brown and Harvard University.
This product includes software developed by Adam Ciarcinski for the NetBSD project.
This product includes software developed by Adam Glass.

This product includes software developed by Adam Glass and Charles M. Hannum.
This product includes software developed by Alex Zepeda.
This product includes software developed by Alex Zepeda, and Colin Wood for the NetBSD Project.
This product includes software developed by Allen Briggs.
This product includes software developed by Amancio Hasty and Roger Hardiman
This product includes software developed by Ben Gray.
This product includes software developed by Berkeley Software Design, Inc.
This product includes software developed by Bill Paul.
This product includes software developed by Bodo Möller.
This product includes software developed by Boris Popov.
This product includes software developed by Brini.
This product includes software developed by Bruce M. Simpson.
This product includes software developed by Causality Limited.
This product includes software developed by Charles Hannum.
This product includes software developed by Charles M. Hannum.
This product includes software developed by Charles M. Hannum, by the University of Vermont and State Agricultural College and Garrett A. Wollman, by William F. Jolitz, and by the University of California, Berkeley, Lawrence Berkeley Laboratory, and its contributors.
This product includes software developed by Christian E. Hopps.
This product includes software developed by Christian E. Hopps, Ezra Story, Kari Mettinen, Markus Wild, Lutz Vieweg and Michael Teske.
This product includes software developed by Christopher G. Demetriou.
This product includes software developed by Christopher G. Demetriou for the NetBSD Project.
This product includes software developed by Chuck Silvers.
This product includes software developed by Cisco Systems, Inc.
This product includes software developed by Colin Wood.
This product includes software developed by Colin Wood for the NetBSD Project.
This product includes software developed by Computing Services at Carnegie Mellon University (http://www.cmu.edu/computing/).
This product includes software developed by Daan Vreeken.
This product includes software developed by Daishi Kato
This product includes software developed by Daniel Widenfalk and Michael L. Hitch.
This product includes software developed by Daniel Widenfalk for the NetBSD Project.
This product includes software developed by David Jones and Gordon Ross
This product includes software developed by David Miller.
This product includes software developed by Dean Huxley.
This product includes software developed by Emmanuel Dreyfus
This product includes software developed by Eric S. Hvozda.
This product includes software developed by Eric S. Raymond
This product includes software developed by Eric Young (eay@cryptsoft.com)
This product includes software developed by Eric Young (eay@mincom.oz.au)
This product includes software developed by Ezra Story.
This product includes software developed by Ezra Story and by Kari Mettinen.
This product includes software developed by Ezra Story, by Kari Mettinen and by Bernd Ernesti.
This product includes software developed by Ezra Story, by Kari Mettinen, and Michael Teske.
This product includes software developed by Ezra Story, by Kari Mettinen, Michael Teske and by Bernd Ernesti.
This product includes software developed by Frank van der Linden for the NetBSD Project.
This product includes software developed by Gardner Buchanan.
This product includes software developed by Garrett D'Amore.
This product includes software developed by Gary Thomas.
This product includes software developed by Gordon Ross
This product includes software developed by Harvard University.
This product includes software developed by Harvard University and its contributors.
This product includes software developed by Hellmuth Michaelis and Joerg Wunsch
This product includes software developed by Henrik Vestergaard Draboel.
This product includes software developed by Herb Peyerl.
This product includes software developed by Hidetoshi Shimokawa.
This product includes software developed by Hubert Feyrer for the NetBSD Project.
This product includes software developed by Ian W. Dall.
This product includes software developed by Intel Corporation and its contributors.
This product includes software developed by Internet Initiative Japan Inc.
This product includes software developed by Internet Research Institute, Inc.
This product includes software developed by James R. Maynard III.
This product includes software developed by Jared D. McNeill.
This product includes software developed by Jason L. Wright
This product includes software developed by Jason R. Thorpe for And Communications, http://www.and.com/
This product includes software developed by Joachim Koenig-Baltes.

This product includes software developed by Jochen Pohl for The NetBSD Project.
This product includes software developed by Joerg Wunsch
This product includes software developed by John Birrell.
This product includes software developed by John P. Wittkoski.
This product includes software developed by John Polstra.
This product includes software developed by Jonathan R. Stone for the NetBSD Project.
This product includes software developed by Jonathan Stone.
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 Julian Highfield.
This product includes software developed by K. Kobayashi
This product includes software developed by K. Kobayashi and H. Shimokawa
This product includes software developed by Kazuhisa Shimizu.
This product includes software developed by Kazuki Sakamoto.
This product includes software developed by Kenneth Stailey.
This product includes software developed by Kiyoshi Ikehara.
This product includes software developed by Klaus Burkert,by Bernd Ernesti, by Michael van Elst, and by the University of California, Berkeley and its contributors.
This product includes software developed by Kyma Systems.
This product includes software developed by Leo Weppelman and Waldi Ravens.
This product includes software developed by Lloyd Parkes.
This product includes software developed by Lutz Vieweg.
This product includes software developed by Marc Horowitz.
This product includes software developed by Marcus Comstedt.
This product includes software developed by Mark Brinicombe.
This product includes software developed by Mark Brinicombe for the NetBSD Project.
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 Marshall M. Midden.
This product includes software developed by Masanobu Saitoh.
This product includes software developed by Masaru Oki.
This product includes software developed by Matt DeBergalis
This product includes software developed by Matthew Fredette.
This product includes software developed by Michael Smith.
This product includes software developed by Microsoft
This product includes software developed by Mika Kortelainen
This product includes software developed by Mike Pritchard.
This product includes software developed by Mike Pritchard and contributors.
This product includes software developed by Minoura Makoto.
This product includes software developed by MINOURA Makoto, Takuya Harakawa.
This product includes software developed by Niels Provos.
This product includes software developed by Niklas Hallqvist.
This product includes software developed by Niklas Hallqvist, Brandon Creighton and Job de Haas.
This product includes software developed by Paolo Abeni.
This product includes software developed by Paul Kranenburg.
This product includes software developed by Paul Mackerras.
This product includes software developed by Paul Mackerras <paulus@samba.org>.
This product includes software developed by Pedro Roque Marques <pedro_m@yahoo.com>
This product includes software developed by Per Fogelstrom.
This product includes software developed by Peter Galbavy.
This product includes software developed by Phase One, Inc.
This product includes software developed by Philip A. Nelson.
This product includes software developed by QUALCOMM Incorporated.
This product includes software developed by Ravikanth.
This product includes software developed by RiscBSD.
This product includes software developed by Roar Thronæs.
This product includes software developed by Rodney W. Grimes.
This product includes software developed by Roger Hardiman
This product includes software developed by Rolf Grossmann.
This product includes software developed by Ross Harvey.
This product includes software developed by Ross Harvey for the NetBSD Project.
This product includes software developed by Scott Bartram.
This product includes software developed by Scott Stevens.
This product includes software developed by Shingo WATANABE.
This product includes software developed by Softweyr LLC, the University of California, Berkeley, and its contributors.
This product includes software developed by Stephan Thesing.

The End