NAME

INSTALL — Installation procedure for NetBSD/pmax.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 11.1 on the *pmax* 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 11.1 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 10 and 11 Releases

The NetBSD 11.1 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 11.1 release. The complete list of changes can be found in
CHANGES:

https://cdn.NetBSD.org/pub/NetBSD/NetBSD-11.1/CHANGES

CHANGES-11.1:

https://cdn.NetBSD.org/pub/NetBSD/NetBSD-11.1/CHANGES-11.1

which are also present in the top level directory of the NetBSD 11.1 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 11.1 Release Contents

The root directory of the NetBSD 11.1 release is organized as follows:

`.../NetBSD-11.1/`

`CHANGES` Changes between the 10.0 and 11.0 releases.

`CHANGES-11.0` Changes between the initial 11.0 branch and final release of 11.0.

`CHANGES-11.1` Changes between the 11.0 and the 11.1 release.

`CHANGES.prev` Changes in previous NetBSD releases.

`LAST_MINUTE` Last minute changes and notes about the release.

`README.files` README describing the distribution's contents.

`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 11.1 has a binary distribution.

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

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

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

src This set contains all of the base NetBSD 11.1 sources which are not in **gnusrc**, **sharesrc**, or **syssrc**.

syssrc This set contains the sources to the NetBSD 11.1 kernel for all architectures as well as the `config`(1) utility.

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

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

The source sets are distributed as compressed tar files. Individual 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/pmax subdirectory structure

The pmax-specific portion of the NetBSD 11.1 release is found in the `pmax` subdirectory of the distribution: `.../NetBSD-11.1/pmax/`. It contains the following files and directories:

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

`binary/`
- `kernel/`
  - `netbsd-GENERIC.gz` A gzipped NetBSD kernel containing code for everything supported in this release.
- `sets/` pmax binary distribution sets; see below.

`installation/`
- `misc/` Miscellaneous pmax installation utilities; see installation section below.

Binary distribution sets

The NetBSD pmax binary distribution sets contain the binaries which comprise the NetBSD 11.1 release for pmax. The binary distribution sets can be found in the `pmax/binary/sets` subdirectory of the NetBSD 11.1 distribution tree, and are as follows:

base The NetBSD 11.1 pmax **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-GENERIC
This set contains a NetBSD/pmax 11.1 `GENERIC` kernel, named `/netbsd`. You *must* install this distribution set.

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

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

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

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 XFree86 4.5.0. 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. This includes the Xpmax server with man page.

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

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

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

NetBSD/pmax System Requirements and Supported Devices

Supported machines

- DECstation 2100 and 3100, also known as *pmin* and *pmax*
- DECsystem 5100
- Personal DECstations (5000/20, /25 and /33) also known as *MAXINE*
- DECstation 5000/120, /125, and /133, also known as *3MIN*
- DECstation 5000/200, also known as *3MAX*
- DECstation 5000/240 and DECsystem 5900, also known as *3MAXPLUS*
- DECstation 5000/50, 5000/150, 5000/260 (and DECsystem 5900-260) – R4000 and R4400 variants of the *MAXINE, 3MIN* and *3MAXPLUS*

Unsupported machines

- DECsystem 5400 and 5500 (Q-bus based systems, similar to the Vax *Mayfair* or *Mayfair II*, but with an r2000a or r3000 CPU instead of a CVAX CPU.)
- DECsystem 5800 (xbi-based multiprocessor, a Vax 6200 with Vax CPU boards replaced with Mips CPU boards)

The minimal configuration requires 8 MB of RAM and 170 MB of disk space. To install the entire system requires much more disk space, and to run X or compile the system, more RAM is recommended, as NetBSD with 8 MB of RAM feels like Ultrix with 8 MB of RAM - slow. Note that until you have around 16 MB of RAM, getting more RAM is more important than getting a faster CPU.

Supported devices

- DECstation 2100 and 3100 baseboard video
  - one-bit mono or 8-bit pseudocolor frame buffers.
- DECstation 5000 series TURBOchannel video
  - PMAG-DV Personal DECstation baseboard 1024x768 frame buffer
  - PMAG-B (CX) 1024x768 8-bit color frame buffer
  - PMAGB-B (HX or SFB) 1280x1024 8-bit color frame buffer
  - PMAG-A (MX) 1280x1024 four-bit greyscale frame buffer
  - PMAG-C (PX) 2D 8-bit accelerated frame buffer.
  - PMAG-D (PXG) 3D 8-bit accelerated frame buffer.
  - PMAG-E (PXG+) 3D 24-bit accelerated frame buffer.
  - PMAG-F (PXG+ Turbo) 3D 24-bit accelerated frame buffer.
- PMAG-JA (TX) 1024x1280 color frame buffer
- PMAG-RO (TX) 1024x1280 color frame buffer

  Note: All supported DECstation video produces sync-on-green. Be sure to use either a DEC-compatible fixed-sync monitor or a multisync monitor that supports sync-on-green.
- Serial ports
  - ttya and ttyb (can be used as console if needed)
- Ethernet
  - on-board AMD Lance ethernet (*le0*) ,
  - TURBOchannel AMD Lance ethernet cards (the *PMAD*)

- SCSI
  - on-board DEC *sii* SCSI controller (2100 and 3100)
  - on-board *asc* SCSI controller (5000 series machines)
  - TURBOchannel *asc* SCSI controller (the *PMAZ*)
- DEC LK-201 (or compatible) keyboard
- DEC *hockey puck* or compatible mouse

Unsupported devices

- Q-bus based DECsystems
- PrestoServe NVRAM on DECsystem 5100 and 5000/2xx machines
- Audio drivers for Personal DECstation machines
- Floppy driver for Personal DECstation machines
- TURBOchannel audio hardware (LoFi)

Note: The primary obstacle to supporting the above is non-availability of sample hardware for development.

Getting the NetBSD System on to Useful Media

First-time installation on a bare machine is not supported, because most DECstations do not have any suitable load device. Some versions of the DECstation PROM are buggy and will not boot via TFTP and bootp; other versions are buggy and do not boot via MOP.

The only DECstation with a floppy-disk drive is the Personal DECstation, and that device is not supported as a boot device.

The recommended installation procedure is to boot an install kernel via TFTP, or to use a "helper" system to write a miniroot diskimage onto a disk, move that disk to the target installation system, and then boot the miniroot.

Installation is supported from several media types, including:

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

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

CD-ROM / DVD / USB stick Find out where the distribution set files are on the CD-ROM, DVD or USB stick. Likely locations are `binary/sets` and `pmax/binary/sets`. (You only need to know this if you are mixing installer and installation media from different versions - the installer will know the proper default location for the sets it comes with).

Proceed to the instructions on installation.

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

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

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

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

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

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

Note: the tape devices with which NetBSD/pmax is believed to work is the DEC TK-50. This is a very slow device. Installation via disk or network is recommended if possible.

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_sets
```

where *tape_device* is the name of the tape device that represents the tape drive you're using. This might be `/dev/rst0`, or something similar, but it will vary from system to system. In the above example, *dist_sets* is a list of filenames corresponding to the distribution sets that you wish to place on the tape. For instance, to put the **kern-GENERIC, base,** and **etc** distributions on tape (the absolute minimum required for installation), you would do the following:

```
# cd .../NetBSD-11.1
# cd pmax/binary
# tar -cf tape_device kern-GENERIC.tgz
base.tgz etc.tgz
```

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

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

Preparing your System for NetBSD installation

First and foremost, before beginning the installation process, *make sure you have a reliable backup* of any data on your hard disk that you wish to keep. Mistakes in partitioning your hard disk may lead to data loss.

Installing the NetBSD System

Note: There may be updates to this procedure available from the NetBSD/pmax web page at http://www.NetBSD.org/ports/pmax/ .

To install or upgrade NetBSD, you need to first boot an installation program and then interact with the screen-menu program **sysinst**. The installation program actually consists of the NetBSD kernel plus an in-memory file system of utility programs.

From most convenient to least convenient, the installation methods are:

1. Installation from an existing NetBSD or Ultrix system by booting the install kernel from an existing root file system (/). See the **Installing from an existing system** section later in these INSTALL notes.
2. Copying a bootable diskimage onto the beginning of a disk and installing onto that disk. See the **Install via diskimage** section later in these INSTALL notes.
3. Booting the install kernel over the network and installing onto a local disk. See the **Install via netboot install kernel** section later in these INSTALL notes.
4. Using a helper machine with a SCSI controller to copy the bootable diskimage onto the beginning of a disk, and moving the disk to the target machine. See the **Install via diskimage** section later in these INSTALL notes.
5. For machines with some PROMs that cannot netboot standard kernels, you will need to set up an NFS server with a diskless root file system for trimmed-down network install kernel and run the installation system from the NFS root file system. See the **Install via diskless boot** section later in these INSTALL notes.

You should familiarize yourself with the console PROM environment and the hardware configuration. The PROMs on the older DECstation 2100 and 3100 use one syntax. The PROMs on the TURBOchannel machines use a completely different syntax. Be sure you know how to print the configuration of your machine, and how to boot from disk or network, as appropriate.

To boot from disk, use:

2100 or 3100:	**boot -f rz(0,*X*,0)netbsd**
5000/200:	**boot 5/rz*X*/netbsd**
other 5000s:	**boot 3/rz*X*/netbsd**

To boot via TFTP, use:

2100 or 3100:	**boot -f tftp()**
5000/200:	**boot 6/tftp**
other 5000s:	**boot 3/tftp**

To boot via MOP, use:

2100 or 3100:	**boot -f mop()**
5000/200:	**boot 6/mop**
other 5000s:	**boot 3/mop**

You will also need to know the total size (in sectors) and the approximate geometry of the disks you are installing onto, so that you can label your disks for the BSD fast file system (FFS). For most SCSI drives (including all SCSI-2 drives), the kernel will correctly detect the disk geometry. The **sysinst** tool will suggest these as the default.

If you're installing NetBSD/pmax for the first time it's a very good idea to pre-plan partition sizes for the disks on which you're installing NetBSD. Changing the size of partitions after you've installed is difficult. If you do not have a spare bootable disk, it may be simpler to re-install NetBSD again from scratch.

Assuming a classic partition scheme with separate `/` (root) and `/usr` file systems, a comfortable size for the NetBSD `/` partition is about 100 MB. A good initial size for the swap partition is twice the amount of physical memory in your machine (though, unlike Ultrix, there are no restrictions on the size of the swap partition that would render part of your memory unusable). The default swap size is 64 MB, which is adequate for doing a full system build.

Install via a NetBSD CD-ROM

You can obtain the disk image or diskless boot tar file from the NetBSD 11.1 CD-ROM.

To mount the CD-ROM from a NetBSD/pmax host, type

mount -r -t cd9660 /dev/cd*X*c /mnt

and from an Ultrix host

mount -r -t cdfs -o nodefperm,noversion /dev/rz*X*c /mnt

where *X* is the SCSI-ID of the CD-ROM.

Note: Ultrix does not have Rock Ridge extensions so leave out everything between the first and last period ('.') in the paths on the CD. For example, the path `NetBSD-1.4.3` would show up as `NetBSD-1.3`.

The diskimage file can be found on the CD-ROM at the following location (relative to the mount point of the CD)

`pmax/installation/diskimage/diskimage.gz`

and the diskless boot tar file can be found at

`pmax/installation/netboot/diskimage.tgz`

Once you have located these files, continue on to either **Install via netboot install kernel**, **Install via diskless boot**, **Install via diskimage** or **Installing from an existing system** section later in the INSTALL notes.

Installing from an existing system

With the NetBSD bootblocks or using the Ultrix bootloader, you can boot the kernel located in:

`pmax/binary/kernel/netbsd-INSTALL.gz`

On Ultrix systems you will need to **gunzip** this kernel before booting it.

Then boot using one of:

2100 or 3100:	**boot -f rz(0,*X*,0)netbsd-INSTALL**
5000/200:	**boot 5/rz*X*/netbsd-INSTALL**
other 5000s:	**boot 3/rz*X*/netbsd-INSTALL**

where *X* is your boot disk SCSI-ID, and continue to the **Once you've booted the diskimage** section

Install via diskimage

The diskimage file is in

`pmax/installation/diskimage/diskimage.gz`.

It is shipped compressed and is around 1760 kbytes, uncompressing to exactly 2100 kbytes.

To install the diskimage onto disk sd*X* on a NetBSD/pmax system, do:

```
# disklabel -W /dev/rsdXc
# gunzip -c diskimage.gz | dd of=/dev/rsdXc bs=10240
```

When installing on a disk with no NetBSD or Ultrix label, you may get a message like:

```
sdX: WARNING: trying Ultrix label, no disk label
```

or when installing on an old Ultrix disk, you may get a message like:

```
sdX: WARNING: using ULTRIX partition information
```

when issuing the **disklabel -W /dev/rsd*X*c** command. This can safely be ignored.

On NetBSD/i386, the "raw disk partition" is the 'd' partition, so do:

```
# disklabel -W /dev/rsdXd
# gunzip -c diskimage.gz | dd of=/dev/rsdXd bs=10240
```

On NetBSD, be sure to use **disklabel -W** to enable writing to the label area of the disk. If you forget this and/or use the 'block' device, the **dd** command may silently fail.

On Ultrix systems, do:

```
# gunzip -c diskimage.gz | dd of=/dev/rrzXc bs=10240
```

On MS-DOS, use an 'unzip' utility, then use **rawrite**.

Then boot using one of:

2100 or 3100:	**boot -f rz(0,*X*,0)netbsd**
5000/200:	**boot 5/rz*X*/netbsd**
other 5000s:	**boot 3/rz*X*/netbsd**

where *X* is your boot disk SCSI-ID, and continue to the **Once you've booted the diskimage** section.

Install via netboot install kernel

Booting NetBSD/pmax 11.1 install kernel over a network requires a BOOTP or DHCP server and a TFTP server. (These are usually all run on the same machine.) There are two basic stages to the boot:

- The pmax PROM software sends a BOOTP request to get its own address, the address of the TFTP server and the file to download.
- It downloads the file name obtained from BOOTP, which is the install kernel, via TFTP and then executes it.

You will need to set up servers for BOOTP and TFTP.

For the BOOTP server you need to specify the:

- hardware type (Ethernet)
- hardware address (Ethernet MAC address)
- IP address of the client
- subnet mask of the client
- address of the TFTP server
- name of the kernel loaded via TFTP

Here's an example for a UNIX-like system running **bootpd**:

```
myhost.mydom.com:\
         :ht=ethernet:ha=08002b123456:\
         :ip=192.168.1.2:sm=255.255.255.0:\
         :sa=192.168.1.1:bf=install:\
         :rp=/usr/export/pmax:
```

And here's an example for a UNIX-like system running the ISC **dhcpd**:

```
host pmax {
        hardware ethernet 08:0:2b:12:34:56;
        fixed-address 192.168.1.2;
        option host-name "myhost.mydom.com";
        filename "install";
        option domain-name-servers 192.168.1.1;
        option broadcast-address 255.255.255.0;
        option domain-name "my.domain";
        option root-path "/usr/export/pmax";
}
```

For the TFTP server, You will need to copy the **install.ecoff** kernel to the directory used by the TFTP server. This file must be gunzipped.

Then boot using one of:

2100 or 3100:	**boot -f tftp()**
5000/200:	**boot 6/tftp**
other 5000s:	**boot 3/tftp**

and continue to the **Once you've booted the diskimage** section.

Install via diskless boot

The file pmax/installation/netboot/diskimage.tar.gz contains a suitable set of files for installing on an NFS server to set up a diskless root filesystem. (It is a tar copy of the contents of an installation ramdisk file system contained in the install kernel.) You will need to find an NFS server, unpack the tarfile, and setup BOOTP/dhcp service for your pmax. Instructions for setting up an NFS server and diskless booting are on the NetBSD/pmax netboot webpage at

http://www.NetBSD.org/ports/pmax/netboot.html

Since the system install utility, **sysinst**, requires a read/write root, installing via diskless boot is only feasible if your NFS server exports the diskless root read-write. If this is not possible, you should install via other of the other installation procedures.

Once you have booted the kernel, continue to the **Once you've booted the diskimage** section.

Once you've booted the diskimage

Once you've booted the installation kernel you will need to select your terminal type. Use **vt100** for a framebuffer console or a serial console with a vt100-compatible terminal, or **xterm** or **xterms** for a `tip`(1) or `cu`(1) connection running in an `xterm`(1).

The system will then start the **sysinst** program.

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 minimum of 8 MB of memory installed.
     - An optical drive.
     - A hard drive with at least 600 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-11.1 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.
- 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.**
- 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*

   You may want to read the boot messages, to notice your disk's name and capacity. Its name will be something like `sd0` and the geometry will be printed on a line that begins with its name. As mentioned above, you may need your disk's geometry when creating NetBSD's partitions. You will also need to know the name, to tell **`sysinst`** which disk to use. The most important thing to know is that `sd0` is your first SCSI disk, `sd1` the second, etc.

   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 `rz0` or `rz1`

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*

   - Choosing which portion of the disk to use.

     You will be asked if you want to use the entire disk or only part of the disk. If you decide to use the entire disk for NetBSD, **`sysinst`** will check for the presence of other operating systems and you will be asked to confirm that you want to overwrite these.

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**.

   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-h` 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. Nothing has been written to your disk yet, but if you confirm that you want to install NetBSD, your hard drive will be modified. 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 `rz1e`), 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 11.1. 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 the correct terminal type as discussed in the **Once you've booted the diskimage** section. 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. 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.

4. 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.

5. 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**:

6. 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 pmax/11.1/All subdir. If you installed pkgin(1) in the **sysinst** post-installation configuration menu, you can use it to automatically install binary packages over the network. Assuming that /usr/pkg/etc/pkgin/repositories.conf is correctly configured, you can install them with the following commands:

     ```
     # pkgin install tcsh 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.

7. 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 11.1 is with binaries, and that is the method documented here.

To do the upgrade, you must install new bootblocks and boot off the **install** kernel as described in the **Installing from an existing system** 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 11.1 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 11.1.

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.

A number of things have been removed from the NetBSD 11.1 release. See the "Components removed from NetBSD" section near the beginning of this document for a list.

Using online NetBSD documentation

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

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

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

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

```
# man passwd
```

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

```
# man 5 passwd
```

instead.

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

```
# apropos subject-word
```

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

Administrivia

If you've got something to say, do so! We'd like your input. There are various mailing lists available via the mailing list server at `majordomo@NetBSD.org`. 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.
This product includes software developed by Steven M. Bellovin
This product includes software developed by Takashi Hamada.
This product includes software developed by Takumi Nakamura.
This product includes software developed by Tatoku Ogaito for the NetBSD Project.

Some files have the following copyright:

Some files have the following copyright:

Some files have the following copyright:

Some files have the following copyright:

The End