NAME

INSTALL — Installation procedure for NetBSD/sparc64.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 10.2_STABLE on the *sparc64* platform. It is available in four different formats titled `INSTALL.ext`, where `.ext` is one of `.ps`, `.html`, `.more`, or `.txt`:

`.ps` PostScript.

`.html` Standard Internet HTML.

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

`.txt` Plain old ASCII.

You are reading the *PostScript* version.

Quick install notes for the impatient

This section contains some brief notes describing what you need to install NetBSD 10.2_STABLE on a machine of the sparc64 architecture.

- Fetch files needed to install NetBSD. This is either a CD image (`NetBSD-10.2_STABLE-sparc64.iso`) which can be booted directly, or the installation kernel and bootloader (`sparc64/binary/kernel/netbsd-INSTALL.gz` and `sparc64/installation/misc/ofwboot`) which can be booted from a Solaris or NetBSD partition. Alternatively, you may netboot the installation kernel. This process is covered below, in detail.
- The actual binary distribution is in the `sparc64/binary/sets/` directory. When you boot the install CD-ROM or installation kernel, the installation program can fetch these files for you (using, e.g., ftp) if you have a network connection. There are several other methods to get the binary sets onto your machine.

  You will at a minimum need the following sets: `kern-GENERIC.tar.xz`, `base.tar.xz` and `etc.tar.xz`. In a typical workstation installation you will probably want all the installation sets.
- Burn the CD or put the installation kernel and bootloader at the root level of a bootable Solaris or NetBSD partition.

  The media you just prepared will be used to boot the installation kernel, which contains all the tools required to install NetBSD.

- You will need to get to the OpenFirmware "**ok**" prompt. After your system first powers on and displays some initial information, press the STOP-A keys, or send a BREAK if you're on a serial console. At the "**ok**" prompt, type the command to boot your system into NetBSD. The command to boot from CD is: "**boot cdrom**". The command to boot the NetBSD kernel from a Solaris or NetBSD partition depends on which disk and partition it is on. To boot from the first partition of the first (primary) disk: "**boot disk:a /ofwboot -a**". When it asks you for a kernel, specify "**netbsd-INSTALL.gz**"
- For third-party programs which are not part of the base NetBSD distribution, you will want to explore the **pkgsrc** package management system, which contains thousands of third-party software applications.

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.tar.xz
```

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/sparc64 subdirectory structure

The sparc64-specific portion of the NetBSD 10.2_STABLE release is found in the `sparc64` subdirectory of the distribution: `.../NetBSD-10.2_STABLE/sparc64/`. 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 every system supported in this release. This kernel supports SMP on systems with more than one CPU.
  - `netbsd-GENERIC.UP.gz`
    A gzipped NetBSD kernel containing code for every system supported in this release. This kernel disables SMP support.
  - `netbsd-INSTALL.gz`
    The install kernel.
- `sets/` sparc64 binary distribution sets; see below.

`installation/`
- `miniroot/` sparc64 miniroot file system image; see below.
- `misc/` Miscellaneous sparc64 installation utilities; see installation section below.
- `netboot/` Netboot bootloader; see below.

Binary distribution sets

The NetBSD sparc64 binary distribution sets contain the binaries which comprise the NetBSD 10.2_STABLE release for sparc64. The binary distribution sets can be found in the `sparc64/binary/sets` subdirectory of the NetBSD 10.2_STABLE distribution tree.

Note: If you want to install 32-bit only binaries, you will still need to boot using the sparc64 installation tools. When sysinst asks you for the distribution set to install, provide the NetBSD/sparc `binary/sets`. Be sure to use the 32 bit sparc `kern-GENERIC_SUN4U.tgz` kernel distribution. Otherwise, continue to follow the sparc64 installation procedure, not the sparc instructions.

Most people will want the 64-bit sparc64 distribution tree:

base The NetBSD 10.2_STABLE sparc64 **base** binary distribution. You *must* install this distribution set. It contains the base NetBSD utilities that are necessary for the system to run and be minimally functional.

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/sparc64 10.2_STABLE `GENERIC` kernel named `/netbsd`. You *must* install this or the kern-GENERIC.UP distribution set.

kern-GENERIC.UP
This set contains a NetBSD/sparc64 10.2_STABLE `GENERIC.UP` kernel named `/netbsd`, with SMP support disabled.

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. This includes the modular Xorg and Xvfb servers.

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

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 sparc64 binary distribution also has its own checksum files, just as the source distribution does.

NetBSD/sparc64 System Requirements and Supported Devices

Supported machines

The minimal configuration requires 32 MB of RAM and ~180 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. Note that until you have at least 64 MB of RAM, getting more RAM is more important than getting a faster CPU.

Not all of the machines listed here have been tested. Often Sun used the same motherboard design in multiple models. For example, there might be a Blade ("workstation") model, Enterprise or Fire ("server") model, and Netra or ft ("telco") model with essentially the same devices, as far as NetBSD is concerned. If one model in this scheme works, it is highly likely another will work.

Sun has also released a few models with names that might imply the systems are UltraSPARC-based but actually have an i386/amd64 class CPU. These systems might be supported by the i386 or amd64 ports of NetBSD.

- Sbus-based UltraSPARC I or II systems
  - Ultra 1 family (Ultra 1, Ultra 1 Creator, Ultra 1 Creator 3D, Netra i 150, Netra nfs 150, Enterprise 1, Enterprise 150)
  - Ultra 2 family (Ultra 2, Ultra 2 Creator, Ultra 2 Creator 3D, Enterprise 2)
  - Enterprise (3000, 3500, 4000, 4500, 500x, 5500, 600x, 6500, 10000)
- PCI-based UltraSPARC II, IIi, IIe, III, or IIIi systems
  - Ultra 5/10 family (Ultra 5, Ultra 10, Enterprise 5, Enterprise 10)
  - Ultra 30 family (Ultra 30, Netra T 1100)
  - Ultra 60 family (Ultra 60, E220R, Netra T 1120 and T 1125)
  - Ultra 80 family (Ultra 80, E420R, Netra T 1400 and T 1405)
  - Ultra 250 family (Enterprise 250 and E250R)
  - Ultra 450 family (Ultra 450, Enterprise 450, Netra ft 1800)
  - SunBlade 100 and 150
  - SPARCengine CP1400 or SPARCengine CP1500 system boards (Netra T1 (models 100 and 105), CT400, CT800, CT1600)

- SPARCengine Ultra AXi family (Netra X1, Fire v100, Fire v120, Netra 120, Netra AC 200, Netra DC 200)
- SPARCengine Ultra AXe system board
- SunBlade 1000 and 2000
- SunBlade 1500, 2500 (Red and Silver)
- Ultra 25 and 45
- Sun Fire v125, v210, v215, v240, v245, v250, v440, v445, v480

Unsupported machines

- Systems with an UltraSPARC IV CPU
- Systems with an UltraSPARC T1-T5 CPU
- Systems with a Fujitsu SPARC64 CPU

Supported devices

- Ethernet devices
  - Sbus LANCE ethernet (*le*)
  - Sbus BigMac ethernet (*be*)
  - On-board, PCI, and Sbus HME ethernet (*hme*)
  - PCI ERI (*gem*)
  - Many other PCI and PCMCIA Ethernet interfaces, including:
    - Tulip-compatible (*tlp*)
    - Intel (*fxp*)
    - Intel gigabit (*wm*)
    - NE2000-compatible (*ne*)
    - Realtek (*rtk* or *re*)
    - SysKonnect XMAC II (*sk*)
    - Marvell GMAC (*msk*)
  - Several USB Ethernet interfaces, such as (*aue*, *cue*, and *kue*)
- Wireless network adapters
  - Atheros wireless cards (*ath*)
- Video devices
  - All on-board frame buffers
    Unless otherwise specified, only text console is supported
  - On-board and SBUS card (GX, GX+, TGX, TGX+) (*cgsix*)
    8 bpp accelerated framebuffer. Accelerated X via the integrated Xorg.
  - UPA devices (ffb and afb, also called Creator3D) 24 bpp accelerated framebuffer. Accelerated X via the integrated Xorg.
  - On-board ATI mach64 framebuffers (found in Ultra 5/10).
- SCSI host controllers
  - ncr53c9x based controllers (*esp*)
  - ncr53c8xx based controllers (*siop* or *esiop*)
  - ISP10x0 based controllers (*isp*)
  - Adaptec AHA-2x4x[U][2][W] cards and onboard PCI designs using the AIC-7770, AIC-7850, AIC-7860, AIC-7870, AIC-7880 and AIC-789x chipsets.

    Note: Some Adaptec cards that support booting on MacOS may be bootable on sparc64 systems.

- Many other PCI SCSI controllers should work, but no one has tried them
- Most SCSI disk/tape/CD-ROM devices should work

- PCI IDE host controllers
  - Generic IDE controllers (*pciide*)
  - Acard ATP-850/860 based IDE controllers (*pciide*)
    (including the Acard AEC-6210/6260 and the Sonnet Tempo ATA/66 cards)
  - CMD Tech PCI064[3689] and Silicon Image 0680 IDE Controllers (*cmdide*)
  - Silicon Image 3112, 3114, and 3512 SATA Controller (*satalink*)

    Note: Some IDE cards that support booting on MacOS may be bootable on sparc64 systems.
  - Many other PCI IDE controllers should work, but no one has tried them
  - Most IDE and ATAPI disk/tape/CD-ROM devices should work

- Floppy drives
  - On-board Floppy Disk Controller -- SBus machines only for now (*fdc*)

- Serial devices
  - On-board Sbus z8530 serial (*zs*)
  - On-board PCI 16550-based su and su-pnp serial (*com*)
  - On-board PCI SAB82532 dual UART serial (*sab*)

- Parallel devices
  - On-board and Sbus parallel ports (*bpp*)
  - On-board PCI/Ebus parallel ports (*lpt*)

- Audio devices
  - On-board, PCI, and SBus based CS4231-based 44 KHz, 16 bit, stereo, PCM audio (*audiocs*)

- Other PCI devices
  - PCI "universal" communications cards (*puc*)

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

Unsupported devices

Smart card readers.

Getting the NetBSD System on to Useful Media

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.

Note: If you are installing the 32-bit sparc distribution sets, you will need to rename `sparc/binary/sets/kern-GENERIC_SUN4U.tar.xz` to `kern-GENERIC.tar.xz` since the sparc64 installation tools would otherwise attempt to install the kernel for 32-bit sparc computers which does not boot on sparc64 systems.

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 `sparc64/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.

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-10.2_STABLE
# cd sparc64/binary
# tar -cf tape_device kern-GENERIC.tar.xz
base.tar.xz etc.tar.xz
```

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

Deciding on partition sizes

If you're installing NetBSD/sparc64 for the first time it's a good idea to look at the partition sizes of the disk you plan to install NetBSD on. Will you be installing NetBSD onto the same disk as Solaris, on its own disk, or will you be netbooting? While NetBSD can work just fine on a disk shared with Solaris, the NetBSD installer does not currently support this. The limitation is that the NetBSD `disklabel`(8) writes partition info that Solaris is not familiar with. Therefore, if you are sharing a disk with Solaris, any time you change a partition table, you must do it from Solaris.

Assuming a classic partition scheme with `/` (root) and `/usr` file systems, a comfortable size for the NetBSD `/` partition is about 200 MB. A full binary installation including X takes over 600 MB in `/usr`. Since the pkgsrc binaries are typically installed in `/usr/pkg` you may want a significantly larger `/usr` partition. A good initial size for the swap partition is the amount of physical memory in your machine if you've got more than 128 MB RAM. If you've got less RAM, you may want swap to be at least 128 MB.

Note that the OBP on Ultra 1 and Ultra 2 machines can only boot from the first 4Gb of the disk, so this limits the size of the root partition on these models.

Setting up Open Firmware

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

If the ethernet address of your sparc64 system is `ff:ff:ff:ff:ff:ff` (check with the Open Firmware "banner" command), then your NVRAM battery is dead and you will have trouble using ethernet (among other problems). Read the *Sun NVRAM/Hostid FAQ:*

https://web.archive.org/web/20150611091616/https://www.squirrel.com/squirrel/sun-nvram-hostid.faq.html.

If you have a valid ethernet address and you plan to netboot, write down your system's ethernet address.

You cannot use the security modes of the SPARC OpenFirmware.

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

If you are using a serial console, the NetBSD/sparc64 installer defaults to using 9600 bps, 8N1 settings. You may want to configure your system and serial terminal like this prior to booting the installer. A new installation of NetBSD/sparc64 will default to these settings as well.

SCSI devices are specified by an Open Firmware **`devalias`** which provides simple mnemonics for the full path to the device. Type **`devalias`** to get a list of all of the available aliases. At a minimum, the alias and partition are necessary when booting.

Therefore, to boot from the swap partition on the internal hard drive one would use:

ok **boot disk:b**

To boot from a CD-ROM (Open Firmware assumes SCSI CD-ROMs are at target 6), one would use:

ok **boot cdrom**

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

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

If you get ".... Fast Data MMU Miss" when booting after the NetBSD installation, your OpenBoot PROM may need updating. It has been reported that version 3.31 lead to a successful boot on an Ultra Enterprise 420R, while version 3.23 did not. Exact values may vary, depending on your hardware, current OpenBoot PROM version and moon phase.

Determining how to boot from an SBUS or PCI card

Some SBUS and PCI cards have firmware that lets you use them as boot devices. These cards do not automatically create **devalias** entries, so you must traverse the device tree to figure out what Open Firmware calls your card. You will be using Open Firmware commands at the '`ok`' prompt. First 'cd' to the top of the device tree and list the nodes there. The following is the procedure to boot from an IDE card in an UltraSPARC 30.

```
ok cd /
ok ls
f006cf08 SUNW,ffb@1e,0
f006c32c SUNW,UltraSPARC-II@0,0
f006002c counter-timer@1f,1c00
f005f410 pci@1f,2000
f005eb54 pci@1f,4000
f004cf84 virtual-memory
f004c9a4 memory@0,0
f002ce38 aliases
f002cdc8 options
f002cc90 openprom
f002cc24 chosen
f002cbb4 packages
```

Usually, you can simply type in the name before the at (@) sign and OpenFirmware will fill in the rest.

```
ok cd pci@1f,4000
ok ls
f0081524 ide@2
f007be50 scsi@3
f0074688 network@1,1
f0060324 ebus@1
ok cd ide@2
ok ls
0081fe4 cdrom
f0081938 disk
ok cd disk
ok ls
ok pwd
/pci@1f,4000/ide@2/disk
```

OK, now we know the path to the IDE device in this example. Now, we need to determine if it's capable of booting. If it is, it will have the **word** 'open'.

```
ok words
close         load          write         read          seek
open          write-blocks  read-blocks   max-transfer
block-size    dma-free      dma-alloc     spin-down     spin-up
```

Great! Also, in case you're interested in further details about your hardware, you can use the **.properties** command.

So, when it's time to type in a boot command, use the shortened version of the **pwd** command. You need to be more specific if there are two devices with the same name (in this case, two /pci entries). In this example, you'd type:

```
ok boot /pci@1f,4000/ide/disk@0,0
```

You can also store this device path across reboots using the **nvalias** command.

```
ok nvalias wd0 /pci@1f,4000/ide/disk@0,0:a
```

And when the kernel is done booting, it may not automatically use your card as the root device -- you may need to type in the NetBSD/sparc64 name for that device:

```
root on sd0a dumps on sd0b
no file system for sd0 (dev 0x700)
cannot mount root, error = 79
root device (default sd0a): ?
use one of: hme0 sd0[a-h] wd0[a-h] halt
root device (default sd0a): wd0a
dump device: wd0a
file system (default generic): ffs
root on wd0a
```

The root device can also be specified in your kernel config file.

Configuration of network interfaces

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

Installing the NetBSD System

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

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

If you have problems with these or you are installing NetBSD onto the same disk as Solaris, see the section below on **Manual Installation of NetBSD using Solaris**

Installing NetBSD from CD-ROM

Installing from CD-ROM is the least painful way to install NetBSD. Simply insert the CD-ROM in the drive, power up the computer, and type:

ok **boot cdrom**

This Open Firmware boot command will cause the NetBSD kernel contained in the CD-ROM to be booted. After the initial probe messages you'll be asked to start the install or upgrade procedure. Proceed to the section **Running the sysinst installation program** below.

Installing NetBSD by using the NetBSD miniroot

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

Loading the miniroot onto your raw partition is simple using the `dd`(1) command. Just remember to first uncompress the miniroot image and boot your OS with the `-s` flag so that it runs "single-user" and does not attempt to start swapping.

ok **boot -s**

On Solaris you use a command like:

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

On NetBSD the command is:

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

Replace `/dev/rdsk/c0t0d0s1` or `/dev/rsd0b` with your swap partition.

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

halt

Then boot the miniroot by typing the appropriate command at the OpenFirmware prompt:

ok **boot disk:b netbsd**

If you've loaded the miniroot onto some other disk than `sd0` use the correct `devalias`, such as

ok **boot disk1:b netbsd**

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

Installing NetBSD by using a NetBSD kernel on a Solaris partition

This procedure is very straightforward. You will be putting the NetBSD installation kernel (kernel with a RAM disk installer) on your Solaris partition and telling Open Firmware to boot the NetBSD kernel.

CAVEAT: this method is a bit fragile. Depending on physical partition layout, partition size, and parameters used to create the filesystem of your Solaris root partition, bugs in the bootloader might be triggered and cause this method to fail.

First, copy the `netbsd-INSTALL.gz` kernel and bootloader to the root level of your hard drive and halt your system

```
# cp binary/kernel/netbsd-INSTALL.gz /
# cp installation/misc/ofwboot /
# halt
```

At the Open Firmware prompt, boot NetBSD.

```
ok boot disk:a /ofwboot -a
```

The -a flag is needed so that the bootloader will ask you to find your installation kernel.

```
Rebooting with command: boot disk:a /ofwboot -a
Boot device: /pci@1f,4000/scsi@3/disk@0,0:a  File and args: /ofwboot -a
Enter filename [/ofwboot]: <return>
>> NetBSD/sparc64 OpenFirmware Boot, Revision 1.7
>> (autobuild@tgm.netbsd.org, Thu May 20 16:29:20 UTC 2004)
Boot: netbsd-INSTALL.gz
```

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

Installing NetBSD by using a netboot setup

1. Introduction

   To netboot a sparc64, you must configure one or more servers to provide information and files to your sparc64 (the 'client'). If you are using NetBSD (any architecture) on your netboot server(s), the information provided here should be sufficient to configure everything. Additionally, you may wish to look at the diskless(8) manual page and the manual pages for each daemon you'll be configuring. If the server(s) are another operating system, you should consult the NetBSD Diskless HOW-TO, which will walk you through the steps necessary to configure the netboot services on a variety of platforms:

   https://www.NetBSD.org/docs/network/netboot/

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

   Briefly, the netboot process involves discovery, bootstrap, kernel and file system stages. In the first stage, the client discovers information about where to find the bootstrap program. Next, it downloads and executes the bootstrap program. The bootstrap program goes through another discovery phase to determine where the kernel is located. The bootstrap program tries to mount the NFS share containing the kernel. Once the kernel is loaded, it starts executing. For RAM disk kernels, it mounts the RAM disk file system and begins executing the installer from the RAM disk. For normal (non-RAM disk) kernels, the kernel tries to mount the NFS share that had the kernel and starts executing the installation tools or init(8). All sparc64 systems use a combination of RARP and DHCP for the discovery stage. TFTP is used in the bootstrap phase to download the bootstrap program, ofwboot.net, which has been linked to a file name appropriate to the client's IP address as described in the TFTP section below. NFS is used in both the kernel and file system stages to download the kernel, and to access files on the file server.

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

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

2. dhcpd(8) in bootpd(8) compatible mode

   Put the following lines in your /etc/dhcpd.conf (see dhcpd.conf(5) and dhcp-options(5) for more information):

```
ddns-update-style none;
                # Do not use any dynamic DNS features
                #
allow bootp;    # Allow bootp requests, thus the dhcp server
                # will act as a bootp server.
                #
authoritative;  # master DHCP server for this subnet
                #
subnet 192.168.1.0 netmask 255.255.255.0 {
                # Which network interface to listen on.
                # The zeros indicate the range of addresses
                # that are allowed to connect.
}
group {
                # Set of parameters common to all clients
                # in this "group".
                #
        option broadcast-address        192.168.1.255;
        option domain-name              "test.net";
        option domain-name-servers      dns.test.net;
        option routers                  router.test.net;
        option subnet-mask              255.255.255.0;
                #
                # An individual client.
                #
        host client.test.net {
                hardware ethernet       CC:CC:CC:CC:CC:CC;
                fixed-address           192.168.1.10;
                #
                # Name of the host (if the fixed address
                # doesn't resolve to a simple name).
                #
                option host-name        "client";
                #
                # Name of the bootloader or kernel
                # to download via tftp.

                #
                # The path on the NFS server.
                #
                option root-path "/export/client/root";

                #
                # The host address of the NFS server. This is mandatory for
                # NetBSD kernels even it's the same host as the DHCP server.
                #
                next-server server.test.net;
        }
```

```
 #you may paste another "host" entry here for additional
 #clients on this network
}
```

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

touch /var/db/dhcpd.leases

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

/etc/rc.d/dhcpd restart

3. rarpd(8)

   Create an /etc/ethers file with the following line:

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

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

   ```
   192.168.1.10 client
   ```

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

   # **/etc/rc.d/rarpd restart**

4. tftpd(8)

   The default configuration of the TFTP server is to run in a chroot(8) environment in the /tftpboot directory. Thus, the first order of business is to create this directory:

   # **mkdir -p /tftpboot**

   Next, edit /etc/inetd.conf and uncomment the line with the TFTP daemon:

   ```
   tftp  dgram  udp  wait  root  /usr/libexec/tftpd tftpd -l -s /tftpboot
   ```

   Now, restart inetd(8). If the server is running NetBSD, you can achieve this with:

   # **/etc/rc.d/inetd restart**

   Now, you need to copy the bootloader for your sparc64 machine to /tftpboot. Get ofwboot.net from the installation/netboot directory of the distribution.

   # **cp ofwboot.net /tftpboot**

   Now, you need to link ofwboot.net to the filename that your sparc64 will look for. It will look for a filename composed of the machine's IP address (in hexadecimal). For example, a machine which has been assigned IP address 192.168.1.10, will make a TFTP request for C0A8010A.

   You can use bc(1) to help calculate the filename:

   ```
   # bc
   obase=16
   192
   C0
   168
   A8
   1
   1
   10
   ```

```
A
quit
# cd /tftpboot
# ln -s ofwboot.net C0A8010A
```

Just to be sure, let's make everything readable.

```
# chmod -R a+rX /tftpboot
```

Sometimes, the `arp`(8) table gets messed up, and the TFTP server can't communicate with the client. In this case, it will write a log message (via `syslogd`(8)) to `/var/log/messages` saying: '`tftpd: write: Host is down`'. If this is the case, you may need to force the server to map your client's ethernet address to its IP address:

```
# arp -s client CC:CC:CC:CC:CC:CC
```

5. `nfsd`(8), `mountd`(8), and `rpcbind`(8)

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

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

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

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

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

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

6. NetBSD kernel and installation tools

   Now, if you place a kernel named `netbsd` in `/export/client/root` your client should boot the kernel. Use `binary/kernel/netbsd-GENERIC.gz`.

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

   If you are netbooting the installer, copy the distribution files to the client's root directory and extract the tools from `installation/misc/instfs.tar.xz`.

   ```
   # cp *.tar.xz /export/client/root
   # cd /export/client/root
   # tar -xpzf instfs.tar.xz
   ```

7. Client file system

   You can skip this step if you do not plan to run your client diskless after installation. Otherwise, you need to extract and set up the client's installation of NetBSD. The Diskless HOW-TO describes how to provide better security and save space on the NFS server over the procedure listed here. See *for details:*
   https://www.NetBSD.org/docs/network/netboot/nfs.html.

   - Extracting distribution sets

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

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

- Set up swap

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

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

- Create device nodes

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

  This procedure only works on NetBSD hosts.

- Set up the client's `fstab`(5)

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

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

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

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

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

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

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

  Edit `/export/client/root/etc/hosts`

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

8. Setting up the server daemons

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

```
dhcpd=YES        dhcpd_flags="-q"
rarpd=YES        rarpd_flags="-a"
nfs_server=YES         # enable server daemons
mountd=YES
rpcbind=YES      rpcbind_flags="-l"   # -l logs libwrap
```

   Also, you'll need to make sure the `tftpd` line in `/etc/inetd.conf` remains uncommented.

Now, netboot your system from the server by entering the appropriate **boot** command at the Open Firmware prompt.

ok **boot net netbsd**

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

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 32 MB of memory installed.
     - An optical drive.
     - A hard drive with at least 700 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 are installing onto a disk which you want to use with Solaris, stop here. You will need to perform a manual installation as **sysinst** overwrites the Solaris partition table. See the section on **Manual Installation of NetBSD using Solaris**
- 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.**
- Once the system reaches the Open Firmware prompt, you will need to type the correct command to boot from your hard 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*

   You may want to read the boot messages, to notice your disk's name and capacity. Its name will be something like `sd0` or `wd0` 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 `wd0` is NetBSD's name for your first IDE disk, `wd1` the second, etc. `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 `wd0`, `wd1`, `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*

   - 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, `g` 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 `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. *Ensure you have the correct kernel installed*

    If you are installing from the 32-bit sparc distribution set, make sure that you installed the correct kernel. The sparc64 installation tools do not by default copy the correct 32-bit kernel. Unless you prepared ahead of time by renaming the `kern-GENERIC_SUN4U.tgz` to `kern-GENERIC.tar.xz` then you will need to follow the next few instructions.

    Go to the main installation menu, and select **Utility menu** and then select the **Run /bin/sh** option, which will give you a shell prompt. 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 '^?'
    ```

    Type the following command (replacing `wd0a` with the partition name of your destination root partition):

    ```
    # mount /dev/wd0a /mnt
    # cd /mnt
    ```

    Now you need to mount the location of your distribution sets:

    ```
    # mount /dev/cd0a /mnt2
    # tar xpzvf /mnt2/sparc/binary/kernel/kern-GENERIC_SUN4U.tgz
    # umount /mnt
    # umount /mnt2
    # exit
    ```

20. *Finalizing your installation*

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

Skip down to the section on **Booting NetBSD for the first time**

Manual Installation of NetBSD using Solaris

(Adapted from Murray Stokely's <murray@osd.bsdi.com> instructions)

You can use Solaris to prepare the NetBSD user-friendly installer or to perform a full manual installation of NetBSD. If you want to use the user-friendly miniroot installer or RAM disk installation kernel, follow the sections **Installing NetBSD by using the NetBSD miniroot** or **Installing NetBSD by using a NetBSD kernel on a Solaris partition**.

Manual installation from Solaris 10 is not possible because NetBSD cannot use the resulting UFS file system. It is possible to install Solaris 10 and NetBSD on the same disk. To do so, partition the disk with the Solaris format command, then boot NetBSD and perform a manual installation. Be careful not to write a NetBSD disklabel. Use the disklabel command to read the partition size, as constructed from the Solaris disklabel. By default the NetBSD newfs command writes a NetBSD disklabel. Avoid this by using the -F and -s arguments to newfs.

- *Preparing the disk in Solaris*

  The first step is to format and label the disk that you would like to use with NetBSD. This can be accomplished with the `format`(1M) command in Solaris, which allows you to partition a disk and write a disklabel. It also is used to perform a low-level format on SCSI drives. You will want to create a root partition and a swap partition. Depending on your preferences, you may also wish to create separate /usr or /var partitions.

```
# /usr/sbin/format
Searching for disks...
Mode sense page(3) reports nsect value as 280, adjusting it to 218
done

c0t1d0: configured with capacity of 16.95GB

AVAILABLE DISK SELECTIONS:
       0. c0t0d0 <SUN4.2G cyl 3880 alt 2 hd 16 sec 135>
          /pci@1f,4000/scsi@3/sd@0,0
       1. c0t1d0 <IBM-DXHS18Y-0430 cyl 8152 alt 2 hd 20 sec 218>
          /pci@1f,4000/scsi@3/sd@1,0
Specify disk (enter its number): 1
selecting c0t1d0
[disk formatted]
Disk not labeled.  Label it now? y

format> format
Ready to format.  Formatting cannot be interrupted
and takes 114 minutes (estimated). Continue? y
Beginning format. The current time is Sat May 29 22:15:13 2004

Formatting...
done
Verifying media...
        pass 0 - pattern = 0xc6dec6de
   8151/19/208
        pass 1 - pattern = 0x6db6db6d
   8151/19/208

Total of 0 defective blocks repaired.
format> partition
partition> print
Current partition table (original):
Total disk cylinders available: 8152 + 2 (reserved cylinders)

Part      Tag    Flag     Cylinders        Size            Blocks
  0       root    wm       0 -   60      129.86MB    (61/0/0)      265960
  1       swap    wu      61 -  121      129.86MB    (61/0/0)      265960
  2     backup    wu       0 - 8151       16.95GB    (8152/0/0) 35542720
  3 unassigned    wm       0                0        (0/0/0)            0
  4 unassigned    wm       0                0        (0/0/0)            0
  5 unassigned    wm       0                0        (0/0/0)            0
  6        usr    wm     122 - 8151       16.69GB    (8030/0/0) 35010800
  7 unassigned    wm       0                0        (0/0/0)            0

partition> label
Ready to label disk, continue? y
partition> quit
```

format> **quit**

After your disk has been labeled you need to create file systems on your slices. The Solaris `newfs`(1M) command will create ffs file systems that can be used by NetBSD.

```
# /usr/sbin/newfs /dev/dsk/c0t1d0s0
# /usr/sbin/newfs /dev/dsk/c0t1d0s6
```

- *Installing NetBSD Software from Solaris*

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

  ```
  # /usr/sbin/mount /dev/dsk/c0t1d0s0 /mnt
  # mkdir /mnt/usr
  # /usr/sbin/mount /dev/dsk/c0t1d0s6 /mnt/usr
  ```

  Now extract the distribution file sets

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

  Now you should copy the NetBSD second stage bootloader into your new root partition and install the bootblocks using Solaris's `installboot`(1M) command.

  ```
  # cp ~/netbsd/installation/misc/ofwboot /mnt
  # /usr/sbin/installboot ~/netbsd/installation/misc/bootblk /dev/rdsk/c0t1d0s0
  ```

- *Creating NetBSD Device Nodes under Solaris*

  This is not a necessary step. If your `/dev` directory is empty, `init`(8) will create a RAM disk with all of the essential device nodes each time the system boots. If you want to create the device nodes on disk, you will need to use the Solaris `mknod`(1M) command. Look in `/dev/MAKEDEV` for the correct names, major and minor numbers, ownership, and permissions.

- *Configuring the NetBSD system from Solaris*

  To save effort, you may want to use your favorite Solaris editor to configure some of the files in `/etc` before booting into NetBSD the first time. In particular, you should look at `/etc/fstab`, `/etc/rc.conf`, `/etc/resolv.conf`, and `/etc/hosts`. See the section below on **Post installation steps** before **Booting NetBSD for the first time**.

Booting NetBSD for the first time

Now it is time to boot NetBSD for the first time. You will boot from your disk using similar syntax as described above in **Setting up Open Firmware** and **Determining how to boot from an SBUS or PCI card**. To boot from your first disk, type:

```
ok boot 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 **sun** for a local console, or whatever is appropriate for your serial console (some systems display garbage with a **sun** terminal type, you may need to use **sun-ss5**) and press RETURN. You may need to type one of the following commands to get your delete key to work properly, depending on your keyboard:

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

At this point, you need to configure at least one file in the /etc directory. You will need to mount your root 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_hme0="inet 192.0.2.123 netmask 255.255.255.0"
```

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

```
ifconfig_hme0="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

   If you are using a serial console, you will have to edit the /etc/ttys file and change sun-ss5 to the appropriate terminal type, such as vt220.

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 `sparc64/10.2_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 boot the install kernel using one of the methods described above. 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.
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:

The End