NAME

INSTALL — Installation procedure for NetBSD/amiga.

CONTENTS

DESCRIPTION

About this Document

This document describes the installation procedure for NetBSD 11.1 on the *amiga* 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://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-11/latest/CHANGES

CHANGES-11.1:

https://nycdn.NetBSD.org/pub/NetBSD-daily/netbsd-11/latest/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.
`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/amiga subdirectory structure

The amiga-specific portion of the NetBSD 11.1 release is found in the `amiga` subdirectory of the distribution: `.../NetBSD-11.1/amiga/`. 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/` amiga binary distribution sets; see below.

`installation/`
- `floppy/` amiga boot and installation floppies; see below.
- `miniroot/` amiga miniroot file system image; see below.
- `misc/` Miscellaneous amiga installation utilities; see installation section below.

Miniroot file system

The Amiga now uses a single miniroot file system for both an initial installation and for an upgrade. A gzipped version is available, for easier downloading. (The gzipped version has the **.gz** extension added to their names.)

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

Binary distribution sets

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

base The NetBSD 11.1 amiga **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/amiga 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.

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

NetBSD/amiga System Requirements and Supported Devices

NetBSD 11.1 runs on any Amiga that has a 68020 or better CPU with some form of MMU, and on 68060 DraCos.

NetBSD does not, and will never, run on run on A1000, A500, A600, A1200, A2000, A4000/EC030, CDTV and CD32 systems that are not enhanced by a CPU board.

For 68020 and 68030 systems, a FPU is recommended but not required for the system utilities. 68LC040, 68040V and 68LC060 systems don't work correctly at the moment.

The minimal configuration requires around 24 MB of RAM (not including CHIPMEM!) and about 250 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. (24 MB of RAM will actually allow you to compile, however it won't be speedy. X really isn't usable on a 24 MB system.)

You will probably want to compile your own kernel. `GENERIC` is large and bulky in order to accommodate all people. For example, most people's machines have an FPU, so you do not need the bulky FPU_EMULATE option.

If you have less than 8 MB of fast memory, you should make your swap partition large, as your system will be a lot of swapping. In addition, do *not* place your swap partition onto a old small (and normally slow) disk!

Supported devices

- IDE controllers
  - A4000/A1200 IDE controller, including ATAPI devices
  - ELBOX FastATA 1200 Mk-III/Mk-IV
  - Individual Computers ACA500 (both CF slots)
  - Individual Computers X-Surf IDE
- SCSI host adapters
  - 33c93 based boards: A2091, A3000 builtin, A3000 builtin modified for Apollo accelerator board (please note that this is not the same as the SCSI host adapter on the Apollo accelerator board itself, which is unsupported), and GVP series II.
  - 53c80 based boards: 12 Gauge, IVS, Wordsync/Bytesync and Emplant The Emplant SCSI adapter has been reported by a party to hang after doing part of the installation without problems
  - 53c710 based boards: A4091, BlizzardPPC 603e+, Magnum, Warp Engine, Zeus and DraCo builtin
  - FAS216 based SCSI boards: FastLane Z3, Blizzard I and II, Blizzard IV, Blizzard 2060, CyberSCSI Mk I and II
  - 53c770 based SCSI boards: Cyberstorm Mk III SCSI, Cyberstorm PPC SCSI
- PCI bridges
  - ELBOX Mediator PCI 1200 series (original model, LT2, LT4, SX, TX)
  - ELBOX Mediator PCI 4000 series (4000D, 4000Di, 3000D, 3/4000T, also Mk-II models)
  - Prometheus PCI bridge
  - Phase5 PCI bridge: CyberVision PPC, BlizzardVision PPC
- Clockports
  - Amiga 1200 on-board clockport
- Video controllers
  - ECS, AGA and A2024 built in on various Amigas
  - Retina Z2 (no X server available), Retina Z3 and Altais
  - Cirrus CL GD 54xx based boards: GVP Spectrum, Picasso II, II+ and IV, Piccolo and Piccolo SD64
  - Tseng ET4000 based boards: Domino and Domino16M proto, oMniBus, Merlin
  - A2410 (no X server available)
  - Cybervision 64
  - Cybervision 64/3D
- Audio I/O
  - Amiga builtin
  - Melody MPEG-audio layer 2 board

  - Repulse audio board

- Ethernet controllers
  - A2065 Ethernet
  - Hydra Ethernet
  - ASDG Ethernet
  - A4066 Ethernet
  - Ariadne Ethernet
  - Ariadne II Ethernet
  - Quicknet Ethernet
  - X-Surf Ethernet
  - X-Surf 100 Ethernet

- ARCnet controllers
  - A2060 ARCnet

- Most SCSI tape drives, including Archive Viper, Cipher SCSI-2 ST150

- SCSI-2 scanners behaving as SCSI-2 scanner devices, HP ScanJet II, Mustek SCSI scanner. SCSI scanner support is machine independent, so it should work, but hasn't been tested on most Amiga configurations. There are reports that the Mustek and HP ScanJet hang if accessed from the A3000. This might apply to other 33C93-Adapters, too.

- Most SCSI CD-ROM drives

- Serial/Parallel cards
  - HyperCom 3Z, HyperCom 4, HyperCom 3+ and 4+
  - MultiFaceCard II and III
  - A2232 (normal and clockdoubled)
  - SilverSurfer clockport-based serial port
  - IOBlix Zorro-Bus

- Amiga floppy drives with Amiga (880/1760kB) and IBM (720/1440kB) encoding. Our floppy driver doesn't notice when mounted floppies are write-protected. Your floppy will stay unchanged, but you might not notice that you didn't write anything due to the buffer cache. Also note that HD floppy drives only get detected as such if a HD floppy is inserted at boot time.

- Amiga parallel port

- Amiga serial port

- Amiga mouse

- DraCo serial port, including serial mouse

- DraCo parallel printer port

- Real-time clocks
  - A2000, A3000, A4000 builtin (r/w)
  - DraCo builtin (r/o)

If it's not on the above lists, there is no support for it in this release. In particular, there are no drivers for: Blizzard III SCSI option, Ferret SCSI, Oktagon SCSI.

Getting the NetBSD System on to Useful Media

Installation is supported from several media types, including:

- AmigaDOS
- 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.

AmigaDOS partition

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

Note where you place the files as you will need this later.

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

CD-ROM / DVD / USB stick

To install NetBSD from a CD-ROM drive, make sure it is a SCSI CD-ROM on a SCSI bus currently supported by NetBSD (refer to the supported hardware list) or an ATAPI CD-ROM connected to the A1200 or A4000 internal IDE connector. If it is a SCSI CD-ROM on a non-supported SCSI bus like Blizzard-3 SCSI or Apollo SCSI you must first copy the distribution sets to an AmigaDOS partition as described above.

If your SCSI CD-ROM is connected to a supported SCSI host adapter, or it is an ATAPI cd-rom connected to the A1200/A4000 internal IDE connector, simply put the CD into the drive before installation.

Find out where the distribution set files are on the CD-ROM, DVD or USB stick. Likely locations are `binary/sets` and `amiga/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-11.1
# cd amiga/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

You will need an AmigaDOS hard drive prep tool to prepare your hard drives for use with NetBSD/amiga. HDToolBox is provided with the system software and on floppy installation disks since Release 2.0 of AmigaDOS, so we will provide instructions for its use.

Note that NetBSD can't currently be installed on disks with a sector size other than 512 bytes (e.g., "640 MB" 90mm M-O media). You can, however, mount ADOSFS partitions on such M-O's.

Preparing your hard disk with HDToolBox

A full explanation of HDToolBox can be found with your AmigaDOS manuals and is beyond the scope of this document.

The first time you partition a drive, you need to set its drive type so that you have working geometry parameters. To do this you enter the "Change drive type" menu, and either use "read parameters from drive" or set them manually.

Note that you will be modifying your HD's. If you mess something up here you could lose everything on all the drives that you configure. It is therefore advised that you:

1. Write down your current configurations. Do this by examining each partition on the drive and the drives parameters (from Change drive type.)
2. Back up the partitions you are keeping.

What you need to do now is partition your drives, creating at least root and swap partitions.

This should be done as the HDToolBox manual describes. One thing to note is that if you are not using a Commodore controller you will need to specify the device your SCSI controller uses, e.g., if you have a Warp Engine you would:

From cli

hdtoolbox warpdrive.device

From wb Set the tooltype

SCSI_DEVICE_NAME=warpdrive.device

The important things you need to do above and beyond normal partitioning include (from Partition Drive section):

1. Marking all NetBSD partitions as non-bootable, with two exceptions: the root partition (/) if you want to boot NetBSD directly, or the swap partition if you want to boot the installation miniroot directly.
2. Changing the file system parameters of the partitions to NetBSD ones. This must be done from the partitioning section and "Advanced options" must be enabled.

To make the needed changes:

1. Click the "Adv. Options" button
2. Click the "Change file system" button
3. Choose "Custom File System"
4. Turn off "Automount" if on.
5. Set the dostype to one of these three choices:

   ```
   root partition  : 0x4e425207  (NBR\007)
   swap partition  : 0x4e425301  (NBS\001)
   other partitions: 0x4e425507  (NBU\007)
   ```

   Here '`other`' refers to other partitions you will format for reading and writing under NetBSD (e.g., `/usr`)

   Make sure you press RETURN to enter this value as some versions of HDToolBox will forget your entry if you don't.
6. Turn custom boot code off
7. Set Reserved Blocks start and end to 0.

8. Click Ok.

On the root (/) (and, for installation, swap) partition:

1. Turn custom boot code on
2. Set Reserved Blocks start and end to 0.
3. Set Number of Custom Boot Blocks to 16.
4. Set Automount This Partition on
5. Click Ok.

   Mask and maxtransfer are not used with NetBSD.

Caveat: The swap (for installation) and the root partition (if you plan to use the bootblocks) *must be within the first* 4 gigabytes of the disk! The reason for the former is that xstreamtodev uses trackdisk.device compatible I/O-calls, the reason for the latter is that the bootblock gets a 32bit partition offset from the operating system.

Once this is done NetBSD/amiga will be able to recognize your disks and which partitions it should use.

Transferring the miniroot file system

The NetBSD/amiga installation or upgrade now uses a *miniroot* file system which is installed on the partition used by NetBSD for swapping. This removes the requirement of using a floppy disk for the file system used by the installation or upgrade process. It also allows more utilities to be present on the file system than would be available when using an 880 KB floppy disk.

Once the hard disk has been prepared for NetBSD, the miniroot file system (`miniroot.fs`) is transferred to the swap partition configured during the hard disk prep (or the existing swap partition in the case of an upgrade). The xstreamtodev utility provided in the `amiga/installation/misc` directory can be used on AmigaDOS to transfer the file system for either a new installation or an upgrade. The file system can also be transferred on an existing NetBSD system for an update by using dd. This should only be done after booting NetBSD into single-user mode. It may also be possible to shutdown to single-user, providing that the single-user mode processes are not using the swap partition.

On AmigaDOS, run the command:

`xstreamtodev --input=miniroot.fs --rdb-name=<swap partition>`

where <swap partition> is the name you gave to the NetBSD partition to be used for swapping. If **`xstreamtodev`** is unable to determine the SCSI driver device name or the unit number of the specified partition, you may also need to include the option

`--device=<driver.name>`

and/or

`--unit=<SCSI unit number>`

To transfer the miniroot using NetBSD, you should be booted up in single user mode on the current NetBSD system, or use the **`shutdown now`** command to shutdown to single-user mode. Then copy the miniroot using **`dd`**:

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

where `/dev/rsd0b` should be the device path of the swap partition your system is configured to use. Once the file is copied, reboot back to AmigaDOS to boot the upgrade kernel.

Caveat: Once you have started installation, if you abort it and want to retry you must reinstall the `miniroot.fs` on the swap partition.

Installing the NetBSD System

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

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

The following is a walk-through of the steps necessary to get NetBSD installed on your hard disk. If you wish to stop the installation, you may press `CONTROL-C` at any prompt, but if you do, you'll have to begin again from scratch.

Transfer the miniroot file system onto the hard disk partition used by NetBSD for swapping, as described in the "Preparing your System for NetBSD Installation" section above.

Booting

NetBSD, with boot blocks installed

[This description is for V40 (OS 3.1) ROMs. For older ROMs, there might be small differences. Check your AmigaDOS documentation to learn about the exact procedure.] Using bootblocks may not work on some systems, and may require a mountable file system on others.

Reboot your machine, holding down both mouse buttons if you have a 2-button mouse, the outer mouse buttons if you have a 3-button mouse. On the DraCo, press the left mouse button instead, when the boot screen prompts you for it.

From the boot menu, select **`Boot Options`**. Select the swap partition with the miniroot, and then **`ok`**. Select **`Boot`** now. The machine will boot the bootblock, which will prompt your for a command line. You have a few seconds time to change the default. Entering an empty line will accept the default.

The bootblock uses command lines of the form:
    `file` [*`options`*]
where *`file`* is the kernel file name on the partition where the boot block is on, and [options] may contain the following:

If you have an AGA machine, and your monitor will handle the dblNTSC mode, you may include the **`-A`** option to enable the dblNTSC display mode.

If your machine has a fragmented physical memory space, as, e.g., DraCo machines, you should add the **`-n2`** option to enable the use of all memory segments.

Once your kernel boots

You should see the screen clear and some information about your system as the kernel configures the hardware. Note which hard disk device(s) are configured (`sd0, sd1,` etc.) Then you will be prompted for a root device. At this time type `sd0b`, where `sd0` is the device which contains the swap partition you created during the hard disk preparation. When prompted for a dump device, answer '**`none`**' for the install (normally, you would tell it one of the swap devices). When prompted for the root file system type, confirm '**`generic`**', which will auto-detect it.

If the system should hang after entering the root device, try again with

    `netbsd -I ff -b`

This disables synchronous transfer on all SCSI devices on the first bus.

The system should continue to boot. For now ignore "WARNING" messages about bad dates in clocks, and a warning about `/etc/rc` not existing. Eventually you will be be asked to enter the pathname of the shell, just press RETURN. After a short while, you will be asked to select the type of your keyboard. After you have entered a valid response here, the system asks you if you want to install or upgrade your system. Since you are reading the *install* section, '**i**' would be the proper response here...

The installer starts with a nice welcome messages. Read this message carefully, it also informs you of the risks involved in continuing! If you still want to go on, type '**y**'. The installer now continues by trying to figure out your disk configuration. When it is done, you will be prompted to select a root device from the list of disks it has found.

You should know at this point that the disks are *not* numbered according to their SCSI-ID! The NetBSD kernel numbers the SCSI drives (and other devices on the SCSI bus) sequentially as it finds them. The drive with the lowest SCSI-ID will be called `sd0`, the next one `sd1`, etc. Also, any ATAPI disk drives (e.g. ZIP) will be configured as "SCSI" drives, too, and will be configured before any 'real' SCSI drives if connected to the Amiga internal port on A4000/A1200 (if any are present). Real IDE drives will be configured as `wd0`, `wd1`, etc.

The installer will offer you to look at the NetBSD disk label of the disks at this point. You should do this, to find out what partition letters the NetBSD kernel assigned to the partitions you created, and as a check of whether the disk number you are going to use is right.

you are now at the point of no return. If you confirm that you want to install NetBSD, your hard drive will be modified, and perhaps its contents scrambled at the whim of the install program. Type `Control-C` *now* if you don't want this.

At this time, you will need to tell the installer which partition will be associated with the different file systems. Normally, you'll want to add a partition for `/usr`, at least.

Caveat: Do not use the `rsd`*N*`c` or `sd`*N*`c` partitions for anything! They are for access to the whole disk only and do *not* correspond to any Amiga partition!

The install program will now make the file systems you specified. There should be only one error per file system in this section of the installation. It will look like this:

```
newfs: ioctl (WDINFO): Invalid argument
newfs: /dev/rsd0a: can't rewrite disk label
```

If there are any others, restart from the beginning of the installation process. This error is ok as the Amiga does not write disklabels currently. You should expect this error whenever using newfs.

The install will now ask you want to configure any network information. It will ask for the machine's host name, domain name, and other network configuration information.

Since the network configuration might have lead to additional (NFS) file system entries, you get another chance to modify your fstab.

You are finally at the point where some real data will be put on your freshly made file systems. Select the device type you wish to install from and off you go....

Some notes:

- If you want to install from tape, please read the section about how to create such a tape.
- Some tapes (e.g. Archive Viper 150) refuse to operate with the default tape density `(nrst0)`. Try `nrst0h`, `nrst0m`, or `nrst0l` instead.
- Install at least the **base** and **etc** sets.
- If you have to specify a path relative to the mount-point and you need the mount-point itself, use '`.`'.

Next you will be asked to specify the timezone. Just select the timezone you are in. The installer will make the correct setup on your root file system (/). After the timezone-link is installed, the installer will proceed by creating the device nodes on your root file system under `/dev`.

Be patient, this will take a while...

Next, the installer will copy your keymap settings to the new system. After this, it will copy the kernel from the installation miniroot to the newly installed / upgraded system. If the installed system already has a kernel, it will ask you for confirmation.

kern.tgz distribution set, this is an old kernel, and you should answer "y" to install a working (although restricted) INSTALL kernel.

If you did install the kern.tgz kernel, you normally should answer "n".

Finally, the installer asks you if you want to install the bootblock code on your root disk and, if yes, what boot command it should execute. This is a matter of personal choice and can also be done from a running NetBSD system. See the `installboot`(8) manual page about how to do this.

Once the installer is done, halt the system with the **halt** command (wait for `halted` to be displayed) and reboot. Then again boot NetBSD this time selecting the root partition (/) from the boot menu, and tell it to boot

```
netbsd -s
```

You need to do your final tweaks now. First mount your file systems like so:

```
mount -av
```

Your system is now complete, and it is up to you to configure the rest. You may want to start by looking at `/etc/rc.conf`.

Once you are done with the rest of configuration unmount your file systems and halt your system, then reboot:

```
# cd /
# umount -av
# halt
```

Finally you can now boot your system and it will be completely functional:

```
netbsd
```

When it boots off of the hard drive, you will have a complete NetBSD system! *Congratulations!* (You really deserve them!!!)

Post installation steps

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

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

2. Configuring `/etc/rc.conf`

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3. 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 `amiga/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 upgrade path to NetBSD 11.1 documented here is by binary sets.

To do the upgrade, you must have the NetBSD kernel on AmigaDOS and you must transfer the miniroot file system `miniroot.fs` onto the swap partition of the NetBSD hard disk. You must also have at least the **base** binary distribution set available. Finally, you must have sufficient disk space available to install the new binaries. Since the old binaries are being overwritten in place, you only need space for the new binaries, which weren't previously on the system. This is typically not more than a few megabytes.

Since upgrading involves replacing the kernel 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 YOUR DISK, whether on the NetBSD partition or on another operating system's partition, before beginning the upgrade process.

To upgrade your system, begin by transferring the miniroot file system onto the hard disk partition used by NetBSD for swapping, as described in the "Preparing your System for NetBSD Installation" section above.

Now boot up NetBSD, with bootblocks installed

Note: This description is for V40 (OS 3.1) ROMs. For older ROMs, there might be small differences. Check your AmigaDOS documentation to learn about the exact procedure. Booting using bootblocks doesn't work at all on some systems, and may require a mountable file system on others.

Reboot your machine, holding down both mouse buttons if you have a 2-button mouse, the outer mouse buttons if you have a 3-button mouse. On the DraCo, press the left mouse button instead, when the boot screen prompts you for it.

From the boot menu, select **`Boot Options`**. Select the swap partition with the miniroot, and then **`ok`**. Select **`Boot`** now. The machine will boot the bootblock, which will prompt you for a command line. You have a few seconds to change the default. Entering an empty line will accept the default.

The bootblock uses command lines of the form:

`file` [*`options`*]

where *`file`* is the kernel file name on the partition where the boot block is on, and *`options`* may contain the same as described in the INSTALL section.

For installing, use

`netbsd -b`

If you machine has a split memory space, like, e.g., DraCo machines, use this instead:

`netbsd -bn2`

Once your kernel boots

You should see the screen clear and some information about your system as the kernel configures the hardware. Note which hard disk device is configured that contains your root `(/)` and swap partitions. When prompted for the root device, type `sd0b` (replacing '`0`' with the disk number that NetBSD used for your root/swap device). When prompted for a dump device, answer '**`none`**' for the upgrade. (For a normal boot, you would tell it one of the swap devices). When prompted for the root file system type, confirm '**`generic`**', which will auto-detect it.

You will be presented with some information about the upgrade process and a warning message, and will be asked if you wish to proceed with the upgrade process. If you answer negatively, the upgrade process will stop, and your disk will not be modified. If you answer affirmatively, the upgrade process will begin, and your disk will be modified. You may press `CONTROL-C` to stop the upgrade process at any time. However, if you press it at an inopportune moment, your system may be left in an inconsistent (and possibly unusable) state.

You will now be greeted and reminded of the fact that this is a potentially dangerous procedure and that you should not upgrade the **etc** set.

When you decide to proceed, you will be prompted to enter your root disk. After you've done this, it will be checked automatically to make sure that the file system is in a sane state before making any modifications. After this is done, you will be asked if you want to configure your network.

You are now allowed to edit your fstab, but normally you don't have to. Note that the upgrade-kit uses its own copy of the fstab. Whatever you do here *won't* affect your actual fstab. After you are satisfied with your fstab, the upgrade-kit will check all file systems mentioned in it. When they're ok, they will be mounted.

You will now be asked if your sets are stored on a normally mounted file system. You should answer '**y**' to this question if you have the sets stored on a file system that was present in the fstab. The actions you should take for the set extraction are pretty logical (we think).

After you have extracted the sets, the upgrade kit will proceed with setting the timezone and installing the kernel and bootcode. This is all exactly the same as described in the installation section.

Your system has now been upgraded to NetBSD 11.1.

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. There are several things that you should do, or might have to do, to insure that the system works properly.

You will probably want to get the **etc** distribution, extract it, and compare its contents with those in your `/etc` directory. You will probably want to replace some of your system configuration files, or incorporate some of the changes in the new versions into yours.

You will want to delete old binaries that were part of the version of NetBSD that you upgraded from and have since been removed from the NetBSD distribution.

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:

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.
This product includes software developed by Tommi Komulainen <Tommi.Komulainen@iki.fi>.
This product includes software developed by TooLs GmbH.
This product includes software developed by Trimble Navigation, Ltd.
This product includes software developed by Waldi Ravens.
This product includes software developed by WIDE Project and its contributors.
This product includes software developed by Winning Strategies, Inc.
This product includes software developed by Yen Yen Lim and North Dakota State University
This product includes software developed by Zembu Labs, Inc.
This product includes software developed by the Alice Group.
This product includes software developed by the Computer Systems Engineering Group at Lawrence Berkeley Laboratory.
This product includes software developed by the Computer Systems Laboratory at the University of Utah.
This product includes software developed by the Harvard University and its contributors.
This product includes software developed by the Kungliga Tekniska Högskolan and its contributors.
This product includes software developed by the Network Research Group at Lawrence Berkeley Laboratory.
This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.OpenSSL.org/)
This product includes software developed by the PocketBSD project and its contributors.
This product includes software developed by the RiscBSD kernel team
This product includes software developed by the RiscBSD team.
This product includes software developed by the SMCC Technology Development Group at Sun Microsystems, Inc.
This product includes software developed by the University of California, Lawrence Berkeley Laboratories.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory.
This product includes software developed by the University of California, Lawrence Berkeley Laboratory and its contributors.
This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

Some files have the following copyright:

The End