Frequently Asked Questions for FreeBSD 2.X, 3.X and 4.X

Chapter 3 Installation

Nik Clayton

3.1. Which file do I download to get FreeBSD?

Prior to release 3.1, you only needed one floppy image to install FreeBSD, namely `floppies/boot.flp`. However, since release 3.1 the Project has added out-of-the-box support for a wide variety of hardware, which takes up more space. For 3.x and later you need two floppy images: `floppies/kernel.flp` and `floppies/mfsroot.flp`. These images need to be copied onto floppies by tools like `fdimage` or `dd(1)`.

If you need to download the distributions yourself (for a DOS filesystem install, for instance), below are some recommendations for distributions to grab:

- bin/

- manpages/

- compat*/

- doc/

A 3.5 inch (1.44MB) floppy can accommodate 1474560 bytes of data. The boot image is exactly 1474560 bytes in size.

Common mistakes when preparing the boot floppy are:

- Not downloading the floppy image in *binary* mode when using FTP.

  Some FTP clients default their transfer mode to *ascii* and attempt to change any end-of-line characters received to match the conventions used by the client's system. This will almost invariably corrupt the boot image. Check the size of the downloaded boot image: if it is not *exactly* that on the server, then the download process is suspect.

  To workaround: type *binary* at the FTP command prompt after getting connected to the server and before starting the download of the image.

- Using the DOS `copy` command (or equivalent GUI tool) to transfer the boot image to floppy.

  Programs like `copy` will not work as the boot image has been created to be booted into directly. The image has the complete content of the floppy, track for track, and is not meant to be placed on the floppy as a regular file. You have to transfer it to the floppy ``raw'', using the low-level tools (e.g. `fdimage` or `rawrite`) described in the [installation guide to FreeBSD](#).

3.3. Where are the instructions for installing FreeBSD?

Installation instructions can be found in the [Handbook entry on installing FreeBSD](#).

3.4. What do I need in order to run FreeBSD?

You will need a 386 or better PC, with 5 MB or more of RAM and at least 60 MB of hard disk space. It can run with a low end MDA graphics card but to run X11R6, a VGA or better video card is needed.

See also [Chapter 4](#)

3.5. I have only 4 MB of RAM. Can I install FreeBSD?

FreeBSD 2.1.7 was the last version of FreeBSD that could be installed on a 4MB system. FreeBSD 2.2 and later needs at least 5MB to install on a new system.

All versions of FreeBSD will *run* in 4MB of RAM, they just cannot run the installation program in 4MB. You can add extra memory for the install process, if you like, and then after the system is up and running, go back to 4MB. Or you could swap your disk into a system which has >4MB, install onto the disk and then swap it back.

FreeBSD 2.1.7 will not install with 640 kB base + 3 MB extended memory. If your motherboard can remap some of the ``lost'' memory out of the 640kB to 1MB region, then you may still be able to get FreeBSD 2.1.7 up. Try to go into your BIOS setup and look for a ``remap'' option. Enable it. You may also have to disable ROM shadowing. It may be easier to get 4 more MB just for the install. Build a custom kernel with only the options you need and then remove the 4MB out. You can also install 2.0.5 and then upgrade your system to 2.1.7 with the ``upgrade'' option of the 2.1.7 installation program.

After the installation, if you build a custom kernel, it will run in 4 MB. Someone has even successfully booted with 2 MB, although the system was almost unusable.

3.6. How can I make my own custom install floppy?

Currently there is no way to *just* make a custom install floppy. You have to cut a whole new release, which will include your install floppy.

To make a custom release, follow the instructions in the [Release Engineering](#) article.

3.7. Can I have more than one operating system on my PC?

Have a look at [the multi-OS page](#).

3.8. Can Windows 95/98 co-exist with FreeBSD?

Install Windows 95/98 first, after that FreeBSD. FreeBSD's boot manager will then manage to boot Win95/98 and FreeBSD. If you install Windows 95/98 second, it will boorishly overwrite your boot manager without even asking. If that happens, see the next section.

3.9. Windows 95/98 killed my boot manager! How do I get it back?

and the boot manager will be reinstalled.

- Boot the FreeBSD boot floppy again and go to the Custom installation menu item. Choose Partition. Select the drive which used to contain your boot manager (likely the first one) and when you come to the partition editor for it, as the very first thing (e.g. do not make any changes) select (W)rite. This will ask for confirmation, say yes, and when you get the Boot Manager selection prompt, be sure to select ``Boot Manager''. This will re-write the boot manager to disk. Now quit out of the installation menu and reboot off the hard disk as normal.

- Boot the FreeBSD boot floppy (or CDROM) and choose the ``Fixit'' menu item. Select either the Fixit floppy or CDROM #2 (the ``live'' file system option) as appropriate and enter the fixit shell. Then execute the following command:

      Fixit# **fdisk -B -b /boot/boot0** *bootdevice*

  substituting *bootdevice* for your real boot device such as ad0 (first IDE disk), ad4 (first IDE disk on auxiliary controller), da0 (first SCSI disk), etc.

3.10. My A, T, or X series IBM Thinkpad locks up when I first booted up my FreeBSD installation. How can I solve this?

A bug in early revisions of IBM's BIOS on these machines mistakenly identifies the FreeBSD partition as a potential FAT suspend-to-disk partition. When the BIOS tries to parse the FreeBSD partition it hangs.

According to IBM[1], the following model/BIOS release numbers incorporate the fix.

Model BIOS revision

Model	BIOS revision
T20	IYET49WW or later
T21	KZET22WW or later
A20p	IVET62WW or later
A20m	IWET54WW or later
A21p	KYET27WW or later
A21m	KXET24WW or later
A21e	KUET30WW

It has been reported that later IBM BIOS revisions may have reintroduced the bug. This message from Jacques Vidrine to the FreeBSD laptop computer mailing list <freebsd-mobile@FreeBSD.org> describes a procedure which may work if your newer IBM laptop does not boot FreeBSD properly, and you can upgrade or downgrade the BIOS..

If you have an earlier BIOS, and upgrading is not an option a workaround is to install FreeBSD, change the partition ID FreeBSD uses, and install new boot blocks that can handle the different partition ID.

First, you will need to restore the machine to a state where it can get through its self-test screen. Doing this requires powering up the machine without letting it find a FreeBSD partition on its primary disk. One way is to remove the hard disk and temporarily move it to an older ThinkPad (such as a ThinkPad 600) or a desktop PC with an appropriate conversion cable. Once it is there, you can delete the FreeBSD partition and move the hard disk back. The ThinkPad should now be in a bootable state again.

With the machine functional again, you can use the workaround procedure described here to get a working FreeBSD installation.

1. Download boot1 and boot2 from http://people.FreeBSD.org/~bmah/ThinkPad/. Put these files somewhere you will be able to retrieve them later.

2. Install FreeBSD as normal on to the ThinkPad. *Do not* use Dangerously Dedicated mode. *Do not* reboot when the install has finished.

3. Either switch to the ``Emergency Holographic Shell'' (**ALT**+**F4**) or start a ``fixit'' shell.

4. Use fdisk(8) to change the FreeBSD partition ID from 165 to 166 (this is the type used by OpenBSD).

5. Bring the boot1 and boot2 files to the local filesystem.

6. Use disklabel(8) to write boot1 and boot2 to your FreeBSD slice.

       # **disklabel -B -b boot1 -s boot2 ad0sn**

3.11. Can I install on a disk with bad blocks?

Prior to 3.0, FreeBSD included a utility known as `bad144`, which automatically remapped bad blocks. Because modern IDE drives perform this function themselves, `bad144` has been removed from the FreeBSD source tree. If you wish to install FreeBSD 3.0 or later, we strongly suggest you purchase a newer disk drive. If you do not wish to do this, you must run FreeBSD 2.x.

If you are seeing bad block errors with a modern IDE drive, chances are the drive is going to die very soon (the drive's internal remapping functions are no longer sufficient to fix the bad blocks, which means the disk is heavily corrupted); we suggest you buy a new hard drive.

If you have a SCSI drive with bad blocks, see this answer.

3.12. I have just upgraded from 3.X to 4.X, and my first boot failed with ``bad sector table not supported"

FreeBSD 3.X and earlier supported `bad144`, which automatically remapped bad blocks. FreeBSD 4.X and later do not support this, as modern IDE drives include this functionality. See this question for more information.

To fix this after an upgrade, you need to physically place the drive in a working system and use disklabel(8) as discussed in the following questions.

3.13. How do I tell if a drive has `bad144` information on it before I try to upgrade to FreeBSD 4.0 and it fails?

Use disklabel(8) for this. `disklabel -r drive device` will give you the contents of your disk label. Look for a `flags` field. If you see `flags: badsect`, this drive is using bad144. For example, the following drive has `bad144` enabled.:

```
# disklabel -r wd0
# /dev/rwd0c:
type: ESDI
disk: wd0s1
label:
flags: badsect
bytes/sector: 512
sectors/track: 63
```

3.14. How do I remove `bad144` from my pre-4.X system so I can upgrade safely?

Use `disklabel -e -rwd0` to edit the disklabel in place. Just remove the word `badsect` from the flags field, save, and exit. The bad144 file will still take up some space on your drive, but the disk itself will be usable.

We still recommend you purchase a new disk if you have a large number of bad blocks.

3.15. Strange things happen when I boot the install floppy! What is happening?

If you are seeing things like the machine grinding to a halt or spontaneously rebooting when you try to boot the install floppy, here are three questions to ask yourself:-

1. Did you use a new, freshly-formatted, error-free floppy (preferably a brand-new one straight out of the box, as opposed to the magazine cover disk that has been lying under the bed for the last three years)?

2. Did you download the floppy image in binary (or image) mode? (do not be embarrassed, even the best of us have accidentally downloaded a binary file in ASCII mode at least once!)

3. If you are using Windows95 or Win98 did you run `fdimage` or `rawrite` in pure DOS mode? These operating systems can interfere with programs that write directly to hardware, which the disk creation program does; even running it inside a DOS shell in the GUI can cause this problem.

There have also been reports of Netscape causing problems when downloading the boot floppy, so it is probably best to use a different FTP client if you can.

3.16. I booted from my ATAPI CDROM, but the install program says no CDROM is found. Where did it go?

The usual cause of this problem is a mis-configured CDROM drive. Many PCs now ship with the CDROM as the slave device on the secondary IDE controller, with no master device on that controller. This is illegal according to the ATAPI specification, but Windows plays fast and loose with the specification, and the BIOS ignores it when booting. This is why the BIOS was able to see the CDROM to boot from it, but why FreeBSD cannot see it to complete the install.

Reconfigure your system so that the CDROM is either the master device on the IDE controller it is attached to, or make sure that it is the slave on an IDE controller that also has a master device.
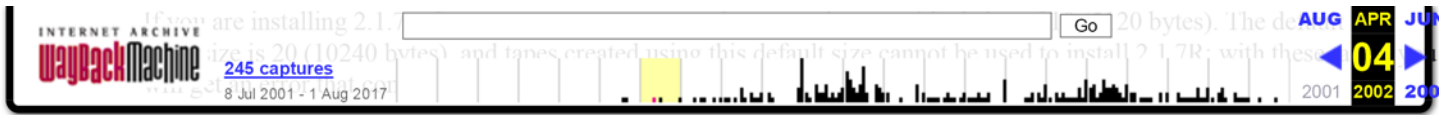
3.18. Can I install on my laptop over PLIP (Parallel Line IP)?

Yes. Use a standard Laplink cable. If necessary, you can check out the [PLIP section of the Handbook](#) for details on parallel port networking.

If you are running FreeBSD 3.X or earlier, also look at the [Mobile Computing page](#).

3.19. Which geometry should I use for a disk drive?

Note: By the ``geometry'' of a disk, we mean the number of cylinders, heads and sectors/track on a disk. We will refer to this as C/H/S for convenience. This is how the PC's BIOS works out which area on a disk to read/write from.

This causes a lot of confusion among new system administrators. First of all, the *physical* geometry of a SCSI drive is totally irrelevant, as FreeBSD works in term of disk blocks. In fact, there is no such thing as ``the'' physical geometry, as the sector density varies across the disk. What manufacturers claim is the ``physical geometry'' is usually the geometry that they have determined wastes the least space. For IDE disks, FreeBSD does work in terms of C/H/S, but all modern drives internally convert this into block references.

All that matters is the *logical* geometry. This is the answer that the BIOS gets when it asks the drive ``what is your geometry?'' It then uses this geometry to access the disk. As FreeBSD uses the BIOS when booting, it is very important to get this right. In particular, if you have more than one operating system on a disk, they must all agree on the geometry. Otherwise you will have serious problems booting!

For SCSI disks, the geometry to use depends on whether extended translation support is turned on in your controller (this is often referred to as ``support for DOS disks >1GB'' or something similar). If it is turned off, then use N cylinders, 64 heads and 32 sectors/track, where N is the capacity of the disk in MB. For example, a 2GB disk should pretend to have 2048 cylinders, 64 heads and 32 sectors/track.

If it *is* turned on (it is often supplied this way to get around certain limitations in MSDOS) and the disk capacity is more than 1GB, use M cylinders, 63 sectors per track (*not* 64), and 255 heads, where 'M' is the disk capacity in MB divided by 7.844238 (!). So our example 2GB drive would have 261 cylinders, 63 sectors per track and 255 heads.

If you are not sure about this, or FreeBSD fails to detect the geometry correctly during installation, the simplest way around this is usually to create a small DOS partition on the disk. The BIOS should then detect the correct geometry, and you can always remove the DOS partition in the partition editor if you do not want to keep it. You might want to leave it around for programming network cards and the like, however.

Alternatively, there is a freely available utility distributed with FreeBSD called `pfdisk.exe`. You can find it in the `tools` subdirectory on the FreeBSD CDROM or on the various FreeBSD FTP sites. This program can be used to work out what geometry the other operating systems on the disk are using. You can then enter this geometry in the partition editor.

3.20. Are there any restrictions on how I divide the disk up?

Yes. You must make sure that your root partition is below 1024 cylinders so the BIOS can boot the kernel from it. (Note that this is a limitation in the PC's BIOS, not FreeBSD).

For a SCSI drive, this will normally imply that the root partition will be in the first 1024MB (or in the first 4096MB if extended translation is turned on - see previous question). For IDE, the corresponding figure is 504MB.

3.21. Is FreeBSD compatible with any disk managers?

FreeBSD recognizes the Ontrack Disk Manager and makes allowances for it. Other disk managers are not supported.

If you just want to use the disk with FreeBSD you do not need a disk manager. Just configure the disk for as much space as the BIOS can deal with (usually 504 megabytes), and FreeBSD should figure out how much space you really have. If you are using an old disk with an MFM controller, you may need to explicitly tell FreeBSD how many cylinders to use.

If you want to use the disk with FreeBSD and another operating system, you may be able to do without a disk manager: just make sure the FreeBSD boot partition and the slice for the other operating system are in the first 1024 cylinders. If you are reasonably careful, a 20 megabyte boot partition should be plenty.

3.22. When I boot FreeBSD I get ```Missing Operating System`''. What is happening?

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.