Ftape−HOWTO

Table of Contents

Table of Contents

Table of Contents

Ftape−HOWTO

Claus−Justus Heine, <heine@math1.rwth−aachen.de>

v3.0, August 1998

This HOWTO discusses essential do's and dont's for the `ftape` floppy tape driver under Linux. It focusses on the newest version which is `ftape−4.02` at the time of this writing. This HOWTO is to be intended as first step help and source of information. The `ftape` driver interfaces to QIC−40, QIC−80, QIC−3010 and QIC−3020 compatible drives, and to the Iomega Ditto 2GB and Ditto Max drives. The QIC−3010 and QIC−3020 standards are also known as `Travan' (TR−2 and TR−3). These drives connect via the floppy disk controller (**FDC**) which may be either an internal FDC or inside of certain parallel port floppy tape drives. Please refer to the section Supported drives for further information. `ftape`**does not** cover SCSI or QIC−02 tape drives. DAT tape drives usually (always?) connect to a SCSI controller. This is but one of the Linux HOWTO documents. You can get an index of the HOWTOs from the Linux HOWTO index, while the real HOWTO's can be fetched (using `ftp`) from `sunsite.unc.edu:pub/Linux/doc/HOWTO` (this is the ``official'' place) or via the World Wide Web from the Linux Documentation Project home page.

5.The Care and Feeding of Tape and Tape Drives

6.Hardware support

7.Backing up and restoring data

8.Creating an emergency boot floppy for `ftape`

9.Frequently Asked Questions

10.FAQ: "Compiling and installing Ftape" related questions !

11.FAQ: "Using Ftape" related questions !

12.FAQ: "Tape and Drivers" related questions !

13.FAQ: Miscellaneous !

14.Debugging the `ftape` driver

15.Contributions

1.Legalese

If you have questions or comments, please contact the author at

heine@math1.rwth−aachen.de

2.Revision History

version 3.0 (August, 1998)

Additions to list of supported hardware.
New section about differences between ftape versions.
Pointers to the Ftape−FAQ and the Ftape manual.
Updated to ftape−4.02.
Additions to the FAQ.
Update all URLs.

version 2.0 (March 15, 1997)

Updated to ftape v2.11 and v3.xx
Lots of updates.

version 1.9 (September 20, 1996)

New maintainers of ftape and the HOWTO.
A few minor formatting and spelling fixes.
Updated for Linux v2.0.
Started to integrate some of Andrew Martin's ftape info.

version 1.8 (May 22, 1996)

Copyright policy changed to GNU GPL v2
The maintainer's email address has changed.
Updated to ftape−2.08
`ftape` is now a part of the kernel distribution.

version 1.7.1 (February 13, 1996)

Updated to ftape−2.06b

version 1.7 (January 28, 1996)

Updated to ftape−2.06 and modules−1.3.57

version 1.6.2 (January 23, 1996)

Connor TST3200R drive added
Updated 2Mbps fdc information.

version 1.6.1 (January 16, 1996)

minor corrections

version 1.6 (January 10, 1996)

New maintainer of `ftape`
updated to v2.05
added new drives

3.The preliminaries

3.1 Other sources of information

ftape version 3

`ftape−3.x` came with a manual of its own, which is contained in the `ftape−3.04d` package available from the usual places. See Getting Ftape.

ftape version 4

`ftape−4.x` also has a documentation package `ftape−doc` which is available from the usual places. This Ftape−HOWTO, however, also focusses on `ftape−4.x` and is meant as an entry point to the available documentation. See Getting Ftape.

ftape−tools

The `ftape−tools` package (including useful utilities for `ftape`) comes with its own manual. See Getting Ftape.

Ftape−FAQ

The `Ftape−FAQ` is included wordly in this manual, but more recent versions may be found at http://www.correct.nl/~ftape.

3.2 Contacts

The maintainer of the source for `ftape` is Claus Heine <heine@math1.rwth−aachen.de>. He has a web page at http://www−math.math.rwth−aachen.de/~LBFM/claus/ftape/.

If you have a problem or questions about ftape, try posting to the `Linux Tape` mailing list `linux-tape@vger.rutger.edu` (see Following the ftape development below). There also used to be a newsgroup that mirrored the mailing list traffic but it has vanished some time ago.

I use `ftape` (it is my sole means of backing up on my linux box :−). I hesitate to make recommendations on what hardware to buy. See the section Supported drives and Unsupported drives for a list of supported and unsupported drives.

You should try to post a summary of your problems and its solution(s), after you've got it working, even if you only got it partially working. Please also send a copy copy of your solution to the Linux Tape mailing list at <linux−tape@vger.rutgers.edu> so that it can be added to the HOWTO and/or the FAQ.

If you receive this as part of a printed distribution or on a CD−ROM, please check out the Linux Documentation home page or ftp to ftp://sunsite.unc.edu:/pub/Linux/doc/HOWTO to see if there exists a more recent version. This could potentially save you a lot of trouble.

If you email me, please include the string `ftape` in the subject line. This will help ensure the mail doesn't inadvertently get buried. But preferrably you should email to the Linux Tape mailing list at <linux−tape@vger.rutgers.edu> instead of contacting me directly.

3.3 What is `ftape`

`ftape` is a driver program that controls various low−cost tape drives that connect to the floppy controller.

`ftape` is not a backup program as such; it is a device driver, which allows you to use the tape drive (just like the SoundBlaster 16 driver let you use your sound card) through the device files `/dev/[n]qft[0−3]`.

`ftape` was originally written by Bas Laarhoven `<bas@vimec.nl>`, with ``a little help from his friends'' to sort out the ECC (Error Correcting Code) stuff. `ftape` is copyrighted by Bas under the GNU General Public License, which basically says: ``go ahead and share this with the world, just don't disallow other people from copying it further''.

`ftape` has undergone several changes since then. While the Linux−2.0.x kernel series still contains `ftape−2.08` the v2.1.x and soon the v2.2.* kernel series come with `ftape−3.x` (hopefully even with `ftape−4.02`, but this wasn't clear at the time of this writing) which differs in some points from the `ftape−2.x` driver. Since version `3.00` the `ftape` driver has been maintained by me (Claus−Justus Heine); it has been changed and improved in several respects and support for new hardware has been added.

`ftape` is quite stable, and has been that for some time now. It is reliable enough for critical backups (but it's always a good idea to check your backups, so you won't get a nasty surprise some day).

`ftape` supports drives that conform to the QIC−117 and one of the QIC−80, QIC−40, QIC−3010, and QIC−3020 standards as well as the Iomega Ditto 2GB and Ditto Max drives which no longer strictly conform to the QIC standards in all respects.

`ftape` can drive floppy tape drives that connect to the internal FDC as well as certain parallel port floppy tape drives.

`ftape` supports neither QIC−02, IDE (ATAPI), nor SCSI tape drives. SCSI drives are accessed as `/dev/[n]st[0−7]` and are supported by the kernel through the SCSI drivers. If you look for help on SCSI tape drives, you should read the `SCSI−howto`. ATAPI tape drives are supported by the kernel since 1.3.46. See section Supported drives and Unsupported drives for a list of supported and unsupported drives.

4.Getting and installing `ftape`

4.1 Getting `ftape`

The v2.0.x versions of the kernel include version 2.08 of `ftape` I recommend, however, that you grab the latest version of the full source code package for `ftape`. It is a newer version, includes files that are not included in the kernel v2.0.X distribution, and includes much better documentation about how to install `ftape`. The v2.1.x and later versions of the kernel include the version 3.04 of `ftape`.

I recommend that you download the latest stable version of `ftape` which is 4.02 at the time of this writing and is available from

http://www−math.math.rwth−aachen.de/~LBFM/claus/ftape/archives.html

as well as from

ftp://sunsite.unc.edu/pub/Linux/kernel/tapes/.

You probably should also grab the `ftape−doc` and the `ftape−tools` package that are available from the same locations.

If you still want to use the `ftape−2.08` which is shipped with the v2.0.x kernels, then you get a version of the driver which is really out of date and doesn't support QIC−3020 tapes at 2Mbps correctly, neither does it support the Ditto 2GB drives nor the Ditto Max drives nor any kind of parallel port tape drive. The section Supported drives gives detailed information about which version of the `ftape` driver supports which hardware.

4.2 Differences between `ftape−2.x`, `ftape−3.x` and `ftape−4.x` versions

`ftape−3.x` and `ftape−4.x` use the file system interface that was implemented for a branch release which was called `zftape`. Actually, the module that implements the **VFS** (`Virtual File System`) interface of `ftape−3.x` and `ftape−4.x` still is called `zftape.o` and its `C`−sources inside the kernel tree reside in `[/usr/src/linux/]drivers/char/ftape/zftape/`.

`ftape−2.x` (i.e. the version still contained in the v2.0.x kernel) uses another file system interface, that was implemented by `ftape's` original author Bas Larhoven.

File Marks

The conceptional difference between `ftape−2.x` and later versions of `ftape` is the way *file marks* are implemented.
Floppy tape devices don't have real file marks. **File marks** are used to distinguish different backup sets if you write multiple backup sets to a tape. SCSI and QIC−150 tapes have *real* file marks, i.e. between two different backup sets there is a region on the tape that is written special data to so that the drive logic can detect that marker when the tape is wound with (possibly) high speed over those file marks. Because the goal of `ftape's` file system interface was from the beginning on to provide an interface that could be used with standard Unix−like tape utilities (i.e. `mt`) the developers of `ftape` started to emulate file marks by storing the positions on the tape where a file mark should be located in certain fields of the header segments. **header segments** refers to a region at the beginning of the tape sized two times 29k to hold some important information about the tape format and size and some status information.
However, the `QIC` standards already designate a special region to store such information in, the so called **volume table segment**. Since `ftape−3.x` this *volume table segment* is used instead of using unused data fields in the header segment. As a result it is possible to use your tape cartridge with different operating systems in the sense that your Win or DOS backup program will realize that certain regions of the tape cartridge are already occupied with data, and `ftape−3.x` and later will detect the regions used by those DOS and Win programs. However, you can't extract a DOS backup set under Linux or extract a volume written by `ftape` under DOS, safe you write your own software to do that.

IOCTL interface

There are certain differences in the **IOCTL**This *IO control* interface is used by e.g. `mt` to rewind the tape or skip to the next file mark or do any other tape operation. interface between `ftape−2.x` and `ftape−3.x` and later. A detailed description can be found in the `ftape−manual` contained in the `ftape−doc` package. See Getting Ftape.

Formatting

Formatting of cartridges is supported with `ftape−3.x` and later **only**. Please get the `ftape−tools` package that contains the `ftformat` program that interfaces to the driver to format cartridges. See Getting Ftape. The `ftape−tools` package comes with (more or less) detailed documentation, so the case of formatting cartridges is not dealt with in this document.

Compression

`ftape−3.x` supported user transparent on−the−fly compression in software. This feature (or *bug*) has vanished in `ftape−4.x` as it made further improvements concerning the realiability of backups very very hard. This means, `ftape−4.x` comes without compression support.
However **de−compression** of compressed archives produced with `ftape−3.x`**is** supported in order not to brake existing backup programs where a user−level filter would not suffice to preserve compatibility. Think, e.g., of `taper` which calls the `MTIOC` ioctls itself instead of relying on the `mt` program to perform tape operations.

The `ftape−manual` contained in the `ftape−doc` package contains much more detailed information about `` ftape`s `` file system interface as well as implementation notes which by far exceed the scope of this HOWTO. See Getting Ftape for informations about where to obtain the manual.

4.3 Installing the driver with v2.0.x and earlier kernels

The following section provides some useful information to get you going with the installation of v4.x which is **not** shipped with the kernel source tree yet but has to be downloaded separately, see the section Getting ftape above.

Once you've downloaded the source code (probably `ftape−4.02−tar.gz`), untar it. You can do this by determining what directory you want the source code to be located in. I recommend `/usr/src/` or `~/src`. When the tar file is extracted, it will dump everything into a `ftape−4.02` subdirectory, so that you'll end up, in the example I've given, with something like `/usr/src/ftape−4.02` or `~/src/ftape−4.02`.

NOTE: you cannot compile `ftape−4.02` into your v2.0.x kernel. Instead, configure your kernel to **not** compile the `ftape` driver and follow the installation instructions in the `ftape−4.02` distribution and install `ftape−4.02` as a module.

Read the `README` file. The `README` is required reading. It's the top of the tree, so to speak. If there are specific files that the `README` tells you to read then read them. It will make the process much less complicated.

Do NOT proceed with compiling the package until you have read the appropriate `README` files and the `INSTALL` file.

Afterwards you need to edit the `MCONFIG` file and configure you package according to your hardware. The `MCONFIG` file contains lots of explanations so it should be fairly easy to go along with it.

However, most of the hardware configuration can be done via setting parameters during module load time so most parameters specified in the file `MCONFIG` simply give the default configuration, but you don't need to recompile the driver to change IO addresses or interrupt settings. The file `INSTALL` and the file `modules/insert` contain examples how to specify the proper module parameters when loading the kernel modules, so I won't go into further detail here.

If you are using a Linux−v1.3.x kernel, you should consider moving to v2.0.x. v1.3.x was the development release prior to the production release v2.0.x.

4.4 Installing the driver with v2.1.x and later kernels

Comment

Maybe `ftape−4.02` will be included into the v2.2.x kernel, but this isn't clear at the time of this writing. This HOWTO will be revised appropriately when this has become clear. So long you have to refer to the previous section Installing the driver with v2.0.x and earlier kernels and disregard the contents of this section.

The Linux kernel v2.1.x and later already include `ftape−4.x` so you don't need to download the `ftape−4.x` kernel driver package.

`ftape−4.x` as included in the v2.1.x versions of the kernel can be completely configured using the kernel configuration menus (either with `make menuconfig` or `make xconfig`. Also, there is online help available that documents each parameter setting which I won't repeat here.

The various boot− and loadtime parameter settings are explained in the file

`[/usr/src/linux/]Documentation/ftape.txt`

of the Linux−v2.1.x and later kernel distributions.

4.5 Following the development of the `ftape` driver

If you want to follow the development of the `ftape` driver, you should subscribe to the Linux Tape mailing list `linux−tape@vger.rutgers.edu`. To do so you need to send an email saying `` `subscribe linux−tape`' (*in the body*) to `majordomo@vger.rutgers.edu`. When you subscribe, you will be sent a greeting mail, which will tell you how to submit real mails and how to get off the list again. **Store this email in a safe place**. Please.

Please note that I do not, repeat **DO NOT**, have any special powers with regard to this mailing list. If you're stuck on the list, don't bother to tell me that. I can only shrug and send you my sympathy (but that won't get you off the list).

4.6 Mixing `ftape` and floppies

If you use your floppy tape drive with the standard FDC then the floppy drive and the floppy tape drive cannot run concurrently as they share the same hardware, the FDC, and the `floppy` and the `ftape` driver do not talk to each other. Thus, if you have mounted a floppy and then try to access the tape drive, `ftape` will complain that it cannot grab IRQ6 and then die. This is especially a problem when designing a emergency disk for use with ftape. This solution is to either load the boot/root disk into a ramdisk and then unmount the floppy, or have two floppy drive controllers.

5.The Care and Feeding of Tape and Tape Drives

5.1 Formatting

Before a tape can be used, it must be formatted. The formatting process lays out sector information onto the tape. Other tape interfaces don't typically require formatting. The reason floppy tapes do is that they need to look like a floppy (kinda gross, but what the hey − it works :−).

Can I format my tapes under Linux?

Yes, you can, if you use `ftape−3.04d` or above. To format a floppy tape cartridge you need a user level tool called `ftformat` as well which is contained in the `ftape−tools` distribution (see section Getting ftape).

The `ftape−tools` package comes with its own manual, so I do not need to repeat here how to use `ftformat`.

Which formatting programs can I use under DOS?

The following are known to work:

- Colorado Memory System's software (`tape.exe`)
- Conner Backup Basics v1.1 and all Windows versions
- Norton Backup
- QICstream version 2
- Tallgrass FileSecure v1.52
- Escom Powerstream 3.0 (`qs3.exe` −− QICstream v3?)

These programs are known to be more or less buggy:

- Conner Backup Basics 1.0
- Colorado Windows tape program
- CP Backup (wastes tape space, but is OK apart from that)

As a general rule, most software under DOS should work. The Conner Backup Basics v1.0 has a parameter off by one (someone could not read the QIC−80 specs right!), which is corrected in version 1.1. However, `ftape` detects this, and will work around it. Dennis T. Flaherty (`<dennisf@denix.elk.miles.com>`) report that Conner C250MQ owners can obtain the new v1.1, by calling Conner at 1−800−4Conner (in the US) and ask for an upgrade (for a nominal fee for the floppy). The Windows versions should work fine. Some versions of Colorado's tape program for windows, has an off−by−one error in the number of segments. `ftape` also detect and work around that bug.

Central Point Backup can be used, but it wastes precious tape space when it encounters a bad spot on the tape.

NOTE: If you are running a formatting software under DOS, which is not mentioned here, please mail the relevant info to me (<heine@math1.rwth−aachen.de>), so I can update the list.

5.2 Retensioning

QIC tapes are particularly sensitive to tape stretch. The reason is that floppy tapes are pre−formatted with sector information, whereas other tape types have their sync information written as the data is written to the tape. If the floppy tape stretches and the sync fields get out of sync the result will be read errors. The problem is worse with longer tapes.

It is a good idea to retension new tapes a few times before using them and before formatting them. You should also try retensioning the tape if you are start getting read errors. It might also be a good idea retension the tape before a backup.

5.3 Drive Cleaning

The coating on the tape is an oxide compound. As the tape is dragged across the tape head it has a tendency to leave tiny amounts of residue on the head. You should periodically use a tape cleaner − following the specs for the drive in question. Tape cleaners should be available from any distributer of tapes.

One more additional note about tape cleaning. You might want to clean the drive after the first use of a brand new tape. A brand new tape will typically leave quite a bit of residue the first time it's used.

Thanks to Neal Friedman for the explanation and suggestion that this information be included in the HOWTO.

5.4 Repairing de−spooled cartridges

In rare occasions it can happen that the tape drive doesn't detect the **EOT** (End Of Tape) markers correctly. These markers are simply holes in the tape which are detected by the tape drive with means of a little photo−transistor (or the like).

The manual of your tape drive will probably give you proper hints how to clean those EOT detectors.

However, if the EOT detection fails, then the tape drive despooles the cartridge because the tape isn't glued to the wheels, but hold by friction only.

There are detailed instructions how to fix such a despooled tape at the Iomega WWW pages at http://www.iomega.com/support/techs/ditto/3006.html

and at the Hewlett Packard WWW pages at

http://www.hp.com/isgsupport/cms/docs/lpg12020.html

If the pages shouldn't be in the exact locations as given above, then please try to browse a little bit through the web pages of HP or Iomega until you find the needed information.

6.Hardware support

6.1 Supported tape drives

All drives that are both QIC−117 compatible *and* one of the QIC−40, 80, 3010, and 3020 standards should work. QIC−WIDE and Travan drives are also supported (TR−1 is just QIC−80 with 8mm tapes, while TR−2 and TR−3 is a.k.a QIC−3010 and 3020 respectively). Iomega Ditto 2GB and Ditto Max drives are supported, too, though they no longer conform to the QIC standards in every respect. Some parallel port tape drives are supported as well.

Some of the comments given below about possible problems with certain tape drives are very old, and I don't have access to all of the hardware, so I couldn't check everything.

Some of the reports below have been commented by me (<heine@math1.rwth−aachen.de>) like this:

> This is a comment.

Currently, the list of drives that are known to work with `ftape` is:

Alloy Retriever 250

Archive 5580i, XL9250i

Colorado DJ−10, DJ−20 (aka: Jumbo 120, Jumbo 250)

Colorado 1400

<kosowsky@bellini.harvard.edu> reported a problem doing a 1G backup using taper.

Colorado Trakker parallel port tape drive

Support added by Jochen Hoenicke <Jochen.Hoenicke@Informatik.Uni−Oldenburg.DE>.

HP Colorado T1000

The problem reports are probably totally out−dated. In particular, the `zftape` the people talk about doesn't exist any more, and the `ftape` driver is the very `ftape-2.08`.
Works with 3M Travan 400M (TR−1) tapes with 120M tapes. Also reported that mt dies, but with backups using tar it works ok. With cpio, ftape is recommended rather than zftape. (<millner@millner.bevc.blacksburg.va.us>)
Problems have been reported with the drive continually stopping and starting with zftape (<75104.1756@compuserve.com>). This appears to be a problem with the tape going too fast for the computer; the DMA buffers are getting flushed before getting filled again. Newer versions of zftape don't do this any more is a suitably fast backup program or large DMA buffers are used (<millner@millner.bevc.blacksburg.va.us>).

Conner C250MQ(T)

The 250Q is reported to generate write error and frequent repositioning. (Frank Stuess at Nacamar Data Communications)
Write errors need not be caused by the tape drive, but also by bad tape cartridges. Frequent repositioning can be caused by bad cartridges, too, but can also be caused by overrun errors which would indicate that the FDC and DMA controller have problems to talk to each other.

Conner TSM420R, TSM850R

The 400 and 800 models only work with TR−1 tapes.
I don't know whether it was meant that named drives doesn't work with ordinary 120MB DC−2120 cartridges, or that TR−3 tapes can't be read. The tape drives weren't designed for the latter. So what.

Conner TST3200R

Works with TR−3 tapes at 1Mbps (ie. 1600M capacity only). Wirks with QIC−WIDE 400M tapes (Sony 5122's?) (<chris@cs.wmich.edu>). Works with TR3, QIC−3010, and QIC−3020 tapes. Comes with a 2MB FDC which the Promise 2300+ 1Mbps controller works (<kjh@pollux.usc.edu>).
Reported that the floppy disk can no longer read low−density floppies. May have to fiddle with IRQ/ports/dma channels (<chris@yakkocs.wmich.edu>).
The TST3200R works well with `ftape`.

Conner TST800R

The TST800R works with TR−1, Sony QW5122F (210M) and DC2120 tapes.
Works well with `ftape` since `ftape-2.07` at least. Used it myself until the drive died with a melted transistor. Probably caused by over−heating it previously.

Conner CTT3200

The CTT3200 is supposedly identical to the Iomega Ditto 3200. It works with the supplied 2Mbps controller, but reported not to work under DOS on some machines. (<jmorris@dtx.net>)

Conner 1.7G Tapestor (TSM1700R)

Works with QIC−WIDE tapes (<pschmidt@slip.net>). Partially works with QIS−3200. Using the HSC−2 controller, the DMA channel needs to be changed (incremented by 1, channel2?, Modify the Makefile). You then need to modify the ftape Makefile to reflect this change. However, ftape seems to be a bit flaky with this (no version number supplied) (<ttait@tiac.net>). It may not work at 2Mbps (QIC−3020) with the HSC controller. The tape died with a messages like "dumb tape stop" and has since been unreliable (<ttait@tiac.net>). No recent informations available

Escom or Archive (Hornet) 31250Q

Exabyte EXB−1500

Work with QIC−3010 tapes.

Exabyte TR−3

Irwin 80SX, Insight 80Mb

Iomega 250

Iomega Ditto Tape Insider 420, 1700

Iomega Ditto Tape Insider 3200

This is the unit, that I use. The default jumper settings don't work. Leave the irq and ioport address at the default (6 and 0x370, respectfully), but change the DMA from 3 to 2. (Kevin Johnson <kjj@pobox.com>).
Refer to the file `MCONFIG` of recent `ftape` distributions for other suggestions for ioport, irq and DMA channel.
May require the having `{0x08882, 80, wake_up_colorado, "Iomega 3200"},` added to `vendors.h` on older versions of `ftape`.
Problems reported with ftape 2.07 and kernel 1.12.13. With all sorts of combinations of accelerator, etc, the drive may (on some systems) only be accessed once (<erwin@box.nl>). Also, after the first access, the next use of the tape says it is write protected (<erwin@box.nl>, <M.J.Ammerlaan@dutiwy.twi.tudelft.nl>).
There has been one report of a problem where the tape got wound off the end of the spool. This may be caused by a dirty EOT sensor, and need not be a real hardware bug (except when it was a bug that dirtied the EOT sensor ...)
Another problem has been reported with writing archives (with dd) to the tape. It may start fine, but when the driver catches up with dd, it stops the tape and rewinds it to the beginning.

Then it starts winding on through the tape ad infinitum. It appears to occur when the driver asks the tape to pause which should cause the tape to move back by 3 segments, but instead is moves back to the beginning of the tape. A bug fix submitted is reported to not solve the problem.
Should have been fixed somewhere between `ftape−3.00` and `ftape−4.00`. Unluckily, the fast−skipping facilities of all Iomega floppy tape drives are **really** poor. Recent `ftape` versions work around this problem. I suggest getting the latest version of the `ftape` driver when you experience this problem.

Iomega Ditto 800 Insider

Works with Travan TR1, TR2, or DC2120 tapes (<klein@informatik.uni−rostock.de>).

Iomega Ditto 2GB

Support added by Jochen Hoenicke <Jochen.Hoenicke@Informatik.Uni−Oldenburg.DE> to `ftape−3.xx` and later.
Can't format cartridges, writing is only possible with special Ditto 2GB cartridges (hardware limitation, not a lacking feature of `ftape`).

Iomega Ditto Max

Iomega Ditto Max Pro

Supported since `ftape−4.00`. Thanks to Tim Jones <tjones@estinc.com>.
Can't format cartridges, writing is only possible with special Ditto Max cartridges (hardware limitation, not a lacking feature of `ftape`)
I wasn't able to get the Ditto Max to work with any other device than `/dev/[n]qft0`. I don't know whether this is a feature of the Ditto Max or the Ditto EZ controller I had plugged the Ditto Max into.
Comment You don't need to buy a `Ditto Max Pro` to use the 5/10GB cartridges. With `ftape` there is no real difference between the `Ditto Max` and the `Ditto Max Pro`.

Iomega Ditto 800/3200/2GB/Max/Max Pro Easy (parallel port)

Supported since `ftape−4.00` with the `bpck−fdc` FDC driver.

Mountain FS8000

Reveal TB1400

Reported not to work with kernel 1.3.79 and ftape (no version given) or with kernel 1.2.13 and zftape 1.04 (<colin@colina.demon.co.uk>).
The mentioned `ftape` driver versions are out of date. If you still have such a beast try the more recent versions of the `ftape` driver.

Summit SE 150, SE 250

Tallgrass FS300

If you have a Tallgrass FS300 and an AHA1542B, you need to increase the bus−on / bus−off time of the 1542B. Antti Virjo (`<klanvi@uta.fi>`), says that changing `CMD_BUSON_TIME` to 4 and `CMD_BUSOFF_CMD` to 12 in `linux/drivers/scsi/aha1542.c` will do the trick.

Teac 800

Memorex tape drive backup system

Wangtek 3040F, 3080F

You can always check out the newest list of drives that are recognised by `ftape`, by looking in the file `vendors.h` in the `ftape` distribution.

Although I do not want to endorse one drive type over another, it has been reported that the Colorado DJ−20 drive is rather noisy, when compared to, say, a Conner C250MQ drive ('tis said that the Colorado is 5−10 times as noisy as the Conner drive. Since I have neither, I can't tell for sure).

If you have a drive that works fine, but it is not listed here, or if you have corrections to the above information, please send a mail to the HOWTO maintainer (`<heine@math1.rwth-aachen.de>`).

6.2 Supported special controllers

These dedicated high−speed tape controllers are supported by `ftape`:

- Colorado FC−10, FC−20
- Mountain MACH−2
- Iomega Tape Accelerator II
- 2Mbps controllers (using the i82078−1 fdc)
- Iomega Ditto EZ 4Mbps PnP controller

Colorado FC−10, FC−20

Support for the FC−10 controller has been merged into the `ftape` driver in version 1.12. See the `RELEASE−NOTES` and the `Makefile` files in the `ftape` distribution. Since of version 2.03 of `ftape`, the FC−20 controller will work, but only at 1Mbit/sec (check the Release notes!).

Mountain MACH−2

The support for the MACH−2 controller was added in `ftape−1.14d`.

Iomega Tape Accelerator II

To use the Iomega Tape Accelerator II (**not** to be mistaken as the Iomega Ditto Dash!), use `−DMACH2`, and set the right settings for I/O base, IRQ and DMA. This works (by the empirical testing of Scott Bailey <sbailey@xcc.mc.xerox.com>), with at least `ftape−2.02`.

Iomega Ditto Dash and other 2Mbps controllers

The Iomega Ditto Dash, and all other known 2Mbps controllers, use the Intel 82078−1 chip, which can run at 2Mbps. This is supported properly since `ftape−3.00`.

Iomega Ditto EZ PnP controller

This controller requires the use of e.g. the `isapnptools` package to configure it. You may get it from

http://www.roestock.demon.co.uk/isapnptools/

The controller will cause too many overrun errors when used at the highest possible speed of 4Mbps. Neither Tim Jones <tjones@estinc.com> nor I <heine@math1.rwth−aachen.de> have been able to find but a single system which could run the controller at 4Mbps. 3Mbps seems to be fine.

If you configure the Ditto EZ to use DMA 2 (the DMA channel used by the floppy controller) then your floppy drive will no longer work. It doesn't help to disable the controllers DMA gate (as is the case with other hight speed controllers) so this can't be helped from inside `ftape`.

6.3 Unsupported tape drives

- Some parallel port floppy tape drives still not work. Others do.
- Irwin AX250L / Accutrak 250. (not a QIC−80 drive)
- IBM Internal Tape Backup Unit (identical to the Irwin AX250L drive)
- COREtape light

The Irwin AX250L (and the IBM Internal Tape Backup Unit) does not work the `ftape`. This is because they only support QIC−117, but not the QIC−80 standard (they use Irwin's proprietary ``servoe (Rhomat)'' format). I know nothing about the Rhomat format, nor where to get any info on it. Sorry.

The COREtape light does not accept the initialisation commands, we're feeding it. This pretty much leaves the drive unusable.

6.4 Using an external tape drive with `ftape`

If you have a floppy controller which has a female DB37 connector on the bracket (and some means of delivering power to the drive), you can use it with `ftape`. OK, that sentence was not very obvious. Let's try it this way: Some FDC's (the very ancient one's), have a DB37 connector on the bracket, for connecting to external floppy drives.

If you make a suitable cable from the DB37 connector (on the FDC) to your external tape drive, you can get `ftape` to control your tape drive.

This is because that from a program's view there is no difference between the internal and the external connectors. So, from `ftape`'s point of view, they are identical.

- Pins 20−37: GROUND
- 1: +12 Volt (POWER)
- 2: +12 Volt return (GROUND)
- 3: +5 Volt return (GROUND)
- 4: +5 Volt (POWER)
- 5: 2
- 6: 8
- 7: 10
- 8: 12
- 9: 14
- 10: 16
- 11: 18
- 12: 20
- 13: 22
- 14: 24
- 15: 26
- 16: 28
- 17: 30
- 18: 32
- 19: 34

The power connector is of the "mini" type, sitting on 3.5" floppy drives. The idea appears to be that you plug one of the power connectors from the PSU to this connector on the board. If you want to use just a single cable, you might want to get a 50 wire cable, and use multiple wires for the power lines (and ground, for that matter).

I have received no confirmation from anyone that this works. Let me know your results if you try it.

6.5 PCI motherboards and `ftape`

Unfortunately, some PCI motherboards cause problems when running `ftape`. Some people have experienced that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX machine. If you have such a problem, please read the `README.PCI` file in the `ftape` distribution.

> A floppy disk controller needs the ISA bus DMA controller for its memory transfers. Seemingly the ISA DMA controller doesn't get control over the memory bus often enough on some PCI based systems.

7.Backing up and restoring data

This section describes some simple uses of `tar` and `mt`. Other examples can be found in the `ftape−manual` of the `ftape−doc` package. The `ftape−tools` contains some simple automated `DejaGnu`

> Package for writing automated tests.

test−suites. See section Getting ftape for informations about where to download those additional packages from.

7.1 Writing an archive to a tape

You can use \``tar`', \``dd`', \``cpio`', and \``afio`'. You will need to use \``mt`' to get the full potential of your tapes and the `ftape` driver. For a start I'd recommend using \``tar`', as it can archive lots of directories and let you pick out separate files from an archive. `cpio` creates smaller archives and is more generally more flexible than `tar`, but is missing some features like volume labels. \``afio`' creates backups where each file is compressed individually and then concatenated. This will allow you to access the files ``after'' the point of the error. If you use `gzip`ped `tar` files, all data after the point of the error is lost! (to me, this is a pretty good reason for NOT using compression on backups). The choice of which is most appropriate depends on the situation and the features and malfeatures of each of the packages. I recommend taking a look at each package at reviewing the options that each provides. It's possible that this HOWTO may provide more detail on this subject at some point in the future.

There are more links pointing to backup software at http://www−math.math.rwth−aachen.de/~LBFM/claus/ftape/ in the `software` section of that page.

To make a backup of your kernel source tree using `tar`, do this (assuming you have the sources in `/usr/src/linux`):

```
# cd /usr/src
# tar cf /dev/ftape linux
```

This will not compress the files, but gives you a smoother tape run. If you want the compression (and you've got `tar` 1.11.2), you just include the `−z` flag(*), eg: `` `tar czf /dev/ftape linux' ``

For further instructions on how to use `tar`, `dd` and `mt` look at the man pages and the texinfo files that comes with the respective distributions.

(*) `tar` assumes that the first argument is options, so the `` `−' `` is not necessary, i.e. these two commands are the same: `` `tar xzf /dev/ftape' `` and `` `tar −xzf /dev/ftape' ``

7.2 Restoring an archive

OK, let us restore the backup of the kernel source you made in section Writing an archive to a tape above. To do this you simply say

```
tar xf /dev/ftape
```

If you used compression, you will have to say

```
tar xzf /dev/ftape
```

When you use compression, `gzip` will complain about trailing garbage after the very end of the archive (and this will lead to a `broken pipe' message). This can be safely ignored.

For the other utilities, please read the man page.

7.3 Testing the archive

tar has an option (`−d`) for detecting differences between two archives. To test your backup of the kernel source say

```
tar df /dev/ftape
```

If you do not have the man page for `tar`, you are not lost (yet); tar has a built−in option list: try `` `tar --help 2>&1 | less' ``

7.4 Putting more than one backup on a tape

To put more than one backup on a tape you must have the `mt` utility. You will probably have it already, if you got one of the mainline distributions (eg. Slackware or Debian).

Programs like `tar` and `cpio` generate a single Tape ARchive and know nothing about multiple files or positioning of a tape, it just reads or writes from/to a device. `mt` knows everything about moving the tape back and forth, but nothing about reading the data off the tape. As you might have guessed, combining `tar` or `cpio` with `mt` does the trick.

By using the `nqft[0−3]` (`nftape`) device, you can use `` `mt' `` to position the tape the correct place (`` `mt −f /dev/nqft0 fsf 2' `` means step over two ``file marks'', i.e. `tar` files) and then use `tar` or `cpio` to read or write the relevant data.

The most common use of the non−rewinding device is to append another backup to an existing tape. Here are the specific steps with a little explanation thrown in for good measure.

- Insert a tape into the drive. On some devices this may cause the tape to be rewound.
- Issue an End−of−Tape command to the NON−rewinding device.

  ```
  mt −f /dev/n???? eof
  ```

  The tape should now be positioned at the End−of−Data (**EOD**). The tape won't move unless a program opens the device, closes the rewinding device, removes the device driver from kernel memory (rmmod) or ejects the tape. Using `` `mt eof' `` may be faster on QIC tapes.
- The next tape operation will start at the EOD mark. If you perform a write, it will append a new `file'. If you perform a read it will fail with EOF. The EOD mark on most tape formats is actually two consecutive EOF marks, however, since version 3.xx `ftape` uses the volume table as specified in the `QIC−113` standard to emulate file marks, thus there aren't two consecutive file marks at EOD. Writing the EOF marks is handled by either the device driver or the hardware when a close() is performed.

* Here's where you write the actual data to the tape.
* Here's the important part. **Now rewind the tape**. Both `ftape` caches some information that belongs in the header segments on the tape and update those header segments **only when the tape is rewound**. This caching is necessary because rewinding the tape and updating the header segments takes a conspicuous amount of time. The drawback of this caching is that you will lose information if you have written to the tape and not rewound the device.

7.5 Appending files to an archive

``Is there a way to extend an archive −− put a file on the tape, then later, add more to the tape?''

No. The `tar` documentation will tell you to use `` `tar −Ar' ``, but it does not work. This is a limitation of the current `ftape` driver.

7.6 Mount/unmounting tapes

Since a tape does not have a ``filesystem'' on it, you do not mount / unmount the tape. To backup, you just insert the tape and run your `` `tar' `` command (or whatever you use to access the tape with).

8.Creating an emergency boot floppy for `ftape`

Comment

As of the time of this writing (August 1998) I remember that I have read about several emergency disk sets in the **c.o.l.a** (`comp.os.linux.announce`) news group since the time this section has been written. Some of those packages actually might produce rather sophisticated emergency boot floppy sets. Please check out yourself. I didn't try to create an emergency boot floppy with recent versions of `ftape`.

This section was written by Claus Tøndering `<ct@login.dknet.dk>`.

Once you are the happy owner of a tape drive and several tapes full of backups, you will probably ask yourself this question: ``If everything goes wrong, and I completely lose my hard disk, how do I restore my files from tape?"

What you need is an emergency floppy disk that contains enough files to enable you to boot Linux and restore your hard disk from tape.

The first thing you should do is to read ``The Linux Bootdisk HOWTO'' written by Graham Chapman <grahamc@zeta.org.au>. That document tells you almost everything you need to know about making an emergency floppy boot kit. The paragraphs below contain a few extra pieces of information that will make your life a bit easier when you follow Graham Chapman's procedures:

- You don't really need `/etc/init`, `/etc/inittab`, `/etc/getty`, and `/etc/rc.d/*` on your floppy disk. If Linux doesn't find `/etc/init`, it will start `/bin/sh` on your console, which is fine for restoring your system. Deleting these files gives you extra space on your floppy, which you will probably need.
- Find a small version of `/bin/sh`. They are frequently available on the boot floppies that come with a Linux distribution. This again will give you extra space. I'd suggest `ash`, which is extremely small (approx 62Kbytes), and yet very `bash` compatible.
- The `/etc/fstab` you include on your floppy disk should look something like this:

```
/dev/fd0        /               minix   defaults
none            /proc           proc    defaults
/dev/hda        /mnt            ext2    defaults
```

  Once you have booted from your floppy, give the command:

```
mount −av
```

- Make sure your floppy drive is not mounted when you access the streamer tape! Otherwise you may get the following error message:

```
Unable to grab IRQ6 for ftape driver
```

  This means that you **MUST** load the floppy into a RAMDISK. This has the unfortunate consequence that the programs needed to restore the files from the tape can not be located on a separate floppy disk. You have two options here:

  1. You place `tar` (or `cpio` or `afio` or whatever other backup program you use) on your root floppy disk. (This is where you'll need all the extra space created in the steps above.)
  2. Before you start restoring from tape, copy `tar` (or `cpio` or `afio` or whatever) to your hard disk and load it from there.
- Apart from your backup program, you will probably need `mt` on your root floppy as well.
- Make sure your ftape device (typically `/dev/nqft0`) is present on your boot floppy.
- Finally: **TRY IT OUT!** Of course, I don't recommend that you destroy your hard disk contents to see if you are able to restore everything. What I do recommend, however, is that you try booting from your emergency disks and make sure that you can at least make a file listing of the contents of your backup tape.

9.Frequently Asked Questions

Comment

This is the literal inclusion of the Ftape Frequently Asked questions collection which is maintained by Johan De Wit <jo@correct.nl> and which may be viewed on the web at http://www.correct.nl/~ftape. As Linuxdoc SGML doesn't include sub−sub−sections into the table of contents, I have prepended the word `FAQ` to the sections of the original FAQ document.

This FAQ collection might be slightly out of data as it was collected while version 3.04d of the `ftape` driver was the newest one. If any answer given in the FAQ contradicts any other statement of this HOWTO, then please disregard the answer in the FAQ and drop me (<heine@math1.rwth−aachen.de>) as well as the maintainer of the `Ftape−FAQ` (Johan De Wit <jo@correct.nl>) a note

You might encounter references to the following addresses while reading this document:

- The maintainer of the Ftape FAQ : Johan De Wit <jo@correct.nl>
- The Ftape maintainer : Claus−Justus Heine <claus@momo.math.rwth−aachen.de>
- The Ftape Home Page : <http://www−math.math.rwth−aachen.de/~LBFM/claus/ftape/>
- Mirrors of the Ftape Home Page :

  <http://www.torque.net/ftape/>

  Thanks to Grant R. Guenther <grant@torque.net>

  <http://www.info−systems.com/ftape/>

  Thanks to Jakob Curdes <jc@info−systems.com>

  <http://www.newwave.net/~joshg/ftape/>

  Thanks to Josh Goins <joshg@newwave.net>
- The Ftape HOWTO : <http://sunsite.unc.edu/LDP/HOWTO>
- The Ftape Mailing List : <linux−tape@vger.rutgers.edu>

There is surely quite a lot missing. Please feel free to improve this FAQ. The preferred way of doing this is to post to the Ftape Mailing List in case you have a question that isn't answered here.

Also, if you are already reading the list regularly and have the impression that some questions occur again and again, feel free to send that question and possibly an answer in the format indicated below to the maintainer of the *Ftape FAQ* AND to Ftape Mailing List.

If you make FAQ related postings, then please DON'T FORGET to prepend the word "**[FAQ]**" to the subject of your posting. Please don't add the word "FAQ" to the subject if you post something that isn't related to the FAQ.

That's all for now.

Claus−Justus Heine.

10.FAQ: "Compiling and installing Ftape" related questions !

10.1 What Ftape version should I use?

Always the latest stable version which is _supposed_ to be available from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes and http://www−math.math.rwth−aachen.de/~LBFM/claus/ftape/

At time of this writing the latest stable version is `ftape−4.02`.

<answer from Claus Heine>

10.2 I'm having problems getting my XYZ drive to run under the 2.0.xx kernel with the built−in driver. How do I fix this?

The default version of *Ftape* included in the 2.0.xx kernel sources is 2.08 or 2.09 and is very out of date. Please update the *Ftape drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

10.3 I'm running Linux/SMP and the system just freezes when trying to access the Ftape devices!

You need to add `−D__SMP__` to the `KERNEL_OPT` variable in the file `MCONFIG`. In newer `ftape` versions you only need to uncomment a certain line in the `MCONFIG` file.

<answer from Claus Heine>

10.4 Why does depmod complain about "undefined symbols"?

Ignore the depmod error messages. The problem is that the *Ftape modules* have to be compiled without the version checksum feature (i.e. `CONFIG_MODVERSIONS`) with 2.0. kernels. This causes no problem, even when the modules are used with a kernel that has support for this feature; only that `depmod` wrongly complains about undefined symbols. Ignore the complaints of `depmod` and try to insert the modules despite of these complaints:

```
modprobe zftape
```

If this fails, something is wrong.

<answer from Claus Heine>

10.5 "insmod" says the kernel version is wrong

The `insmod` program can check the kernel version against the version that *Ftape* was compiled for in two ways: It can directly compare the kernel version number recorded in the *Ftape module* against the version of the running kernel, or, if both the kernel and *Ftape* is compiled with versioned symbols, compare the version of the used kernel symbols.

If you have upgraded your version of GCC to v2.7.0 or later, you must recompile the modules utilities with gcc v2.7.x.

Newer versions of `insmod` allows you to "force" insertion of a module into the kernel, even though the version string is incorrect.

<from the Ftape−Howto>

10.6 "insmod" says that kernel 1.2.0 and 1.2.0 differ

Did you remember to apply the `ksyms.c` patch to the kernel? If not, read the `README.linux−1.2` file in the source distribution.

<from the Ftape−Howto>

10.7 Trying to compile Ftape gives me the error "modversions.h: no such file or directory"

The `modversions.h` file is created when the kernel is compiled with the configuration item `CONFIG_MODVERSIONS` turned on. With this option enabled, the file will be created during the `make dep` step.

One more handy tip is that a `make mrproper` will remove `/usr/include/linux/modversions.h`. You will need to reconfig the kernel and do a `make dep` to get the file back.

<from the Ftape−Howto>

10.8 What is this versioned symbols stuff anyway?

When you say `yes' to `CONFIG_MODVERSIONS` during `` `make config' ``, all the symbols exported by the kernel, i.e: the symbols that the loadable modules can "see", are augmented to include a checksum across the types of the call/return parameters. This allows `insmod` to detect whether the definition of a variable or function in the kernel has changed since the time when *Ftape* was compiled.

This ensures a high degree of safety, such that you do not crash the kernel because you used an outdated module with your kernel.

If you enable `CONFIG_MODVERSIONS` in the kernel, make sure you have

```
-DMODVERSIONS -include /usr/include/linux/modversions.h
```

uncommented in the MODULE_OPT line in the *Ftape* Makefile. Conversely, if you do not have CONFIG_MODVERSIONS enabled, make sure you have it commented out.

<from the Ftape−Howto>

10.9 I seem to be getting sftape instead of zftape. When I run "ftmt status" command, I get output that the Ftape docs says corresponds to sftape (/dev/qft0: Invalid argument). Why?

There are (at least) two possible sources of the problem:

- All Ftape−3. versions prior to 3.04 install the modules into

  ```
  /lib/modules/misc
  ```

```
instead of
/lib/modules/uname -r/misc
```

As `modprobe` searches in `/lib/modules/misc/` last there might be an old `ftape.o` module floating around in `/lib/modules/` *uname* `−r/misc` which `modprobe` finds first (`uname −r` stands for the kernel version). Remove the old `ftape.o` module.
- Your kernel has support for *Ftape* compiled in. Reconfigure your kernel without support for *Ftape* (`CONFIG_FTAPE`) and recompile and install it.

<answer from Claus Heins>

10.10 My Ditto DASH/FC−20/Exabyte Accelerator card works under Microsoft Windows, but I get a drive not found type of error in /var/log/messages when trying to use it under Linux.

You are probably trying to use the same IRQ and DMA settings as your on−board FDC. This does not work in versions of *Ftape* prior to 3.03b. Please update the *Ftape Drivers* to the latest from the Ftape Home Page.

<answer from Tim Jones>

10.11 Ftape DMA transfers gives ECC errors

Sadly to say there are some SVGA cards and Ethernet cards that do not decode their addresses correct. This typically happens when the *Ftape* buffers are in the range `0x1a0000` to `0x1c0000`. Somehow, the DMA write cycles get clobbered and every other byte written gets a bad value (`0xff`). These problems are reported to happen with both SVGA and Ethernet cards. We know of at least one (bad?) ATI 16bit VGA card that caused this.

The easiest solution is to put the card in an 8bit slot (it is often not enough to reconfigure the card to 8bit transfers). Moving the *Ftape* buffer away from the VGA range is only a partial solution; All DMA buffers used in Linux can have this problem! Let us make this one clear: This has nothing to do with the *Ftape* software.

<from the Ftape−Howto>

10.12 Help! I'm getting 'dmaalloc() failed' in my syslog file.

You should only see this is you are trying to `insmod` the `ftape.o` module. Try running `swapout` first. It is provided with the standalone *Ftape* source. It doesn't appear in the *Ftape* source that's provided with the kernel.

Here's an example of how you can set your rc.local file to use it.

```
# Install the Floppy Tape Driver
if [ -f /boot/modules/`uname -r`/misc/ftape.o ]; then
   echo Installing ftape for Linux `uname -r`
   swapout
   insmod /boot/modules/`uname -r`/misc/ftape.o
fi
```

Please note that you won't have this type of problem if you compile the *Ftape driver* into the kernel.

<from the Ftape−Howto>

10.13 Syslogd works overtime when running Ftape

The compile−time options `NO_TRACE` and `NO_TRACE_AT_ALL` in *Ftape* control the amount of system logging. Add whichever is appropriate to the `FTAPE_OPT` line in the Makefile and recompile.

<from the Ftape−Howto>

10.14 How do I change the trace−level?

There are three ways you can do this (in order of personal preference).

While we're at it, here are the meanings of the various trace levels.

- 0 Bugs
- 1 + Errors
- 2 + Warnings
- 3 + Information
- 4 + More information
- 5 + Program flow
- 6 + FDC/DMA info
- 7 + Data flow
- 8 + Everything else

1. Using insmod to change trace−level If you are using the modules mechanism to load the *Ftape* driver, you can specify the tracing level as an option to the insmod command.

```
/sbin/insmod ftape.o tracing=<tracing-level>
```

2. Using mt to change trace−level The *Ftape* driver has a hack in it that allows the `fsr` option in `mt` to be used to set the tracing level. `zftape` does not have this hack.

```
mt -f /dev/ftape fsr <tracing-level>
```

   The use of the `fsr` command in `mt` is a *hack*, and will probably disappear or change with time.
3. Recompiling to change trace−level The file `tracing.c` contains a line `int tracing = 3;`. Change the 3 to whatever is appropriate and recompile.

<From the Ftape−Howto>

10.15 I'm having problems with Ftape. I'm using the latest version of Ftape from the Ftape Home Page and believe that I've located a real bug. What should I do?

Check the Ftape Home Page. for an even newer version. Then check the FAQ contained in the that package if your problem is listed there. Next, try to check if the manual that comes with the *Ftape distribution* mentions your problem.

There is no need to read the entire manual, simply check the "Concept Index" if it contains a keyword that might be related to your problem, then jump to the proper location in the manual.

If you are still convinced you've found a bug, then post a general question describing the problem to the Linux−Tape Mailing List , but do not attach your entire *Ftape* error−log. If we've seen the problem before, we'll let you know where the resolution effort stands. If we haven't, the Ftape maintainer will most likely request that you send him the entire *Ftape* error−log (snipped from your system messages file).

<answer from Tim Jones>

11.FAQ: "Using Ftape" related questions !

11.1 How fast is Ftape ?

You can achieve quite respectable backup and restore speeds with *Ftape*: a Colorado DJ−20 and an Adaptec 1542CF controller, has been measured at 4.25Mbyte/min sustained data transfer rate (no compression) across a 70Mbyte tar archive, while comparing the archive on the tape with data on an IDE disk. The speed of *Ftape* is mostly dependent on the data transfer rate of your FDC: The AHA1542CF has a ``post−1991 82077''

FDC, and it will push 1Mbit/sec at the tape drive. If you have an FDC which can only deliver 500Kbit/sec data rates, you will see half the transfer rate (well, roughly).

11.2 When I write to some of my tapes, they seem to spend a lot of time "shoe−shining," or repositioning instead of streaming. Is something wrong with my system?

There has been a few reports of "shoeshining". This is when the tape just seems to run back and forth endlessly. This has been seen on a Jumbo 250 (74407.3051@compuserve.com) and on an Iomega 250 Ditto Insider (tom@opus.cais.com). In the latter case it has been narrowed own to using an ELF Linux and running off a SCSI hard disk (connected to an Adaptec 1542cf). Please contact me if you have an update to this problem.

<from the Ftape−Howto>

Probably not. If you are backing up a large number of < 2K files, you're just going to have to live with it. In this event, the repositions are caused by file system access overhead. If you are backing up a normal system's files, this may be caused by slop or media stretching in the tape cartridge. By simply retensioning the tape, you should see this go away. Try

```
ftmt -f /dev/zqft0 reten
```

to retension the tape. If retensioning doesn't solve this, and it's only happening on certain tapes, it might be wise to replace the tapes in question.

<answer from Tim Jones>

If you use afio as your backup tool you can set it to write a very large number of buffers in one hit by using the −c flag. Make it large enough so that you supply enough data for most of a single end−to−end pass over the tape. For my system, the following streams quite nicely − stopping relatively few times per tape pass on an unloaded system:

```
find /usr/local -xdev -print | afio -o -v -f -b 10240 -c 800 /dev/qft0
```

In my case I'm writing 800 x 10240 bytes per tape write, i.e. about 8MB. haven't experimented that much with these settings − so someone might like to establish more optimal ones.

Presumably other backup utilities could be modified to use a similar technique.

<answer by Michael Hamilton>

GNU tar doesn't use buffering in this way. The commercial backup program "bru" is able to multi−buffer using shared memory; this works only when writing compressed archive with bru (regardless whether you use *Ftape's* builtin compression)

Another way to overcome the problem might be to use more dma buffers in the *Ftape kernel driver* like:

```
mt −f /dev/qft0 setdrvbuffer $((632786))
```

$((632786)) should be expanded by your shell when using a Bourne compatible one. This has a negative impact on the system's memory pool: *Ftape's* dma buffers cannot be used by any other part of the kernel nor by any other application. And kernel memory cannot be swapped out. If you decide to use this kind of multi−buffering then you should unload the driver as soon as it isn't needed any longer.

<answer by Claus Heine>

11.3 Do I have to reboot to the DOS world to format tapes?

Not if you are using the latest version of the *Ftape drivers* from the Ftape Home Page.

To format a QIC−80, TR−1, TR−3, QICWide 3010 or 3020 tape, get the latest version of `ftape` and the latest version of the `ftape−tools` package (from the same location) and read the documentation of the `ftformat` utility which is included in the `ftape−tools` package.

Comment

Do not try to format Ditto 2GB tapes.

Comment

Do not try to format Ditto Max or Max Pro tapes.

<answers from Tim Jones and Claus Heine>

11.4 Is it possibly to format Ditto 2GB tapes with ftape?

It isn't possible to format `Ditto 2GB` tapes with `Ditto 2GB` tape drive, and it isn't possible at all to re−format `Ditto 2GB` tapes in a way that they still can be used by a `Ditto 2GB` tape drive.

This is a hardware limitation of the `Ditto 2GB` tape drive. It can't be helped at the software level, i.e. it isn't `ftape's` fault.

11.5 Is it possibly to format Ditto Max or Max Pro tapes with ftape?

No, the `Ditto Max` can't format tapes.

This is a hardware limitation of the `Ditto Max (Pro)` tape drive. It can't be helped at the software level, i.e. it isn't `ftape's` fault.

11.6 Ftape detects more bad sectors than DOS on QIC−3020 tapes

If you look at the difference, you will notice that *Ftape* always detects 2784 sectors more than DOS.

The number that *Ftape* reports is correct (of course `:-)`. Each correctly formatted QIC−3020 tape has 2784 sectors at fixed positions that are marked in the bad sector map. To quote from the specs:

> Tracks 5,7,9,11,13,15,17,19,21,23,25 and 27 within 4 segments of either EOT or BOT are prone to increased error rates due to hole imprints. Therefore, these regions shall be mapped as bad at format time and entered in the bad sector map by indicating that all sectors within the identified segments are bad.

This gives 12 tracks 2 4 segments 29 sectors == 2784 sectors.

So *Ftape* choose to report the real number of sectors that cannot be used on the tape, while DOS gives a more optimistic number giving a better indication of tape quality. (*Ftape's* behavior might change in the future to detect correct formatting and display the separate numbers. It has rather low priority though).

QIC−3010 are alike QIC−3020 tapes regarding this.

<from the Ftape−Howto>

11.7 Is it ok that I'm not hearing the tape move when I do a fsf or a bsf with mt?

Yes. The driver merely updates an internal counter when those commands are issues. The tape should move to the proper location on the next read or write access to the tape drive.

<from the Ftape−Howto>

11.8 Why does my XYZ backup program complain about "Invalid argument" errors?

`zftape` requires the data to be written in multiples of a fixed minimal block size. This is a very usual behavior for a tape device. There are three ways to get rid of those errors:

- set *Ftape's* block size to the block size used by the backup program. The example below works for "afio":

  ```
  mt -f /dev/qft0 setblk 5120
  ```
- If you don't want to use *Ftape's* built in compression you can also use

  ```
  mt -f /dev/qft0 setblk 0
  ```

  to switch *Ftape* to variable block size mode and be able to write the data in arbitrary portions to the tape (BUT: the builtin compression doesn't work with this setting). When you intend to use "KBackup" then this is the only way to make it work together with *Ftape* (it _may_ work, don't know if it does)
- tell your backup program about *Ftape's* default block size of 10k (which is also the default of GNU tar). For "afio" you can use the following command line switch:

  ```
  afio -b 10k ...
  ```

You may want to read the section "Tape blocks" of the manual (use its "Concept index" to directly jump to that section)

When using GNU tar's builtin compression with GNU tar versions prior to tar−1.12 one needs to run tar with the `--block-compress` switch to `re-block` the output to the tape. Otherwise tar will compress the data it reads, and write it in arbitrary portions to the tape.

```
Example :

tar -czvf /dev/qft0 --block-compress /etc
```

WARNING: One shouldn't use tar's builtin compression with large backups as it makes the entire data stream one huge compressed block. If such archives are corrupted right at the beginning it will be very difficult to recover.

<answer by Claus Heine>

11.9 I/O errors and FDC − some explanations.

When you get next messages, this could be interesting for you !

- fdc−io.c (ft_handle_perpend) − Your FDC does not support QIC−3020.
- Cannot write to /dev/qft0: I/O error

The explanations:

"FDC" menas "Floppy Disk Controller". The problem is that your floppy disk controller must be able to support something that is called "perpendicular mode" to be able to read and write QIC−3020/QIC−3010 cartridges (i.e. TR−3 cartridges). To my knowledge all FDCs that are capable of at least 1Mbit/sec data transfer rate also support "perpendicular mode" ("perpendicular" refers to the direction of magnetization of the ferro−magnetic particles on the tape).

This means that you need to purchase another FDC. Either look around some computer stores and ask for an IO controller cards that is able to support 2.88 Mb floppies (which imlies 1Mbit data transfer rate and perpendicular mode).

Or get one of the so called "high speed" controllers that even support 2Mbit/sec data transfer rate. Those controllers are based on an Intel 82078 FDC. Iomega sells such a card under the name "Ditto Dash". I think Exabyte sells their 2Mbit controllers separately, too, whereas Seagate ships its TR−3 drives (i.e. the TST−3200) together with such a controller.

<answer from Claus Heine>

11.10 Why do I get "/dev/qft0: No such device" errors?

I assume that the following is the problem: The *Ftape* module is loaded OK into the kernel:

```
/usr/src/ftape-3.03b-970603# lsmod
 Module         Pages    Used by
 ftape             22           0
```

but then this happens:

```
$ ftmt -f /dev/qft0 status
ftmt: /dev/qft0: No such device
```

Solution You need to load the *zftape.o* module as well. With Ftape−3. the *ftape.o* module doesn't implement the VFS interface. This is done by *zftape.o*.

<answer from Claus Heine>

11.11 I get "device busy" when I make multiple backups on a tape using some script.

The "device busy" messages can only occur while the *Ftape devices* are still held open by some program. As soon as the close() system call has completed the busy flag is cleared. May be "bru" or some other program has still forked off a child that dies delayed?

Yes, this will reproduce the problem, it seems:

```
tar -cvvzf /dev/nqft0 --block-compress ; mt rewind
```

You can skip the "−−block−compress" if using the most recent version of GNU tar.

However, this is not a bug of *Ftape*. It seems that the parent tar process exits before its child has closed the tape device. I know, however, from hacking the tar code ages ago, that tar properly waits for its parent to die.

However, the busy message simply means that the "busy" variable is still held at 1 (zftape/zftape−init.c). And this simply means that there still is a process hanging around that holds the tape device open.

I think I have it (only for the case of tar 'cause I have the source code.

If on uses tar with compression, then it forks a child which will become the compressor bei execing "gzip" or whatever. Before the call to execlp() the child will fork off a grand child of its parent tar. That grandchild will do the actual tape I/O.

```
tar - fork() - write to child tar
        |
      child tar - fork() - gzip (will pipe to grand child tar)
                    |
                  grand child tar - open archive.
```

Now, parent tar only waits for its child to die. gzip surely doesn't wait for the grand child as the gzip is a result of an execlp().

What I don't know is whether the grand child should be implicitly waited for by the parent tar, or if the wait() function also waits for grand childs.

But this seems to be the problem: the parent tar already has exited while its grandchild still is busy closing the archive. One hardly will notice this problem if the close() happens fast (i.e. regular files, block devices, also other tape devices?), but it isn't a bug in *Ftape*, but either in the backup programs or in the kernel or maybe libc exit code.

Don't know if the considerations above also apply to bru. If there is no grandchild and the parent process properly waits for its childs then there shouldn't be a problem.

<answer from Claus Heine>

11.12 How do I "..." with tar?

These are really `tar` questions: Please read the `man` page and the `info` page. If you have not got it either, try

```
tar  --help 2>&1 | less
```

If your version of `tar` is v1.11.1 or earlier, consider upgrading to v1.11.8 − This version can call `GNU zip` directly (i.e.: it supports the `−z` option) and has an elaborate help included. Also, it compiles right out of the box on Linux.

<from the Ftape−Howto>

11.13 What block−size should I use with tar ?

When using compression, and in all general, it can be a benefit to specify to `tar`, that it should block the output into chunks. Since *Ftape* cuts things into 29Kbyte blocks, saying `` `−b58 ``' should be optimum.

"Why 29Kbyte?", I hear you cry. Well, the QIC−80 standard specifies that all data should be protected by an Error Correcting Code (ECC) code. The code specified in the QIC−80 standard is known as a Reed−Solomon (R−S) code. The R−S code takes 29 data bytes and generates 3 parity bytes. To increase the performance of the ECC code, the parity bytes are generated across 29 1Kbyte sectors. Thus, *Ftape* takes 29Kbytes of data, adds 3Kbytes of ECC parity, and writes 32Kbytes to the tape at a time. For this reason, *Ftape* will always read and write 32K byte blocks to be able to detect (and correct) data errors.

If you are curious, and wish to know more, look in the `ecc.c` and `ecc.h` files, for an explanation of the code and a reference to a textbook on Reed−Solomon codes.

<from the Ftape−Howto>

11.14 Where can I find the tar/mt/cpio/dd binaries − sources − manpages?

All of these tools have been developed by the GNU project, and the source (and man page) can be fetched from just−about any ftp site in the world (including `ftp.funet.fi`, `tsx−11.mit.edu`, and `sunsite.unc.edu`). In any case they can be fetched from the official GNU home site: `prep.ai.mit.edu [18.71.0.38]:/pub/gnu`. The latest versions (as of September 12 1996) are:

```
cpio:   2.4.2 (cpio−2.4.2.tar.gz)
dd:     3.13 (fileutils−3.13.tar.gz)
mt:     2.4.2 (cpio−2.4.2.tar.gz)
tar:    1.11.8 (tar−1.11.8.tar.gz)
gzip:   1.2.4 (gzip−1.2.4.tar.gz)
```

They all compile out of the box on Linux `v1.0.4` / `libc v4.5.19` / `gcc v2.5.8`.

<from the Ftape−Howto>

11.15 If I use tapers compression, is it a bad idea to use the compression with zftape, or would it be better to not use tapers compression, and let zftape do it?

It is not bad as such to compress data twice (which would be the case when using tapers compression together with *zftape's* compression) but it doesn't make any sense. You won't gain much further compression, but only waste CPU cycles.

Tapers compression should be quite safe, as taper compresses single files; in contrast to *tar −czf* ... which makes the entire data stream a large compressed block of data, which is really a bad thing with serious backups as a single bad byte at the beginning of the archive can make the entire archive unusable, well, it will be at least quite difficult to recover.

<Answer from Claus Heine>

11.16 How does zftape compression compare to say gzip −9?

gzip −9 is better (i.e. one gains higher compression). *zftape's* compression is comparable with the Un*x *compress* program, but should be faster, and is faster than *gzip*.

<Answer from Claus Heine>

11.17 I don't trust compression, but hear that the sftape interface is going away. What should I do?

Use the *zftape interface*, but don't load the *zft−compressor module*. The device then becomes `/dev/qft0`.

<answer from Tim Jones>

11.18 Ftape says "This tape has no 'Linux raw format"

You get this complaint if you haven't *erased* your freshly formatted tape. This is because *Ftape* expect a "magic header" on the tape, to be able that it is allowed to interpret the header segment in its own way (eg: file marks). To remove the problem, say

```
mt −f /dev/nftape erase
```

<from the Ftape−Howto>

11.19 Can I exchange tapes with someone using DOS?

No. The DOS software conforms to the QIC−80 specs about the layout of the DOS filesystem, and it should(?) be a small problem to write a program that can read/write the DOS format. In fact, I'd bet that creating a nice user interface would be a bigger problem.

<From the Ftape−Howto>

11.20 How does `mt eom' work when you've started overwriting a tape in the middle?

(EOM is "End Of recorded Media", the position right after all data already recorded to the tape)

One cannot use tape "files" like files on an ordinary file system.

In principle, a tape doesn't allow anything but appending new data at EOM. However, if one positiones just in the middle of the already recorded data AND starts writing, then the driver first deletes all following files (thus moving the EOM to the actual position) and then starts writing.

Thus, the new EOM after finishing the write process, is then after the newly recorded data.

One of the consequences of the above is, of course, that writing to the tape in the middle of the already recorded area, is destructive in the sense, that it not only overwrites the "file" the tape is positioned at, but also deletes all following files.

<from the Ftape−Howto> <Answer from Claus Heine>

11.21 When I made backups before using taper, under the 2.0.29 ftape my drive didn't support fsf, under the new zftape it does, why would this be, and what exactly is fsf ?

It probably didn't work before because you didn't use a

```
mt −f /dev/rft0 erase
```

before writing data to the cartridge. THIS ISN'T necessary any more.

But, hey, what does *mt fsf*? Tape drives don't store files in the sense that you can use

```
cp somefile /dev/my_what_ever_tape
```

or be able to mount the tape drive like you could mount a harddisk. You can't do nothing with a tape drive but write data to it in a sequential manner.

As this is quite inconvenient, somebody invented something which is known under the name *file mark* or *eof mark* (eof == End Of File). Those marks don't separate files that have been backed up to the tape device, but only separate blocks of data (whatever data that might be).

Normally, the kernel tape device drivers take care of writing file marks when the tape device is closed, i.e.

```
tar −cf /dev/nqft0 /bin
tar −cf /dev/nqft0 /etc
mt −f /dev/nqft0 rewind
```

would result in a backup of all files under */bin* and */etc*. When the first *tar* finishes, the kernel driver will take care of writing a file mark to the tape at the the current tape position, and when the second *tar* process has finished, another file mark is written to the tape cartridge at that position.

Now, the sense of those file marks is, that it is possible to skip between different archives on the tape more quickly than would be possible with reading the data back.

The commands to do that are:

mt fsf

fast skip to the next file mark towards EOT (End Of Tape)

mt bsf

fast skip to the next file marks towards BOT (Begin Of Tape)

Thus, to extract the second archive in the example above, one doesn't need to read the first archive back, but can proceed as follows:

```
mt −f /dev/nqft0 rewind
```

```
mt −f /dev/nqft0 fsf
tar −xvf /dev/nqft0
```

<Answer from Claus Heine>

11.22 What exactly is the difference between ftape, and zftape?

When *Ftape* was young there were two versions of the floppy tape driver, one of them was called *zftape* because of its built−in user−transparent on−the−fly compression. Whether such a thing is a feature or a bug ('cause this needn't be done in kernel space) is another question. However, the ioctl interface and file mark handling provided by *zftape* was much better and had less bugs. And *zftape* allows to use floppy tape cartridges with different OS. Well, you can't exchange data, but *zftape* won't overwrite volumes created by your Windoze program, and vice versa.

Nowadays, *Ftape* is name of the entire floppy tape driver package AND *ftape.o* is the file−name of the kernel module that implements the low−level hardware support. *zftape* has ceased to exist as a separate package, but the new *Ftape* versions (since ftape−3.00) contain a *zftape.o* module that needs to be loaded on top of *ftape.o* (i.e. you need to load BOTH modules to be able to access your floppy tape drive) and implements the file system interface and the advanced (?) features of the previous verions *zftape*.

<Answer from Claus Heine>

11.23 What is the difference between a rewinding, and non rewinding drive?

Well, the rewinding tape devices rewind the tape to BOT (Begin Of Tape) when the device is closed, i.e.

```
tar −cvf /dev/qft0 /bin
```

will rewind the tape cartridge when the tar job has finished. In contrast,

```
tar −cvf /dev/nqft0 /bin
```

will NOT rewind the tape cartridge and leave the tape R/W head at its current position.

Rewinding devices should be used when performing a single backup, non−rewinding devices can be useful when doing multiple backups as one doesn't need to space to EOM (End Of recorded Media) before appending another archive.

Non−rewinding devices MUST be used when sending any of the tape motion command to the tape drive, such as

```
mt −f /dev/nqft0 fsf
```

, because when the *mt* process finishes then the tape device is closed which would result in rewinding the cartridge with the rewinding devices.

<Answer from Claus Heine>

11.24 Can someone tell me how to use mt to rewind my TR−3 drive one record using zftape record, so I can verify it?

Well, it depends. If the tape is still positioned inside the volume just written, "mt bsf 1" (or equivalently "mt bsf") will backspace just to the beginning of that volume (this is how "tar −−verify" works). If the tape is already positioned AFTER the filemark that marks the end of the last written volume, then you need to issue "mt bsf 2"

The logic behind this is as follows: "MTBSF count" backspaces over count file marks, stops, and then positions on the `EOT` side of the last skipped file mark. This means, an "mt bsf 2" will position right at the beginning of the previous volume.

<answer form Claus Heine>

11.25 By non−rewinding, they mean that it doesn't automatically rewind, correct? It doesn't mean that under no circumstances it will rewind, right? I tried using /dev/zqft0, and it instantly rewinds the tape.

You are right: auto−rewind means, the tape is rewound when the tape device is closed, non−rewinding means, the tape isn't automatically rewound when the tape device is closed (but you can, of course, use the tape motion commands BSF/FSF etc. to position the tape head at every position you like).

<answer form Claus Heine>

11.26 What is the difference between what mt considers a record and what it considers a file?

A record is the minimal amount of bytes that will be accepted by the tape in one read/write operation (except in "variable block size mode" where it just should be the amount of data actually written in a single write operation??).

For *zftape* every read and write access has to be a multiple of a fixed block size (fixed, but tunable with `MTSETBLK`). This block size is a "tape record" (as mentioned in the GNU mt man page and defaults to 10kb for *zftape*.

A "file" (in the sense of the mt man page) is a, well, misleading terminus. What is meant is an area of the tape between two file marks. This is not a file like a file on the file system, in the sense that it could have a name, file access modes, could be moved or copied with cp, mv, rm etc.

Instead, It simply is the area of the tape that was recorded in one backup session, its end is marked by a tape file mark, and its beginning is delimited by either `BOT` or the file mark of the previous tape "file". That tape "files" are the things that can be skipped with the `MTBSF/FSF` commands.

<answer form Claus Heine>

11.27 Reusing tapes with zftape without reformatting the tape.

We try to answer the followong questions :

- Is there a good way to erase, as in remove the data or at least the volumes from a tape, without reformating?
- Can you overwrite the last volume on a tape with making a mess out of it?
- Can you overwrite the last several volumes without making a mess?
- Can you delete the last volume?

If you want to "erase" an entire cartridge, then simply do:

```
mt −f /dev/qft0 erase
```

This will erase the volume table (i.e. the "file marks").

Pre−ftape−3.x releases of zftape and ftape used to allow overwriting of already existing volumes on a cartridge. I have removed this feature as it was reported that it already has caused data−loss with some backup programs.

If you indeed need to remove some volumes on the tape then you should use the

```
vtblc
```

program that comes with the `ftape−tools` package which can be down−loaded from the same locations as the `ftape` kernel driver package. Please refer to the documentation which is contained in the `ftape−tools` package for more information.

If you simply want to reuse old tapes, then it suffices to do

```
mt rewind
```

If the tape is at BOT (Begin Of Tape) then every write access to the tape will silently erase all file marks and overwrite the data already existing on the tape.

<answer by Claus Heine>

11.28 This script implements a simple contents listing for the zftape package using the "MTIOCVOLINFO" ioctl.

Here is as little perl/bash script that lists the contents of a cartridge using the *zftape* specific "volinfo" ioctl. Hope this shows how to handle this kind of stuff.

What it basically does is the following:

1. Rewind the cartridge
2. Issue the volinfo command:

    ```
    claus@thales:~$ mt volinfo
    file number          = 1
    block size           = 10240
    physical space used  =  522.0 kilobytes
    real size of volume  =  520.0 kilobytes
    ```

    Parse the ouput and place the values in appropriate variables
3. Skip to the next volume with "mt fsf"
4. Exit if this gives an error (EOD), otherwise "goto 2)"

The Perl Script

```
#!/usr/bin/perl
#
#     Copyright (C) 1997 Claus−Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
```

```
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

$version = <<EOT;
listtape−1.0 −− a perl script to list the contents of a floppy tape cartridge
under Linux using the zftape driver

RCS \$Revision: 1.2 $
RCS \$Date: 1998/08/30 13:44:03 $
EOT

$tapedev = "/dev/tape";
$usage = <<EOT;
Usage: listtape [options ...]

Mandatory or optional arguments to long options are mandatory or optional
for short options too.

−f, −−file=FILE       Tape device to use. Default is  "/dev/tape".
−h, −−help            Print this help.
−?                    Same as '−h'.
−−usage           Same as '−h'.
−V, −−version         Print version information.

Author: Claus−Justus Heine <claus\@momo.math.rwth−aachen.de>
EOT

while ($ARGV[0] =~ /^−/) {
$_ = shift;
if (/−−file/) {$_ = shift; $tapedev = $_; next;}
if (/−f/) {$_ = shift; $tapedev = $_; next;}
if (/−−help/) { print $usage; exit 0; }
if (/−h/) { print $usage; exit 0; }
if (/−−usage/) { print $usage; exit 0; }
if (/−\?/) { print $usage; exit 0; }
if (/−−version/) { print $version; exit 0; }
if (/−V/) { print $version; exit 0; }
die $usage;
}

&open_tape($tapedev, "status");
while(<FTMT>)
{
$online = 1 if (/.*online.*/);
}

if (! $online) { die "No cartridge present.\n"; }

&mtop($tapedev, "rewind");
```

```
printf "%11s%12s%20s%20s\n",
"file number", "block size", "volume size", "tape space";

while (1)
{
&open_tape($tapedev, "volinfo");
while (<FTMT>) {
if (/^file number\s*=\s*([0-9]*)$/) { $filenumber = $1; }
if (/^block size\s*=\s*([0-9]*)$/) { $blocksize = $1; }
if (/^physical space used\s*=\s*([[0-9]*.*)/) { $rawsize = $1; }
if (/^real size of volume\s*=\s*([[0-9]*.*)/) { $size = $1; }
}
close(FTMT);
if (&mtop($tapedev, "fsf 1") != 0) {
&mtop($tapedev,"rewind");
print "\nRemaining space: $rawsize\n";
print "Tape block size: $blocksize\n";
exit 0;
}
printf "%6d          %5d  %20s%20s\n",
    $filenumber, $blocksize, $size, $rawsize;
}

sub mtop
{
local ($tape, $operation) = @_;
local ($exitval);
system "ftmt -f $tape $operation > /dev/null 2>&1";
}

sub open_tape
{
local ($tape, $operation) = @_;
local ($command);

$command = "ftmt -f " . $tape . " " . $operation . " |";
open(FTMT, $command) || die "Couldn't open $command -- $!\n";
}
```

The Bash Script

```
#! /bin/bash
#
#     Copyright (C) 1997 Claus-Justus Heine
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2, or (at your option)
# any later version.
#
# This program is distributed in the hope that it will be useful,
# but WITHOUT ANY WARRANTY; without even the implied warranty of
# MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
# GNU General Public License for more details.
#
```

```
# You should have received a copy of the GNU General Public License
# along with this program; see the file COPYING.  If not, write to
# the Free Software Foundation, 675 Mass Ave, Cambridge, MA 02139, USA.
#
#   This script implements a simple contents listing for the zftape
#   package using the MTIOCVOLINFO ioctl.
#

#
# insert better option parsing here
#
TAPEDEV=${1-/dev/tape}

if ! echo $TAPEDEV | grep "/dev/n"
then
TAPEDEV=/dev/n$(basename $TAPEDEV)
fi

if ! [ -c $TAPEDEV ]
then
echo $TAPEDEV is not a character device!  1>&2
exit 1
fi

if ! mt -f $TAPEDEV rewind
then
echo Could not rewind $TAPEDEV - no cartridge present?  1>&2
exit 1
fi

echo -e "\nContents of $TAPEDEV:\n"

printf "%11s%12s%20s%20s\n" "file number" "block size" "volume size" "tape space"

trap "rm -f /tmp/$0.$$" exit

while true
do
if ! foo=$(mt -f $TAPEDEV volinfo |cut -f 2 -d =)
then
echo $TAPEDEV doesn\'t seem to be a floppy tape device 1>&2
exit 1
fi
#
# "echo foo | read foo" will not work as the "read foo" is executed in
# another shell.
#
echo $foo > /tmp/$0.$$
read file blksz used usedunit size sizeunit < /tmp/$0.$$
if ! mt -f $TAPEDEV fsf 1 > /dev/null 2>&1
then
echo -e "\nRemaining space: $used $usedunit"
echo -e "Tape block size: $blksz"
if ! mt -f $TAPEDEV rewind
then
    echo Rewind of $TAPEDEV failed 1>&2
    exit 1
fi
exit 0
fi
printf "%6d          %5d  %20s%20s\n"\
$file $blksz "$size $sizeunit" "$used $usedunit"
```

```
done
```

<answer from Claus Heine>

12.FAQ: "Tape and Drivers" related questions !

12.1 What are good makers of Travan tapes?

I was the UNIX Product Manager at Archive Corp (Prior to the Conner/Seagate mess) and we performed extensive tests of tape media for compatibility certification, including retentivity, flaking and length consistancy. Based on the results of the tests, we selected the best of these certified manufacturers' products to private label as our own media. Here is the order in which we selected vendors up through 1995 (when I lost contact with the ATI group):

QIC

3M (now known as Imation)
QMaxell/Sony (tie)
(BTW − Iomega uses Sony private−labelled media)

4MM

Fuji
Maxell/Sony (tie − is this a trend?)

8MM

Fuji/Exabyte − which we believed to be OEM'd Fuji (tie − so much for trend!)
Sony
Maxell

DLT

Maxell
Sony

Otherwise, we had entries in our search from other vendors which were generally a private−labelled version of one of the major labels above. The exceptions were Verbatim and DIC. Both of these manufacturer's media had fall−out rates and length discrepancies that were so high that we would not certify them and even warned customers about them indicating that we could not offer any sort of guarantee that they would get a good backup using the media from these manufacturers.

In addition, since coming to EST, I've found that Verbatim media is still not worth the money saved in purchasing it. We have 11 of their TR−Extra and QIC−Extra (QICXL) tapes that were useless after fewer than 20 passes each.

While this is my personal opinion, it is based on over 9 years of experience with this very question, I strongly recommend Imation/3M media for QIC/Travan user, Fuji media 4MM users, Exabyte/Fuji for 8MM and DEC labelled media for DLT users.

<answer from Tim Jones>

12.2 Where can I obtain the QIC standards?

If you wish to help developing *Ftape*, or add some utility (e.g. a tape formatting program), you will need that appropriate QIC standards. The standard(s) to get is: QIC−80, −117, −3010, and 3020. QIC−117 describes how commands are sent to the tape drive (including timing etc), so you would probably never need it. QIC−80/3010/3020 describes higher level part, such as tape layout, ECC code, standard filesystem. You can get the QIC standards from the following address:

```
Quarter Inch Cartridge Drive Standards, Inc.
311 East Carrillo Street
Santa Barbara, California 93101
Phone: (805) 963−3853
Fax:   (805) 962−1541
```

Note: They are registered as `Freeman Associates, Inc' in the phone book.

<from the Ftape−Howto>

12.3 Is the Iomega Ditto 2GB drive supported?

Yes, if you are using version `ftape−3.x` or later of the *Ftape drivers* from the Ftape Home Page or from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes.

<answer from Tim Jones>

As the Ditto 2GB is a Tr−3 tape (though it can only store 1GB instead of the 1.6GB you get with a regular Tr−3 drive) you need an FDC (FDC means: Floppy Disk Controller) that is capable of at 1Mbit/sec transfer rate. You don't need to worry about this if you have an accellerator card (i.e. the Ditto Dash controller). Otherwise try to purchase an FDC that claims to be capable of driving 2.88Mb floppies because this implies that the FDC is capable of 1Mbit transfer rate.

Ftape prints the maximum data rate of the FDC to the kernel log files like this:

```
ftape-ctl.c (ftape_init_drive) - Highest FDC supported data rate: 500 Kbps.
```

<answer from Claus Heine>

12.4 Is the Iomega Ditto Max drive supported?

Yes, if you are using version `ftape−4.02` or later of the *Ftape drivers* from the Ftape Home Page or from ftp://sunsite.unc.edu/pub/Linux/kernel/tapes.

<answer from Claus Heine>

12.5 Is the Iomega Ditto Max Pro drive supported?

Yes. But if you want to use the 5GB (10GB with compression) cartridges you don't need it. With `ftape` there doesn't seem to be any difference between the `Ditto Max` and the `Ditto Max Pro`.

<answer from Claus Heine>

13.FAQ: Miscellaneous !

13.1 How to subscribe to the Ftape Mailing List?

You can subscribe to that list by sending mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
subscribe linux-tape
```

in its body. Please store the answer you get from majordomo in a safe place because it contains instructions how to `UNSUBSCRIBE` from the mailing list.

<answer from Claus Heine>

13.2 How to un−subscribe from the Ftape Mailing List?

Send mail to

```
majordomo@vger.rutgers.edu
```

with the single line

```
unsubscribe linux-tape MY@EMAIL.ADDRESS
```

where `MY@EMAIL.ADDRESS` has to be replaced by the the email address that you used when subscribing to the list. Note that you must have received an email with instructions how to unsubscribe from the mailing list at the time you subscribed to it.

<answer from Claus Heine>

13.3 Links to related information.

<http://www.uwsg.indiana.edu/usai/library/backups.html>

More links wanted !!!

14.Debugging the `ftape` driver

14.1 The kernel/`ftape` crashes on me when I do `...' − is that a bug?

No, that is a feature

Seriously, reliable software do not crash. Especially kernels do not or rather **should** not crash. If the kernel crashes upon you when you are running `ftape`, and you can show that it is `ftape` that is messing things up, regard it as a Bug That Should Be Fixed. Mail the details to the maintainer

(<heine@math1.rwth−aachen.de>) and to the tape list.

14.2 OK, it's a bug ...ehhh... feature − How do I submit a report?

First, make sure you can reproduce the problem. Spurious errors are a pain in the ass, since they are just about impossible to hunt down `:−/` This is a quick check list:

- Kernel version, and patches applied
- `ftape` version
- tape drive model / manufacturer
- Expansion bus type (EISA, ISA, PCI, or VL−bus)
- What you did to expose the problem
- What went wrong on your system.
- Do not delete the kernel and the `ftape.o` file. I might want you run try some patches out or run a different test on your system.

Increase the tracing level to 4 or 5 and run the command that caused problems again (don't do it if your fear that you loose data or damage your hardware, there is absolutely no warranty for neither data loss nor hardware damage caused by `ftape`, remember this). Increasing the trace level beyond 5 probably doesn't make any sense as it affects the timing of the driver in a way that it doesn't work well any more. Get the tracing data from the kernel log or `/proc/kmsg`, depending on where you harvest your error messages. Try to look at what `ftape` spews out at you. It may look in−comprehensible to you at first, but you can get valuable information from the logfile. Most messages have a function name prepended, to make it easier to locate the problem. Look through the source, don't just cry ``WOLF!'', without giving it a try. If your version of the kernel (or `ftape` for that matter), is ``old'', when compared to the newest version of the kernel, try to get a newer (or even the newest) kernel and see if the problem goes away under the new kernel. When you post your problem report, include the information about ftape version, kernel version, expansion bus type (ISA, VL−bus, PCI or EISA), bus speed, floppy controller, and tape drive. State exactly what you did, and what happened on your system. Some people have experienced that `ftape` would not run in a PCI based box, but ran flawlessly in a normal ISA based 386DX machine (see section Getting PCI motherboards to work with <tt/ftape/ on PCI machines above)

Also, please think of the poor souls who actually *pay* the their Internet access (like me): avoid posting a (huge) log from the `ftape` run, without reason. Instead, you could describe the problem, and offer to send the log to the interested parties.

Send your bug report to <linux−tape@vger.rutgers.edu>. You might also want to mail the bug to <heine@math1.rwth−aachen.de>.

15.Contributions

The following is a list of notable folks that have contributed to `ftape`'s HOWTO document. This is a recent addition added by someone coming in midstream. My sincerest apologies if I've inadvertently left someone important off the list. You can view anoterh attempt to collect such kind of information at Ftape's Hall of Fame

`Johan De Wit` <jo@correct.nl>: The maintainer of the Ftape FAQ.

`Kevin Johnson` <kjj@pobox.com>: The previous maintainer of the `Ftape−HOWTO`

`Kai Harrekilde−Petersen` <khp@dolphinics.no>: The previous maintainer of `ftape` and the HOWTO.

`Andrew Martin` <martin@biochemistry.ucl.ac.uk>: Many additions to the HOWTO.

`Bas Laarhoven` <bas@vimec.nl>: The original author of `ftape`.