

**xf**

**COLLABORATORS**

	<i>TITLE :</i> xf		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 25, 2024	

**REVISION HISTORY**

NUMBER	DATE	DESCRIPTION	NAME

# Contents

<b>1</b>	<b>xfx</b>	<b>1</b>
1.1	xfx user (friendly) guide . . . . .	1
1.2	disclaimer . . . . .	1
1.3	distribution . . . . .	1
1.4	introduction . . . . .	2
1.5	system requirements . . . . .	2
1.6	installation . . . . .	3
1.7	usage . . . . .	3
1.8	readdir notes . . . . .	4
1.9	read notes . . . . .	4
1.10	ql notes . . . . .	5
1.11	spectrum notes . . . . .	5
1.12	msdos notes . . . . .	6
1.13	archimedes notes . . . . .	6
1.14	amiga notes . . . . .	7
1.15	minix notes . . . . .	7
1.16	macintosh notes . . . . .	7
1.17	cpm notes . . . . .	8
1.18	Workbench notes . . . . .	8
1.19	ZIP drive notes . . . . .	9
1.20	x-code usage . . . . .	9
1.21	other bits and pieces . . . . .	10
1.22	todo . . . . .	11
1.23	acknowledgements . . . . .	11
1.24	the author . . . . .	12
1.25	bugs . . . . .	12
1.26	history . . . . .	13

# Chapter 1

## xfs

### 1.1 xfs user (friendly) guide

xfs - XFileSystem © 1998 frans (francis swift)

user guide

[disclaimer](#)

[distribution](#)

[introduction](#)

[system requirements](#)

[installation](#)

[usage](#)

[todo](#)

[acknowledgements](#)

[the author](#)

[bugs](#)

[history](#)

### 1.2 disclaimer

This program is provided "as is" without warranty of any kind, either expressed or implied, including, but no limited to, the implied warranty of fitness for a particular purpose. The entire risk as to the results, reliability and performance of this program is assumed by you.

### 1.3 distribution

This program is freely distributable. Distribution is allowed if the following conditions are met:

1. Program and documentation must not be changed in any way,
-

except for archiving with an archiver, for which the corresponding unarchiver will work on any CPU type used in Amigas at the time of distribution.

2. Program, documentation and the icons for the program and the documentation must be distributed together.
3. The file names of the program, documentation and icons must not be changed. However, you ARE allowed to change the default tool of the documentation icon. You are also allowed to snapshot the icons if you really need to.
4. Only a nominal and small fee for copying is charged, unless some sort of media is supplied with xfs, in which case it is allowed to add the cost of that media to the charge.
5. xfs or parts of it must not be sold in combination with or as part of commercial software without my written permission.

## 1.4 introduction

XFileSystem (xfs) is a package consisting of a filesystem (xfsd) and an exec device (fd.device) that enables your amiga to recognise multiple disk types.

Basically I had written two handlers (QL and Spectrum), and had nearly completed another (an 8.3 style PC one) and I just thought it would be more convenient to integrate them. Of course, a low level device which could understand different disk track encoding was also necessary. Luckily I was already writing one (to read 256/512/1024 byte sectors), and that eventually became known as fd.device.

Eventually, when I have finalised the internal API (just at this moment undergoing a major revision to adapt it to Mac disks), I hope to release the source code, or at least enough to allow further additions to the disks that are recognised.

I should make it clear at this point that internally the handlers don't know they're running on an amiga, as they only see a 'virtual' machine interface. This will allow me to port the code to other environments. It should be possible to get it to run on the PC, maybe even the QL.

## 1.5 system requirements

xfs is designed to be compatible with the amiga operating system from WB1.3 upwards.

Note that you must use the non-XL fd.device on WB1.3, it's not as system-friendly, but it should work ok.

---

## 1.6 installation

Note that in the instructions below, FD0 / FD1 refers to the mountlists for xfsd using fd.device, and FX0 / FX1 for using mfm.device.

Quick installation notes:

FIRST

Open the drawer "L" in the XFS directory and drag the program "xfsd" to your system "L:" drawer.

Open the drawer "DEVS" in the XFS directory and drag the program "fd.device" to your system "DEVS:" drawer.

THEN for Workbench 2.1 and later

Drag the icons FD0:, FD1:, FX0: and FX1: into your system "Storage/DOSDrivers" drawer or "Devices/DOSDrivers" drawer.

Double click the FD0: or FD1: to mount the volume using fd.device  
Double click the FX0: or FX1: to mount the volume using mfm.device

OR for old Workbenches

Copy the updated mount.xfsd file to DEVS: and add a line to your startup for each drive you want to connect to XFS, as follows:

```
MOUNT FD0: FROM DEVS:mount.xfsd ;to use fd.device
MOUNT FD1: FROM DEVS:mount.xfsd
```

or

```
MOUNT FX0: FROM DEVS:mount.xfsd ;to use mfm.device
MOUNT FX1: FROM DEVS:mount.xfsd
```

## 1.7 usage

Usage should be (generally) fairly transparent, but here are a few disk type specific notes.

	read	write	delete	rename	readpart	format	
ql		+		+	+		+
spectrum		+		+			
msdos		+			+	+	+
archimedes		+		+			
amiga		+		+	+	+	+
minix		+		+			
macintosh		+		+			+
cpm		+		+			

Workbench  
 ZIP drives  
 x-code  
 other...

## 1.8 readdir notes

By 'readdir' I mean that commands like List, Dir etc should work, and hopefully directory utilities should as well.

The Examine/ExamineNext calls are supported, but currently the ExamineAll packet is disabled, which means it should work ok. Yes, you did read that correctly. If a handler doesn't support ExamineAll the Dos emulates it via Examine and ExamineNext. I'll re-enable it when it's fixed.

Note that (internally) file names are converted from however they are stored on disk into utf-8, which is a byte-sequence implementation of unicode. This should allow any filename to be read correctly. Of course, the amiga is limited in its own character set, so the internal utf-8 representation is further converted to either normal 8 bit ascii (iso) if that is possible, or to a sort of hybrid www/utf-8 sequence, eg

Unicode	Entity Name	utf-8 sequence	in amiga directory listings
0x0192	&fnof;	0xC6,0x92	%C6%92

Note that standard AmigaDos commands don't allow '%' characters in filenames, but any third party programs like NewList, Ls etc, and directory utilities like DirWork should have no problem accessing these files through xfsd.

As an added bonus, I have included my program x-code which is a utility that converts between the various character representations shown above.

## 1.9 read notes

By 'read' I mean that commands/programs that open files to read should work, so you can use Copy, Type etc, and various viewer programs. For this to work, Lock/Open/Read style calls are supported.

In general, xfsd does no translation of contents, as I don't consider a disk handler the proper place to be converting files from one locale, code page, cr/lf etc to another system. The only (slight) exception to this is for spectrum disks, where the files are converted to ZX82 format, which is in any case just a

'wrapper' format, and doesn't change the contents.

## 1.10 ql notes

Since this was the first handler I wrote, most things are complete. Only thing left to do is that rename only works as an actual rename, that is it doesn't allow you to rename things to move them.

Directory listings of ql disks show the following info:

Names - can be up to 36 characters on the ql, whereas the amiga (officially) allows only 32. I just ignore this because most amiga programs can accept up to around 40 characters. Still, a possible source of random crashes, so I'll be looking at a fix.

Size - Doesn't include the (hidden) 64 byte file header, so files may appear to take more blocks than you would think.

Flags - r/w/d ignored at present, 'e'xecutable flag is used to indicate a type 1 (program) file, with the data space held in the comment.

Dates - translated from base 1961

Note that ql directories are implemented in such a stupid way I can scarcely bring myself to explain it. Oh go on then... The whole path, including separators, is stored! So if you have a directory with for example a 10 byte name, the files inside can only have 25 character names (36 - 10 - 1 separator). And the deeper into the directory structure, the worse it gets...

## 1.11 spectrum notes

Currently read-only, this handler is different from all the rest in that the files are translated into a common format as they are read. The format used is called ZX82. This has the advantage of allowing, for instance, the use of datatypes to automatically recognise particular Spectrum file types, in particular screen shots.

The spectrum has a veritable cornucopia of different disk types, falling into two main categories. Here is a list of some of the types that should be usable.

What	Where	Who
BetaDOS	Spectrum and +D	BetaSoft
GDOS	Sinclair Spectrum	MGT Disciple Interface

G+DOS	Sinclair Spectrum	MGT/Datel +D interface
MasterDOS	SAM Coupe	MGT and BetaSoft
SAMDOS	SAM Coupe	Miles Gordon Technology

In addition, the Spectrum +3 uses its own version of CP/M. This, of course, is handled via the CP/M section of xfsd, and means +3 files (currently) are NOT in ZX82 format.

## 1.12 msdos notes

Since this handler has been getting the most attention, it should be the most reliable. Just a few things you should note.

Accented characters in 'short' names (ie 8+3) are always translated (when reading a directory) as code page 850. Of course, characters in 'long' names need no translation, but where they are not supported on the amiga, they will be displayed in a sort of combined utf-8 / 'www' format, eg %C6%92 for the curvy letter f, (also known as &fnof;).

The process followed to generate 'short' names (aliases) from 'long' names is similar to W95, except that in order to get rid of the need for code pages, the 'short' names will be strictly 7-bit ascii, with accented characters mapped onto their uppercase unaccented equivalents. Again, this doesn't affect the proper 'long' names.

Note also that the 'short' names appear in the comment field, what's more they can be directly changed there. This can be useful if you want the 'short' names to appear as they would on other handlers. For instance, if you save a disk icon onto a pc disk using xfsd, the name 'disk.info' would be saved as the 'long' name, but since this is not a legal 'short' (ie 8+3) name, a new one would be generated, in fact DISK~1.INF would be used. Now, since other handlers expect the disk icon to be called DISK.INF they wouldn't recognise it, so just change it via the comment, eg

```
FileNote FD0:disk.info DISK.INF
```

(if the disk was in FD0: of course).

## 1.13 archimedes notes

At present only 'E' format disks are supported. I hope to add 'D' format disks at some future date.

Note that since the only info I have on the archimedes consists of a two Basic programs written by Richard Kettlewell in SuperBasic for the ql, as supplied to me by Simon Goodwin, support is limited to whatever those programs could do. In particular, dates were not supported, so I don't know how they work, so the

dates on directory listings will be wrong.

## 1.14 amiga notes

Why an amiga handler?

Well, this allows you to read/write FastFileSystem disks on WB1.3, and, because fd.device is used to decode the tracks, you can transparently read diskspare formatted floppies.

In addition, having amiga support in xfsd lets you have just one mountlist entry for other types of removable media that may be formatted in a variety of ways, like ZIP drives for instance.

Currently the amiga handler is read / write on all media with the exception of Directory Cacheing (DC) disks, which will remain read-only until I fix that part of the handler (next release).

V2.11b2 contains a completely re-written sector allocation system (to allow writing to large media) and consequently will require substantial testing before I can recommend it for anything other than experimental use. But it should be safe just reading. Hopefully it will be thoroughly tested by the next release.

## 1.15 minix notes

This is based on info taken from linux. I probably will add writing support at some future date, but really it's more of a priority to support more popular linux filesystem types like ext2. I also hope to support BerkeleyFastFileSytem at some point.

Note that minix support has only been tested on 720k floppies.

## 1.16 macintosh notes

Characters in file names are considered to be MacOS\_Roman.

Understands mac-style partitions, so ZIP media partitioned and written on a mac should be readable.

Currently reads only the data fork of files.

Hopefully writing support should appear quite soon. Although this section of the handler is based on the hfsutils package by Robert Leslie, it didn't prove possible to just drop it into xfsd, only the btree routines being directly usable, others needing major changes to fit in with the way xfsd works. Nevertheless, having an example of working code shortened the development time considerably (as did actually having a macintosh).

---

## 1.17 cpm notes

Bit of a mistitle this, as the number of CP/M disks supported is only two out of the many hundreds of types, specifically the PCW and the +3. At least it works on the PCW disks I've got, and on the one +3 disk I have as well.

Hopefully it won't take too long to implement the ZX82 file format for +3 disks, now that I have an example disk to experiment with.

## 1.18 Workbench notes

You should note that disk icons only appear on the Workbench when VOLUME entries appear for them in the internal dos list. Unfortunately, you can't have two volumes with the same name, so if you have another handler mounted that has e.g. recognised a pc disk AND xfsd also recognises it and they BOTH put volume entries in the dos list, you WILL see two icons appear. USUALLY this doesn't immediately crash the machine, but opening the icon(s) or removing the disk may well do so.

So? Well, in order to let you try xfsd without having to take all your other handlers out, you can disable the mounting of the volume entries via the Flags entry in the mountlist.

Here is how to control it (excerpt from mount.xfsd)

8<-----

Now Flags = 0 stops the volume being mounted by name. This prevents the WorkBench recognising the disk but still allows you to access it using the device name (ie FD0:). There is no longer a separate handler called xfsd-ami. To get the volume to be mounted (and the Workbench icon to appear) the bits in the top byte represent each different section of the handler.

```

                8 4 2 1 - 8 4 2 1
                | | | | | | | |
ql -----+ | | | | | | |
spectrum -----+ | | | | | |
msdos -----+ | | | | | |
archimedes -----+ | | | |
amiga -----+ | | |
minix -----+ | |
mac -----+ |
cpm -----+

```

So if you wanted the handler to only mount MSDOS or MAC volumes you would use Flags = 0x22000000

REMEMBER, even if the volume isn't mounted, the disk is still accessible via the device (eg FD0:).

ALSO to disable a filesystem completely, the same bits in the next byte can be set, so to disable the amiga file handler section of xfsd use `Flags = 0x00080000`

ALSO only the top word of the flags is used by the handler, this being masked off before the flags are sent to the low level device, so you can put any values you like in the lower word, whatever is required for the .device you are using. This means you can use the xfsd handler with `mfm.device`, or with `catweasel` if you like.

8<-----

I know that those of you who use the Workbench for everything would really like a more sophisticated support for disk icons, for example giving different disk formats a different default icon, but this isn't possible at the moment. However, for the disk types that are writeable using xfsd, you can of course copy any disk icon of your choice onto the disk, and the Workbench will read it when you insert the disk.

## 1.19 ZIP drive notes

There is an example mountlist called `ZIPX` in `devs/DosDrivers`, but you **MUST** remember to set the device/unit (in the tooltypes).

Also, if you are using another ZIP mount entry at the same time you **MUST** also read the explanation (in Workbench) about setting the flags field correctly (ie to not let xfsd put up a disk icon).

## 1.20 x-code usage

When using xfsd to list file names, any characters unavailable on the amiga are printed in a sort of hybrid `www percent / utf-8` style, so for instance, if a file name includes the 'L with a slash' character in it (otherwise known as `&Lstrok;`), it is displayed as `%C5%81`.

To assist you with these strange codes x-code can convert and display the various code formats. So...

```
x-code &Lstrok;
```

```
prints out...
```

```
&Lstrok;    0x0141  %C5%81
```

where `0x0141` is the unicode representation, and...

```
x-code %C3%B6
```

prints out...

```
&ouml;    0x00F6    %C3%B6
```

V1.01

-----

x-code now lists ALL entities that even partially match, so

```
x-code &Z
```

prints out...

```
&ZHcy;      0x0416      %D0%96
&Zacute;    0x0179      %C5%B9
&Zcaron;    0x017D      %C5%BD
&Zcy;       0x0417      %D0%97
&Zdot;      0x017B      %C5%BB
&Zeta;      0x0396      %CE%96
&Zgr;       0x0396      %CE%96
```

Note that it's still case-sensitive, so

```
x-code &z
```

would be required to print out the lower case entities

## 1.21 other bits and pieces

As you can see, this guide has been thrown together rather quickly, and as a result certain details may have been missed out. I suggest you examine the history for more details.

Disk recognition

As if all this was not enough, `fd.device` inserts an input stream handler to allow you to force it to re-read the disk, at the same time sending a disk change interrupt to whatever dos handler is using it.

The key combination(s) for `fd.device` are

```
left_alt + numeric_pad(unit#)
```

ie left alt with numeric pad 0 would force a disk change on unit 0.

The same is true for the `file.device`, although in this case it also allows you to select another disk image file for access, via a requester.

The key combination(s) for `file.device` are

```
left_alt + left_shift + numeric_pad(unit#)
```

ie left alt plus left shift plus numeric pad 1 would force a disk change on unit 1.

NOTEZ BIEN

This key usage may change - especially the particular key combinations (because they may clash with other programs).

---

#### POWER XL

The current fd.device doesn't work with the Power XL high density drive. I've included a more system-friendly version (fd.device.XL) that seems to work ok.

If you need it, just rename it to fd.device and put it in devs:

#### RAWPATCH / MFM

---

Two utilities to experiment with raw disk data. So obscure as to be almost unusable. Use rawpatch to read in raw track data, then mfm to decode it.

Hint: ALWAYS use -I -W with rawpatch (index sync & word sync). At the very least mfm likes its input wordsync'ed, and you can only get an idea of the sector skewing if you read starting from the index pulse.

Hint: use -v on mfm to get a verbose output (or even -vv ).

#### GEOMETRY

---

This program sends a TD\_GETGEOMETRY command to any device to find out the media layout AND with fd.device it gets some hidden extra information about the interleave, skew factors etc.

#### CRC

---

Just a simple program to generate CRC's for files, but I've added an option to generate a CRC for each block (by using eg -b512), which by default is the one used on disk sectors (pc MFM that is, not amiga style).

## 1.22 todo

First thing to do is get the Mac handler finished.

Then maybe I can write a Format utility.

Then perhaps upgrade the Spectrum handler to allow writing.

## 1.23 acknowledgements

Simon Goodwin <simon@studio.woden.com>

- keeps me supplied with spectrum/ql/cpm disks and info.
- also for writing the original documentation.

Mark Weller

- for supplying Archimedes 'D' and 'E' format disks.
-

Richard Kettlewell

- for his two Archimedes (QL Superbasic) reader programs.

William James

- (author of Speculator) for his co-operation over ZX82.

Robert Leslie

- for his excellent hfsutils package.

Chris Thornley <osu036@sos.bangor.ac.uk>

- for e-mailing me example disk images.

Brian Gaff <briang@bgserv.demon.co.uk>

- for sending me the +3 disk / info

Plus a few others with useful bug reports/suggestions:

Dirk Neubauer	<neubauer@rz.uni-greifswald.de>
Guido Mersmann	<geit@studst.fh-muenster.de>
Andreas Paul	www.student.uni-kl.de/~apaul
Lars Nelson	<lars@infohwy.com>
Adrian Siemieniak	<sauron@pwr.wroc.pl>
Marcel Timmermans	<mtimmerm@worldaccess.nl>
Christian Kersting	<kerstic@uni-muenster.de>
Marko Seppanen	<marko.seppanen@wwnet.fi>
Ken J Powell	<ken@dra.ex.ac.uk>
Michel De Meerleer	<michel.demeerleer@iname.com>
Tim Jackson	<timjackson@radiolink.net>
Ralph Debusmann	<rade@coli.uni-sb.de>
Jernej Pecjak	<jernej.pecjak@kiss.uni-lj.si>

Plus I got postcard from Heidelberg, which, after careful examination with a magnifying glass, I managed to discover as coming from (I think) Jorg <jc7@ix.urz.uni-heidelberg.de>

Your name here!

## 1.24 the author

francis swift AKA frank AKA (in computerspeak) frans

- e-mail: (via my brother Mark)

msw@blackpool.ac.uk

- snail mail:

325 Charlestown Road, Blackley, Manchester M9 7BS, UK.

bug reports, suggestions, ideas, banker's drafts etc are welcome.

## 1.25 bugs

---

Most of the current 'bugs' are caused by the program being incomplete. For instance, the Spectrum handler doesn't write, but it isn't disallowed from doing it, so weird things can happen if you try to save something.

Still, if you do have a disk which shows up a particular bug, make a disk image file of it and e-mail it to my brother Mark.

You can make a disk image file with the DEV: (dos) device and fd.device.

Mount DEV: from devs:mount.dev (or wherever you keep it)

Copy DEV:fd.device,0 Ram:dodgydisk

Note: this is my DEV: device - others may operate differently. (e.g. the ,0 means unit 0)

## 1.26 history

----- xfs211b2 -----

I've completely re-written the sector access / allocation on the Amiga handler, so as to allow you to write to media larger than floppy disks. Unfortunately I haven't had time to test it, so I don't recommend you use it for anything crucial yet, just media you can afford to re-format. Oh, and I found a few things were missing in the directory cacheing (DC) part of the handler, so I've made DC disks read-only for the moment till I sort it out. Also, for some reason the utf-8 internal names were being sent to the Amiga name hashing routines, causing all files whose names contained accented characters to be put on the wrong hash chain. If you have any disks written using this I suggest you use the old version of the handler to copy the files off the disks then use 211b2 or later to write them back (if it works!).

No further additions to any other handlers, 'cause I've been adapting various compression routines, to see how easy it would be to add archive-handling to xfsd.

I might as well mention that I had hoped to replace the rather awkward DMS uncrunching in file.device, but I couldn't get access to the source on the Aminet. Oh, didn't you know file.device could read DMS files? Well I can't recommend it. Or DMS for that matter. But you will need both, as file.device calls DMS to do the work.

----- xfs211b (only released on cover cd) -----

MAC handler does something at last. You can now do directory listings and read the data forks of files.

Temporarily disabled ExamineAll support so that ExamineAll should now work properly. Yes you did read that right. If a handler doesn't support ExamineAll the Dos emulates it via Examine and ExamineNext. I'll re-enable it when it's fixed.

On the QL, accented characters should now be completely usable in filenames, previously they would only read correctly.

Oh, and I've upgraded the x-code utility (to V1.01), and I've replaced the geometry program with a more general device command utility called devio (no docs for it but it's simple to use).

Oh, there seems to be some confusion as to the names of the mountlists. I've made sure they're all named in the same way (devs/mount.#?).

Finally, I've changed all the flags fields in the mountlists to make all volumes appear on the Workbench. So if you use xfsd at the same time as another handler for the same disk, something strange may happen (unless you edit the flags field to stop xfsd mounting volume entries - see mount.xfsd).

----- xfs210b2 -----

Amiga PARTITIONS now really looked at properly (a last minute change to the checksum code stopped them working). Anyway, it gave me a chance to look at MAC PARTITIONS. Unfortunately, although the partition info is understood by xfsd, the actual mac volumes aren't (yet).

Enhanced FINDER.DAT support to deal with names that are the same on both mac and pc but with the case changed.

Finally squashed the bug that only cropped up on the A4000. It only appeared in Zorro3 fast memory, so I never saw it.

NEXT RELEASE the archive name will be just xfs.lha with the version number in the short description. There will also be an xfs.guide (just working on this at the moment). Also (hopefully) a working(ish) mac handler. When that's done I can finally get round to writing a format facility.

----- xfs210b -----

Finally got round to fixing WRITING of Amiga disks (OFS & FFS). This used to work (ages ago) but got messed up when I added a few things. Anyway, while I was looking at the code I added something else. Amiga PARTITIONS are now examined. Unfortunately the actual handler code was only designed for floppies, and doesn't look at the BitMapExtend entry, so there is a limit of around 50MB on the volume size. Should be fixed by the next release.

----- xfs209b4 (unreleased) -----

Last minute bug (re-)stomped - on writing out the fat on pc disks sometimes one extra byte of memory got trashed causing random crashing. Second time I've fixed this!

Oh perhaps I should remind you to change the ZIPX mount entry to match your own device / unit, otherwise it obviously won't work. Now that's what I call documentation.

---

----- xfs209b3 (unreleased) -----

The fd.device write bug should be cured (it only affected high density disks anyway - made the gaps too small).

There is now also a new (slightly experimental) XL version of fd.device (fd.device.V42.77.XL). This tries to be more system friendly, so it may also work with the raw disk images used by UAE. If you need this or just want to see if it works, rename it to fd.device and put it in devs:.

The reason for the delays in releasing the write version of the PC filesystem was it kept crashing on the A4000, it worked ok on the A500, and just suffered harmless anomalies on the A1200 (it left the disk spinning after a write). Since this A4000 bug hasn't really been fixed, I can't guarantee anything yet.

Oh, the handler (from xfs209b2 onwards) DOES understand at least the first partition on PC media - try the ZIPX mount entry.

----- xfs209b2 -----

For users who wish to use mfm.device in conjunction with xfsd instead of fd.device, the appropriate mountlist entries and workbench icons are now provided.

However, using mfm.device has the disadvantage of only being able to access disks with a fixed dos-like format of 9 or 18 sectors per track (as pre-defined in the mountlist). This will prevent you from accessing some foreign disk formats.

Updated quick installation notes:

Open the drawer "L" in the XFS directory and drag the program "xfsd" to your system "L:" drawer.

Open the drawer "DEVS" in the XFS directory and drag the program "fd.device" to your system "DEVS:" drawer.

Procedure for Workbench 2.1 and later

Drag the icons FD0:, FD1:, FX0: and FX1: into your system "Storage/DOSDrivers" drawer or "Devices/DOSDrivers" drawer.

Double click the FD0: or FD1: to mount the volume using fd.device  
Double click the FX0: or FX1: to mount the volume using mfm.device

Procedure for old Workbenches

Copy the updated mount.xfsd file to DEVS: and add a line to your startup for each drive you want to connect to XFS, as follows:

```
MOUNT FD0: FROM DEVS:mount.xfsd ;to use fd.device
MOUNT FD1: FROM DEVS:mount.xfsd
```

```
MOUNT FX0: FROM DEVS:mount.xfsd ;to use mfm.device
MOUNT FX1: FROM DEVS:mount.xfsd
```

\* Note - There is a known bug in fd.device which shows up on some Amiga models when WRITING to disks. If you experience problems, you should use mfm.device until the bug is fixed in a later release.

----- xfs209b -----

(Never actually released)

The PC/95/NT filesystem should now WRITE as well. For files with both long and short names the short name appears in the comment and can be changed via SetComment / FileNote.

Short names on disk are assumed to be code page 850.

Long names (in unicode) are converted to / from amiga codes.

Any characters unavailable on the amiga are printed in www style percent format using their UTF-8 coding, so &Lstrok; is displayed as %C5%81.

To assist you with these strange codes a utility (x-code) is included to convert / display the various code formats. So...

```
x-code &Lstrok;
```

```
prints out...
```

```
&Lstrok; 0x0141 %C5%81
```

```
and...
```

```
x-code %C3%B6
```

```
prints out...
```

```
&ouml; 0x00F6 %C3%B6
```

You can enter FILENAMES including the percent format characters and the handler converts them to the correct character on the filename on disk.

Where SHORT NAMES have to be automatically generated from the long names, the accented characters are mapped to their uppercase un-accented equivalents, &lstroke; (entered as %C5%82) becoming the letter L.

Some support for MAC users is included in the PC handler, with mac style long file names stored on PC disks in FINDER.DAT being used if possible.

Since at present the handler doesn't understand PARTITIONS, to use with any media that has been partitioned the mountlist must

point directly to the volume, not the start of the partition.  
If you can figure out a hacked mountlist like this you may be  
able to get the handler to access ZIP drives. In any case, this  
would be a temporary solution until partition support is  
implemented (probably ten minutes after this release).

----- xfs208b -----

MINIX FileSystem now working (read-only).

New system for enabling/disabling the mounting of volumes, and  
for disabling recognition of certain disks (no more xfsd-ami)

...see mount.xfsd for details

CPM3 Filesystem (well PCW 3.5 inch 720K disks) (read-only)

I have precisely one PCW disk and it works perfectly, so there.

Sorry, the mac handler just recognises that the disk is indeed a  
Mac disk, reads the volume name, then just sits there and refuses  
to do anything else. Maybe next time.

----- xfs207b2 -----

ALL NEW INTRO!!!

Well, just a few extra words.

There was a hash clash (try saying that quickly) in the internal  
name cache (or should that be hash cache?) between the short names  
and the long names. Basically, the dummy short name entries were  
displacing the true long name entries, thus losing all the internal  
pointers etc, so the directory listings (which don't use the name  
cache) showed all the files, but the handler couldn't access them.

I'm still working on the MAC handler, based on code from the  
hfsutils package on linux, written by Robert Leslie.  
Hopefully, because I'm basing it on working code, it should work  
as a complete read/write handler as soon as it's finished :-)

EXTRA BONUS \*\*\* x-hfsdir \*\*\* EXTRA BONUS

Mount DEV: (you've already got it - see the mount list for details)  
then you can get a directory of a MAC disk with

```
x-hfsdir dev:mfm.device,1
```

or, if there's a subdirectory called hyper

```
x-hfsdir dev:mfm.device,1 hyper
```

and a subdirectory in there called new

```
x-hfsdir dev:mfm.device,1 hyper/new
```

you get the picture, just remember the ,1 means drive unit 1 (df1:)

---

----- xfs207b -----

first release!