

MultiFileSystem

COLLABORATORS

	<i>TITLE :</i> MultiFileSystem		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY		July 22, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	MultiFileSystem	1
1.1	MultiFileSystem	1
1.2	introduction	1
1.3	installation	2
1.4	Format and DiskCopy	2
1.5	The MFS Commodity	3
1.6	Tricks & Tips	3
1.7	author	4
1.8	distribution	4
1.9	acknowledgements	4
1.10	history	5

Chapter 1

MultiFileSystem

1.1 MultiFileSystem

MultiFileSystem

by Nicola Salmoria

Introduction
Installation
Format and DiskCopy
The MFS Commodity
Tricks & Tips

Author
Distribution
Acknowledgements

History

1.2 introduction

Thanks to the versatility of AmigaDOS, we are able to use many different file systems on our disks. The most popular is of course CrossDOS (tm), but many other file systems can be found in the PD.

Adding a new file system to the Amiga is simplicity itself: just copy a file in L:, provide a suitable mountlist, and there you go.

This works flawlessly if every physical unit is accessed by a single file system, but if multiple file systems share the same device, some problems arise:

- You have to call the same unit with different names (e.g. DF0: and PC0:) depending of the kind of disk inserted.
- When you insert a PC disk, the Workbench shows a DF0:???? icon, and vice versa; this can be very annoying especially for owners of more than one drive.

- When a file system is busy (for example during a DiskCopy), the other file systems can still access the disk; this goes against the concept of "inhibiting" a device.

MultiFileSystem solves all of these problems. It isn't a file system in the usual way: it is an interface to other file systems. When you mount it, you tell it which file systems it should use; afterwards, when a disk is inserted, MultiFileSystem will recognize it and pass the commands to the appropriate file system. This means that with a single device name, for example DF0:, you can access any conceivable file system!

You are not limited to floppy drives, of course: I successfully installed MultiFileSystem over a 128Mb magnetooptical drive.

1.3 installation

The provided Installer script will perform a standard installation, placing MultiFileSystem on top of all available floppy drives. It will also copy the MFS commodity to SYS:Tools/Commodities, and optionally modify the system commands Format and DiskCopy to make them take advantage of MFS features.

NOTE: to make the changes effective, you must reset after the installation. The script will not do that automatically.

The script will create, in DEVS:DOSDrivers, a drawer for every available floppy drive (i.e. DF0, DF1 and so on). If icons named PC0 and so on are present in DEVS:DOSDrivers, they will be moved to the new drawer (only PCx icons will be moved, therefore if your CrossDOS devices have different names you'll have to move them manually).

During the system startup, DEVS:DOSDrivers will be scanned for directories (instead of the plain files used by the standard file systems) and for every directory present MFS will create a device with the same name. MFS will share the filesystems contained in that directory. If a device with the same name of the directory is already present (for example DF0), MFS will "steal" it and use it as the default file system for the newly created device.

Therefore, if you want to add another filesystem to the ones recognized by e.g. DF0, all you have to do is drag its icon into the drawer named DF0.

1.4 Format and DiskCopy

Since the system commands don't know about MFS, I provide two patches for Format and DiskCopy which allow them to take advantage of MFS features. The patches are installed on request by the install script.

Features of Format:

- When you click on the icon of a MFS disk and pick the menu command Icons/Format Disk... a requester will ask you which filesystem to use

for the format.

- If you load Format by double clicking on its icon, selecting a MFS device will not bring up the requester. You have to use the MFS commodity .
- When run from CLI, the new FS=FILESYSTEM argument is available. It allows to specify a dostype as it appears in the MFS commodity window.

Example:

```
> Format DRIVE DF0: NAME "MS-DOS" FS MSD\0
```

Specifying one of the Amiga specific keywords (OFS,FFS,INTL,NOINTL and so on) will automatically select DOS\0.

- These features are active only if the device is in "Automatic" mode, so you can still lock a specific filesystem using the MFS commodity .

Features of DiskCopy:

- Dragging a disk icon over the icon of a MFS disk will automatically choose the appropriate filesystem on the target device.
- Same when used from CLI.
- These features are active only if the device is in "Automatic" mode, so you can still lock a specific filesystem using the MFS commodity .

1.5 The MFS Commodity

Originally, this commodity was written to provide a way to lock a specific file system before formatting or copying a disk. Starting with MFS 1.1, the Format and DiskCopy patches make these operations much easier.

The commodity may still be useful when precise control over the file system choice is required. It also shows the file system used by the disk currently inserted.

The user interface is very simple: you just have to select the device from the list, and choose which file system it should use. The default setting is "Automatic": this means that MultiFileSystem automatically chooses the file system depending on the disk inserted. The other entries in the cycle gadget are the DOS Types of the file systems, for example "DOS\0" for Amiga and "MSD\0" for CrossDOS.

1.6 Tricks & Tips

- Some people requested the ability to use different default disk icons for different file systems, so one can recognize the file system used by a disk by just looking at its icon. This is possible via another program, which was incidentally written by me ;-). It's called DefIcons, and is part of the NewIcons package. When DefIcons is installed, disks without a disk.info file will use icons named def_XXXXdisk.info, where XXXX is the file system
-

DosType; for example def_DOS0disk.info for Amiga disks and def_MSD0disk.info for MS-DOS disks.

The NewIcons package can be found on Aminet: util/wb/NewIcons.lha

• To speed up the recognition of a MS-DOS disk, lower the tdu_RetryCnt field of trackdisk.device. Users of NickPrefs can use the Floppy preferences for the purpose. Since NickPrefs doesn't work under Kickstart 3.0, an alternative program must be used, for example SetTDRetry by S. Koszewski. SetTDRetry can be found on Aminet: misc/emu/SetTDRetry.lha

1.7 author

Nicola Salmoria
Via Piemonte 11
53100 Siena ITALY

Internet : MC6489@mclink.it

I try to reply to all E-Mail I receive. If you wrote and I didn't answer, it's likely that either your original message or my reply have been lost, or that the header of the message didn't contain your correct E-Mail address. Try again; don't forget to put your complete E-Mail address in the message body.

1.8 distribution

This software is FreeWare. It is freely distributable as long as the archive remains intact, and only a nominal fee is charged for its distribution.

This software is provided "AS IS" without warranty of any kind, either expressed or implied. By using it, you agree to accept the entire risk as to the quality and performance of the program.

1.9 acknowledgements

Many thanks to the countless people who sent me E-Mail suggesting improvements and reporting bugs. In particular:

Dan E Babcock
Todd Deery
Jeremy Elgin
Andreas Gutzwiller
Ernst Lehmann
Daniel O'Connor
Dante Profeta
Steve Sim
Andrea Suatoni

1.10 history

Version 1.2

mfs-handler:

- Now works with DiskExpander. If you install DiskExpander on DF0:, it will act on every disk inserted, regardless of the file system used. Even if the underlying file system is not compatible with DiskExpander!
- Now CrossMAC displays icons in Workbench windows. Thanks to Michael Simons for his help.
- Now works with the MaxDOS commodity.
- Now completely "disguises" itself as the underlying file system, copying not only the FileSysStartupMsg but the other DevInfo fields too. This is the maximum compatibility which can be obtained. The only field different from the underlying file system is, for obvious reasons, dol_Task. A side effect is that "version df0:" will now return the version of the underlying file system, not the version of mfs-handler.
- ACTION_MORE_CACHE now passes the command only to the active file system, not to all of them. This way you can add memory to the ROM file system without giving it to all of the others. The easiest way to increase the buffers of other file systems is to change their mountlist.
- If none of the underlying file systems recognizes a disk, now tries to at least pick a file system which is able to read the disk. This allows usage of disk editors (or FLAT:) on disks which are readable (typically by mfm.device) but don't use a known file system - or any file system at all. Note that for this new feature to work as expected, all of the underlying file systems must be correctly written, returning as DiskType BAD\0 if the disk is not readable and NDOS if the disk is readable but not recognized. Some of the available file systems do not do this. Of course you can always use the MFS commodity to force a specific file system.
- Now most packets are handled asynchronously. The only exceptions, for technical reasons, are ACTION_DISK_INFO and ACTION_INHIBIT.
- The "You MUST replace volume xxx in device yyy!" error requester showed the wrong device name (e.g. EF0 instead of DF0). Sorry, I didn't notice this before because I'm a good guy and never remove disks when the light is on! ;-)
- Fixed memory loss with ACTION_FH_FROM_LOCK.
- Now handles failure of ACTION_INHIBIT on the underlying devices.
- Fixed problems when relabeling a PFS disk.

MFSMount:

- When it couldn't mount a device, it was displaying the "Couldn't mount
-

device xxx:" error message, but xxx was truncated to the length of the MFS device name (for example, it would say 'MAC:' instead of 'MAC0:' if the MFS device was DF0:)

DiskCopy patch:

- Didn't work from CLI if you specified a colon-terminated source device name.

Version 1.1

mfs-handler:

- This is the most important bugfix of this release: mfs-handler wasn't parsing correctly the startup string generated by some versions of Mount, and it was therefore impossible to mount it. This bug probably affected all Workbench 2.1 users, so if MFS 1.0 didn't work on your system, 1.1 very likely will. If it still doesn't work, please E-Mail me. Many thanks to Steve Sim for helping me sort this out.
- ACTION_CREATE_DIR wasn't handled correctly (but thanks to the clever organization of the code, this wasn't causing major problems)
- Changing the locked filesystem is now handled faster.
- Sometimes, after de-inhibiting a device, it wasn't possible to access the disk inside it via the device name (e.g. df0:): the requester "please replace volume xxxx" would appear. This was caused by a bug in FastFileSystem. Thanks to Andrea Suatoni for the report.
- If the MFS commodity was used to lock a specific filesystem, setting it back to automatic while the device was inhibited (for example during a Format) would cause problems. Thanks to Jeremy Elgin for the report.
- ACTION_READ and ACTION_WRITE are now handled asynchronously. This means that while copying files to/from a floppy disk, other tasks are still free to access the filesystem, in particular the Workbench will not lock up until the disk operation is finished as it did before. Thanks to Ernst Lehmann for pointing this out.

MFSMount:

- The version number is now correctly 38.x, to show that it needs at least Workbench 2.1 to work.
 - Now checks the version number of C:Mount, and fails if it is less than 38.
 - Since it cannot install MFS correctly otherwise (the DF0:???? icon would continue to appear when a non-Amiga disk was inserted), now fails when the Workbench is already loaded.
 - Would not mount correctly DosDrivers with a name length different from the drawer name. Therefore a drawer named DF0 would allow only DosDrivers
-

named for example PC0 or PF0, not MAC0: or A:. SORRY!

Thanks to Dan E Babcock and Andreas Gutzwiller for the report.

- Due to a buffer too short, would crash if one tried to share more than four filesystems under a MFS device. Thanks to Todd Deery for the report.
- Is now more informative when something goes wrong.
- If a dosdriver couldn't be mounted (for example, if CrossDOSFileSystem was not in the L: directory), MFSMount would print a cryptic "device DF0: already mounted" error message, and MFS would not be mounted correctly. Now a more appropriate "Couldn't mount device 'xxx:'" message is printed, where xxx is the name of the dosdriver which caused the error. MFS is mounted correctly, but of course it is not able to read disks which use the xxx filesystem. Thanks to Daniel O'Connor for the report.

MFS Commodity:

- Now shows the filesystem currently used.

Install script:

- Wasn't moving the PC1 icon to the DF1 drawer.

Format and DiskCopy patches:

- These are new, weren't included in MFS 1.0. See the Format and Diskcopy section for a detailed description.
-