MSF

MS-DOS File Manager
for OS9 Level Two

Clearbrook Software Group

MSF

MS-DOS File Manager for OS9 Level Two
OS9 System Series

Release A
August, 1987

Clearbrook Software Group
USA: P.O. Box 8000-499, Sumas, WA, 98295-8000
CANADA: P.O. Box 8000-499, Abbotsford, B.C., V2S 6H1
Telephone: (604)853-9118

TABLE OF CONTENTS

Introduction

MSF is an MS-DOS file manager for OS9 Level Two. It organizes the information on floppy disks in the same way that MS-DOS does, allowing you to interchange disks between your OS9 computer and other computers which use MS-DOS.

Two device descriptors are provided with MSF. **/A** is the descriptor for the first MSF device (like **A:** under MS-DOS) and **/B** corresponds to the second MSF device. **/A** shares the first physical drive with RBF (Random Block File—i.e. OS9 format) device **/d0** and **/B** shares the same drive as **/D1**. Several device descriptors are supplied so you can choose the ones which fit your drive capabilities.

System Requirements

To use the MSF MS-DOS file manager, you need a Tandy Color Computer 3 with two floppy disk drives. You must be using the OS9 Level Two operating system with the SDisk3 disk driver.

Differences Between MSF and RBF

File Names

One of the main differences between MSF and RBF files is file naming.

An RBF name can be up to 29 characters long containing upper and lower case letters, numbers, periods, and underscores. The first character of the file name must be a letter.

An MSF file name consists of eight characters, a period, and three additional characters. The characters can consist of letters (converted to upper case), numbers and **#**, **$**, **%**, **&**, **'**, **'**, **(**, **)**, **{**, **}**, **-**, **_**, **@**, **,** **^**, or **!**. The file name may begin with any one of these characters. If you specify a name longer than eight but shorter than twelve characters with no period in it, MSF will automatically place the period after the eighth character.

OS9 Name	*MS-DOS Name*
`long_file_name`	`long_fil.e_n`
`mail.list`	`mail.lis`
`read.me.first`	`read.fir`

Reserved File Names

Both MSF and MS-DOS do not allow certain names to be given to files. These reserved names are: AUX, COM1, COM2, CON, LPT1, LPT2, LPT3, NUL, and PRN.

The OS9 shell treats #, &, (,), <, >, and ! as special and will not allow them in a file name. You must rename MS-DOS files with these characters in them before using those names as parameters in a Shell command.

Device Names

MS-DOS uses **A:** and **B:** as floppy disk drive names. MSF uses **/A** and **/B**.

Directory Separator Character

MS-DOS uses \ to separate directory names in a pathlist while OS9 and MSF use /.

File Attributes

MSF files have different attributes than RBF files. There is no *Sharable* attribute so all files are assumed to be sharable. Files are always readable and can be set to read only status with the *Read Only* attribute. All attributes are public—all users can access MSF files.

Sector Length

RBF disks use a sector length of 256 bytes while MSF disks have a sector length of 512 bytes. This causes some undesirable side effects. Most disk drive controller chips and the software which drives them will read an entire physical sector from the disk regardless of the length of sector expected. When the software is expecting a 256 byte sector but reads a 512 byte sector instead, the last 256 bytes gets loaded into memory, overwriting whatever is currently there! Depending on what it overwrites, it could crash the system or cause other serious problems. This situation occurs when you place an MS-DOS disk in a drive and try to access it as an RBF disk—*please* avoid this situation. For example, don't place an MS-DOS disk in drive /B and then try to access a file on **/d1**.

End of Line Termination

OS9 normally terminates a line of text with a carriage return character ($0D). MS-DOS terminates a line of text with a carriage return and a line feed ($0A). You can add or remove line feeds from a file using the the 'lf' filter provided with this package.

End of File Termination

MS-DOS has two ways of detecting the end of a file. For a binary file the end is detected by comparing the current file position with the file size, in the same way as OS9. The end of a text file is marked with a CTRL-Z character ($1A). To maintain compatibility with MS-DOS, MSF will return an EOF error (#211) when the first character read by an I$ReadLn system call is CTRL-Z. When writing past the end of a file, CTRL-Z is appended to the end of the file unless another cluster would be required by this action.

BASIC Languages

The BASIC languages on MS-DOS systems are quite different from Basic09. However, you can modify Microsoft Basic programs to run in Basic09 if you have the time and inclination. You will have to determine the differences between the BASICs yourself. Remember to save the Microsoft Basic source programs in the ASCII format if you want your OS9 text editor to be able to read them. You will probably have to run the files through the 'lf' filter as well.

Root Directory

The root directory on an MS-DOS disk is of a fixed size. A maximum of 112 files may exist in the root directory on a double-sided MS-DOS format disk and 64 on a single sided disk.

Disk Management

Other differences between MSF and RBF disk structures such as space and directory management are taken care of by the file manager and utility programs.

Utility Programs for MSF

Because MSF organizes disks in a different way than OS9's RBF file manager some new utility programs are required. For example, the directory structures are different so a new utility is provided to get a directory of MS-DOS files. Other RBF utilities which also do not support MS-DOS disks are Attr, Pwd, Pxd, and Rename (these functions should have been implemented as I$SetStt calls in RBF).

MSF prevents damage to MS-DOS disks when unintentionally using the RBF Attr and Rename commands by preventing writes to the directories and the entire disk (@). Error #203 (bad mode) will be reported.

lf

Usage:

```
lf [option]
```

Purpose and Operation:

Add or remove line feed characters from standard input. Output is sent to standard output. When line feeds are being added to a file they are added immediately after a carriage return. When line feeds are being removed from a file they are removed only if they immediately follow a carriage return.

Option	*Function*
-	remove line feeds
+	add line feeds (default)

Example:

```
list /b/game.bas ! lf - >/d0/basic/game↵↵←————remove line feeds
list /d0/basic/game ! lf + >/b/game.bas↵↵←————add line feeds
```

MSAttr

Usage:

```
MSAttr pathname [options]
```

Purpose and Operation:

Change the attributes of an MS-DOS file where **pathname** specifies the desired file and **option** is one of:

Option	*Function*
R	read only
-R	not read only
H	hidden (don't show in directory)
-H	not hidden (show in directory)
S	system file
-S	not a system file
V	volume name
-V	not volume name
A	set archive bit (to indicate file has been updated)
-A	clear archive bit (to indicate file has not been updated)
-D	not a subdirectory

Note:

MSF ignores the *System* and *Volume* attributes but MS-DOS systems don't. MS-DOS systems allow only one file with the *Volume* attribute set and that file is empty.

When changing the attribute of a subdirectory to make it a regular file, the subdirectory must be empty.

Example:

```
MSAttr /b/dog -r↵                    ←allow writing to the file
MSAttr /b/cmds/copyright h r↵        ←hide file and make it read only
```

MSCopy

Usage:

```
MSCopy source[wildcard] [destination] [-options]
```

Purpose and Operation:

Copy multiple files between MSF and RBF devices. The **source** parameter specifies the directory to copy from. The **wildcard** parameter is the optional name matching specification. It consists of valid file characters and/or asterisk (*) and question mark (?) characters. Where an asterisk is used in a file name, multiple characters at that point in the file name are forced to match. A question mark will force a single character at its position in the file name to match. The **destination** parameter is the directory to copy to (default is current data directory).

Option	*Function*
A	copy subdirectories
C	copy only files which don't match
L	list files names as they are copied
P	prompt before copying
N	don't delete existing files
Y	delete existing files
F	copy all files (including subdirectories) to one directory
M	make subdirectories

Note:

If neither the **n** nor **y** options are used, you will be prompted to delete an existing file. If neither the **f** nor **m** options are used, you will be prompted to make a subdirectory or skip copying the subdirectory. The **f** and **m** options are only useful when the **a** option is used. If the **c** option is used with no wildcard, no files will be copied. If a **wildcard** is specified and the **a** option is used, only subdirectories which match the wildcard will be copied.

Example:

```
MSCopy /d0 /b -lam↵ <-- list files while copying them all from /d0 to /b
MSCopy /d0/cmds/MS* -p #40k↵ <-- prompt before copying files which
                                 begin with MS from the /d0/cmds
                                 directory to the current data directory.
MSCopy /b/basic/?a* /d0/basic09 -c #40k↵ <-- copy files which don't
                                             have a second character of 'a'.
```

MSDir

Usage:

```
MSDir [directory] [options]
```

Purpose and Operation:

Get a directory of MS-DOS files.

Option	*Function*
-H	show files with Hidden attribute set
-W	list multiple files on one line

When listing detailed information about the directory entries, the first column lists the file name. The second column lists the file attributes in the following order:

Attribute	*Description*
A	Archive Flag
D	Sub-directory
V	Volume Name
S	System File
H	Hidden File
R	Read Only

The third column lists the file size followed by the creation date and time.

Example:

```
MSDir /b -w↵ ←————————————list only the names of non-hidden files on drive B
 Directory of /b

OS9BOOT         CMDS            A               B               STARTUP
MSF             AA              ABC             NEWSDISK.3      MSF.BAK
MSDIR           MSF.A           MSFDEFS         MSFORMAT        MSFORMAT.SYM
MSATTR          MSRENAME        MSCOPY          START.UP


MSDir /b/cmds -h↵ ←————————————————————list detailed information for all files
 Directory of /b/cmds

.               -D----        0  08-05-87  10:01:42
..              -D----        0  08-05-87  10:01:42
COPYRIGH.T      ---SHR     1077  08-05-87  10:02:00
ATTR            A-----      645  08-21-87  11:21:11
BACKUP          A-----     1202  08-21-87  11:21:43
```

```
BUILD       A-----      84  08-21-87  11:22:02
CMP         A-----    4121  08-21-87  11:22:27
UTD         -D----       0  08-21-87  12:10:26
MSDIR       A-----     663  08-21-87  13:32:43
MSFORMAT    A-----    2023  08-21-87  13:33:56
MSCOPY      A-----    2008  08-21-87  13:46:12
MSRENAME    A-----     472  07-23-87  13:50:58
```

MSFormat

Usage:

```
MSFormat devname [options]
```

Purpose and Operation:

Format an MS-DOS diskette.

Option	*Function*
-1	Single Sided
-8	8 sectors/track (default 9)
-I=n	set sector interleave (n=1-8) (default 5)
-N=name	set the volume (disk) name
-R	don't prompt for disk ready

Note:

If the -N option is not used, you will be prompted for the volume name.

Example:

```
MSFormat /b -n=basic↵  ←format the disk in drive B and name it BASIC
MSFormat /b -18↵  ←format an MS-DOS version 1 diskette
```

MSRD

Usage:

```
MSRD directory
```

Purpose and Operation:

Remove an MS-DOS directory. If the **directory** is not empty, an error is reported.

Example:

```
msrd /b/basic↵←————————————remove the directory called 'basic'
```

MSRename

Usage:

```
MSRename [filename newname]
```

Purpose and Operation:

Rename an MS-DOS file. If `filename` and `newname` are not specified on the command line, you will be prompted for them. This allows you to rename files whose names contain characters which the Shell would remove.

Note:

MSRename does not check if the new name is unique in the directory so you could end up with two files named the same. You will only be able to access the first one. To access the second file with the same name, the first one will have to be renamed.

Example:

```
MSRename↵ ←————————————————— prompt for file to rename
MSRename /b/cmds/cat dog↵ ←————————— rename file cat to dog
```

Installing MSF

The original MSF disk contains the following directories and files:

/CMDS	/SRC	/MODS
MSAttr	MSAttr.a	msf_man
MSCopy	MSCopy.a	A_1s35t
MSDir	MSDir.a	A_1s40t
MSFormat	MSFormat.a	A_2s40t
MSRename	MSRename.a	A_2s80t
		B_1s35t
		B_1s40t
		B_2s40t
		B_2s80t

The first step is to copy the MSF disk to your system disk. Place the MSF disk in drive **/d1** and your system disk in drive **/d0**. Use the MSCopy command to copy:

```
OS9:/d1/cmds/MSCopy /d1 /d0 -ma #32k
```

or if you want to copy only the commands:

```
OS9:/d1/cmds/MSCopy /d1/cmds /d0/cmds #32k
```

Now make a merged module containing the MSF file manager and device descriptors. The names of the descriptors in the MODS directory are self explanatory as the capabilities of the drives they are for. Choose the ones which match your drives.

```
OS9:chd /d1/mods
OS9:merge msf_man a_2s40t b_2s40t >/d0/msf
```

The above example is for double sided 40 track drives.

The MSF file manager and devices can be installed in two ways. To install them temporarily, simply 'Load' them.

```
OS9:load msf
```

The disadvantage with this method is that it uses a complete memory block (8k) of system memory. This reduces the total number of processes that you can run at the same time.

To install MSF permanently you must add it to your boot file. Use the following procedure to accomplish this after having performed the steps above:

1. Format a blank disk—leave your system disk in drive **/d0** and place an unformatted disk in drive **/d1**.

```
OS9:format /d1
```

2. Use OS9's OS9Gen command to make a boot file on drive **/d1**.

```
OS9:os9gen /d1
/d0/os9boot
/d0/cmds/msf
[Esc]
```

3. Use MSCopy to copy the files from your system disk to the new disk.

```
OS9:MSCopy /d0 /d1 -man #32k
```

4. Label the disk in drive **/d1** as the new boot disk and use it to boot up your system.

Note:

If you are not already using the SDISK3 disk driver software, follow the instructions in the SDISK3 documentation to make a bootable disk. Then use the instruction above to install MSF on the bootable SDISK3 disk.

If you are using the Sardis Technologies DMC floppy disk controller make sure you run the KERNLPATCH program on your new boot disk after running OS9GEN and before copying any files to it. An early version of SDISK3 for the DMC controller will not work with MSF. Contact Sardis Technologies for an update.

New OS9 I$SetStt Functions

The MSF MS-DOS file manager adds some new **I$SetStt** functions.

SS.Attr — Set File Attribute

On entry:

A = path number
B = function code ($85)
X = file attribute (LS Byte)
bit 0 = Read Only
bit 1 = Hidden File
bit 2 = System File
bit 3 = Volume Name
bit 4 = Sub Directory
bit 5 = Archive Flag

If error:

Carry set and error code in B.

SS.Renam — Rename File

On entry:

A = path number
B = function code ($86)
X = points to new name

On exit:

X points past new name.

If error:

Carry set and error code in B.

SS.Allow — Allow Writing to Directories and Entire Disk (@)

On entry:

A = path number
B = function code ($87)

If error:

Carry set and error code in B.

New OS9 I$GetStt Function

SS.Attr — Get File Attribute

On entry:

A = path number
B = function code ($85)

On exit:

X = file attribute (LS Byte)
bit 0 = Read Only
bit 1 = Hidden File
bit 2 = System File
bit 3 = Volume Name
bit 4 = Sub Directory
bit 5 = Archive Flag

If error:

Carry set and error code in B.

CLEARBROOK SOFTWARE GROUP

Printed in Canada