MDOS commands

Filename is divided to two parts **-** *name* and *extension.* Name can be up to 10 chars long, extension is 1 char long. Symbols *, ?, . , cannot be used in filename.
Extensions:
P - BASIC program
N - Number array
C - Character array
B - Bytes
Q **-** Sequence
S - Snapshot
Filename example:
MYGRAPHICS.B
Myprog.P
ALCATraZ.S
aaa.P and **AAA.P** are two different files, **bbb.P** and **bbb.p** is the same file.
File attributes:
H - Hidden file (file does'nt appear in disk catalogue)
S - System file
P - Protected file(not implemented yet)
A - Archive file(not implemented yet)
R - Readable file
W - Writeable file
E - Executable file
D - Deleteable file
Wildcards:
Wildcarrds are very similar to standart MS-DOS wildcards. Examples:
mypic*.B
ko*.*
***.S**
.
test?.B
T??T01.P
?sound??.B
????.N
?123*.P

FORMAT

Description: Formatting floppy disk

Syntax: **FORMAT "*drive:nameofdisk"***

Example: **FORMAT "a:DISK01"**

CAT

Description: Catalogue of actual floppy disk

Syntax: **CAT, CAT "*drive*:*"*, CAT "*nameofdisk:*", CAT "*filename*", CAT "*drive:filename*", CAT "*nameofdisk:filename*", CAT -** (minus)

Example: **CAT "a:", CAT "DISK02"**

CAT- is same command as CAT, but filesize and attributes doesn't appear

LIST*

Description: Prints actual state of diskdrive and system.

Syntax: **LIST***

Example: **LIST***

LOAD*, SAVE*, MERGE*

Description: Load, save or merge program, data, array. Filename must be entered.

Syntax: **LOAD* "*drive:filename",* LOAD* "*filename*", LOAD* "*drive:filename*" CODE, LOAD* "*filename*" CODE**

SAVE* same as above

MERGE* "*drive:filename*", MERGE* "*filename"*

Example: **LOAD* "a:R-TYPE", SAVE* "b:MyPic99" CODE 32768,6912 , MERGE* "TOWER"**

MOVE

Description 1: File copying

Syntax: **MOVE "*nameofdisk1:filename1","nameofdisk2[:filename2]*"**

Example: **MOVE "disk01:robocop$.b","disk99"**

This works on one drive too. MDOS asks for change disks. If ***filename2*** is not used, target file will have the same filename as source file. Wildcards can be used.

Description 2: Actual drive selection

Syntax: **MOVE "*drive*"**

Example: **MOVE "a:", MOVE "B:"**

ERASE

Description: Erase specified file(s) from actual device

Syntax: **ERASE "*[drive:]filename"***

Example: **ERASE "A:OLDPIC.P", ERASE "*.*"**

Wildcards can be used. File couldn't be erased if attribute ***D*** is not set.

LET FN

Description: Rename file

Syntax: **LET FN("*oldfilename*")="*newfilename*"**

Example: **LET FN("11codeOLD.B")="11codeNEW.B"**

LET ATTR

Description: Set attributes for specified file

Syntax: **LET ATTR(*filename)*="*attributes"***

Example: **LET ATTR(BACKUP.P)="RWED"**

Wildcards can be used.

POKE#

Description: Write data to diskdrive's 2kB shadow RAM.

Syntax: **POKE#*adr*,*data***

Example: **POKE#97,50**

Shadow RAM is mapped to adress 3800h and is 2048 bytes long. Adressing begins 0000h, so you enter POKE#0000h,XX not POKE#3800h,XX.

READ*

Description: Read specified sector (512 bytes) from disk and save it to specified adress

Syntax: **READ*"*drive:[filename]*",*sector*,*adress***

Example: **READ*"a:",1,32768**

RESTORE*

Description: Read specified 512 bytes long data from specified adress and write it to disk on specified sector.

Syntax: **RESTORE**"drive:[filename]",sector,adress***

Example: **RESTORE*"b:data123.b",10,40000**

OPEN#

Description: Connect specified file(s) to the specified channel

Syntax: **OPEN#*n*,*["drive:INfilenameQ"]*,*["drive:OUTfilenameQ"]***

Example: **OPEN#2,"A:dataIN.Q","B:dataOUT.Q" , OPEN#3,"a:prndata.Q"**

With this command you can connect specified file(s) to the specified channel. Then it is possible to read or write from/to file with standart commands (PRINT#n;"blablabla" etc.) File must have the **.Q** extension. CAT output cannot be send to the channel.

CLOSE#

Description: Close specified channel

Syntax: **CLOSE#*[n]***

Example: **CLOSE#3**

PRINT*

Description: Print specified sequention file to display.

Syntax: **PRINT*"*[drive:]filenameQ*"**

Example: **PRINT*"B:Sequent3.Q"**

RUN

Description: Simple boot - LOAD *"run.p"

Syntax: **RUN**

Example: **RUN**

If BASIC program is in memory, command starts this program. If it's not, command loads ***RUN.P*** or ***run.P*** from actual device. RUN.P must be saved with LINE xx command (SAVE *"run.p" LINE 10).

Simple MDOS description

MDOS supports max. two diskdrives. BASIC commands, as you can see above, are very strong and comfortable. Another disadvantage - MDOS doesn't support machine code routines. It is posible to call MDOS and run his subroutine but this is very incompatible. There is TWO different versions of MDOS. Firstly Didaktik manufactured external diskdrive called D40. It is based on WD2797. Later came external D80. Few pieces was with WD2797, but then Didaktik completely rebuilded D80 hardware and the heart of new D80 was some INTEL 8XXX, so the MDOS must be rewriten. Then came DIDAKTIK KOMPAKT, that is combination of DIDAKTIK M with a kind of D80. From BASIC all diskdrives are compatible, but machine code progs where are direct calls to MDOS doesn't functioning on some configurations of D40, D80 and probably some Kompakts too. Anyway Kompakt is very good and stable machine with (imho) excellent diskdrive.

Disk capacity is 720kB, but it is posible to format disk to around 800kB. Disk must be double sided. Standartly there is 80 tracks on disk, every track contains 9 sectors, every sector has 512 bytes. Max. count of sectors on disk can be 1640(= 82Tx10S), but this cannot be read by every drive .... MDOS using logical adressing of sectors, begins by sector 0 on track 0. FAT is 12 bits long andthere can be ONE directory and max. 128 files on disk. Directory occupies sectors 7-14. Directory is simple 32 bytes long table.

Boot sector description

bytes	*description*
128-191	System variables with format informations (this section is a BIG garbage, only few people can understand it)
192-201	Name of disk (10 chars)
202,203	randomly generated 16-bit number for recognize two disks with same name
204-207	MDOS identification bytes, here staying text "SDOS", so MDOS can identify his format on disk
177	bit3= log1 if 40 tracks drive is connected, bit4=always log1 - double sided disk
178	number of tracks on one side
179	number of sectors on one track
180	always 0
181-183	copy of bytes 177-179

Other bytes of bootsector are not used, so you can write there your own infomations.

Directory table description

bytes	*description*
0	File extension (P,B,N,C,S,Q)
1-10	Filename
11,12	Length of file
13,14	Start adress if it is bytes or start line if it is BASIC program
15,16	Length of BASIC program without variables
17,18	FAT entry point, logical adress of 1st sector that file occupies
19	always 0
20	Attributes, every bit stays for one attribute (HSPARWED > bit7=H, bit6=S, bit5=P .....)
21	Third byte for file length if it is bigger than 65535 bytes (useful for sequential files)
22-31	filled with 229

Assembled by z00m in 8/2001

with a litlle help of the original D40 manual.