Command Reference Manual

(Volume Two)

Also for:
VIC20,
Plus4, C16,
C128

Copyright Notice

The Professional version of 64HDD is available for purchase from the 64HDD website or authorised dealers.

Code is being developed by Nicholas Coplin,
Email: ncoplin@lycos.com
Website: www.64hdd.com

Table of Contents

Disk Access Command Reference Guide

Directory and Disk Image Access /Loading

Loading and saving, reading and writing from disk images and the native PC file system should be completely transparent to most Commodore programs. The standard access commands used with your 1541 will work, however there are some extensions provided to access new 64HDD specific features.

Command: Select and/or view 64HDD directory
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
LOAD”$drive:/path=type”,device
LOAD”$drive:/path=type[,options]”,device
OPEN fileno,device,15,”$drive:/path=type”:CLOSE
```

(Professional version only)

Where types allowed include:

P	for PRG files in MSDOS mode (PRG can also be used) and N64 and P00 files *(Professional version Only)*
S	for SEQ files in MSDOS mode (SEQ can also be used) and S00 files *(Professional version Only)*
U	for USR files in MSDOS mode (USR can also be used)
R	for REL files in MSDOS mode (REL can also be used)
C	for CBM files in disk image file systems
B	for directories in MSDOS and disk image file systems
G	gives GEOS information such as type/purpose
DIR	for MSDOS directories
D64	for D64 images
D71	for D71 images
D81	for D81 images
D2M	for D2M images (from FD2000 made by 1581COPY)
D*	for all supported disk images
T64	for T64 images
LNX	for LNX images
MSD	for all other files
Z	for all ZIP files *(Professional version Only)*

Wildcards ? and * permitted for filtering the directory listing

Options include:

I	for giving tally of blocks used by displayed files
II	for giving tally and other extended information, including subtotals for file(s), dir(s), image(s)

Examples:

Load current directory assigned to this device for viewing (use LIST as per CBM practice to see it)

```
LOAD"$",11
```

Load only directory header and blocks free information

```
LOAD"$$",11
```

Load current directory but only files matching this type

`LOAD"$,D64",11` *(only works in MSDOS mode)*

`LOAD"$,D64,P",11` *(only in MSDOS mode, multiple types)*

```
LOAD"$T?ST*,P",11
```

Set C:\ as current directory for future commands

```
LOAD"$C:/",11
```

Set C:\ as current directory for future commands

```
OPEN1,11,15,"$C:/"
INPUT#1, EN,EM$,T,S
CLOSE1
PRINT "NEW PATH:";EM$
```

Change to D:\ as current directory for future commands

```
LOAD"$D:/",11
```

Set "this-dir" as directory for future commands

```
LOAD"$THIS-DIR",11
```

Get directory listing with extended information

```
LOAD"$*,II",11
```

Attach "this D64 image" as an emulated disk for future commands

```
LOAD"$THIS.D64",11
```

Dettach "this D64 image" and set root on current drive

```
LOAD"$/",11
```

Errors:

- With the `OPEN` format of the command the error message returns the new path attached, but error code will be set to [0].
- If a partition change is requested, but not available, then error [77] may result
- If a partition change or drive is requested that is not available or would normally result in a "Drive Not Ready" message in MSDOS then error [78] may result

Notes:

- Image files will be listed with the file extension in both the name and the type. The image can only be attached if the file extension is given in the name (else a file of that name is assumed).

- The MSDOS directory listing allocates an extra digit for block size information (this is because files and disk images can have 4digit values for their size – not possible on a 1541!)
- The MSDOS directory separator "\" is exchanged for "/" on the CBM machine as the CBM keyboard has no "\" key.
- When an image or MSDOS directory is attached the directory is sent if a LOAD was issued. It is not sent if the command channel was used.
- An image can be detached by setting the directory to a known MSDOS location, eg. `LOAD"$/",11`. Be sure to set to a known /existing directory. `LOAD"$..",11` can also be used to return to the MSDOS directory the disk image was attached from.
- In MSDOS modes, relative paths may be used (eg. `LOAD"$..",11`) but remember that path length is limited to 255 characters by some versions of MSDOS. It is recommended to frequently set directory to root when using relative paths.
- In MSDOS mode, a new drive may be logged by giving the absolute path to a directory, for example the root directory: "D:\"
- Pressing RUN/STOP will abort the directory load.
- With the OPEN command format the directory is set for the device, but no listing is sent on this channel. The device's directory can be retrieved using the normal CBM directory read from secondary address 0 for the device with file name `$`, eg `OPEN fileno,device,0,"$"` and use `GET#`
- In MSDOS mode, "blocks free" is number of blocks available on the drive, limited to 63999 because of BASIC line number limitations
- In LNX and T64 modes, "blocks free" will always be zero as write support is not provided for these image types.
- Incompatibility: because $name is now used to change to the directory or disk image, if "name" is not a directory a file not found message or default directory load will be performed. This will have some incompatibility with programs that try to identify "blocks free" by reading the information after they execute a `OPEN1,11,0,"$!#$%"` command. To patch such instances, change the last character to a wildcard, eg `"$!#$*"`

Command: Request EZ-Load format for 64HDD directory **(Professional version only)**

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
LOAD"$!EZ:drive:/path=type",device
LOAD"$drive:/path=type,EZ",device
```

Where types allowed include:

As for regular directory requests

Examples:

Load current directory assigned to this device for viewing (use LIST as per CBM practice to see it)

```
LOAD"$!EZ:*",11
```

Errors:

- Same as for normal directory requests

Notes:

- Complete LFN name displayed for files, disk images and directories (LFN given on second line – use short name to access)
- Professional users can activate EZ-Load permanently by command- options
- Display of prefix for FastLoader, X-Loader or PwrLoad can be requested by setting command-line options

---

Command: Fast 64HDD Directory Reading

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `OPEN fileno,device,3,”$drive:/path”`

Examples:

Get current directory using fast read

```
1 OPEN 1,11,3,”$”
2 INPUT#1,D$
5 INPUT#1,BL,FL$,TY$
6 IF TY$=”FRE” THEN 10
7 PRINT BL,FL$,TY$
8 GOTO 5
10 PRINT BL;FL$
15 CLOSE 1
```

Errors:

- See previous

Notes:

- Directory access with secondary address of 3 will initial fast directory read format (this is not supported by real CBM DOS).
- The fast format means that directory information only is sent and not the normal BASIC format detail as per a read of $ with secondary channel of 0
- First INPUT# gets the directory or disk header. This contains information such as disk id, disk name, path or file system depending upon what is attached. The DOS version in the header can be used to identify which file system is active according to the following:

1T	T64 image attached
1L	LNX image attached
PC	MSDOS directory
LF	MSDOS directory, but with LongFileName support
2A	D64 image (unless disk image has corrupt DOS Version)
2A	D71 image (unless disk image has corrupt DOS Version)
3D	D81 image (unless disk image has corrupt DOS Version)
1H	D2M image (unless disk image has corrupt DOS Version)

- Second and subsequent INPUT# fetch three items in the format:
  Blocks, Filename$, Type$

- Filetype FRE is reserved for use as a flag that the details just received is block free information and there is no further data to be sent
- The notes for normal directory access should also be reviewed, but the 63999 block free limitation does not apply.
- LFN file type is used to signify that entry is the Long File Name of the previous file shown.

File Handling Commands

Command: Open File for LOAD/SAVE/Read/Write

Applicability:

☐ All ■ MSDOS ■ D64 ■ D71 ▨ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
OPEN fileno,device,sa,“filename,type,mode”
LOAD “filename,type,mode”,device,sa
```

Examples:

Open file to write ASCII data to it

```
OPEN 1,11,1,”TESTFILE,S,W”
FOR X=1TO10:PRINT#1,”NUMBER:”;X:NEXT
CLOSE 1
```

Open file to read ASCII data from it

```
OPEN 1,11,1,”TESTFILE,S,R”
FOR X=1TO10:INPUT#1,A$:PRINT A$:NEXT
CLOSE 1
```

Save currently loaded basic program

```
SAVE”TESTFILE”,11
```

Load program file

```
LOAD”TESTFILE”,11
```

Errors:

- errors returned for file not able to be open or found or existing

Notes:

- In the MSDOS file system, the name is limited to 8characters unless LFN support is selected and installed, in which case the limit is 16characters. If a type is specified, then this is converted to a 3character extension. If no file type is specified, PRG is assumed.
- PRG file load supported in all file systems.
- SEQ file reads supported in MSDOS and disk image file systems.
- RUN/STOP key press supported for PRG files.
- FILE NOT FOUND supported for PRG and SEQ formats
- File creation supported for MSDOS, D64, D71 and for root partition for D81 files. In the supported file systems, file exist warning is given, but if not tested by the user program the file will be overwritten
- Writes to disk images do not use the standard CBM-DOS interleave. The interleave is irrelevant for a virtual disk, and instead blocks are written in the next available order starting at Track=1, Sector=0. The only thing you may notice is that these disk may be slightly slower to read when copied back to real media.

Error Channel Reading

Command: Reading the error channel
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: See example,
Return format is:
err_number, err_string, err_track, err_sector

Examples:

Read error information on this "device" (using INPUT#1)

```
OPEN 1,11,15
INPUT#1, EN, EM$,ET,ES
PRINT EN, EM$,ET;ES
CLOSE 1
```

Read error information on this "device" (using GET#1)

```
10 OPEN 1,11,15
20 GET#1, A$:PRINT A$;
30 IF ST<>64 THEN 20
40 CLOSE 1
```

Errors:

Notes:

- Error information returned regardless of mode in which the emulator is operating in
- Only last error flagged is return
- Error may not be a fault, but informational. Error number will be set to 0 in this case, or error 99 if emulator interrogated before error flag set.
- Errors 8x are new and indicate if error was a result of the 64HDD emulator not supporting the previous command channel request.
- Dependent Devices only respond with errors compatible with error messages to the real CBM device plus the appropriate error 8x and 99 codes.

Standard Commodore Drive Commands

Below are listed the standard commands supported by 64HDD. These work identically to your regular 1541 or other Commodore drive, however 64HDD provides for these to work on both disk images and the native MSD/LFN file system.

Command: Scratch file(s) from current device /directory

Applicability:

☐ All ■ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“S:filename[;M]”
“S0:filename[;M]”
“SCRATCH:filename[;M]”
“SCRATCH0:filename[;M]”

    Where:
         ;M   specified that filenames should be treated as
              explicitly give, as is the case when working
              with MSDOS files.
              This option is only valid in the MSDOS file
              system.
```

Examples:

Scratch this file only

```
OPEN 1,11,15,”S:TESTFILE”
CLOSE 1
```

Scratch this MSDOS file only

```
OPEN 1,11,15,”S:TESTFILE.JPG;M”
CLOSE 1
```

Scratch files matching this spec

```
OPEN 1,11,15,”S:TE?TFIL*.PRG”
CLOSE 1:
```

Scratch files matching this spec (all extensions /types)

```
OPEN 1,11,15,”S:TE?TFIL*”
CLOSE 1:
```

Errors:

- Error [62] returned for file not able to be found or can not be scratched
- File delete count may not be valid if using LFN mode (this is a limitation of the external LDEL program)

Notes:

- In the MSDOS file system, 8.3 filename validity is omitted an error 62 returned if file not found

- In the MSDOS file system, if ";M" has not been specified attempts will be made to delete the given filename with extension ".PRG", ".SEQ" and ".USR". This should not disturb existing CBM programs as they would only expect one file of a given name in each as type checking is not done by CBM DOS.

Command: **Rename file on /in current device /directory**
Applicability:

☐ All ■ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“R:newfilename=oldname[;M]”
“R0:newname=oldname[;M]”
“RENAME:newname=oldname[;M]”
“RENAME0:newname=oldname[;M]”

Where:
      ;M   specified that filenames should be treated as
           explicitly give, as is the case when working
           with MSDOS files.
           This option is only valid in the MSDOS file
           system.
```

Examples:

Rename this file

```
OPEN 1,11,15,”R:NEWFILE=OLDFILE”
CLOSE 1
```

Rename this MSDOS file

```
OPEN 1,11,15,”R:NEWFILE.JPG=OLDFILE.JPG;M”
CLOSE 1
```

Errors:

- Errors [62] and [60] returned as appropriate
- If using LFNs Ok is always reported (limitation of the external LREN program)

Notes:

- In the MSDOS file system, if ";M" has not been specified, three attempts are made to rename the file using extensions PRG, SEQ and USR. Each rename attempt is to the same "file type", eg PRG will be renamed to PRG.
- In the MSDOS file system, 8.3 filename validity is omitted for both oldfile and newfile as a robust "check and approve" strategy still needs development, hence correct renaming is up to the user and presently requires filename extensions to be given.
- In the MSDOS file system, do not put space characters on either side of the "="

Command: Copy file on /in current device /directory to new name
Merge multiple sequential files to one combined file

Applicability:
☐ All ■ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax (copy):

```
“C:backup=original[;M]”
“C0:backup=original[;M]”
“COPY:backup=original[;M]”
“COPY0:backup=original[;M]”
```

Where:

;M	specified that filenames should be treated as explicitly give, as is the case when working with MSDOS files. This option is only valid in the MSDOS file system.

Syntax (merge):

```
“C:combined=file1,file2,file3,file4”
“C0:combined=file1,file2,file3,file4”
“COPY:combined=file1,file2,file3,file4”
“COPY0:combined=file1,file2,file3,file4”
```

Examples:

Copy this file

```
OPEN 1,11,15,”C:BACKUP=ORIGINAL”CLOSE 1
```

Copy this file

```
OPEN 1,11,15,”C:BACKUP.TAS=ORIGINAL.TAS;M”CLOSE 1
```

Copy this disk image from current location to a:

```
OPEN 1,11,15,”C:A:/GAMES.D64=GAMES.D64”
CLOSE 1
```

Merge these files

```
OPEN 1,11,15,”C:All=FILE1,FILE2,FILE3”
CLOSE 1
```

Merge these files

```
OPEN 1,11,15,”C:All.SEQ=FILE1.SEQ,FILE2.SEQ,FILE3.SEQ”
CLOSE 1
```

Errors:

- Errors [62] and [25] returned as appropriate

Notes:

- In the MSDOS file system, if “;M” has not been specified, three attempts are made to copy the file using extensions PRG, SEQ and USR. Each copy attempt is to the same “file type”, eg PRG will be renamed to PRG.
- In the MSDOS file system, 8.3 or 16.3 filename validity is checked for <u>merge</u> and SEQ file type is forced (ie only SEQ files can be merged).
- In the MSDOS file system, 8.3 or 16.3 filename validity is omitted for <u>copy</u> as a robust “check and approve” strategy still needs development, hence correct naming is up to the user and presently requires filename extensions to be given.

- In the MSDOS file system, 8.3 or 16.3 filename name may be the name of a disk image and if a full pathname given for the destination, a disk image may be copied between drives (note: the source must be in the default location and no path given)
- In the MSDOS file system, do not put space characters on either side of the "=" or "," unless it is truly part of the LongFileName.
- In the MSDOS file system, there is no limitation on the number of files which may be merged.
- For disk image formats, only the simple COPY version of the command is so far supported. The MERGE version is not yet available.

Command: Initialize device
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
"I"
"I0"
"INITIALIZE"
"INITIALIZE0"
```

Examples:

Initialize this device
```
OPEN 1,11,15,"I"
CLOSE 1
```

Errors:
- Always error [0] returned

Notes:
- In the MSDOS file system, the command is acknowledged, but not acted upon.

Command: Validate device /disk
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
"V"
"V0"
"VALIDATE"
"VALIDATE0"
```

Examples:

Validate this device
```
OPEN 1,11,15,"V"
CLOSE 1
```

Errors:
- Always error [0] returned

Notes:

- In the MSDOS file system, the command is acknowledged, but not acted upon.

Command: Warm boot / Cold boot
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `“UI”`
`“UJ”`
`“U9”`

Examples:

Warm boot this device

```
OPEN 1,11,15,”UI”
INPUT#1, EN,EM$,T,S
CLOSE 1
```

Errors:

- Error [73] returned with EM$ containing 64HDD version message. The last four letters of EM$ are “EMUL” can be used to detect that this is an emulated drive and as such advanced 64HDD features can be implemented.

Notes:

- Currently the error message is identical regardless of file system being used
- Active devices and partitions remain unchanged
- `UI-` will set protocol timings to 1540 (ie VIC20 or blanked screen required on C64/C128). With this timing the data valid pulse width is shorter and does not take into account the C64’s bad-line requirements.
- `UI+` will return timings to 1541 protocol

Command: 1581 Partition Selection
Applicability:

☐ All ☐ MSDOS ☐ D64 ☐ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `“/”`
`“/:”`
`“/0:”`

Examples:

Select partition called “SUB-PART”

```
OPEN 1,11,15,”/SUB-PART”
INPUT#1, EN,EM$,T,S
CLOSE 1
```

Select root partition

```
OPEN 1,11,15,”/”
INPUT#1, EN,EM$,T,S
CLOSE 1
```

Errors:

- Error [77] returned if partition invalid, else error [2] if partition is selected with track and sector detailing the start/end track for the partition.

Notes:

- Refer to the 1581 User Manual for further detail
- Attaching a new D81 disk image always logs the root directory

---

Command: 1581 Partition Creation

Applicability:

☐ All ☐ MSDOS ☐ D64 ☐ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“/name,”+CHR$(starting track)+CHR$
     (starting sector)+CHR$(<# of sectors)+CHR$(>#
          of sectors)+",C"
“/:name,”+CHR$(starting track)+CHR$
     (starting sector)+CHR$(<# of sectors)+CHR$(>#
          of sectors)+",C"
“/0 name,”+CHR$(starting track)+CHR$
     (starting sector)+CHR$(<# of sectors)+CHR$(>#
          of sectors)+",C"
```

Examples:

Create partition called “SUB-PART” starting Track=60, 120blocks long

```
OPEN 1,11,15
PRINT#15,"/0:SUB-PART,"
     +CHR$(60)+CHR$(0)+CHR$(120)+CHR$(0)+",C"
INPUT#1, EN,EM$,T,S
CLOSE 1
```

Errors:

- Error [0] returned if partition successfully created.
- Error [62] if name exists
- Error [65] if block allocation fails (already allocated)
- Error [72] if trouble creating directory entry

Notes:

- Refer to the 1581 User Manual for further detail, particularly about sizing limitations, etc
- Note: the created sub-partition needs formatting before it can be used as a directory. 64HDD may not support N: command at this present time.

Direct Access Commands

Drive Simulation Model

Those who have extensively programmed the 1541 drive will realise that it is possible to interrogate the drive for information or to manipulate the buffers, and memory locations to support functions not normally available by user commands. Many commercial programs make use of Block-Read and buffer Memory-Reads to identify which drive is attached to the system and to access directory information.

Even though 64HDD does not as yet support CPU emulation, each drive attached to the system can have a virtual map associated with it. For the virtual map to be available the following conditions must be satisfied:

- A disk image must be attached (a T64 file – or tape unit – does not have a memory map!) Note: the MSDOS system is also presently without a memory map.
- The specific ROM file must be available in the +sysdir location. Each disk image type has its own specific ROM. A generic ROM is supplied with the 64HDD distribution as shown in the table:

Image Type	**Drive Type**	**ROM filename**	**Included in ZIP distribution**
D64	1541	DOS1541.ROM	Yes
D71	1571	DOS1571.ROM	Yes
D81	1581	DOS1581.ROM	Yes
D2M	FD2000	DOS2000F.ROM	No
MSD/LFN	CMDHD	DOSCMDHD.ROM	No

- XMS must be available. XMS is extended memory, which is available in DOS only when a driver such HIMEM.SYS is included in the config.sys settings. 64HDD assigns just over 64k of RAM for each device number attached to the 64HDD system. Therefore if there are 8 possible devices, 512kb of XMS is needed. 64HDD will work without XMS, but direct access commands will fail.

Every virtual map has a 64k linear memory layout. ROM is implemented as write-protected RAM in this 64k layout in the locations specific to the drive. The rest of the space is allocated as RAM, and presently simulation of the I/O devices is not included. Likewise, the RAM contents in locations $0000-$2FFF are not as yet fully simulated.

Direct Access Commands normally operate through buffers. Buffer 0 is at hex location $0300-$03FF, buffer 1 at $0400-$04FF, etc. 64HDD supports 15 such buffers regardless of the device type being emulated. The buffer number is normally requested by the # filename; for example OPEN 1,11,2,”#2” would request /assign buffer #2 and commands such as U1 (disk block read) will then fill location $0500-$05FF. Request of a NULL buffer assigns buffer #0, not the next “free” buffer, so care must be taken when programming specifically for 64HDD. Typically, any serious drive programming requests buffers by number or only uses one NULL buffer thus 64HDD can generally cope without buffer conflicts. Buffers reserved for BAM blocks are the same as the associated drive type, eg D64 ➔ 1541 map.

The details about specific Direct Access Commands will not be described below, instead the keen user is directed to the user manuals which come with the Commodore disk drives or to reference books such as “Inside Commodore Dos”.

Command: Block Read
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ■ D2M ☐ T64 ☐ LNX

Syntax:
```
U1
B-R
BLOCK-READ
```

Notes:

- The B-R and BLOCK-READ versions work exactly as the U1 version does. The original bugs have been corrected.
- The drive number parameter is always ignored and 64HDD refers only to the drive that received the command.

Command: Block Write
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ■ D2M ☐ T64 ☐ LNX

Syntax:
```
U2
B-W
BLOCK-WRITE
```

Notes:

- The B-W and BLOCK-WRITE versions work exactly as the U2 version does. The original bugs have been corrected.
- The drive number parameter is always ignored and 64HDD refers only to the drive that received the command.

Command: Block Allocate
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
B-A
BLOCK-ALLOCATE
```

Notes:

- The original bugs have been corrected as BAM is updated directly on disk image and not only to memory.

- The drive number parameter is always ignored and 64HDD refers only to the drive that received the command.

Command: Block Free
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `B-F`
`BLOCK-FREE`

Notes:

- The original bugs have been corrected as BAM is updated directly on disk image and not only to memory.
- The drive number parameter is always ignored and 64HDD refers only to the drive that received the command.

Command: Block Pointer
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ■ D2M ☐ T64 ☐ LNX
Syntax: `B-P`
`BLOCK-POINTER`

Notes:

- The drive number parameter is always ignored and 64HDD refers only to the drive that received the command.

Command: Memory Read
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ■ D2M ☐ T64 ☐ LNX
Syntax: `M-R`

Notes:

- If number of bytes is not specified, 13 may be accidentally transferred as the CR at the end of the PRINT# statement will be parsed.

Command: Memory Write
Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ■ D2M ☐ T64 ☐ LNX
Syntax: `M-W`

Notes:

- If number of bytes is not specified, 13 may be accidentally transferred as the CR at the end of the PRINT# statement will be parsed. In this case the actual data would be invalid values.

- If the write operation would involve over-writing the ROM, then the entire M-W is cancelled. This will not normally be a problem as most CBM drives do not have RAM next to ROM in their memory map.

Extended Drive Commands

Below are listed the special commands supported by 64HDD. These allow use of additional features such as creating and manipulating directories, reading the time and date, and so on. Many of these commands are similar to those used by CMD devices, but often an additional 64HDD syntax is also provided for specifying additional options.

Device Dependent Commands

Command: Copy (Kopy) files between 64HDD virtual devices **(Professional version only)**

Applicability:
☐ All ■ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

(copy):

```
"K:backup#=original"
"KOPY:backup#=original"
```

Where:

backup#	is other 64HDD device number with either a directory or supported disk image attached.

Examples:

Copy this file from the current 64HDD device(11) to another(10)

```
OPEN 1,11,15,"K:10=ORIGINAL"CLOSE 1
```

Errors:

- Errors [62] and [25] returned as appropriate
- Error [74] if target device is not active
- Sometimes the error channel has to be read twice

Notes:

- The file is first read to a temporary file in the 64hdd/system directory, and then to the target drive. The transfer is done entirely by 64HDD.
- Both devices must be 64HDD drives. You cannot KOPY to a non-64HDD drive. You can KOPY from disk image to image, LFN/MSD to image, or image to LFN/MSD.
- Wildcards can be used, but only the first filename match is used
- LFN names are preserved if transferring from disk image to the LFN file system. The filetype is preserved.
- LFN names are only preserved when transferring from LFN to disk image if the full file name is given. The filetype is preserved only if specified, eg ,S for SEQ in the source name.

Command: Make MSDOS directory
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“MD:new_dirname”`
`“MD0:new_dirname”`

Examples:

Create this new directory

```
OPEN 1,11,15,”MD:NEW-DIR”
CLOSE 1
```

Errors:

- Error [77] returned if failure to create

Notes:

- In the MSDOS file system, 8.0 filename validity is enforced. The new directory is created branching from the current path. Wildcards are not permitted.

Command: Change MSDOS directory
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“CD:dirname”`
`“CD0:dirname”`

Examples:

Change to this directory

```
OPEN 1,11,15,”CD:NEW-DIR”
CLOSE 1
```

Change to the root

```
OPEN 1,11,15,”CD:/”
CLOSE 1
```

Change to the root of drive D:

```
OPEN 1,11,15,”CD:D:/”
CLOSE 1
```

Change “up” to parent directory or release disk image

```
OPEN 1,11,15,”CD←”
CLOSE 1
```

or

```
OPEN 1,11,15,”CD..”
CLOSE 1
```

Errors:

- Error [77] returned if failure to create

- Error [2] returned if successful

Notes:

- In the MSDOS file system, 8.3 filename validity checking is omitted so as to be compatible with directory structures created before 64HDD. Therefore it is up to the user to correctly enter valid directory names. Wildcards are not permitted.
- To log a new drive use absolute naming for the directory, eg “d:/”

Command: Remove MSDOS directory

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“RD: dirname”`
`“RD0: dirname”`

Examples:

Remove this directory

```
OPEN 1,11,15,”RD:TEMP”
CLOSE 1
```

Errors:

- Error [77] returned if failure to delete

Notes:

- In the MSDOS file system, 8.3 filename validity checking is omitted so as to be compatible with directory structures created before 64HDD. Therefore it is up to the user to correctly enter valid directory names. Wildcards are not permitted.

Command: Activate a Disk-Flip List

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“XS:”`
`“XS:listfile.DFL”`

Examples:

Clear the disk-flip listing (clear partition definitions 001 though 010)

```
OPEN 1,11,15,”XS:”
CLOSE 1
```

Assign the disks defined in MYGAME.DFL to partitions 001 though 010

```
OPEN 1,11,15,”XS:MYGAME.DFL”
CLOSE 1
```

Errors:

- Error [62] returned if DFL file not found

Notes:

- The command only works in an MSDOS location (since the DFL file must be an MSDOS file)
- listfile.DFL is any 8.3 filename text file listing the name of disk images or paths to be used for a multi-disk application. Each location or disk image name is listed on one line (up to 10 can be listed). If the path to the disk images is not given as 64HDD will make these be the current path.
- This command is compatible with the XS: command used by the SD2IEC device. It need not have the DFL extension.

Command: Locking and Unlocking file(s) from a directory **(Professional version only)**

Applicability:

☐ All ■ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
"F-L:filename[;M]"
"F-L0:filename[;M]"
"F-U:filename[;M]"
"F-U0:filename[;M]"

Where:
          Filename can be a wildcard using * or ?
          [;M] is used to specify an exact MSD filename,
               including extension
```

Examples:

Lock (set write protection) for a file called TESTFILE

```
OPEN 1,11,15,"F-L:TESTFILE"
CLOSE 1
```

Lock the MSDOS file called TESTFILE.JPG

```
OPEN 1,11,15,"F-L:TESTFILE"
CLOSE 1
```

Or

```
OPEN 1,11,15,"F-L:TESTFILE.JPG;M"
CLOSE 1
```

Unlock files matching this spec

```
OPEN 1,11,15,"F-U:TE?TFIL*"
CLOSE 1:
```

Errors:

- Error [3] returned for file(s) successfully locked
- Error [4] returned for file(s) successfully unlocked
- Error Track shows the number of files locked or unlocked (only for disk images)
- Error codes are those used by CBM DOS 10.0 written for the C64DX (otherwise known as the C65)

Notes:

- In the MSDOS file system, 8.3 filename validity is omitted and a wildcard is appended to allow the request to apply regardless of file extension. To specify a particular file extension, use the `;M` option
- In the MSDOS file system, ATTRIB is used to change the state of the read-only attribute. ATTRIB is only compatible with ShortFileNames!!!
- Some DOS wedges such as that in JiffyDOS have a different definition for commands beginning with "F". You will need to use a quote:
    @"F-L:filename

@"F-U:filename

Command: Create and Name Blank Disk Image

(Professional version only)

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
N:diskname.typ[,id]
N0:diskname.typ[,id]

Where:
     "diskname"must be a valid MSD filename
     ".typ"         is .D64 .D71 .D81 .D2M
     id             is ignored
```

Examples:

Create a blank D64 called MY-PROGS

```
OPEN 1,11,15,"N:MY-PROGS.D64"
CLOSE 1:
```

Errors:

- Error [0] if successful, else
- Syntax error [31] if invalid image type given, error [77] if image cannot be created or attached (eg disk is write protected)

Notes:

- Command only works if in a MSD directory. The resultant disk will be created in that directory.
- Diskname can be up 8characters long and should not contain illegal MSD characters
- The disk's header is renamed to diskname, however the disk ID is left as default
- 64HDD will attach the device to the new disk ready for access
- The appropriate BLANK???.COM file must be on the MSDOS path for this command to work.

Device Independent Commands

Command: Get /Set 64HDD Time

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
T:hhmmss
T0:hhmmss
TIME:hhmmss
TIME0:hhmmss
```

Examples:

Get time from 64HDD and set CBM time string

```
OPEN 1,11,15,"T:"
INPUT#1, EN,EM$,ET,ES: TI$=LEFT$(EM$,6)
CLOSE 1:
```

Set 64HDD RTC time to 13:20:30 (hhmmss, 24hr clock)

```
OPEN 1,11,15,"T:132030":CLOSE 1:
```

Errors:

- Current time returned as the error text. EN, ET, ES always 0
- Syntax error [30] returned if format is too long /short

Notes:

- "hhmmss" must be an ASCII string
- works regardless of disk image /directory selected
- no checking that time setting was successful or that time was valid

Command: Get /Set 64HDD Date

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
D:dd/mm/yyyy
D0:dd/mm/yyyy
DATE:dd/mm/yyyy
DATE0:dd/mm/yyyy
```

Examples:

Get date from 64HDD and print string

```
OPEN 1,11,15,"D:"
INPUT#1, EN,EM$,ET,ES: PRINT "DATE: ";EM$
CLOSE 1:
```

Set 64HDD RTC date to 30/01/1999 (dd/mm/yyyy)

```
OPEN 1,11,15,"D:30/01/1999"
CLOSE 1:
```

Errors:

- Current date returned as the error text. EN, ET, ES always 0
- Syntax error [30] returned if format is too long /short

Notes:

- “dd/mm/yyyy” must be an ASCII string
- though “/” are returned other delimiters may be used for setting, eg: space and “-“
- works regardless of disk image /directory selected
- no checking that date setting was successful or that date was valid
- Y2K compliance is a function of the RTC chip used in the PC, and not the 64HDD program

---

Command: Get 64HDD Day
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
DAY:
DAY0:
```

Examples:

Get day-of-week from 64HDD and print string

```
OPEN 1,11,15,”DAY:”
INPUT#1, EN,EM$,ET,ES: PRINT “TODAY IS: “;EM$
CLOSE 1:
```

Errors:

- Current day-of-week returned as the error text. EN, ET, ES always 0

Notes:

- works regardless of disk image /directory selected
- full name of day returned, eg: MONDAY
- Y2K compliance is a function of the RTC chip used in the PC, and not the 64HDD program
- Availability of day-of-week is a function of the RTC type and MSDOS version being used as the emulator O/S

Command: Get 64HDD Device Space Information
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
F:drive_letter
F0:drive_letter
FREE:drive_letter
FREE0:drive_letter
```

Examples:

Get blocks free on drive D

```
OPEN 1,11,15,”F:D”
INPUT#1, EN,EM$,ET,ES: PRINT EM$
BL=VAL(EM$)
CLOSE 1:
```

Errors:

- Current device space (in CBM blocks) is returned. EN, ET, ES always 0
- Syntax error [30] returned if format is too short or drive_letter < A
- If drive not responding then error text indicates this as “ERROR CHECKING DRIVE n”, where for example n = D.
- If drive information is available, VAL can be used to strip number of blocks from error message

Notes:

- works regardless of disk image /directory selected
- only reports what space MSDOS “sees” for the device
- when used whilst a disk image is “logged”, it reports space for the device not the image

---

Command: Exit 64HDD and set error_level upon exit

Applicability:

☒ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
E:error_level
E0:error_level
EXIT:error_level
EXIT0:error_level
```

Examples:

Exit with error_level = 8

```
OPEN 1,11,15,”EXIT:8”
INPUT#1, EN,EM$,ET,ES: PRINT EM$
CLOSE 1
```

Errors:

- Return is Ok or equivalent EN, ET, ES always 0
- Syntax error [30] returned if format is too short or error_level < 0

Notes:

- works regardless of disk image /directory selected
- 64HDD will exit when the next file closure is made, and the emulator returns to idle mode (hence all other CBM files /channels should already be closed before executing this command)
- The error_level is used to control execution of a BATCH file on the PC controller. The batch file can be configured to restart 64HDD, or to restart it with different device assignments (see example batch files).

Command: Read PC Mouse Position and button status (ASCII)
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
M:
M0:
MOUSE:
MOUSE0:
```

Examples:

Read PC mouse position and button status

```
OPEN 1,11,15,"M:"
INPUT#1, EN,EM$,MX,MY
B=VAL(EM$)
CLOSE 1
```

Errors:

- Return is Button information with EN=0, MX=X mouse position (0-511), MY = Y mouse position (0-255)
- B=1 is left Button, B=2 is right Button, B=3 is both Buttons

Notes:

- works regardless of disk image /directory selected

Command: Read PC Mouse Position and button status (Binary)
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
M1:
MOUSE1:
```

Examples:

Read PC mouse position and button status

```
OPEN 1,11,15,"M1:"
GET#1,X1: GET#1,XL :GET#1,XH :GET#1,Y :GET#1,B
X=XL+256*XH
CLOSE 1
```

Errors:

- X mouse position (0-319), MY = Y mouse position (0-199)
- B=1 is left Button, B=2 is right Button, B=3 is both Buttons

Notes:

- works regardless of disk image /directory selected

Command: Run an external MSDOS application
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
MSD:application_name
MSD0:application_name
```

Examples:
Shell to application
```
OPEN 1,11,15,"MSD:application"
CLOSE 1
```

Errors:
- Return is Ok or equivalent EN, ET, ES always 0

Notes:
- works regardless of disk image /directory selected
- 64HDD shells to the *application*, which must be on the path or a path must be given
- an *application* which writes to the screen may corrupt the 64HDD display, this may not be a problem if you do not need to refer to the display (for example in a dedicated system)
- the screen output of many programs is diverted rather than displayed with >MSD.SEQ with the file being written to the +sysdir\MSD.SEQ
- This command may be used for example to run PC based cross compilers

---

Command: Report protocol speed mode and Pentium CPU speed
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
SPEED:
SPEED0:
```

Examples:
Report
```
OPEN 1,11,15,"SPEED:"
INPUT#1, EN,EM$,ES,ET
CLOSE 1
```

Errors:
- Return is Ok, with ET defining the mode, 1=std, 2=faster, 3=fastest, +10 if `UI-` has been previously issued. ES defines CPU MHz if Pentium TSC is being used for microsecond timings. ES=0 if Programmable Interval Timer (PIT) is used for timings.

Notes:
- works regardless of disk image /directory selected

Command: Switch to ShortFileName Mode
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax:

```
SFN:
SFN0:
```

Examples:
Switch to SFN mode for future 64HDD access

```
OPEN 1,11,15,"SFN:"
CLOSE 1
```

Errors:

- Return is Ok, with ES=0 and ET defining the mode, 1=SFN, 2=LFN

Notes:

- works regardless of disk image /directory selected
- filename functions will now be in 8.3 MSDOS format

Command: Switch to LongFileName Mode
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax:

```
LFN:
LFN0:
```

Examples:
Switch to LFN mode for future 64HDD access

```
OPEN 1,11,15,"LFN:"
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:

- Return is Ok if LFN mode was available at 64HDD boot-up, with ES=0 and ET defining the mode, 1=SFN, 2=LFN
- Error [160] returned if LFN was not successfully booted on 64HDD start-up

Notes:

- works regardless of disk image /directory selected
- filename functions will now be in 16.3 MSDOS format
- with JiffyDOS on you CBM computer you will need to enclose the command in quotes as follows so as to avoid the “lock” command which begins with L also: `@"LFN:`

Command: Activate another 64HDD Device
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: U0>+chr$(device_number)

Examples:

Activate device 10

```
OPEN 1,11,15,"U0>"+CHR$(10)
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:

- Return is Ok if device_number within allowable range
- Error [89] if DEVICE_NUMBER is not supported by 64HDD

Notes:

- works regardless of disk image /directory selected
- command will not work when sent by @ in JiffyDOS, use syntax above instead.
- device will be activated to the default path, c:\64hdd

Command: Set /reset "strict" directory mode in MSDOS

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
STR:n
STR0:n
STRICT:n
STRICT0:n

Where n=0, 1 or 2
```

Examples:

Set strict display mode

```
OPEN 1,11,15,"STR:1"
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:

- Return is always [0], with status confirmed in test as either mode clear or set

Notes:

- works regardless of disk image /directory selected
- the higher the value of n, the higher the strictness level
- when strict "set" to 1, the file header is reported strictly and the blocks free is limited to 32000 for compatibility with programs that expect a value in the signed 16bit range
- when strict "set" to 2, no MSDOS directory names or disk /tape images are listed when a directory is requested.
- All directories and images can be attached regardless of whether the directory listing shows them or not

Command: Set /reset “CMD” mode for 64HDD operation

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:
```
CMD:n
CMD0:n

Where n=0, 1 or 2
```

Examples:

Set CMD mode for 64HDD
```
OPEN 1,11,15,”CMD:1”
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:

- Return is always [0], with status confirmed in test as either mode clear or set

Notes:

- works regardless of disk image /directory selected
- when CMD mode is “set”, among other things 64HDD changes the way partitions information is displayed
- when CMD = 0 64HDD uses the 1541’s ROM for the MSD mode memory map
- when CMD = 1 the CMDHD ROM map gets loaded in MSD mode. Having a CMD memory map resident means some CMD compatible programs will now allow you to use additional options. Other file systems use their respective ROM, for example D64 uses the 1541 map
- when CMD = 2 the CMDHD ROM map gets loaded in all file system mode this effectively does three things: 1) 1541 specific software will not detect the drive as 1541 (useful for some programs); 2) allows 64HDD to better emulate the CMDHD; and 3) drive letters are appended to the directory listing.
- the mode change applies to all activated 64HDD devices
- this option can also be set from the command line when starting 64HDD or through function keys

---

Command: Identify what is the current path

(Professional version only)

Applicability:

■ All ☒ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“CWD”`

Examples:

Find the current working directory path for the selected device
```
OPEN 1,11,15,”CWD”
REM READ OUT PATH USING GET# LOOP
CLOSE 1
PRINT EM$
```

Errors:

- Error [0] returned with the EM$ containing the current path

Notes:

-

Command: Activate a new 64HDD Device

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“ON:x”`

Examples:

Activate device #11

```
OPEN 1,11,15,”ON:11”
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:

- Error [91] returned when device is successfully activated /deactivated
- Error [89] returned if device is outside allowable range
- Error [74] returned if requested to activate more devices than 64HDD has memory to handle.

Notes:

- x is an ASCII expression representing the device number. CBM serial devices can only range from 0-30, but not are accessible with the standard kernal ROM.
- Do not activate a device number that is identical to that of another device on the daisy-chain. The serial bus will hang if that device number is accessed as two devices will attempt to respond
- Some DOS wedges such as that in JiffyDOS have a different definition for commands beginning with “O”. You will need to use a quote:
  @”ON:11

Command: De-activate a 64HDD Device

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“OFF:x”`

Examples:

Activate device #11

```
OPEN 1,11,15,”OFF:11”
INPUT#1, EN,EM$,ET,ES
CLOSE 1
```

Errors:
- Error [91] returned when device is successfully activated /deactivated
- Error [89] returned if device is outside allowable range

Notes:
- x is an ASCII expression representing the device number. CBM serial devices can only range from 0-30, but not are accessible with the standard kernal ROM.
- Do not activate a device number that is identical to that of another device on the daisy-chain. The serial bus will hang if that device number is accessed as two devices will attempt to respond.
- Some DOS wedges such as that in JiffyDOS have a different definition for commands beginning with "O". You will need to use a quote:
  @"OFF:11

Command: Sort MSDOS directory filename order
(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `"SORT:x"`

Examples:

Activate device #11

```
OPEN 1,11,15,"SORT:N"
CLOSE 1
```

Errors:
- Return is Ok or equivalent EN, ET, ES always 0

Notes:
- MSDOS, like the 1541, fills the directory in order that files are saved, meaning also that "holes" in the directory are left when files are deleted.
- x is an ASCII expression representing sort sequence required, for example:

N	sort by name
D	sort by date
S	sort by size
E	sort by extensions
–	prefix for reverse sort, eg –N
/S	sort subdirectories recursively

- SORT makes use of an external sort tool called by the batch file LFNSORTR.BAT and which can be configured by the user. Refer to LFNSORT.TXT before using.
- LFNSORT.EXE v1.5 © D.Murdoch is recommended if you want sort LFN files on a FAT32 formatted disks. Other freeware tools are available if you want to use only on FAT16 formatted disks. If there are errors on the disk, it is possible LFNSORT will pause waiting for a Y/N response

- SORT function also available from the 64GUI, but may not work if there is not enough free memory (run 64GUI direct from MSDOS, or use F9 rather than via the F1 key)

Partitions and Unit Support (Short-cuts)

Both the Lt.Kernal and CMD Hard Drives support the concept of multi-unit support. Originally Commodore used the unit number to differentiate the two drives in a dual disk system, for example `LOAD”0:TEST”,11` for drive0 and `LOAD”1:TEST”,11` for drive1. These two third-party hard drives however implemented the unit numbering system to call up a predefined number of “partitions”, or what were effectively images of floppy disk software. Both had disk units 0 through 255 available, with 0 being the current image. CMD uses unit 255 as the system partition.

64HDD also supports partitioning in this regard, but extends the concept further in a couple of ways:

1) units 0 to 999 are available (you should reserve units 900-999 for future usage if you wish them to be compatible)
2) units can point to disk images or MSDOS directories or paths, for example unit 10 can be defined to be your CDROM at d:\, whilst unit 5 can be you favourite game.
3) units 0 to 999 are mapped across all 8 possible device numbers supported by 64HDD, thus unit 8 can be available from device 8 and device 11 at the same time.

Unit numbers can be thought of as “shortcuts” and can be set up to alleviate a lot of keyboard typing.

Unit definitions are saved in the file CMDPRTN.TBL in the 64HDD “System Directory” or “Partition Directory” if specified on command line. This file is in ASCII format and may be edited using a text editor. The first five characters of each line are reserved [abc] for the unit number label. After this is the path definition. The units must be sequential listed and are left bank if undefined. For example:

```
-----
[009]
[010]c:\64hdd\utils
[011]
[012]c:\test.d64
[013]
-----
```

The partition file can also be modified using the appropriate 64HDD commands. Remember, that if the 64HDD “system directory” is stored on a RAMDisk for speed it should be copied back to a real disk by the GO64HDD batch file if changes are to be kept for future use. The following conventions should be adhered to:

Unit 0	Default "partition", re-definitions ignored
Unit 255	System "partition" for CMD compatibility, set to 64HDD "System Directory" location
Units 900-999	Reserved for future 64HDD functions

(Professional version only)

In addition to unit definitions are saved in the file CMDPRTN.TBL, a second file NAMEPRTN.TBL also stored in the 64HDD "System Directory" or "Partition Directory" if specified on command line, contains descriptive information about the partition definition. This is particularly useful when so many disk images are labelled with cryptic 8.3 format names.

This NAMEPRTN.TBL file is in ASCII format and may be edited using a text editor. Like CMDPRTN.TBL, the first five characters of each line are reserved [abc] for the unit number label. After this is the "common name" definition, however "spaces" are not allowed and should be substituted with ".". The units must be sequential listed and are left bank if undefined. Alternatively, with the professional version both TBL files can be edited using 64GUI and this tool will make all necessary substitutions.

A special feature of the NAMEPRTN.TBL is that a suffix can be used to identify disks that are part of a multi-disk set. There are some basic rules to follow. The first disk is identified by the :01 suffix, the second with :02 and so on. 64GUI can also automate this process.

Command: Load partition table

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
LOAD“$=P”,device
LOAD“$=P:=type[>pn][<pn]”,device

    Type can include:   4 for 1541 (D64)
                        7 for 1571 (D71)
                        8 for 1581 (D81)
                        D64, D71, D81, T64
                     D* for all disk images
                        L for LNX

    Where:     >pn  lists only unit definitions > pn
               <pn  lists only unit definitions < pn
```

Examples:

Load complete partition table

```
LOAD”$=P”,11
```

Load partition table only with definitions to disk images

```
LOAD”$=P:=D*”,11
```

Errors:

- Errors typical of a directory read

Notes:

- partition table can currently only be called from a device attached
- partition information is displayed in wide format (not exactly CMD compatible, unless you have activated CMD mode on the professional version)
- *type* only allows restriction of the types to be controlled. No naming pattern match is considered.
- partition definitions longer than the display name are abbreviated and are show with only the start and end portions of the definition
- Professional feature:
  - When CMD mode is set, the partition table closely resembles the layout returned by a CMD device

---

Command: Load “common name” partition table

(Professional version only)

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
LOAD“$=N”,device
LOAD“$=N:,type[>pn][<pn]”,device

Type can include:    4 for 1541 (D64)
                     7 for 1571 (D71)
                     8 for 1581 (D81)
                     D64, D71, D81, T64
                 D* for all disk images
                     L for LNX

Where:     >pn  lists only unit definitions > pn
           <pn  lists only unit definitions < pn
```

Examples:

Load complete partition table

```
LOAD”$=N”,11
```

Load partition table only with definitions to disk images

```
LOAD”$=P:,D*”,11
```

Load selected range of partition table only with definitions to disk images

```
LOAD”$=P:*,D*>100<200”,11
```

Errors:

- Errors typical of a directory read

Notes:

- partition table can currently only be called from a device attached

- partition information is displayed in wide format (not exactly CMD compatible, unless you have activated CMD mode on the professional version)
- name pattern match is used as part of the filtering allowed.
- partition definitions longer than the display name are abbreviated and are show with only the start and end portions of the definition

---

Command: Load /save /open from /to unit path

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
LOAD“pn:$dirspec”,device
LOAD“pn:filename”,device
LOAD“pn//dir1/dir2:filename”,device
SAVE“xpnfilename”,device
OPEN lfn,dev,sec,“x:filename,type,mode”
```

Examples:

Load default program from disk image associated with unit200

```
LOAD”200:*”,11
```

Load program “test” from path associated with unit205

```
LOAD”205:test”,11
```

Load program from path building on definition in unit4, note full progname required!!!

```
LOAD”4//64hdd/utils:errorchk.prg”,11
```

Load directory from path associated with unit100

```
LOAD”100:$”,11
```

Errors:

- error [77] set if partition not found
- error [62] set if file not found
- error [0] or error [2] set if partition was found and change successful

Notes:

- if unit definition is not a found the change is ignored and an attempt is made to load from the current path (ie unit 0)
- with the syntax “pn/dir1/dir2:” the last directory name terminates with a “:” and not “/” and the full progname (including type) is required

---

Command: Change partition to path defined by unit number

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“CPx”
“cPn”
```

Examples:

Change to partition defined as unit 10

```
OPEN 1,11,15,"CP10"
CLOSE 1
```

Change to partition defined as unit 10

```
open 1,11,15,"cP"+chr$(10)
close 1
```

Errors:
- error [77] set if partition not found
- error [0] or error [2] set if partition was found and change successful

Notes:
- x is ASCII partition number and allows 0-999 values to be used
- n is binary partition number and allows only 0-255 values to be used
- Professional feature:
  - If the command is applied to device 8, and the partition's common name identifies the definition as being part of a multi-disk sequence, then it and subsequent "disks" will be attached to the "Disk-Flip" table (units 1-10)

Command: Add current path /image to partition table
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `"APx"`
`"aPn"`

Examples:

Set current path /image to partition table unit 20

```
OPEN 1,11,15,"AP20"
CLOSE 1
```

Set current path /image to partition table unit 20

```
open 1,11,15,"aP"+chr$(20)
close 1
```

Errors:
- error [77] set if partition not found
- error [0] or error [2] set if partition was found and change successful

Notes:

Command: Delete current partition table definition
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“DPx”`
`“dPn”`

Examples:

Delete definition of unit 7

```
OPEN 1,11,15,”DP7”
CLOSE 1
```

Delete definition of unit 7

```
open 1,11,15,”dP”+chr$(7)
close 1
```

Errors:

- error [77] set if partition not found
- error [0] or error [2] set if partition was found and change successful

Notes:

Command: Information about partition

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“IPx”`
`“iPn”`

Examples:

Get definition of unit 7 and assign to EM$

```
OPEN 1,11,15,”IP7”
INPUT #1, EN, EM$, ET, ES
CLOSE 1
```

Get definition of unit 7 and assign to EM$

```
open 1,11,15,”iP”+chr$(7)
INPUT #1, EN, EM$, ET, ES
close 1
```

Errors:

- error [0]

Notes:

- undefined partitions are returned as “.” signifying that no partition change would be made if called
- partitions 990-999 return “RESERVED”
- only partitions 0-255 can be requested using the binary method

---

Command: Assign common name to a partition definition

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `"NPx"`

Examples:
Change "common name" to partition defined as unit 20

```
OPEN 1,11,15,"NP20:name"
CLOSE 1
```

Set up a multi-disk sequence beginning at unit 30

```
OPEN 1,11,15,"NP30:disk1.d64:01"
PRINT #1,"NP31:disk2.d64:02"
PRINT #1,"NP32:disk3.d64:03"
PRINT #1,"NP33:disk4.d64:04"
CLOSE 1
```

Errors:

- error [77] set if partition not found
- error [0] or error [2] set if partition was found and change successful

Notes:

- x is ASCII partition number and allows 0-999 values to be used
- if the common name contains "spaces" 64HDD will replace these with "." (necessary for 64HDD)

Command: Get "common name" assigned to a partition
(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `"INx"`

Examples:
Get definition of unit 7 and assign to EM$

```
OPEN 1,11,15,"IN7"
INPUT #1, EN, EM$, ET, ES
CLOSE 1
```

Errors:

- error [0]

Notes:

- undefined partitions are returned as "." signifying that no partition name has been set
- partitions 990-999 return "RESERVED"

CMD Compatible Time and Date Commands

CMD FD and HD series devices support an option RTC device. 64HDD supports compatible commands, though the preference is to use the native commands as these are simpler to convert (for example to CBM native TI$) and have no Y2K related issues.

The CMD commands are provided only to ensure compatibility with programs written for CMD RTC equipped devices (not including SmartMouse). It must be pointed out that the CMD operating system has Y2K compatibility issues as it uses only a 2-digit year format. As a consequence setting the date using the CMD commands may set the PC's RTC to the wrong date. 64HDD makes the assumption that the year will be 2000+set_year, but this precludes initialising the system at an earlier date. If you wish to do this use the native commands. If you are still using 64HDD in the year 2100 you will also have problems, use the 64HDD native command to avoid these issues.

Command: Read /set RTC in CMD ASCII format
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
"T-RA"
"T-WAdow. mo/da/yr/ hr:mi:se xM"
```

Examples:

See Notes.

```
10 OPEN15,11,15,"T-RA"
20 GET#15,A$:T$=T$+A$:IF ST<>64 THEN 20
30 CLOSE 15

     t$ contains the time in this order:
          "dow. mo/da/yr hr:mi:se xM"+chr$(13)

          day-of-week: 4 letters + one space
          SUN. MON. TUES WED. THUR FRI. SAT.
          mo  month: 01-12
          da  day
          yr  year
          hr  hour: 01-12
          mi  minute: 00-59
          se  second: 00-59
          x   A or P : AM/PM
```

Errors:

- Error [30] returned if time/date setting string the wrong length
- Error is [0] if length Ok

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.

Command: Read /set RTC in CMD decimal format

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“T-RD”
“T-WD”+9bytes
```

Examples:

See Notes, decimal format means byte is 0-99.

```
     byte 0:  day-of-week 0=sun, 1=mon...
     byte 1:  year
     byte 2:  month
     byte 3:  day
     byte 4:  hour (0-12)
     byte 5:  minute
     byte 6:  second
     byte 7:  AM/PM flap 0=AM  <>0 = PM
     byte 8:  chr$(13)
```

Errors:

- Always returns error [0] regardless

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.

Command: Read /set RTC in CMD BCD format

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“T-RB”
“T-WB”+9bytes
```

Examples:

See Notes, BCD format means high/low nibble is 0-9.

```
     byte 0:  day-of-week 0=sun, 1=mon...
     byte 1:  year
     byte 2:  month
     byte 3:  day
     byte 4:  hour (0-12)
     byte 5:  minute
     byte 6:  second
     byte 7:  AM/PM flap 0=AM  <>0 = PM
     byte 8:  chr$(13)
```

Errors:

- Always returns error [0] regardless

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.

Command: Load Directory with TimeStamp
Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `LOAD“$=T:dirspec,type”,device`

Examples:
Load directory from default path with CMD timestamp format
`LOAD”$=T”,11`

Load directory from default path with CMD Long timestamp format
`LOAD”$=T,L”,11`

Load directory from default path with 4digit timestamp format
`LOAD”$=T,LL”,11`

*Load timestamped directory from default path with name matching files**
`LOAD”$=Tfiles*”,11`

Errors:
-

Notes:
- only MSDOS file system has timestamp information currently
- CMD date selection fields are not yet supported

Extended CMD Compatible Drive Commands

(Professional version only)

The CMD FD and HD series devices support a number of additional command channel operations. To enhance compatibility of 64HDD with software written for the CMD drives, these additional commands are supported in the professional version (though there may be alternate 64HDD ways to achieve the same result using native commands).

Command: **Set / Clear Write Protection Mode**

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“W-0”
“W-1”
```

Examples:

Set write protect mode.

```
10 OPEN15,11,15,"W-1":CLOSE 15
```

Errors:

- Error [26] returned attempt made to write with protection set

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.
- T64 and LNX file systems are always in write protect mode as 64HDD does not support creation of these file types.

Command: **SCSI Command**

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“S-C”
```

Examples:

Errors:

- Error [4x] returned upon attempt to issue a particular SCSI command

Notes:

- As 64HDD does not require a SCSI controller, all attempts to perform low level commands are responded to with a SCSI controller error. This should give the CMD application the chance to abort the process.

Command: G-P Get-Partition information

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“G-Pn”`
`“G-D”`

Examples:

Errors:

-

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.
- There are easier ways to access partitions with 64HDD, however, this command is included to provide compatibility with several programs already designed for the CMD series of device, for example FCOPY and FCOPY+

Command: R-H Rename Disk Header

(Professional version only)

Applicability:

☐ All ☐ MSDOS ■ D64 ■ D71 ■ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“R-H:diskname[,id]”`

Examples:

Set current disk’s name to “MY HOMEWORK”

```
10 OPEN15,11,15,“R-H:MY HEADER”:CLOSE 15
```

Errors:

- Errors as would typically be associated with disk block accesses
- Error [0] if successful
- Error [31] is : omitted

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals
- Renaming “disk id” is optional
- Disk contents and BAM remain unaltered

1541COPY and Transfer Commands

(Professional version only)

64HDD supports the direct communication between a 1541 compatible drive and the PC allow the copying of a 1541 disk to a D64 image, or vice-versa, with the issuing of one simple command. The transfer scheme calls two batch files F1541 and T1541, which by default are configured to call the Star Commander utilities (tested with version 0.83.23 beta, though you are free to change this to use your preferred 3rd party tool). If using SC, you will need to download, pre-configure and test the SC utilities before using the F1541 and T1541 commands. This includes adding the path to the SC files to your MSDOS PATH.

For the functions described in this section to work the 64HDD connection and PC setup must be accurately setup and timed.

Command: Transfer D64 image to 1541FDD connected as Device #8

(Professional version only)

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“T1541:filename.D64[,F]”`

Examples:

Transfer D64 image in current directory to 1541

```
OPEN 1,11,15,”T1541:DUCKS.D64”
CLOSE 1
```

Transfer D64 image in current directory to 1541, but format disk first!

```
OPEN 1,11,15,”T1541:DUCKS.D64,F”
CLOSE 1
```

Errors:

- Error [182] returned for any system error
- Other errors as appropriate if disk image has problems
- Error [0] Ok if transfer is successful

Notes:

- This command is only designed to work with a 1541 or a 1571 “locked” in 1541 mode.
- The D64 Disk image has to be in the current directory (else the *relative* path must be specified)
- The transfer is performed by 64HDD directly “talking” and “listening” to a 1541 device connected as #8.

> Note: No other bus activity is allowed during the disk transfer as it will result in a bus contention (due to multiple talkers) and will result in a failure to complete the operation.

Progress is displayed on 64GUI and can also be inferred from the status of the 1541's activity light.

- The transfer scheme used is that configured when you setup SC (or other tool). This allows for normal, fast or warp transfers to be specified.
- Support is only offered in MSDOS file system as a precaution against transferring a disk image that is currently active.
- Error information contained in the disk image is not transferred to the real disk. You must use additional tools to create these errors.
  - CBMDISK will show you a disk images error table
  - Several small disk tools distributed with 64HDD will let you re-create some T&S errors on the real disk

---

Command: Create D64 from 1541FDD connected as Device #8
(Professional version only)

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `"F1541:filename.D64"`

Examples:

Transfer 1541 disk to a D64 image in current directory with given name

```
OPEN 1,11,15,"F1541:TEST.D64"
CLOSE 1
```

Errors:

- Error [183] returned for any error detected
- Other errors as appropriate if disk image has problems
- Error [0] Ok if transfer is successful

Notes:

- This command is only designed to work with a 1541 or a 1571 "locked" in 1541 mode.
- The D64 Disk image will be created in the current directory (else the *relative* path must be specified)
- The transfer is performed by 64HDD directly "talking" and "listening" to a 1541 device connected as #8.

> Note: No other bus activity is allowed during the disk transfer as it will result in a bus contention (due to multiple talkers) and will result in a failure to complete the operation.

Progress is displayed on 64GUI and can also be inferred from the status of the 1541's activity light.

- The transfer scheme used is that configured when you setup SC (or other tool). This allows for normal, fast or warp transfers to be specified.
- Support is only offered in MSDOS file system as a precaution against transferring a disk image that is currently active.

- T&S errors contained on the real disk may cause the 1541 to lock-up!

Command: Interrogate 1541FDD connected as Device #8
(Professional version only)

Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“I1541:”`

Examples:

Transfer 1541 errors disk to 64HD-T&S.ERR in the current directory

```
OPEN 1,11,15,”I1541:”
CLOSE 1
```

Errors:

- Error [183] returned for any error detected
- Other errors as appropriate if 64HDD has problems
- Error [0] Ok if transfer is successful

Notes:

- This command is only designed to work with a 1541 or a 1571 “locked” in 1541 mode.
- The error file is created in the current directory. Typically, you will have already transferred the program files to this directory.
- The transfer is performed by 64HDD directly “talking” and “listening” to a 1541 device connected as #8.

  > Note: No other bus activity is allowed during the disk transfer as it will result in a bus contention (due to multiple talkers) and will result in a failure to complete the operation.

  Progress is displayed on 64GUI and can also be inferred from the status of the 1541’s activity light.
- The transfer scheme employs generic U1 and U2 disk block read commands making it very compatible, but unfortunately very slow. The transfer is however faster than a Commodore doing the work as only a one-way transfer is needed.
- Support is only offered in MSDOS file system as a precaution against transferring to a disk image.

1581COPY Commands

The 1581COPY program by Wolfgang Mosser (Womo) is supported by 64HDD and allows the emulator to treat the 1.44FDD as a 1581 drive indirectly. It should be noted that you must use DD disks as these are most compatible with the real 1581 disk drive. 64HDD will work with disk images transferred using this utility - it cannot directly read a 1581 formatted disk.

Before attempting to use 1581COPY from 64HDD make sure that your system is indeed compatible and capable of producing 1581 readable disks. I have in the past had problems with both incompatible FDD controllers and poor quality floppy disks. As 1581COPY is used as an external resource error information and control is limited. Control of 1581COPY is via two batch files, T1581.BAT and F1581.BAT, and these files together with 1581COPY.EXE need to be on the "path" searched by MSDOS.

The extended error information available is controlled within the two BAT files. If the 64HDD system directory is in a different location to that shown in these files, these files must be modified to reflect the new path. For example, the default is c:\64hdd\system, but if a RAMdisk is used the path may be e:\64hdd\system. Be sure to change all occurrences in both T1581 and F1581 files.

Command: Transfer D81 image to 1.44FDD using 1581COPY
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `"T1581:filename.D81"`
`"T1581:"`

Examples:
Transfer D81 image in current directory to 1.44FDD

```
OPEN 1,11,15,"T1581:DUCKS.D81"
CLOSE 1
```

Format 1.44FDD using 1581COPY but with no transfer

```
OPEN 1,11,15,"T1581:"
CLOSE 1
```

Errors:
- Error [180] returned for any system error
- Error [35] and appropriate message returned for other errors

Notes:
- D81 Disk image has to be in the current directory (else path must be specified)
- The transfer is controlled by shelling to the T1581.BAT file. Modify this file as needed. You require Womo's 1581COPY program (version greater than 0.52) to run this function. The emulator will not respond to other commands whilst the

transfer takes place (this is because 1581COPY is a stand alone program and not integrated into 64HDD)
- Support is only offered in MSDOS file system as a precaution against transferring a disk image which is currently active.

Command: Transfer 1581 disk in 1.44FDD to D81 with 1581COPY
Applicability:

☐ All ■ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX
Syntax: `”F1581:filename.D81”`

Examples:
Transfer 1581 disk from 1.44FDD to D81 image in current directory with given name

```
OPEN 1,11,15,”F1581:TEST.D81”
CLOSE 1
```

Errors:
- Error [181] returned for any error detected
- Error [35] and appropriate message returned for other errors

Notes:
- D81 Disk image is written in the current directory (else path must be specified)
- The transfer is controlled by shelling to the F1581.BAT file. Modify this file as needed. You require Womo’s 1581COPY program (version greater than 0.52) to run this function.
- The emulator will not respond to other commands whilst the transfer takes place (this is because 1581COPY is a stand alone program and not integrated into 64HDD)
- Support is only offered in MSDOS file system as a precaution against transferring a disk image which is currently active.

Device Number Swapping

(Professional version only)

In addition to the use of function keys and hardware button control, device number swapping can be performed by the following command channel commands.

Command: **Device number Swapping**

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
"S-8"
"S-9"
"S-D"
```

Examples:

Swap device 8 and 28.

```
10 OPEN15,11,15,"S-8":CLOSE 15
```

Errors:

-

Notes:

- See examples bundled with the 64HDD software, or refer to the CMD user manuals.
- S-8 swaps with #8. Device #8 has its number changed to the last accessed 64HDD disk device via direct communication between 64HDD and that device. The 64HDD device #8 is activated, and the path definition assigned to 64HDD's device number is transferred to #8.
- Device swapping should only be done whilst the IEC serial bus is idle. This must be the case as during a swap, 64HDD uses the bus to communicate with the other devices. The swapping is performed by 64HDD directly "talking" and "listening" to a 1541 device connected as #8.

  Note: No other bus activity is allowed during the swap operations as it will result in a bus contention (due to multiple talkers) and will result in a failure to complete the operation.

- S-9 is similar to S-9 except the swap is between devices #9 and #29
- S-D restores the original device number drives #8 and #9, cancels 64HDD's activation of these devices and restores the assignments to devices the original device numbers.
- Various beeps and squeaks will sound to signify that the swapping action has occurred. Refer to the 64HDD display for more information.
- Resetting 64HDD will clear all 64HDD assignments, but may not clear the renumbering of real #8 and #9 devices (depends upon the wiring of your reset switches). If your real drive ends up stuck on a "wrong" number, simply issue a "UI" command to it, or reset the Commodore machine.

Print Spooling and PrintEngines

(Professional version only)

Command: Increment to next JobNumber

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“PNEXT:dev”`

Examples:

Increment to next job number on device #4

```
OPEN 1,11,15,”PNEXT:4”
CLOSE 1
```

Errors:

- Always OK or Syntax Error [31]

Notes:

- Must be issued to the command channel of a 64HDD device (can be a drive or printer)
- When issued using JiffyDOS or a DOS-wedge, use a “ before the P, eg: `@”PNEXT:4`
- PNEXT is needed to increment job numbers when a program does not use the secondary channel to talk to a printing device.

Command: Viewing the PrintQueue

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax:

```
“PPAUSE”
LOAD”$”,pdev
```

Examples:

Pause the print spooler and look at what is in the Print Queue for device #4

```
OPEN 1,11,15,”PPAUSE”
CLOSE 1
LOAD”$”,4
LIST
```

Errors:

- Always OK or Syntax Error [31]

Notes:

- Must be issued to the command channel of a 64HDD device (can be a drive or printer)

- When issued using JiffyDOS or a DOS-wedge, use a “ before the P, eg: `@”PPAUSE`
- PPAUSE is needed to allow the $ command to be passed on without being trapped by the spooler routine.
- Only files with the JOBxxxxx.USR name will be listed. APP files and files generated by the Print Engines will not be shown. To see these you will need to change to the directory using one of the drive device numbers.

Command: Printing a specific JobNumber

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“PPRINT:dev:jobnumber:PrintEngine”`

Examples:

Print job number 015 on device #4 to a PrintEngine caller 64PE-TXT

```
OPEN 1,11,15,”PPRINT:4:15:64PE-TXT”
CLOSE 1
```

Errors:

- Always OK or Syntax Error [31]

Notes:

- Must be issued to the command channel of a 64HDD device (can be a drive or printer)
- When issued using JiffyDOS or a DOS-wedge, use a “ before the P, eg: `@”PPRINT:4:15:64PE-TXT`
- 64HDD will not be available until the job has been handled by the PrintEngine.

Command: Delete a specific JobNumber

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `“PDEL:dev:jobnumber”`

Examples:

Delete job number 015 on device #4

```
OPEN 1,11,15,”PDEL:4:15”
CLOSE 1
```

Errors:

- Always OK or Syntax Error [31]

Notes:

- Must be issued to the command channel of a 64HDD device (can be a drive or printer)

- When issued using JiffyDOS or a DOS-wedge, use a " before the P, eg: `@"PDEL:4:15`
- PDEL will delete both the JOB and APP file associated with than job
- *dev* is a given as single digit, eg 4 not 04

Command: Flush the print queue for a device

(Professional version only)

Applicability:

■ All ☐ MSDOS ☐ D64 ☐ D71 ☐ D81 ☐ D2M ☐ T64 ☐ LNX

Syntax: `"PFLUSH:dev"`

Examples:

Delete all jobs in the queue for device #4

```
OPEN 1,11,15,"PFLUSH:4"
CLOSE 1
```

Errors:

- Always OK or Syntax Error [31]

Notes:

- Must be issued to the command channel of a 64HDD device (can be a drive or printer)
- When issued using JiffyDOS or a DOS-wedge, use a " before the P, eg: `@"PFLUSH:4`
- PFLUSH will delete all files in the device spooling directory.
- *dev* is a given as single digit, eg 4 not 04

MSDOS ASCII Compatibility

A special ASCII exchange mode is supported by 64HDD. If the file type is specified with a “shifted” type specifier, basic alphabetic ASCII translation is performed. This means for example, by writing a file with the “shift-S” filetype, a sequential file will be written, but upper and lower case will be correctly viewable in an MSDOS editor. Likewise reading an MSDOS text file by changing filetype to USR and then opening it with a “shift-U” filetype will result in the MSDOS ASCII codes being exchanged for PETASCII equivalents.

Text Exchange (Cross-Platform Compatibility)

Let’s face it; from time to time there is the need to exchange text data between the Commodore machine and the rest of the world. 64HDD is capable of performing on-the-fly ASCII text translations by simply opening the file with a {shift-type} rather than the normal {type} specifier. Alpha-translation is only performed.

The possibilities are endless, and combined with the other cross-platform functions within 64HDD some level of co-operative processing is possible. Consider the following examples:

- Using a Cross-platform Compiler: write a batch file on your 64HDD machine that compiles your source code (saved on-the-fly to true-ASCII) into a PRG file. The batch file is called by 64HDD’s `MSD:` command. Output from the compiler is redirected to the MSD.SEQ file in the system directory, and can be read using the on-the-fly ASCII translator.

- Use a PC finder tool to locate files on a CD: use one of the popular *whereis* utilities available on the PC by calling it with the MSD: command. The results can be viewed in the MSD.SEQ file in the system directory.

Notes:
- on-the-fly translations are modal, meaning that the last file open determines if it is on or off. Therefore the last file open will determine whether the ASCII translation occurs for all other files that are also open.
- 64HDD waits on the drive that has been set to have the system directory. As a result, MSD programs are most easily controlled by running a batch file that can specify the full path name of the utility and any associated input output files. Additionally the batch file can set any local environmental variables.
- Remember, the location of the MSD.SEQ file is in partition 255. If 64HDD users have configured their system properly, the partition number can be used as a universal way of locating the file.

Error Codes and Messages

64HDD reports error codes and messages that closely match those of real CBM drives, such as the 1541, 1571, 1581. These error codes are listed in a number of other sources, but for the convenience of 64HDD users are reproduced below. In addition to standard error codes, 64HDD has some extended error codes, usually to give additional feedback to users about features not built into standard CBM drives. 64HDD Professional users will also have access to a range of codes reported by CMD compatible drives in response to CMD specific commands.

It should be noted that some error codes are used as part of copy protection schemes and are not real errors cause by 64HDD or the PC. In these cases the error code is read from the extended information on some D64 and other disk images, or from special T&S error files in native MSD partitions when using 64HDD Professional.

64HDD Professional also displays the same error codes on its GUI as and when they occur. Reading the error channel clears the error code, as it would on the original CBM drives.

Summary of 64HDD / CBM DOS Error Messages

0	OK, no error exists
1	Files scratched response. Not an error condition
2	Partition selected. Not an error condition
3	Files locked response. Not an error condition
4	Files unlocked response. Not an error condition
5-19	Unused error messages: should be ignored
20	Block header not found on disk
21	Sync character not found
22	Data block not present
23	Checksum error in data
24	Byte decoding error
25	Write-verify error
26	Attempt to write with write protect on
27	Checksum error in header
28	Data extends into next block
29	Disk id mismatch
30	General syntax error
31	Invalid command
32	Long line
33	Invalid filename
34	No file given
39	Command file not found
43	Status Error (pseudo SCSI error)
44-48	Controller Error (pseudo SCSI error)
50	Record not present
51	Overflow in record

52	File too large
60	File open for write
61	File not open
62	File not found
63	File exists
64	File type mismatch
65	No block
66	Illegal track or sector
67	Illegal system track or sector
70	No channels available
71	Directory error
72	Disk full or directory full
73	Power up message, or write attempt with DOS Mismatch
74	Drive not ready
75	Format error
76	Controller error
77	Selected partition illegal
78	Selected path not available
79	SYSDIR error
80-89	Command not supported (actual code depends on mode)
92	Pwr-Load driver missing
94	Fast-Load driver missing
160	LFN mode not available
180	D81 TO 1.44FDD transfer error
181	1.44FDD TO D81 transfer error
182	D64 TO 1541 transfer error
183	1541 TO D64 transfer error
200	CD-AUDIO error

Description of 64HDD / CBM DOS Error Messages

Codes 00-19 are not errors, but status messages. Codes greater than or equal to 20 are true error messages.

00: OK – No error, last requested operation was successfully performed

01: FILES SCRATCHED – returns information about the number of files scratched by the last scratch command

02: PARTITION SELECTED – returns information about the partition number selected or if used on a D81 the successful changed to a partition

03: FILES LOCKED – returns information about the number of files locked by the last F-L command

04: FILES UNLOCKED – returns information about the number of files unlocked by the last F-U command

20: READ ERROR (block header not found) – 64HDD has a problem reading from the PC's device (disk or other). This message may also occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller was unable to locate the header of the requested data block. It may have been caused by an illegal block number, or the header has been destroyed.

21: READ ERROR (no sync character) – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller was unable to detect a sync mark on the desired track when the image was created. It may be caused by misalignment of the read/writer head, no diskette was present, or unformatted or improperly seated diskette.

22: READ ERROR (data block not present) – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller was requested to read or verify a data block that was not properly written. This error message occurs in conjunction with the BLOCK commands and indicates an illegal track and/or block request.

23: READ ERROR (checksum error in data block) – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. This error message indicates that there was an error in one or more of the data bytes. The data has been read into the DOS memory, but the checksum over the data is in error.

24: READ ERROR (byte decoding error) -- This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. The data or header was read into the DOS memory, but a hardware error has been created due to an invalid bit pattern in the data byte.

25: WRITE ERROR (write-verify error) – This message is generated if 64HDD detects a mismatch between the written data and the data in the DOS memory.

26: WRITE PROTECT ON – This message is generated when 64HDD has been requested to write a data block while the write protect mode is selected professional version) or the disk image is of read-only type.

27: READ ERROR (checksum error in header) – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller detected an error in the header of the requested data block. The block had not been read into the DOS memory.

28: WRITE ERROR (long data block) – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller attempted to detect the sync mark of the next header after writing a data block. If the sync mark did not appear within a predetermined time, the error message was generated.

29: DISK ID MISMATCH – This message will only occur on extended disk images containing track and sector error information and is sometimes used as copy protection. In this case the disk controller had requested to access a disk image which had not been initialized.

30: SYNTAX ERROR (general syntax) – 64HDD cannot interpret the command sent to the command channel. Typically, this is caused by an illegal number of file names, or pattems are illegally used. For example, two file names may appear on the left side of the COPY command.

31: SYNTAX ERROR (invalid command) – 64HDD does not recognize the command. The command must start in the first position.

32: SYNTAX ERROR (invalid command) -- The command sent is longer than 58 characters. 64HDD has a longer command buffer than the real CBM DOS, but in some operating modes limits itself to the 1541 limit for improved compatibility.

33: SYNTAX ERROR (invalid file name) – Pattern matching is invalidly used in the OPEN or SAVE command.

34: SYNTAX ERROR (no file given) – the file name was left out of a command or 64HDD does not recognize it as such. Typically, a colon (:) has been left out of the command,

39: SYNTAX ERROR (invalid command) – This error may result if the command sent to command channel (secondary address 15) is unrecognized 64HDD.

43: STATUS ERROR – pseudo SCSI error when in CMD mode. No actual attempt to perform the low level SCSI command is made.

44-48: CONTROLLER ERROR – pseudo SCSI error when in CMD mode. No actual attempt to perform the low level SCSI command is made.

50: RECORD NOT PRESENT – Result of disk reading past the last record through INPUT#, or GET# commands. This message will also occur after positioning to a record beyond end of file in a relative file. If the intent is to expand the file by adding the new record (with a PRINT# command), the error message may be ignored. INPUT or GET should not be attempted after this error is detected without first repositioning.

51: OVERFLOW IN RECORD – PRINT# statement exceeds record boundary. Information is cut off. Since the carriage return is sent as a record terminator is counted in the record size. This message will occur if the total characters in the record (including the final carriage return) exceeds the defined size.

52: FILE TOO LARGE – Record position within a relative file indicates that disk overflow will result.

60: WRITE FILE OPEN – This message is generated when a write file that has not been closed is being opened for reading.

61: FILE NOT OPEN – This message is generated when a file is being accessed that has not been opened in the DOS. Sometimes, in this case, a message is not generated; the request is simply ignored.

62: FILE NOT FOUND – The requested file does not exist on the indicated drive/device.

63: FILE EXISTS – The file name of the file being created already exists on the drive/device/disk-image.

64: FILE TYPE MISMATCH – The file type does not match the file type in the directory entry for the requested file.

65: NO BLOCK – This message occurs in conjunction with the B-A command. It indicates that the block to be allocated has been previously allocated. The parameters indicate the track and sector available with the next highest number. If the parameters are zero (0), then all blocks higher in number are in use.

66: ILLEGAL TRACK AND SECTOR – 64HDD has attempted to access a track or block which does not exist in the format being used. This may indicate a problem reading the pointer to the next block.

67: ILLEGAL SYSTEM T OR S – This special error message indicates an illegal system track or block.

70: NO CHANNEL (available) – The requested channel is not available, or all channels are in use. A maximum of five sequential files may be opened at one time to the 1541 DOS. Direct access channels may have six opened files.

71: DIRECTORY ERROR – The BAM does not match the internal count. There is a problem in the BAM allocation or the BAM has been overwritten in DOS memory. To correct this problem, reinitialize the diskette to restore the BAM in memory. Some active files may be terminated by the corrective action. NOTE: BAM = Block Availability Map

72: DISK FULL – Either the blocks on the diskette are used or the directory is at its entry limit.

73: DOS MISMATCH – This message is available when 64HDD is first started or when it is reset by commands UI or UJ. This message will reveal the current version of the 64HDD core being run and can be used to detect a 64HDD device, for example “CBM DOS V0.78 EMUL”. On the original CBM drives it also identifies an incompatibility between DOS versions. CBM DOS 1 and 2 are read compatible, but not write compatible. Disks may be interchangeably read with either DOS, but a disk formatted on one version cannot be written upon with the other version because the format is different. This error is displayed whenever an attempt is made to write upon a disk which has been formatted in a non-compatible format.

74: DRIVE NOT READY – An attempt has been made to access a drive/disk/disk-image/partition which is not valid or available.

75: FORMAT ERROR – 1581 compatible error. Unlikely to be seen when working with disk images.

76: CONTROLLER ERROR – 1581 compatible error. Unlikely to be seen when working with disk images.

77: SELECTED PARTITION ILLEGAL – requested partition change is not available or partition number illegal. Valid for 1581 emulation mode (D81) and 64HDD partition table.

78: SELECTED PATH NOT AVAILABLE – requested path change is not available or legal

79: SYSDIR ERROR – problem with locating the 64HDD SYSDIR directory

80-89: COMMAND NOT SUPPORTED – various codes as below which can occur if the request function is not supported. Some commands are specific to particular emulation modes, whilst in a few cases some commands are not supported by the current version of 64HDD.

80:	COMMAND NOT SUPPORTED (MSDOS)
81:	COMMAND NOT SUPPORTED (D64)
82:	COMMAND NOT SUPPORTED (D71)
83:	COMMAND NOT SUPPORTED (D81)
84:	COMMAND NOT SUPPORTED (D2M)
85:	COMMAND NOT SUPPORTED (T64)
86:	COMMAND NOT SUPPORTED (LNX)
87:	COMMAND NOT SUPPORTED (H64)
89:	COMMAND NOT SUPPORTED (64HDD)

92: PWR-LOADER MISSING – the request to send the file by the Pwr-Load method is not possible because the required driver file is not located in the SYSDIR directory

94: FAST-LOADER MISSING – the request to send the file by the Fast-Load or X-Load or Z-Load method is not possible because the required driver file is not located in the SYSDIR directory

160: LFN MODE NOT AVAILABLE – LFN mode was not activated during 64HDD start-up and so the requested LFN command cannot be supported

180: D81 TO 1.44FDD ERROR – there was an error encountered whilst using 1581COPY to transfer the requested D81 image to the floppy in A:

181: 1.44FDD TO D81 ERROR – there was an error encountered whilst using 1581COPY to transfer the floppy in A: to the requested D81 image

182: D64 TO 1541 ERROR – there was an error encountered whilst transferring the requested D64 image to the floppy disk in device #8

183: 1541 TO D64 ERROR – there was an error encountered whilst transferring the floppy disk in device #8 requested D64 image

200: CD-AUDIO ERROR – there was an error encountered with the last issued CD-Audio command

Additional extended modes error text which maybe assigned to error/status code 00 instead of the default OK, for example:

MSD==>RESULTS IN [SYSDIR]/MSD
64HDD DRIVE (DE)ACTIVATED
STRICT MODE SET/CLEAR
CBM DRIVE EMU (C) COPLIN
CMD EMU MODE SET/CLEAR
Time, Date and Day output
Mouse commands