Hardbox Manual

Revision 2

small systems engineering limited

2-4 Canfield Place, London NW6 3BT. Telephone: 328 7145 Telex 264538

HARDBOX MANUAL

REVISION 2

Small Systems Engineering · 2-4 Canfield Place · London NW6 · tel 328 7145

Corvus Interface Version
Revision 2

TABLE OF CONTENTS

1. HARDBOX INTRODUCTION

1.1. HardBox features

The HardBox introduces your PET to the world of high-capacity, high speed disk storage. It acts as an intelligent controller for up to four Corvus Winchester disk drives. Each Corvus drive can have a capacity of 5, 10 or 20 million bytes. Here are some HardBox characteristics :

DISK CAPACITY :	5 to 80 Megabytes	
SOFTWARE INTERFACE :	compatible with CBM DOS vers 1 and 2	
MAXIMUM FILE SIZE :	16 Megabytes (sequential or relative)	
MAXIMUM RECORD SIZE :	256 Bytes (relative files)	
MAXIMUM NO. OF RECORDS :	65535 per relative file	
MAXIMUM NO. OF FILES :	Over 2000 on 5 Megabyte drive (more on larger drives)	
MAXIMUM NO. OPEN FILES :	13 per HardBox at any time	
DRIVE ACCESS TIME :	10 or 20 Mbyte drive :	80ms maximum 40ms average
	5 Mbyte drive :	240ms maximum 125ms average

The HardBox is connected to the PET via the IEEE bus, and to the Corvus disk drive by means of a 34-way ribbon cable. The IEEE device number is switch selectable within the range 8 to 15.

Software compatibility

The HardBox is designed to be software-compatible with PET DOS version 1 or 2, so that it will work with existing programs. The HardBox is designed to appear to the PET as a fast, high-capacity floppy disk unit. The only exceptions to this software compatibility are the memory-read, -write and execute commands which have been omitted on the HardBox.

Multi-user capability

When used with the Corvus range of drives, the HardBox provides much more than just a hard disk interface. By using an 8-way multiplexing unit, up to eight PET users equipped with a HardBox may simultaneously address the same hard disk. Each HardBox may be at a distance of up to 20 metres from the multiplexer. By using two levels of multiplexing, up to 64 users may be accommodated. The hard disk may be divided logically into user areas, and independent user work areas and shared central databases may co-exist on the same drive. User areas may be protected from unauthorised access by passwords. A special login command enables a user to log into any user area from any PET.

Video cassette backup (Mirror interface)

The Corvus Mirror option supported by the HardBox provides a fast, inexpensive means of backing up the contents of Corvus hard disks using a commercially available video cassette recorder. The backup speed is about 7.5 Kbytes per second (recording with quadruple redundancy), so that a 10 Megabyte drive may be saved in about 20 minutes. The data capacity of a video cassette is about 100 Megabytes.

Hardware requirements

The minimum system required to support the HardBox is :

1. PET 3000, 4000 or 8000 series computer with BASIC 2 or 4

2. To configure a hard disk drive initially, you will need access to either a cassette unit or a PET floppy disk drive.

3. You will need some form of backup storage for your hard disk. This may be provided by a PET floppy disk unit (or even a cassette unit) for small to medium sized files. Alternatively backup may be provided by a mirror unit (see above) or by having more than one hard disk drive in the system.

2. RUNNING UP YOUR HARDBOX

The HardBox is normally shipped configured for IEEE device 8 (which is the standard PET disk device address) and user number 0. During normal operation you will probably wish the HardBox to have address 8 so that the hard disk appears as the main disk unit. However if the HardBox is to operate in the same system as a PET floppy disk unit (e.g. for file transfer purposes), then either the floppy unit or the HardBox must be set to some device number other than 8. On the HardBox this may be done by altering internal switches on the printed circuit board (see appendix B), or by software. More information about user numbers will be given in later chapters. For configuring the drive, you should leave the user number as 0.

Connect the HardBox to the Corvus disk drive using the 34-way ribbon connector. The maximum ribbon cable length is 6 feet. BE SURE THAT THE CABLE ORIENTATION IS CORRECT, OR DAMAGE TO THE HARDBOX OR DRIVE MAY RESULT. THE RED STRIPE ON THE RIBBON CABLE GOES NEAREST TO THE CENTRE OF THE HARDBOX. ALSO, THE PIN 1 MARKING ON THE CORVUS DRIVE CORRESPONDS TO THE RED STRIPE ON THE RIBBON CABLE. Plug the HardBox into the IEEE bus of your PET system.

WARNING - BEFORE YOU USE YOUR DRIVE :
On the new (revision B) Corvus drives, there is a row of four switches tucked under the front panel of the drive, just below the BUSY, READY and FAULT lights. These are, from left to right, the LSI-11, CONSTELLATION, FORMAT and RESET switches. IMPORTANT- DURING NORMAL OPERATION (WITHOUT A CONSTELLATION) ALL FOUR SWITCHES SHOULD BE IN THE LEFTHAND POSITION. If a Constellation unit is attatched, the CONSTELLATION SWITCH ONLY (second switch from the left) should be switched to the right. The FORMAT switch must always be kept in the left hand position during normal use, or the drive controller firmware may be overwritten. The drive RESET switch should never be used when a disk operation is in progress, or data may be destroyed.

Now power up the the PET, the Hardbox and then the drive. On revision B drives, wait till only the READY light remains on the front of the drive before proceeding. The Hardbox will itself perform some internal RAM/ROM tests and then the front panel light will flash continuously until the drive is ready, and should stay on permanently when the drive comes up to speed. Before a new Corvus drive can be used with the HardBox it must first be configured. To test out a new drive and configure it for use by a single user you should load the CONFIGURE program from floppy or cassette. If you are using a floppy disk, then to avoid having to change the device number of either the floppy disk unit or the HardBox, you could load the CONFIGURE program without the HardBox connected to the PET, and then disconnect the floppy and plug in the HardBox before typing RUN.

When you run the CONFIGURE program, you will be given a list of options. For now, press the Q key (QUICK CONFIGURE) followed by a carriage return. The CONFIGURE program will then execute a formatting check on your drive and configure it with a single user area equal to the physical size of the disk. This will enable you to load and save progams on the hard disk, perform directory listings, create relative record files etc. using the commands described later in this manual.

Should you wish to use the drive in a multi-user environment, or re-configure a drive for a particular application, consult chapter 6 for further details of the CONFIGURE utility.

CARE OF THE DISK DRIVE :

The Corvus drive is considerably more durable than a floppy disk unit, but please observe the following points in order to avoid damaging your drive or data :

1. Always power the drive up after everything else, and switch it off first.

2. Do not obstruct the air holes on the drive or keep it in an enclosed space without adequate ventilation.

3. Keep the drive away from strong magnetic fields.

4. Ensure that the front panel switches are always kept in their correct positions (see paragraphs above).

5. Avoid any excessive mechanical shocks, such as dropping the drive.

3. HARDBOX CONCEPTS

USER AREAS

If required, the Corvus physical drive may be divided up into more than one logical user area, each of which appear to the user concerned as a complete PET disk unit. There may be up to 64 user areas, with user numbers from 0 to 63. Each user area has a name of up to 16 characters, which may be for example the actual name of the user, in the case of an area used for program development, or the particular application in the case of a shared central database.

The number and size of user areas is specified when the disk system is configured using the CONFIGURE utility. It cannot then be altered, however, without destroying the data files stored on the disk. Of course, your application may well be suited to just one user area occupying the whole disk (especially if it is a large database).

Users may log into any user area by means of the LOGIN command. User areas may optionally be protected by a password, which is set by using the !PASSWORD command. To log into such an area, the correct password must be given with the LOGIN command.

User areas may either be single-user, in which case the user is free to read and write files at will, or shared, in which case no user is allowed to create, delete or extend files, or set a password, but all users are allowed to read files and update records in relative files (although in the case of updating records, some form of simple interlocking using the !LOCK and !UNLOCK commands is recommended to prevent two users from simultaneously modifying a record).

BLOCK READ AND WRITE COMMANDS

A certain amount of storage within each user area may optionally be dedicated for use with direct access commands (block read and write). This allows programmers to maintain compatibility with existing programs which have been written using these commands rather than the newer relative record commands. In the HardBox the direct access area is kept seperate from the file storage area for security. (See the description of the CONFIGURE program in chapter 6 for more details).

DIRECTORIES AND DRIVE NUMBERS

Each user area has its own directory. The maximum number of files allowed in each user area is determined at system configuration time, but the larger number of files allowed, the more space takes up by the directory. Each user area is divided further into 10 logical "drives" numbered from 0 to 9. However, unlike the physical drives on a floppy disk unit, these drives

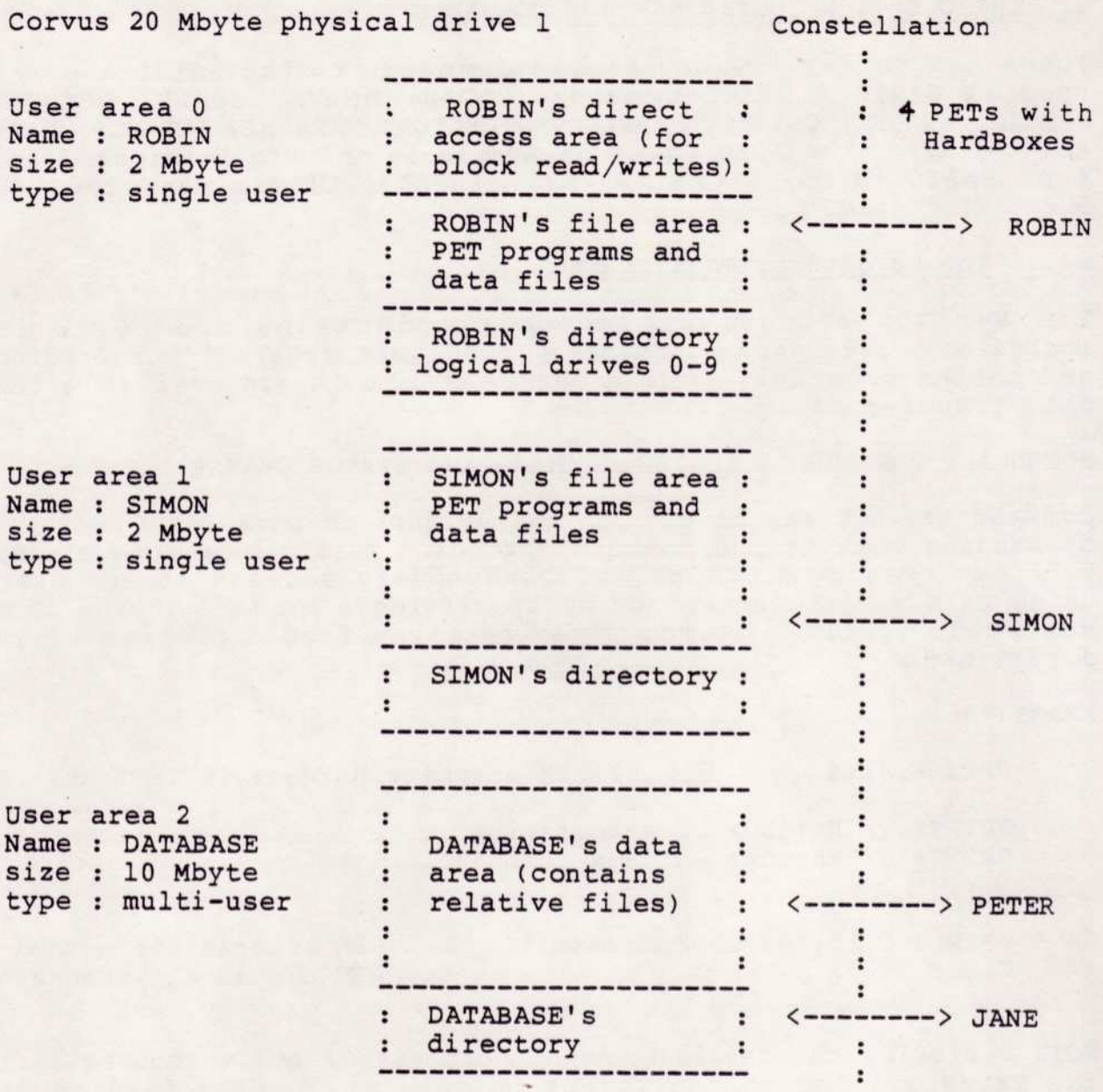
do not have a fixed size. Files may be added or deleted at will on any drive provided that the total space occupied does not exceed the size of the user area. Many existing applications written to work with PET floppies will only require drives 0 and 1, and the remaining eight drives can simply be ignored with no loss of storage space. Within a user area, a file may be global (i.e. the same file may appear on all ten drives), write-protected to prevent accidental erasing, or hidden so that it does not appear in directory listings.

Don't confuse these logical "PET floppy" drive numbers 0-9 with the Corvus physical drive number which determines which hard disk drive (1-4) is being addressed in a daisy-chain, or with the user number (0-63) which specifies the area of the disk you happen to be logged on to.

Here is an example of user areas in action :

The hardware configuration in our example is a 20-Megabyte Corvus drive with a Constellation unit and four PET's equipped with HardBoxes. This enables four users to be on-line to the drive at the same time. Robin and Simon each have their own user area in which they develop or use PET programs. Robin runs a word processing program in his user area, and Simon is testing an accounting program. Meanwhile Peter and Jane have shared access to another user area (called DATABASE) in which they are not allowed to create or delete new files, but can run a database program and update records in existing relative files (or in a direct access area). There is 6 Megabytes left on the drive for additional user areas, if required. The existing three user areas cannot be extended, however.

ROBIN, SIMON and DATABASE each automatically have PET logical drives 0-9 for use by programs and data files if required. The four users may gain access to their respective user areas by using the login command, or by means of the "default user area" switches inside their HardBoxes. Simon, Peter and Jane had better not try to write to ROBIN's single-user directory while Robin is also logged on, or files may be corrupted or lost. To avoid this, Robin could set a password on his user area.


Corvus 20 Mbyte physical drive 1
Constellation
User area 0
Name : ROBIN
size : 2 Mbyte
type : single user
ROBIN's direct access area (for block read/writes)
ROBIN's file area
PET programs and data files
ROBIN's directory logical drives 0-9
4 PETs with HardBoxes
ROBIN
User area 1
Name : SIMON
size : 2 Mbyte
type : single user
SIMON's file area
PET programs and data files
SIMON
SIMON's directory
User area 2
Name : DATABASE
size : 10 Mbyte
type : multi-user
DATABASE's data area (contains relative files)
PETER
DATABASE's directory
JANE

4. INTRODUCTION TO THE HARDBOX COMMANDS

There are three ways of issuing commands to the hardbox - the special BASIC 4 disk commands (DLOAD, DSAVE, DOPEN, DCLOSE, RECORD, HEADER, COLLECT, APPEND, DIRECTORY, RENAME, SCRATCH, COPY and CONCAT), the IEEE commands available in both versions 2 and 4 of BASIC (OPEN, CLOSE, PRINT#, INPUT#, GET#), and the DOS SUPPORT program.

4.1. The HardBox command channel

The HardBox recognises secondary addresses from 0 to 15 inclusive. Secondary addresses 0 and 1 are reserved for loading and saving programs, while channels 2 to 14 are available for data transfer to and from files.

SECONDARY ADDRESS 15 IS THE COMMAND AND STATUS CHANNEL.

Command strings may be sent to the HardBox in both BASIC 2 and 4 by sending them to the command channel. This can be done either by first opening a channel with secondary address 15 and then using PRINT# statements, or by specifying a command in the OPEN statement itself. Commands may be given from a program or in direct mode.

Examples :

```
OPEN 1,8,15                 : REM assuming HardBox is IEEE unit 8
  :
PRINT#1, "HardBox command string...
PRINT#1, "another HardBox command...
  :
  :
OPEN 2,8,15,"HardBox command"       : REM alternative method
CLOSE 2                             : REM closes all channels
```

NOTE - closing the command channel closes all active channels in the HardBox, and so should not be done while file reading or writing is in progress.

4.2. Reading the status channel

In order to find out whether a command was executed successfully, the status channel may be interrogated. This is done by reading from secondary address 15. This must be done from a program however, as the INPUT# command cannot be used in direct mode.

```
10 OPEN 1,8,15
20 INPUT#1, A$,B$,C$,D$
30 PRINT  A$,B$,C$,D$
```

Following completion of a command, the above program segment will read four strings from the HardBox error channel. These are

the error number, error message, and two further numbers called T and S (because on PET floppy units they contained the track and sector number at fault). These are occasionally used by the HardBox to convey further status information. For example after normal completion of a command the status returned is :

```
00,            OK,              00,          00
(error number) (error message)  (T)          (S)
```

Or after executing a SCRATCH command :

```
01,            FILES SCRATCHED,  03,          00
```

The T field tells you how many files were scratched. (in this case 3). See Appendix A for a complete list of error messages.

After executing disk commands in BASIC 4, the entire status string may also be read by displaying the variable DS$. Also the numeric variable DS contains the error number.

4.3. DOS Support Program

While the BASIC 4 disk commands that exist are self contained, to use the additional commands in the HardBox, and to issue commands from BASIC 2, you need to first OPEN disk channels. This can be especially inconvenient when operating in direct mode in BASIC, and so a utility program called DOS SUPPORT exists to communicate directly with the disk by providing direct-mode commands which bypass the normal BASIC input/output. This program will typically be the first program in the user area, so that it may be loaded on power-up by pressing SHIFT-RUN (BASIC 4) or by typing :

```
LOAD "*", 8
RUN
```

in BASIC 2. With DOS SUPPORT loaded the status channel may be read by typing :

```
>
```

followed by a carriage return. Command strings are sent to the disk by prefixing them with the ">" character :

```
>HardBox command
```

Programs may be loaded from the disk by typing

```
/program name
```

or they may be loaded and run by typing

```
^program name
```

DOS Support commands must start in the first screen column. The only disadvantage of using the DOS Support program is that it occupies some locations in high memory. This will not however affect the operation of most programs, unless they are specially written to make use of high memory.

4.4. FILENAMES, USER NAMES and PASSWORDS

These may be from one to 16 characters. They may contain any character, including spaces, except : = , * ? or carriage return.

WILDCARD FILE NAMES

The special characters "*" and "?" are recognised by the HardBox in file names. "*" tells the HardBox that the rest of the file name is insignificant, while "?" will match any single character. For example the wildcard file name :

```
FIL*
```

will match the files :

```
FIL
FILE1
FILEDATA
FILLER
```

and any other file names beginning with the characters "FIL".

```
P???FILE
```

will match :

```
PET FILE
POKEFILE
```

but not :

```
PETFILE
```

5. DESCRIPTION OF COMMANDS

5.1. DIRECTORY LISTINGS

General syntax :

```
BASIC 4 :       DIRECTORY
                CATALOG

BASIC 2 :       LOAD "$", 8
          then  LIST

DOS SUPPORT :   >$
```

The commands DIRECTORY and CATALOG are synonymous in BASIC 4. They are normally abbreviated to DIR or CA (with the R or the A shifted).

This command is used to display a list of the files which exist on a specified drive. For each file it displays :

- the file name
- thesizein blocks (1 block = 256 bytes)
- thefiletype(SEQUENTIAL, PROGRAM, USERor RELATIVE)
- whether the file is write protected
- whether the file is global to all drives

In addition the following information about the drive itself is displayed :

- drive name
- drive number
- drive id
- user name
- user number
- amount of free space left in the userarea

The amount of free space is given in kilobytes rather than in 256-byte blocks, since the PET will not handle line numbers larger than 65335 in a directory listing ! (1 Kbyte = 1024 bytes, i.e. 4 PET floppy disk blocks).

Here is a sample directory listing :

```
0 "DEMO            " XX 03
200  "FILE 1"         SEQ -W
46   "PROGRAM"        PRG G-
435  "RELFILE"        REL --
4589  BLOCKS FREE : TEST
```

First line :	0	- the drive number
	DEMO	- the drive name
	XX	- the drive id
	03	- the user number
Files :	200	- file size in blocks
	FILE 1	- file name
	SEQ	- file type
	W	- means the file is write protected
	G	- means the file is global
Last line :	4589	- number of free kilobytes in user area
	TEST	- the user name

The drive name and id are set by the NEW command (HEADER command in BASIC 4). They are provided purely for documentation purposes and have no special significance to the HardBox operating system.

The BASIC 2 form of the directory command works by loading the directory into the PET memory so that it can be listed as if it were a BASIC program. Thus it destroys any existing program in the PET memory, while the BASIC 4 and DOS support commands leave programs intact.

Listings may be obtained from a drive other than the current drive (from drive 1 for example) by using the syntax :

```
BASIC 4 :       DIRECTORY D1

BASIC 2 :       LOAD "$1", 8
                LIST

DOS SUPPORT :   >$1
```

However the BASIC 4 form only allows D0 or D1 to be specified.

Selective directory listings may be obtained with the BASIC 2 and DOS support commands by giving a wildcard file name :

```
BASIC 2 :       LOAD "$FR*"
                LIST

DOS SUPPORT :   >$FR*
```

This will list all files on the current drive whose names begin with FR. Drive and filename specifications may be combined, for example :

```
>$1:B??
```

Will list all 3-character file names on drive 1 starting with the letter B.

To obtain a directory listing from a device other than device 8 (device 12 for example), you can use the following syntax :

```
BASIC 4 :       DIRECTORY U 12

BASIC 2 :       LOAD "$", 12
                LIST
```

Files with the "hidden" attribute set do not normally appear in directory listings. However they may be listed as follows :

```
BASIC 2:        LOAD "$x:filename,h"
                LIST

DOS SUPPORT :   >$x:filename,h
```

The ",h" tells the DOS to list ALL files on the drive. In this case an extra column appears in the listing to show which files have the "hidden" attribute.

5.2. RENAME

Syntax :

```
BASIC 4 :       RENAME Dx, "old file" TO "new file"

BASIC 2 :       PRINT#n, "RENAME x:newfile=oldfile
                PRINT#n, "Rx:newfile=oldfile

DOS SUPPORT :   >RENAME x:newfile=oldfile
                >Rx:newfile=oldfile
```

This command changes the name of a disk file. X is the drive number containing the file - this must be specified. None of the file's other characteristics are affected. Files may be renamed on a device other than device 8, for example :

```
RENAME Dx, "oldfile" TO "newfile" U 12
```

Renames a file on device 12.

```
Possible errors :   FILE NOT FOUND - oldfile does not exist
                    FILE EXISTS  - newfile already exists
```

5.3. SCRATCH

syntax :

BASIC 4 : SCRATCH Dx, "filename

BASIC 2 : PRINT#n, "Sx:filename,x:filename,...

DOS SUPPORT : >Sx:filename,x:filename,...,x:filename

This command deletes one or more files from the specified drive X. Wildcards may be used in the filename. When the BASIC 4 SCRATCH command is issued from direct mode the computer will ask:

ARE YOU SURE ?

before actually deleting any files. The SCRATCH command may also be used to delete files from a disk unit with a nonstandard device number, for example :

SCRATCH Dx, "filename" U device

The number of files scratched (which can be zero) is returned in the "track number" field of the error channel, with the "FILES SCRATCHED" status code 01. Global files will not be deleted.

Possible errors : WRITE PROTECTED - a file had the "W" attribute set.

5.4. DRIVE NUMBER

Syntax :

BASIC 2 or 4 : PRINT#n,"Dx

DOS SUPPORT : >Dx

This command alters the default drive number (for directory listings etc). X may be a drive number between 0 and 9.

5.5. NEW DRIVE NAME

Syntax :

```
BASIC 4 :       HEADER "new disk name", Dx, Iid

BASIC 2 :       PRINT#n, "NEW x:new name,id
                PRINT#n, "Nx:new name,id

DOS SUPPORT :   >Nx:new name,id
```

X is the number of the drive to be renamed (0 to 9). ID is an optional two-character identification code (included solely for compatibility with PET DOS, where it is used to detect diskette changes).

5.6. WRITE PROTECT

Syntax :

```
BASIC 2 or4 :   PRINT#n, "Wx:filename          - protect
                PRINT#n, "-Wx:filename         - unprotect

DOS SUPPORT :   >Wx:filename                   - protect
                >-Wx:filename                  - unprotect
```

This command is used to set or clear the "write-protect" attribute of a file which prevents it being inadvertently erased or replaced. However a protected file can still be renamed, transferred to another drive, appended to or even updated (if it is a relative file). X is the drive number containing the file(s). Wildcard filenames may be used.

5.7. HIDE A FILE

Syntax :

```
BASIC 2 or4 :   PRINT#n, "Hx:filename          - hidden
                PRINT#n, "-Hx:filename         - revealed

DOS SUPPORT :   >Hx:filename                   - hidden
                >-Hx:filename                  - revealed
```

This command is used to set or clear the "hidden" attribute of a file, which prevents it being listed in normal directory listings. For example in a word processing system the word processor program itself might be hidden so that uncluttered directory listings may be obtained comprising only the document files being worked on. X is the drive number containing the file(s). Wildcard filenames may be used.

5.8. GLOBAL

Syntax :

```
BASIC 2 or4 : PRINT#n, "Gx:filename        - global
              PRINT#n, "-Gx:filename       - not global

DOS SUPPORT : >Gx:filename                 - global
              >-Gx:filename                - not global
```

This command is used to set or clear the "global" attribute of a file, which causes it to appear on all drive numbers simultaneously, while in reality only existing on one. This can save space, as an alternative to keeping several copies of the same file. X is the drive number containing the file(s). Wildcard filenames may be used.

Possible errors : FILE EXISTS - the file name was already in use on one or more drives.

5.9. TRANSFER FILES

Syntax :

```
BASIC 2 or 4 : PRINT#n, "TRANSFERx:filename,y
               PRINT#n, "Tx:filename,y
DOS SUPPORT :  >Tx:filename,y
```

Transfers the specified file from drive X to drive Y. Wildcards may be used.

5.10. INITIALIZE

Syntax :

```
BASIC 2 or 4 : PRINT#n, "Ix

DOS SUPPORT :  >Ix
```

This command is used in PET DOS to initialize drive X. It is provided in the HardBox for compatibility only and has no effect.

5.11. VALIDATE

Syntax :

```
BASIC 4 :       COLLECT Dx
                COLLECT Dx U unit

BASIC 2 :       PRINT#n, "VALIDATE x
                PRINT#n, "Vx

DOS SUPPORT :   >Vx
```

In PET DOS this checks all files on the disk and re-builds the block availability map (BAM) on disk. With the HardBox, no BAM is kept on disk, and this command re-logs into the current user area, deletes any improperly closed files, and re-builds tables in RAM.

5.12. SET PASSWORD

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!PASSWORD:new password
               PRINT#n, "!P:new password

DOS SUPPORT :  >!P:new password
```

This command is used to set or alter the current user's password. Passwords for shared user areas cannot be set using this command - these must be set using the system maintenance utility. To remove the password, just set a null password, for example :

```
    PRINT#n, "!PASSWORD:
or  >!P:
```

5.13. LOGIN

Syntax :

```
BASIC 2 or 4 : PRINT#n, "LOGIN:user name
               PRINT#n, "LOGIN:user name,password
               PRINT#n, "L:user name
               PRINT#n, "L:user name, password

DOS SUPPORT:   >L:user name
               >L:user name,password
```

This command is used to log into a new user area. If theuser area has a password set, then this password must be given, to guard against unauthorised access.

POSSIBLE ERRORS : UNRECOGNISED USER NAME
PASSWORD INCORRECT

5.14. COPY

Syntax :

BASIC 4 : COPY Dx, "oldname" TO Dy, "newname"

BASIC 2 : PRINT#n, "COPYy:newname=x:oldname"
PRINT#n, "Cy:newname=x:oldname"

DOS SUPPORT : >Cy:newname=x:oldname

Copies the named file from drive X to file "newname" on drive Y, which must not already exist.

5.15. CONCAT

Syntax :

BASIC 4 : CONCAT Dx, "file1" TO Dy, "file2

BASIC2or 4 :PRINT#n, "COPYx:file1=y:file2,z:file3...
PRINT#n, "Cx:file1=y:file2,z:file3...

DOS SUPPORT : >Cx:file1=y:file2,z:file3...

The BASIC 4 CONCAT command appends the contents of file1 on drive X to file2 on drive Y. File1 will remain unaltered.

The "C" command concatenates all files listed on the right hand side of the "=", generating file1 on drive X.

5.16. LOAD AND DLOAD

Syntax :

BASIC 4 : DLOAD "filename"

BASIC 2 or 4 : LOAD "filename", 8

DOS SUPPORT : /filename - load only
^filename - load and run program

This command loads a program file from disk into the PET memory. The program may then be run for example using the BASIC "RUN" command. The DLOAD command defaults to disk unit 8 and drive 0.

These defaults may be overriden as follows :

```
DLOAD "filename",D1          - loads from drive 1
DLOAD "filename", U unit     - loads from another unit
```

The LOAD, / and ^ commands default to the current drive number. This may be overriden by prefixing the file name with a drive number, for example :

```
LOAD "x:filename", 8         - load from drive X
```

The quick-load feature of BASIC 4 (pressing shift and run keys simultaneously) will load and run the first program file found on drive 0.

Wildcard filenames can be used. The first matching file in the directory is loaded.

5.17. SAVE AND DSAVE

Syntax :

```
BASIC 4 :       DSAVE "filename"

BASIC 2 or 4 : SAVE "filename", 8
```

This command saves a BASIC program from PET memory onto disk. The DSAVE command defaults to disk unit 8 and drive 0. These defaults may be overriden :

```
DSAVE "filename",D1          - saves on drive 1
DSAVE "filename", U unit     - saves on another unit
```

The SAVE command defaults to the current drive number. This may be overriden by prefixing the file name with a drive number, for example :

```
SAVE "x:filename", 8         - save on drive X
```

An existing file may be replaced by prefixing the filename with the "@" character, for example :

```
SAVE "@x:filename", 8
DSAVE "@filename", Dx
```

5.18. VERIFY

Syntax :

```
BASIC 2 or 4 : VERIFY "filename", 8
```

This command can be used to verify that a program has been correctly saved on disk, by comparing it with what is in the PET memory. The VERIFY command defaults to the current drive number. This may be overriden by prefixing the file name with a drive number, for example :

```
VERIFY "x:filename", 8          - verify file on drive X
```

5.19. OPEN, DOPEN AND APPEND

Syntax :

```
BASIC 4 :       DOPEN# lfn, "filename", Dx
                DOPEN# lfn, "filename", Dx, W
                DOPEN# lfn, "filename", Dx, Lsize
                APPEND# lfn, "filename", Dx

BASIC 2 or 4 : OPEN lfn, unit, sa, "fname,type"
               OPEN lfn, unit, sa, "fname,type,W"
               OPEN lfn, unit, sa, "fname,L,"+CHR$(size)
               OPENlfn,unit,sa,"fname,type,A"
```

These two sets of commands open a disk file for reading or writing. In each case there are four main forms of the command :

Open for reading :

The specified file must exist. The file type is optional in the OPEN command, but if present must correspond with the file type of the file on disk.

Open for writing :

The specified filename is created on disk and becomes open for writing. When the file has been written it must be closed using the CLOSE or DCLOSE command before powering down or RUNning another program, or data will be lost and the file will be flagged as being "incorrectly closed" in the directory (such files are marked by an asterisk in directory listings). The type of file created depends on the file type given in the OPEN command (which may be SEQ or S for sequential, PRG or P for program, or USR or U for user). Only sequential files may be created using the DOPEN command.

Open a relative file :

This form of the open/dopen commands creates or re-opens a relative record file to which data may be written and/or read at random points in the file. If the file does not exist, it is created, and the record length becomes "size" (see the RECORD command). The record size should be between 1 and 255.

Open for appending :

This form of the open command is used to write more data to an existing file on disk. It behaves like "open for read" but leaves the file open for writing additional data to the end of the file instead of for reading.

The DOPEN and APPEND commands default to unit 8 and to drive number 0. Other units and/or drive 1 may also be accessed, for example :

```
DOPEN#lfn, "filename", Dx, U unit
```

X is the drive number on which the file exists or is to be created. The OPEN command defaults to the current drive number. This may be overriden by preceding the file name with a drive number, for example :

```
OPEN lfn, unit, sa, "x:filename
```

LFN is the logical file number, which can be any value from 1 to 255 and must be unique among currently opened files. This number is purely for internal use by the PET and has no significance to the HardBox. However the secondary address SA in the OPEN command determines the channel number used to transmit data to or from the HardBox. This should be unique and in the range 2 to 14. The HardBox allows up to 13 files (any type) to be open at once. The secondary address is generated automatically when using the DOPEN or APPEND commands.

An existing file may be replaced by prefixing the filename with the "@" character, for example :

```
OPEN 2, 8, 2, "@x:filename,S,W
DOPEN#2, "@filename", Dx, W
```

A special form of the OPEN command opens a channel for direct access, for use with block read and write commands :

```
OPEN lfn, unit, sa, "#"
```

5.20. CLOSE AND DCLOSE

Syntax :

```
BASIC 4 :       DCLOSE lfn

BASIC 2 or 4 : CLOSE lfn
```

The DCLOSE command is used to close a file opened by DOPEN or APPEND, while CLOSE is used to close a file opened using OPEN. It is essential to close files opened for writing or appending,

or relative files which have been updated, to avoid data possibly being lost. In any case, CLOSE or DCLOSE will allow the logical channel to be re-used on the PET. The PET allows up to 10 logical files to be open at once. This includes cassette or printer files, etc. as well as disk files.

Closing the disk command channel (secondary address 15) also closes any currently open file channels within the HardBox. The BASIC 4 command :

```
     DCLOSE
or   DCLOSE U unit
```

has the same effect. The only problem is that the PET does not know this and so its corresponding logical files are still open. However they can be closed using CLOSE or DCLOSE as appropriate.

5.21. PRINT#

Syntax :

```
     PRINT# lfn
     PRINT# lfn, list of items
```

This has the same function as the BASIC PRINT statement but can be used to send data to a disk file which has been opened with logical file number LFN. If the LFN is less than 128, the data will be terminated by a carriage return, otherwise by a carriage return and line feed. Terminating the PRINT# statement with a semicolon means that no carriage return or line feed is printed.

When a PRINT# is executed on a relative file with the record pointer positioned past the end of the file, the file is automatically extended to include the new record (provided that there is sufficient space left in the user area).

5.22. INPUT#

Syntax :

```
     INPUT# lfn, list of variables
```

This command may be used to read the next data item(s) from a disk file which has been previously opened for reading using logical file number LFN. The values read in should have been written to the file separated by the carriage return character CHR$(13). This is generated automatically at the end of the PRINT# statement.

5.23. GET#

Syntax :

```
GET# lfn, string variable
```

This command may be used to read the next character from a disk file which has been previously opened for reading using logical file number LFN.

5.24. RECORD#

Syntax:

```
BASIC 4 :      RECORD# lfn, record
               RECORD# lfn, record, byte

BASIC 2 :      PRINT#n, "P";CHR$(sa);CHR$(record low)
                        ;CHR$(record high);CHR$(byte)
```

This command is used to position a relative file at a specified record number, and at a specified byte number within the record. In the BASIC 4 command this byte number defaults to 1 if it is not given. The record number must be between 1 and 65535. This is the only limitation (apart from the size of the user area) on the size of a relative file on the HardBox. The actual size of the file will be the record size times the number of records.

In the BASIC 2 form, the PRINT# must be to the command channel (secondary address 15) of the hardbox. SA is the secondary address associated with the relative file, RECORD LOW and RECORD HIGH are the low and high bytes of the desired record number :

```
RECORD LOW = (record number) AND 255
RECORD HIGH = INT ((record number) / 256)
```

5.25. BLOCK-READ and BLOCK-WRITE

Syntax:

```
BASIC 2 or 4 : PRINT#n, "BLOCK-READ sa,drive,track,sector
               PRINT#n, "B-R sa,drive,track,sector
               PRINT#n, "BLOCK-WRITE sa,drive,track,sector
               PRINT#n, "B-W sa,drive,track,sector

DOS SUPPORT :  >B-R sa,drive,track,sector
               >B-W sa,drive,track,sector
```

These commands are used to access any 256-byte sector from the direct access area within the current user area. SA is the secondary address of a direct access channel which must previously have been opened using the OPEN "#" command.

DRIVE, TRACK and SECTOR give the logical address of the 256-byte sector to be read or written. Virtually any separator may be used between parameters in the BLOCK-READ command, as long as there is no ambiguity. The track and sector numbers must fall within the required range, and provided that the resulting sector number is within the size of the direct access area, a sector is read from or written to disk according to the formula :

sector read or written =

((DRIVE*tracks per drive)+(TRACK-1))*sectors per track+SECTOR

so for example :

B-R sa 0 1 0

would read the first sector from the direct access area.

Otherwise an "ILLEGAL TRACK OR SECTOR" error is returned. The number of logical tracks per drive and sectors per track, and the size of the direct access area are user-specified during system configuration.

Before a sector is written to disk using BLOCK-WRITE, the first byte in the buffer (byte 0) is set to point to the last byte written to the buffer. Prior to a BLOCK-WRITE the user will typically position the buffer pointer to byte 1 using the B-P command and write up to 255 bytes of data to the buffer using PRINT# statements.

When a sector has been read into the channel buffer using BLOCK-READ, the byte counter in the channel is set to the contents of first byte read (byte 0). This counter points to the last valid byte of data in the buffer, and an EOI signal will be returned when this byte is read, causing termination of an INPUT statement. The channel pointer is set to byte 1, which will be the first data byte read by an INPUT# or GET# statement from secondary address SA.

See also the USER read/write commands U1 and U2, which allow access to all 256 bytes of each sector.

5.26. BUFFER-POINTER

Syntax :

BASIC 2 or 4 : PRINT#n, "BUFFER-POINTER sa, position
PRINT#n, "B-P sa, position

DOS SUPPORT : >B-P sa, position

This command is used to set the buffer pointer for a direct access disk channel (opened using the OPEN "#" command). SA is the secondary address of the channel, POSITION is the byte number within the buffer that will next be read/written using INPUT#, GET# or PRINT#. Bytes are numbered from 0 to 255 in the buffer. If you are using the BLOCK-READ/BLOCK-WRITE commands then bytes 1 to 255 may be used for data, if you are using the U1/U2 or ABSOLUTE-READ/ABSOLUTE-WRITE commands then all bytes may be used for data.

5.27. BLOCK-ALLOCATE AND BLOCK-FREE

Syntax : BASIC 2 or 4 : PRINT#n, "B-A drive track sector
PRINT#n, "B-F drive track sector

DOS SUPPORT : >B-A drive track sector
>B-F drive track sector

These commands are used to set and clear a bit in a block availability map (BAM) corresponding to the specified direct access block (DRIVE, TRACK, SECTOR). The BAM facility is provided purely for compatibility with PET DOS and is not used internally by the HardBox. The BAM is stored on disk rather than in the HardBox RAM, and is updated after each block allocate or free operation, so that it can be used in a multi-user environment (possibly making use of the !LOCK and !UNLOCK semaphore commands to ensure that the same block is not allocated simultaneously by two users).

With BLOCK-ALLOCATE, if the specified sector was not free the error status "NO BLOCK" is returned, with the track and sector number of the next free block in the T and S fields. If there is no free block the track and sector number returned will be zero.

5.28. USER

Syntax :

BASIC 2 or 4 : PRINT#n, "U1 sa, drive, track, sector
PRINT#n, "U2 sa, drive, track, sector
PRINT#n, "U:

DOS SUPPORT : >U1 sa, drive, track, sector
>U2 sa, drive, track, sector
>U:

UA, UB and UJ are also synonyms for U1, U2 and U:

U1 is like the BLOCK-READ command but does not use byte 0 in the buffer to determine the amount of data in the buffer. Instead the last data pointer is set to 255 and the buffer pointer is set to zero, allowing access to all 256 bytes in the buffer.

U2 is like BLOCK-WRITE, except that the position of the last byte written is not stored in byte 0 before writing to disk. Thus 256 bytes of data may be written to the disk.

U: or UJ performs a power-on reset of the HardBox.

5.29. ABSOLUTE-READ, ABSOLUTE-WRITE

Syntax :

```
BASIC 2 or 4 : PRINT#n, "ABSOLUTE-READ sa,drive,sector
               PRINT#n, "A-R sa,drive,sector
               PRINT#n, "ABSOLUTE-WRITE sa,drive,sector
               PRINT#n, "A-W sa,drive,sector

DOS SUPPORT :  >A-R sa,drive,sector
               >A-W sa,drive,sector
```

These commands are used to read or write ANY physiçal sector on the hard disk drive. SA is the secondary address of a direct access channel opened using the OPEN "#" command. DRIVE is the Corvus physical drive number (1 to 4), and SECTOR is the 256-byte physical sector number to be read or written. These are privileged commands and may only be used by user 0.

5.30. !LOCK AND !UNLOCK

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!LOCK:id
               PRINT#n, "!L:id
               PRINT#n, "!UNLOCK:id
               PRINT#n, "!U:id

DOS SUPPORT :  >!L:id
               >!U:id
```

These commands may be used to control access to shared files so that no two users can update the same file at the same time. ID is a user-specified name of up to eight characters that is used to identify a lock on the disk drive. The !LOCK command will set the specified lock and return the "OK" status, unless it was already set in which case "LOCK IN USE" will be returned. Thus a user program wishing to access a shared file should first loop trying to set a previously agreed lock, until the "OK" status is returned. Then the user may have sole access to the file until the program sends an !UNLOCK command, which makes the file available to the next user. For example :

```
90   OPEN 1,8,15
100  PRINT#1,  "!LOCK:XYZ
110  INPUT#1,  A$,B$,C$,D$
120  IF A$ <> "00" THEN 100
     :
     :   User accesses file
     :
200  PRINT#1,  "!UNLOCK:XYZ
```

Should two users issue a !LOCK command simultaneously with the same id, only one will receive the "OK" status. UP TO 32 LOCKS MAY BE IN USE AT ANY ONE TIME

5.31. !CLEAR

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!CLEAR
               PRINT#n, "!C

DOS SUPPORT :  >!C
```

This command will clear any locks set on the Corvus drive (see the !LOCK and !UNLOCK commands).

5.32. !VERSION

Return Corvus version number.

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!VERSION n
     or        PRINT#n, "!Vn
     then      INPUT#n, err$, msg$, controller$, rom$

DOS SUPPORT :  >!Vn
     then      >
```

This command returns a status message (number 97 or 98) depending on whether drive N is a "revision A" or "revision B" Corvus. Also returned is the version number of the Corvus controller (in the "track" field), and the ROM version number in the "sector" field for revision B drives (on revision A drives this field gives the number of drives on line).

5.33. !SIZE

Return size of Corvus drive.

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!SIZE n
     or        PRINT#n, "!Sn
     then      INPUT#n, err$, msg$, drive$, size$

DOS SUPPORT :  >!Sn
     then      >
```

This command returns status value 96, with the drive number N echoed in the "track" field, and the "sector" field equal to 05, 10 or 20 depending on the drive size (00 if the drive is not present in the system).

5.34. !HARDBOX VERSION

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!H
     then      INPUT#n, err$, msg$, version1$, version2$

DOS SUPPORT :  >!H
     then      >
```

This command returns a status value (number 99), with the "track" and "sector" fields equal to the major and minor revision numbers of the HardBox firmware.

5.35. !DEVICE NUMBER

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!D device

DOS SUPPORT :  >!D device
```

This command can be used to change the IEEE device number of the HardBox by software. DEVICE is the new device address. The HardBox will revert to its default device number when a reset is executed.

5.36. !NUMBER OF CURRENT USER

Syntax :

BASIC 2 or 4 : PRINT#n, "!N

DOS SUPPORT : >!N

This command returns status value 89 with the "track" field equal to the current user number.

5.37. !MIRROR COMMANDS

Syntax :

```
BASIC 2 or 4 : PRINT#n, "!MB              - backup user area
               PRINT#n, "!MBD drive       - backup drive
               PRINT#n, "!MR              - restore user area
               PRINT#n, "!MRD drive       - restore drive
               PRINT#n, "!MI              - identify a file
               PRINT#n, "!MV              - verify a file
```

These commands are intended for use by the MIRROR utility program and are documented here only for completeness. The status values that may be returned are :

All mirror functions :

```
OK
MIRROR ERROR - the "track" field specifies whichtype.
DRIVE ERROR  - the "track" field specifies which type.
```

Restore or verify commands :

```
TAPE READ ERRORS - the "track" field specifies how many
                   errors
```

Backup command :

```
BACKUP ERRORS - the number of disk read errors is given by
                the "track" field
```

In addition with the IDENTIFY and BACKUP commands, tape header information is passed to or from the Hardbox using channel 14 (which should have been opened as a direct access channel) :

IDENTIFY (read from channel 14 after command completion)

tape image id	1 byte
tape image size	2 bytes
system name	16 bytes
date	16 bytes
time	16 bytes
image name	16 bytes
comments	2 x 64 bytes

BACKUP (write to HardBox channel 14 before sending command)

date	16 bytes
time	16 bytes
comments	2 x 64 bytes

6. HARDBOX UTILITY PROGRAMS

Certain utilities are designed to be used by user 0 only, since vauable data may be destroyed by their misuse. The following utilities fall into this category :

CONFIGURE
SYSTEM MAINTENANCE
CORVUS DIAGNOSTICS

6.1. CONFIGURE

The CONFIGURE utility can be used to build a customised single- or multi-user Corvus system for use with the HardBox. It allows you to (re-)define one or more user areas on the disk, and clears all directories. Thus any files currently stored on the disk will be lost. All passwords will be cleared. When loaded and run, the CONFIGURE program gives you three options -

Q - Quick configure

N - Configure a new system from scratch

M - Modify the existing configuration

The QUICK CONFIGURE option is covered in the chapter on "Running up your HardBox" and may be used where no multi-user capability or drive partitioning is required.

The N option (configure a new system) may be used as an alternative if more than one user area is required.

The M option (modify the existing configuration) enables you to add, delete or modify user areas, for example. It should be used only with drives which have been configured on a previous occasion, and not with new drives.

Using either the M or N option, the CONFIGURE program will display the current number and type of drives in the system, and give you the chance to alter this information if it is incorrect.

When the physical drive configuration has been established, the CONFIGURE program will display a list of any user areas currently in the system (there will be none in the case of a new system). You may now perform any of the following operations until you have the desired logical drive configuration :

C - Create a new user
D - Delete a user
M - Modify a user area
U - Display a list of users
S - Save new user configuration
Q - Quit from program (abort)

The C command will prompt you for details of the new user area :

User name

This is the name that will be assigned to the user area (for use with commands such as LOGIN). It may be 1 to 16 characters long, but don't use the ":", "," or "=" characters. Don't worry, the name can be altered at a later date using the SYSTEM MAINTENANCE utility.

Physical drive #

The drive number of the particular Corvus disk on which the user area is to be placed. This will always be 1 for a single drive Corvus system.

Size of user area in kilobytes

This is the total size of the user area (specified in multiples of 1024 bytes = 1 kilobyte), including the space for file data, directory information and direct access area. A check is made to ensure that there is sufficent room on the drive.

Maximum number of dir entries

This determines the maximum number of files allowed in this user area, as one directory entry will be required per file. Sufficient index blocks are also created to handle the maximum number of files the user may create, giving an overhead of about 500 bytes per file. In addition, because data is allocated on the disk in 2048 byte blocks, the practical limit on the number of files in a user area is around (data area size in Kbytes) / 2.5, allowing for example a total of 2000 files on a 5 Mbyte drive, or 4000 on a 10M byte drive. There there is little point in specifying a directory size larger than this limit.

Single or multi-user ?

As explained in the chapter on "HardBox concepts", single user areas are designed to be accessed only by one user at a time, and that user may set passwords on the user area at will. Attempts by another user to write to the same area simultaneously may result in file or directory information being lost from that user area, as each user's HardBox will then have a different idea of what files are on the disk.

Multi-user areas are designed to allow simultaneous access to files by more than one user, if desired. Operations such as creating, deleting or extending files then become illegal. Note that it is up to the applications programmer to implement a multi-user protocol for writing to records in a shared relative file. This may be done using the !LOCK and !UNLOCK primitives.

Direct access size (in kilobytes)

This gives the amount of space allocated for direct access (using the BLOCK-READ and WRITE commands), in multples of 1024 bytes. This size may be zero if no direct access is required, but if it is nonzero you will be asked to specify the number of logical sectors per track and tracks per drive (for compatibility with PET DOS commands). The direct access area specified will be automatically subtracted from the space made available for sequential, relative and program files.

The M command displays and allows you to modify the parameters of an existing user area.

The D command deletes a user area from the from the table. The remaining areas will then be renumbered as necessary, to form a continuous sequence from 0 upwards.

The U command lists all the user areas and their parameters in table form (this table is designed for an 80-column screen).

The Q command quits to BASIC without re-configuring the drive (In other words, this is an abort command).

The S command saves the re-configured system and clears all directories, headers and passwords. Then the program exits to BASIC. This command MUST be used in order to save any changes made to the drive configuration, otherwise the drive will be left in its previous state.

6.2. SYSTEM MAINTENANCE

This utility is used to effect minor changes in the drive configuration without the need for a complete re-configuration using the CONFIGURE program, which would destroy all files on the disk. THIS UTILITY MUST BE EXECUTED BY USER 0, AND IT IS UNWISE TO USE IT WHILE OTHER USERS ARE ON THE SYSTEM. Functions which may be performed are :

C - Create a new user area
N - Alter user name
P - Set (or clear) a password
T - Alter user type (single or multi-user)
F - Reformat a user area
U - Display a list of users
S - Save updated configuration
Q - Quit (abort) from this program

The C, U, S and Q commands behave as with the CONFIGURE program.

The N command can be used to rename a user area (nothing else in the user area is affected).

The P command allows a password to be set on any user area (or a password may be cleared by setting a null password). This command may be useful if any user forgets his or her password !

The T command lets you alter the single/multi-user status of a user area.

The F command allows you to clear a user directory, either in order to delete all the files or if for some reason the directory becomes corrupted. Using this command also gives you the chance to re-define the directory size (maximum number of files), or the direct access area parameters. Altering the size of the direct access area will result in previous direct access data being lost.

6.3. LOGIN PROGRAM

This program may be used to perform an initial log-in at power up, or subsequently to log into a different user area. The login program will ask for the name of the user area you wish to log into. You will then be prompted for a password if one is required.

6.4. USERS

This utility displays a list of the user areas on a HardBox system, in table form. "USERS (40-COL)" is a version to fit on 40 column screens without wrap-around, which gives a shortened form of the table. The information given for each user is :

User area name
User area size in kilobytes
Type of user area (S = single user, M = multi-user)
Maximum number of files allowed
Size (in kilobytes) allocated for direct access

80 column version only :

Physical drive number (for Corvus drives)
Starting sector number on drive (256-byte sectors)
of "tracks per drive" for direct access
of "sectors per track" for direct access

6.5. SET UNIT NUMBER

This utility will alter the IEEE unit number of the specified HardBox by software. The HardBox will revert to its normal device number on executing a hardware or software reset. The program will ask you for the old device number, and the number you wish to change it to.

6.6. FILE TRANSFER

This utility is normally used to transfer program or data files between the HardBox and PET floppy disks. It may also be used to transfer files between two HardBox systems or two floppy disk units. The two devices must have different IEEE device numbers so that the PET can address each in turn without confusion. You will be asked :

Source unit number ?

followed by

Destination unit number ?

You must specify which IEEE device the files are to be copied from, and which device they are to be copied to. Next the file transfer utility will ask :

File name to copy ?

You may specify the name of a single file, or a wildcard file name specifying a group of files. The source directory will be searched for all matching files. Finally the utility needs to know which "drive" on the unit to read files off and which drive on the destination unit to transfer them to :

Source drive ?

Destination drive ?

On floppy disks the drive number will be 0 or 1. On the HardBox it will be a logical drive number (0 to 9) within the current user area.

6.7. MIRROR UTILITY

This utility provides a backup mechanism for the hard disk using the optional Corvus Mirror video cassette recorder interface. The four commands available are :

BACKUP

This enables a backup copy of a whole Corvus drive (or alternatively just the current user area) to be made on video cassette. If you have more than one drive then each drive must be backed up separately. You will be asked for the date, time and up to two 64-character lines of comments you wish to appear on the tape header. Start the VCR when you are instructed to do so, and then press the RETURN key. There will be a delay of about 10 seconds to allow the tape to come up to speed before writing commences. You will be informed when the operation is complete in the event of any disk read errors occurring during the backup.

VERIFY

Having backed up a version of the current system, you may wish to verify that it has been recorded properly. To do this use the VERIFY command, position the tape at the start of the recording to be verified and start the tape when instructed to do so. The number of hard (unrecoverable) tape read errors will be printed if any occured.

RESTORE

This command is similar to VERIFY but actually restores the data to the disk, and might be needed in the event of a disk failure or accidental erasure. Either a whole disk or just the current user area may be restored. ONLY USER 0 MAY RESTORE A WHOLE DISK.

IDENTIFY

This command reads the header from a Mirror VCR recording so that it can be identified. The following information is displayed :

IMAGE ID	(always 1 for HardBox tapes)
IMAGE SIZE	(the number of kilobytes times 2)
SYSTEM	the type of system which recorded the tape (e.g. HardBox or CP/M)
DATE and TIME	when the recording was made
COMMENTS	2 lines of user supplied comments made at the time of recording

6.8. CORVUS DIAGNOSTICS

This utility allows you to perform any of three functions on the Corvus drive -

1. Head servo test
2. Read version number of controller
3. Format check
4. HardBox version number

Function (1) seeks continuously between the innermost and outermost data tracks to verify correct movement of the head. You will be asked for the physical Corvus drive number (which may range from 1 to 8, with 20 Mbyte drives appearing as two drives). In a single drive system this will normally be drive 1 (or drives 1 and 2 for a 20 Mbyte drive). Any errors encountered during the test will be reported. To stop this test, press any key on the PET keyboard. This function may only be performed by user 0.

Function (2) enables you to read the version number of the Corvus controller, the controller type (revision A or B), and the number and sizes of drives in the system.

Function (3) will check the specified Corvus drive (0-7) for bad sectors, and attempt to correct them if any are found. This takes a few minutes, and when complete the number of bad sectors found (and hopefully corrected) is printed. It is recommended that this format check command should be performed repeatedly until no bad sectors are found. The command may only be issued by user 0, AND SHOULD ONLY BE PERFORMED WHEN NO OTHER USERS ARE ON THE SYSTEM.

Function (4) will display the HardBox version number.

7. MULTI-USER OPERATION

7.1. Setting up the Corvus Constellation

The Corvus Constellation acts as a front-end multiplexing unit for the hard disk drive (or drives), polling up to 64 HardBoxes at high speed. Each user appears to have normal access to the drive, except that slight pauses may occur when several users access the disk simultaneously. When using the Constellation, the Corvus drive(s) should be powered up with the CONSTELLATION SWITCH (the second switch from the left at the front of the drive) switched to the RIGHT. All the other switches should be switched to the LEFT.

Consult the Constellation manual for details of how to connect the Constellation to the HardBoxes and drive(s).

7.2. User area zero

User area 0 is rather special - it allows the user logged into it to access potentially dangerous HardBox commands. In a multi-user system it is a good idea to reserve this user area for special circumstances where system maintenance is required, and to keep the following utilities in the user 0 directory :

```
CONFIGURE
SYSTEM MAINTENANCE
CORVUS DIAGNOSTICS
MIRROR
```

It may also be a good idea to keep a closely-guarded password on this user area to prevent inadvertant or mischievous use of these utilities.

7.3. User area switches

The internal switches on the HardBox printed circuit board may be used to determine the user area (0 to 31) to default to on power-up (see appendix B). User areas 32 to 63 can only be accessed using the login command. The switches will typically need to be changed once a multi-user system has been configured, to allow PET users to boot directly into their own user area or some dedicated appication program (which might be run simply by pressing the SHIFT-RUN keys on BASIC 4 PETs). The user can then log into other user areas if necessary (but must first of course know the corresponding passwords).

7.4. The LOGIN program

The LOGIN program provides a convenient way of allowing users to issue a LOGIN command to the HardBox in order to change user areas. In a system with lots of PETs and users it is usually convenient to have one small dedicated user area containing just the LOGIN program. The switches on HardBoxes may then be set to boot up in this user area initially, allowing any user to walk up to any PET and log into any user area to which he or she knows the password. Logging back into the "login" area would then effectively log-off a user and leave the equipment ready for another user.

A. ERROR MESSAGES

00 - OK
No error condition exists

01 - FILES SCRATCHED
This is not an error condition. The T field tells you how many files (if any) were deleted by a preceding SCRATCH command.

20 - READ ERROR
An error occurred reading from the Corvus disk. The error type and the physical drive number are returned in T and S.

25 - WRITE ERROR
An error occurred while attempting to write to the Corvus disk. The error type and the physical drive number are returned in T and S.

26 - WRITE PROTECTED
Attempt to scratch or replace a write-protected file

30 - SYNTAX ERROR (GENERAL)
The DOS cannot interpret the command sent to the command channel. Typically the wrong number of parameters has been given, or they are the wrong type.

31 - SYNTAX ERROR (INVALID COMMAND)
The DOS does not recognise the command type given. Commands should start in the first column.

33 - SYNTAX ERROR (NO FILE GIVEN)
A file name has been omitted from the command, or the DOS did not recognise it as such because it was not preceeded by a drive number which is necessary in some commands.

50 - RECORD NOT PRESENT
This status message is the result of attempting to read past the last record of a relative file, or of executing a RECORD# command which exceeds the current length of the file. This error may be ignored if the intention is to expand the file by issuing a PRINT# command.

51 - OVERFLOW IN RECORD
Attempt to PRINT# past the end of a record in a relative file. The information written is truncated. The carriage return sent as a record terminator by a PRINT# statment must be included in the record size.

60 - WRITE FILE OPEN
This message warns the user that a file being opened for reading has not been properly closed yet.

62 - FILE NOT FOUND
The requested file could not be found in the directory.

63 - FILE EXISTS
Attempt to create a file when a file with the same name already exists in the directory.

64 - FILE TYPE MISMATCH
The file type of a file being opened for reading does not match that specified by the user.

65 - NO BLOCK
Direct access block specified in BLOCK-ALLOCATE command was not free. T and S give track and sector # of next free block (zero if none).

66 - ILLEGAL TRACK OR SECTOR
Parameters given to the BLOCK-READ, BLOCK-WRITE, BLOCK-ALLOCATE, BLOCK-FREE, U1 or U2 commands were out of range.

72 - DISK FULL
Either the user area is full or there is no more space in the directory.

86 - MIRROR BACKUP ERRORS
Disk read errors occurred during a Mirror backup operation. T gives the number of errors.

87 - TAPE READ ERRORS
This status value is returned by the Mirror verify and restore commands. T gives the number of hard errors detected in the tape image.

88 - MIRROR ERROR
A fatal error occurred during a Mirror command. T gives the error type :

01=IDmismatch (tape ID should alwaysbe 1 for HardBox)
02 = Illegal restore command (should never occur on HardBox)
03 = Illegal retry command (should never occur on HardBox)
04= Size mismatch -- the data restored was the wrongsize to have been saved from the user area specified.
05 = Illegal opcode (should never occur on HardBox)
07 = Start of tape image not found (30 second timeout on playback).
08 = Position error
99 = Tape dropout during replay operation (5 second timeout)

89 - USER #
This status is returned by the !N (read user number) command. T is the current user number.

90 - UNRECOGNISED USER NAME
Attempt to log into an unknown user area.

91 - PASSWORD INCORRECT
Login command does not have the correct password.

92 - PRIVILEGED COMMAND
Attempt to execute a command which can only be issued by user 0. Many commands fall into this category when issued in a shared user area.

93 - BAD SECTORS FOUND
This message is returned from the !F (Corvus format check) command if one or more bad sectors were found and corrected. T is the number of bad sectors found (if more than 63 were found, then T = 63).

94 - LOCK IN USE
Not really an error condition, this status message is the result of a !LOCK command on a lock which has already been set, typically by another user competing for some shared resource. The program concerned should re-try until the lock is !UNLOCKed and this status is no longer returned.

95 - CORVUS DRIVE ERROR
This status is returned when a Corvus controller error is detected, other than during a read/write operation. The actual error code is returned in T.

96 - CORVUS DRIVE SIZE
See the !SIZE command. T = drive number, S = drive size.

97 - CORVUS REV A CONTROLLER VERSION #
98 - CORVUS REV B CONTROLLER VERSION #
This message is used to return the controller version number for a Corvus drive. There are two main types of controller - revision A and revision B. T is the controller version number and S gives the controller ROM version (rev B drives) or the number of drives on-line (rev A drives). See the !VERSION command.

99 - SMALL SYSTEMS HARDBOX VERSION #
This message is used to return the HardBox firmware version. T and S are the major and minor revision numbers.

CORVUS DRIVE ERRORS (returned by status 21, 25 or 96 above)

0 - Disk header fault
1 - Seek timeout
2 - Seek fault
3 - Seek error
4 - Header CRC error
5 - Re-zero (head) fault
6 - Re-zero timeout
7 - Drive not on line
8 - Write fault
9 - - - - - - -
10 - Read data fault
11 - Data CRC error
12 - Sector locate error
13 - Write protected

14 - Illegal sector address
15 - Illegal command
16 - Drive not acknowleged
17 - Acknowlege stuck active
18 - Timeout
19 - Fault
20 - CRC
21 - Seek
22 - Verification
23 - Drive speed error
24 - Drive illegal address error
25 - Drive R/W fault error
26 - Drive servo error
27 - Drive guard band
28 - Drive PLO (phase lock) error
29 - Drive R/W unsafe

B. Printed circuit board switches

There is an 8-bit dual-in-line switch mounted on the printed circuit board which determines the following values :

1. Default user area

Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	User
open	open	open	open	open	0
open	open	open	open	closed	1
open	open	open	closed	open	2
open	open	open	closed	closed	3
open	open	closed	open	open	4
open	open	closed	open	closed	5
open	open	closed	closed	open	6
open	open	closed	closed	closed	7
open	closed	open	open	open	8
open	closed	open	open	closed	9
open	closed	open	closed	open	10
open	closed	open	closed	closed	11
open	closed	closed	open	open	12
open	closed	closed	open	closed	13
open	closed	closed	closed	open	14
open	closed	closed	closed	closed	15
closed	open	open	open	open	16
closed	open	open	open	closed	17
closed	open	open	closed	open	18
closed	open	open	closed	closed	19
closed	open	closed	open	open	20
closed	open	closed	open	closed	21
closed	open	closed	closed	open	22
closed	open	closed	closed	closed	23
closed	closed	open	open	open	24
closed	closed	open	open	closed	25
closed	closed	open	closed	open	26
closed	closed	open	closed	closed	27
closed	closed	closed	open	open	28
closed	closed	closed	open	closed	29
closed	closed	closed	closed	open	30
closed	closed	closed	closed	closed	31

2. HardBox IEEE device address

Switch 6	Switch 7	Switch 8	Address
open	open	open	8
open	open	closed	9
open	closed	open	10
open	closed	closed	11
closed	open	open	12
closed	open	closed	13
closed	closed	open	14
closed	closed	closed	15

Using Corvus Rev H drives with the Hardbox

Corvus have changed their hard disk design to use 5 1/4 drives with 20 sectors per track and 306 cylinders per drive. There are currently three sizes of Rev H drive :

Size	512-byte sectors	Bytes	User area max
"6 Mbyte"	11540	5,908,480	5,754 K
"11 Mbyte"	23700	12,134,400	11,834 K
"20 Mbyte"	35860	18,360,320	17,914 K

These replace the old 6, 11, and 20 Mbyte Rev B drives.

These changes present two problems when used with the HardBox :

Head parking

The heads on the Rev. H drives have to be parked by software before the drive is shipped. You will probably receive a warning notice to this effect with the drive. The diagnostic utility has thus been extended to perform the head parking function. To park the heads on a Rev H drive :

1. Load and run the diagnostic utility (release 9 or later) :

```
LOAD "DIAG", unit
RUN
```

If you have an 8032 or an 8096, use the Petspeed compiled version which runs more quickly :

```
LOAD "DIAG.8032", unit
RUN
```

2. The program will ask for the IEEE unit number of the Hardbox (usually 8) and then display a menu of command options. Use the P function to park the heads (the program will ask for the Corvus drive number, which will normally be 1 unless you have several daisy-chained drives).

3. If the heads are successfully parked, the lights will go out on the drive and the message :

```
OK - THE HEADS ARE PARKED
```

will be displayed. You can then power the drive down and ship it. If you use this command on a Rev B drive the program will hang.

The new drive sizes

The other problem with using the Rev H drives is that their sizes are different to the old versions, and therefore different to what is expected by the CONFIG program. In particular the new 20 Mbyte drive is smaller in capacity than the old one.

When using CONFIG to set up a new drive, the following points need to be borne in mind :

1. If using the Q(uick) configure option with the new 20 Mbyte drive, you cannot use the direct access commands (B-W, B-R, U1, U2) as the direct access area (which is stored last) will overlap the end of the physical disk.

2. If setting up user areas using the N or M options, observe the maximum capacity of the disk rather than relying on the CONFIG program to check this. i.e. on a "20 Mbyte" drive the sum of all user area sizes should not exceed 17900 Kbytes.

3. A new CONFIG program will be prepared in due course, but you will not be able to alter the user area sizes (i.e. reconfigure) without losing the data you have stored on the drive.