

13. PCOS COMMANDS

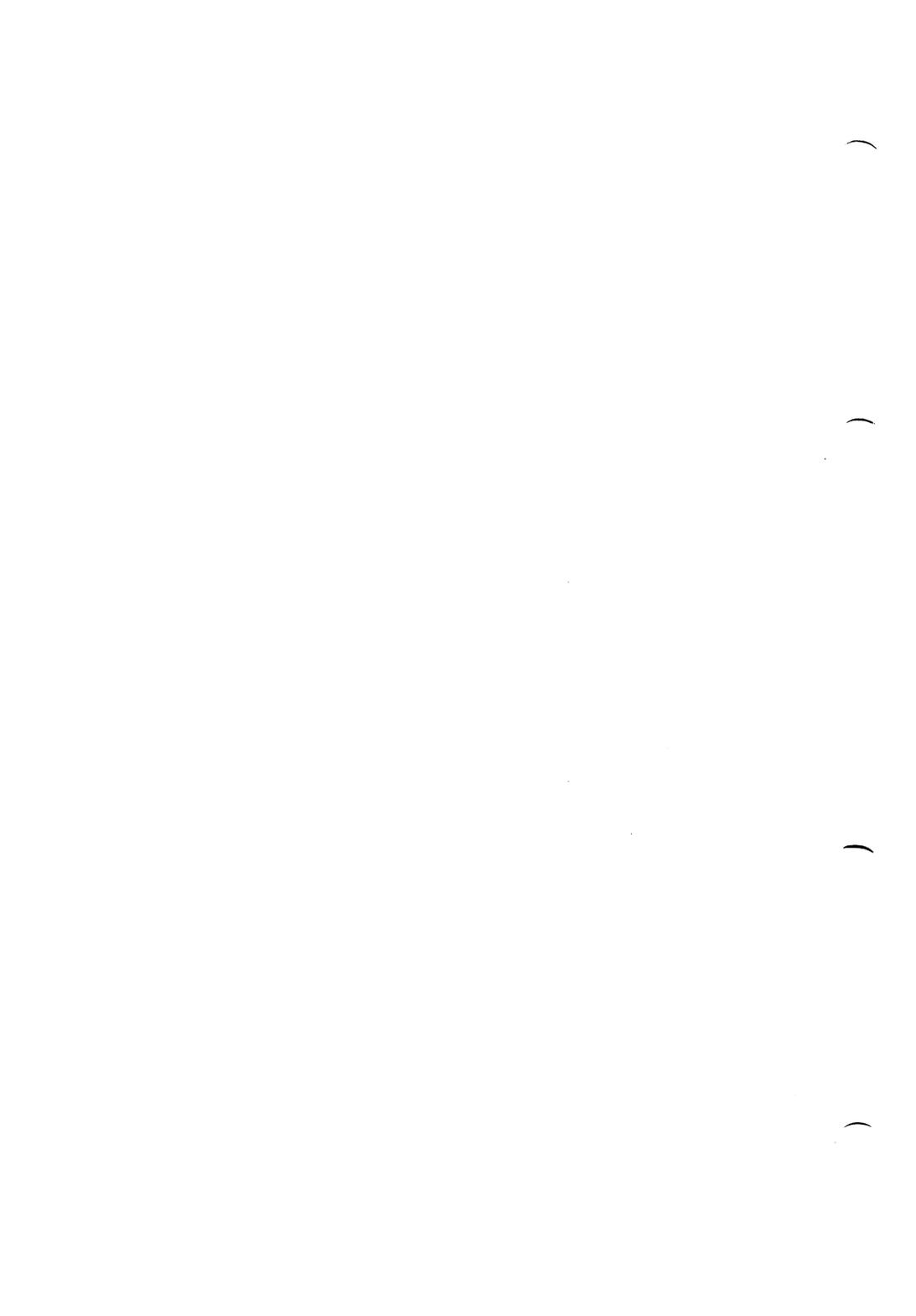
ABOUT THIS CHAPTER

This chapter provides a reference of all PCOS commands. Each command is described in terms of its purpose, a syntax diagram, a description of each syntax element, characteristics of the command, and examples.

CONTENTS

BASIC.CMD	13-1	FKILL.CMD	13-30
BKEYBOARD.BAS	13-2	FLIST.CMD	13-31
BVOLUME.SAV	13-3	FMOVE.CMD	13-33
CI.SAV	13-8	FNEW.CMD	13-35
CKEY.CMD	13-9	FPASS.CMD	13-37
COMMANDS.BAS	13-14	FRENAME.CMD	13-39
DCONFIG.CMD	13-15	FUNPROT.CMD	13-40
EDIT.CMD	13-18	FWPROT.CMD	13-42
EPRINT.SAV	13-18	HELP.BAS	13-44
ERROR.BAS	13-20	IEEE.SAV	13-44
FCOPY.CMD	13-20	LABEL.CMD	13-45
FDEPASS.CMD	13-26	LSCREEN.CMD	13-50
FFREE.CMD	13-28	LTERM (ALWAYS RESIDENT)	13-52

PKEY.CMD	13-53	VQUICK.CMD	13-102
PLOAD (ALWAYS RESIDENT)	13-57	VRENAME.CMD	13-104
PRUN.CMD	13-59	VVERIFY.CMD	13-106
PSAVE.CMD	13-61	WFONT.CMD	13-109
PUNLOAD (ALWAYS RESIDENT)	13-63		
RFONT.CMD	13-65		
RKILL.CMD	13-67		
RS232.SAV	13-69		
SBASIC.CMD	13-69		
SCOMM.CMD	13-72		
SDEVICE.CMD	13-73		
SFORM.CMD	13-75		
SLANG.CMD	13-78		
SPRINT.CMD	13-80		
SSYS.CMD	13-82		
VALPHA.CMD	13-86		
VCOPY.CMD	13-88		
VDEPASS.CMD	13-90		
VFORMAT.CMD	13-91		
VLIST.CMD	13-94		
VMOVE.SAV	13-96		
VNEW.CMD	13-98		
VPASS.CMD	13-100		



Loads the M20 BASIC interpreter into memory and optionally runs a specified BASIC program.

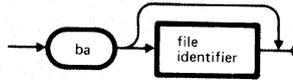


Fig. 13-1 BASIC

Where

SYNTAX ELEMENT	MEANING
file identifier	the file identifier of the BASIC program to be run

Characteristics

If the file is specified, the BASIC Interpreter is loaded, which in turn loads and executes the program stored under that file name. Following program execution the system remains in BASIC command mode.

The no-interaction (%n) flag cannot be used with this command.

Examples

IF you enter...	THEN...
ba my.program /CR/	the BASIC interpreter is loaded into memory, and the file "my.program" is run
ba /CR/	the BASIC Interpreter is loaded into memory and the M20 enters the BASIC environment, command mode

Enables BASIC verbs to be entered at the keyboard by striking the /COMMAND/ key in conjunction with alphabetic keys.



Fig. 13-2 BKEYBOARD

Characteristics

This command is designed for use with the Great Britain and USA ASCII with BASIC keyboards (see Appendix B).

After this command has been executed the BASIC verbs inscribed on the front of the alphabetic keys can be entered by pressing the key in combination with the /COMMAND/ key.

BASIC verbs can be made a permanent feature of your keyboard by using the PSAVE command.

Enables a BASIC program to use the "Search" and "DiskFree" system calls as well as to obtain the name of the current volume.

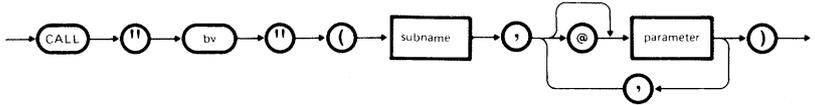


Fig. 13-3 BVOLUME

Where

SYNTAX ELEMENT	MEANING
subname	the name of the function to be used. This value may be one of <ul style="list-style-type: none"> <li data-bbox="434 950 1030 997">- search to search the volume directory for a specified file name string <li data-bbox="434 1020 1030 1067">- diskfree to return the number of free blocks remaining on the specified volume <li data-bbox="434 1091 1030 1138">- getvolname to return the name of the current volume
parameter	a parameter to be passed to the command. This depends on the function. See below for a description of the corresponding function

Characteristics

This command can only be CALLED from BASIC. It cannot be executed directly from the PCOS environment.

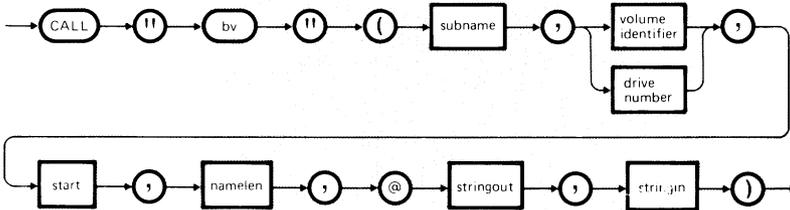


Fig. 13-4 BVOLUME search

Where

SYNTAX ELEMENT	MEANING
subname	the function name. In this case 'search'
drive number	the drive number in which to search, or -1 if the last selected drive is desired
volume identifier	the volume in which to search
start	a flag that determines where the search is to start from. It has two possible values:

	<p>1 - start searching from the start of the directory</p> <p>0 - continue searching from the position established by the last BVOLUME call</p>
namelen	the length of the file name string to be searched for. This value can be set to zero if all names are to be returned
stringout	a dummy string variable in which the file name searched for will be returned. It must be initialised prior to calling BVOLUME by setting it to at least 14 characters
stringin	the name of the file to be searched for. A group of file names may be specified using wild cards

Example

DISPLAY	COMMENTS
<pre> 10 SUBNAME\$="search" 20 DRIVENUM%=0 30 NAMELEN%=5 40 STRINGIN\$="*.cmd" 50 STRINGOUT\$="12345678901234" 60 START%=1T\$=""?Err.cmd " 70 CALL "BV"(SUBNAME\$, DRIVENUM%,START%,NAMELEN%, @STRINGOUT\$,STRINGIN\$) 80 PRINT"returned file name is:";STRINGOUT\$ 90 START%=0 100 GOTO 70 110 END </pre>	<p>Lines 10 to 60 set the parameters for a BVOLUME 'search' CALL on the volume inserted in drive 0. Line 30 sets NAMELEN% to the length of the string to be searched, while line 60 specifies the string; in this case it contains a wild card character. Line 50 sets the length of the output string to 14 characters.</p> <p>Line 80 CALLs the BVOLUME command and line 90 displays the returned file name. Line 90 sets the START% parameter to search for the next occurrence of a file whose name ends with ".cmd"</p>

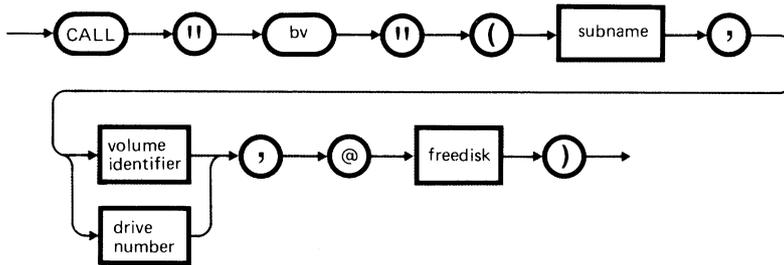


Fig. 13-5 BVOLUME diskfree

Where

SYNTAX ELEMENT	MEANING
subname	the function name. In this case 'diskfree'
drive number	the drive number containing the volume whose number of free blocks is required. If a value of -1 is entered then the last drive accessed is assumed
volume identifier	the volume whose number of free blocks is to be returned
freedisk	an integer variable in which the number of free blocks will be returned

Example

DISPLAY	COMMENTS
<pre> 10 DEFINT F 20 SUBNAME\$="diskfree" 30 V_ID\$="myvol:" 40 FREEDISK=0 50 CALL "bvolume" (SUBNAME\$,V_ID\$, @FREEDISK) 60 PRINT "free blocks ..."; FREEDISK 70 END </pre>	<p>Lines 20 to 40 set the parameters for a BVOLUME "diskfree" CALL on the volume inserted in drive 0.</p> <p>Line 50 CALLs the BVOLUME command.</p> <p>Line 60 displays the number of free blocks</p>

Remark

BASIC always uses signed arithmetic with integer variables. Therefore, if the "diskfree" option of BVOLUME is used on the hard disk, the returned value must be converted to double-precision before being displayed or used in calculations.



Fig. 13-6 BVOLUME getvolname

Where

SYNTAX ELEMENT	MEANING
subname	the function name. In this case 'getvolname'
volume name	a dummy string variable in which the name of the last volume accessed will be returned

Example

DISPLAY	COMMENTS
10 SUBNAME\$="getvolname" 20 VOLNAME\$="12345678901234" 30 CALL "BV"(SUBNAME\$, @VOLNAME\$) 40 PRINT "the current volume is named"; VOLNAME\$ 50 END	Line 10 specifies 'getvolname' as the subname parameter. Line 20 defines the string variable into which the volume name will be returned. Line 30 CALLs the BVOLUME command to perform the required function. Line 40 displays the returned volume name

Allows a BASIC program to interface with the RS-232-C driver. It can only be used from BASIC.

For further details refer to "I/O with External Peripherals User Guide".

Enables the user to change the character codes assigned to the raw key codes generated at the keyboard, or to set the shift lock for the alphanumeric and/or numeric keypads.

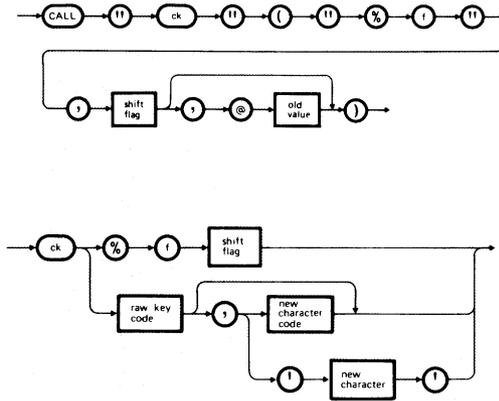


Fig. 13-7 CKEY

Where

SYNTAX ELEMENT	MEANING
f	when present indicates that the numeric value that follows specifies the shift flag value

<p>shift flag</p>	<p>an integer in the range 0 to 3 that determines the setting of the two shift flags - shift lock and cursor lock, where:</p> <p>shift lock - when set infers that all alpha keys on the alphanumeric keypad subsequently take on shifted values. That is, an alpha key struck on its own will generate an upper-case character. Moreover, an alpha key struck at the same time as holding down the SHIFT key will generate a lower case character</p> <p>cursor lock - when set, infers that all keys on the numeric keypad subsequently take on shifted values. That is, if such a key is struck on its own it will generate the code normally associated with pressing the same key in conjunction with the /SHIFT/ key. Likewise, if such a key is pressed in conjunction with the /SHIFT/ key, it will generate the code normally generated by striking the key on its own.</p> <p>The possible values are:</p> <ul style="list-style-type: none"> 0 - both flags cleared 1 - shift lock set, cursor lock cleared 2 - cursor lock set, shift lock cleared 3 - both flags set
<p>old value</p>	<p>a previously defined BASIC integer variable to which the command assigns a value which indicates the previous setting of the shift flag. This facility can only be used when the command is invoked from the BASIC environment using the BASIC CALL statement</p>
<p>raw key code</p>	<p>the code that is immediately generated by striking a particular key of the keyboard. This code is dependent only on the physical position of the key. That is, it is independent of the national keyboard and of any string assignments made by the PKEY command</p>

new character code	the code to be generated when the key or key combination specified by the raw key code is struck. The code may be specified as a decimal integer on its own, or as a hexadecimal integer preceded by &
new character	any character that can be entered from the keyboard

Characteristics

Figure 13-7 shows the raw key codes that are generated for every key whether struck on its own, in conjunction with the SHIFT key, with the CTRL key, or with the COMMAND key. The keys shown correspond to the physical position of the keys on the keyboard.

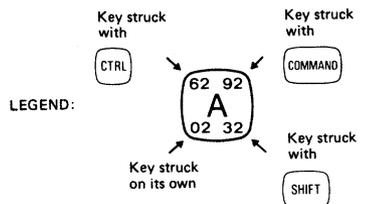
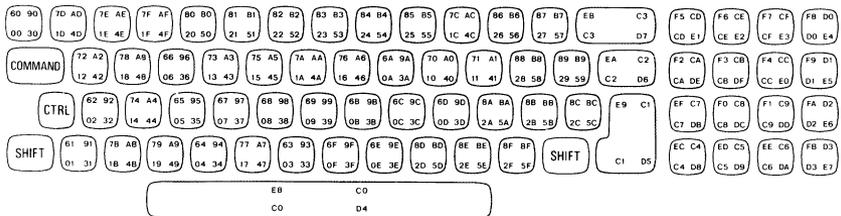


Fig. 13-8 Raw Key Codes

Each raw key code points directly into a table which translates the raw key code into the ASCII code for the keyboard character, or a function for the particular national keyboard. The translated raw key codes correspond to the standard ASCII codes (hexadecimal 0 to 80), while the remainder specify other keyboard functions. But note the special cases shown in the following table:

CODE (hexadecimal)	CORRESPONDING KEY COMBINATION	DESCRIPTION
A0	/CTRL/ /RESET/	Logical reset
A1	-	(reserved)
A2	/CTRL/ /C/	Break facility
A3	/CTRL/ /S/	Halt display
A4	/CTRL/ ///	Cursor lock
A5	/COMMAND/ ///	Shift lock
A6	/00/	Two zeros
A7	↵	End of line (CR in keyboard buffer, zero in LTERM buffer)
A8	/S1/	End of line (CR in keyboard buffer, '1' in LTERM buffer)
A9	/S2/	End of line (CR in keyboard buffer, '2' in LTERM buffer)
AA	↵ (DATEV keyboard)	End of line (CR in keyboard buffer, zero in LTERM buffer)
AB	/S1/ (DATEV keyboard)	End of line (CR in keyboard buffer, '1' in LTERM buffer)
AC	/S2/ (DATEV keyboard)	End of line (CR in keyboard buffer, '2' in LTERM buffer)
AD	-	(reserved)
AE	-	(reserved)
AF	-	No operation

The raw key codes are shown in hexadecimal in Figure 13-7. They may, however, be entered either as hexadecimal values preceded by an & (ampersand), or on their own as their decimal equivalents.

The user may enter the new character code either as a decimal integer, or as a hexadecimal integer preceded by an &, or the ASCII value can be entered simply by striking the corresponding key.

If, when entering a CKEY command, the raw key code is specified without also specifying a character code, then the current code corresponding to the raw key code is displayed.

The character codes assigned to the raw key codes remain until either changed by another CKEY command, or until the entire conversion table is replaced by means of the SLANG command, or the current working session is terminated. The changed values can alternatively be made permanent by use of the PSAVE command.

The no-interaction (%n) flag cannot be used with this command.

ANY CODE ASSIGNED TO A KEY USING THE CKEY COMMAND WILL BE EFFECTIVE EVEN AFTER ENTERING ANOTHER ENVIRONMENT OR APPLICATION PROGRAM. THIS ENABLES THE USER TO, FOR INSTANCE, RE-ARRANGE FUNCTION KEYS AT WILL, BUT THE USER MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT WILL BE REQUIRED ON ENTERING THE NEW ENVIRONMENT/APPLICATION PROGRAM.

Examples

If you enter...	THEN...
ck &C3,8 /CR/	the key S2 becomes backspace. That is, the decimal code for backspace (8) is assigned to the raw key code generated when S2 is struck (hexadecimal C3)

<pre>ck &C3 /CR/</pre>	<p>the raw key code and the corresponding ASCII code are displayed as follows:</p> <pre>KEY = 195 (raw key code) CODE = 8 (ASCII code)</pre> <p>(Note that values are displayed in decimal)</p>
<pre>ck &64,&AF /CR/ . . .</pre> <pre>ck &64, &A2 /CR/</pre>	<p>a no operation code (hexadecimal AF) is assigned to the raw key code generated when /CTRL/ /C/ is struck (hexadecimal 64), thereby disabling the break facility.</p> <p>the break facility is re-enabled</p>
<pre>ck &72,'w' /CR/</pre>	<p>the ASCII code for "w" is assigned to the raw key code 72 (hexadecimal). Entering /CTRL/ /Q/ will subsequently be the same as "w"</p>
<pre>ba /CR/ OLDVALUE%=0 CALL "ck" ("%f", SHIFTFLAG%,@OLDVALUE%) /CR/</pre>	<p>the M20 enters the BASIC environment from where the CKEY command is called, which in turn reads the shift value from integer variable SHIFTFLAG% and returns the old shift flag value in integer variable OLDVALUE%</p>

Displays a list of all PCOS commands.



Fig. 13-9 COMMANDS

Characteristics

The PCOS commands are listed, together with a brief description of their use and purpose. Further information about any particular PCOS command can be selected from a menu provided.

Remark

The COMMANDS utility resides on one of the Help diskettes.

Displays the hardware and/or memory configuration.

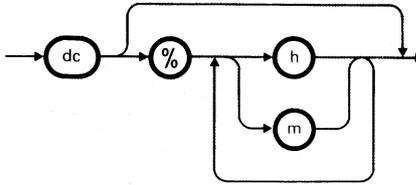


Fig. 13-10 DCONFIG

Where

SYNTAX ELEMENT	MEANING
h	the hardware configuration is to be displayed

m	the memory configuration is to be displayed
---	---

Characteristics

A typical hardware configuration would be displayed as follows :

Memory Configuration:	7.
Floppy Disk Drive(s):	320 KByte.
Drive 0 Diskette:	320 KByte.
Drive 1 Diskette:	320 KByte.
Hard Disk Drive:	Not Present.
IEEE Board:	Not Present.
RS-232 Board:	Not Present.
8086 CPU Board:	Not Present.
Display Type:	Black and White.

Note that on a 320 Kbyte drive, a 320 Kbyte diskette will be indicated even if there is non present. Furthermore, DCONFIG will not recognise a diskette change until the new diskette has been accessed.

The "Memory Configuration" value can be one of the following:

- 0 - eight-colour system with one or more 128K memory expansion boards
- 1 - four-colour system with one or more 128K memory expansion boards
- 2 - eight-colour system with one or more 32K memory expansion boards
- 3 - four-colour system with one or more 32K memory expansion boards
- 5 - black and white system with no memory expansion
- 6 - black and white system with one or more 128K memory expansion boards
- 7 - black and white system with one or more 32K memory expansion boards

Typically, the memory configuration (for a system containing two 32K memory expansion boards) is displayed as follows :

Memory Configuration:

Total memory size:	192 KBytes
Free memory size:	141936 Bytes
Basic memory size:	55000 Bytes

Address (Hex) Segment	Offset	Block Size Hex	Decimal	Filename (Owner)
00	0004	3FFC	16380	PCOS.SAV

Address (Hex) Segment	Offset	Block Size Hex	Decimal	Filename (Owner)
06	0004	7FFC	32764	FREE BLOCK
06	8004	3FFC	16380	PCOS.SAV

Address (Hex) Segment	Offset	Block Size Hex	Decimal	Filename (Owner)
02	0084	0178	376	PCOS.SAV
02	0200	1480	5348	FREE BLOCK
02	1684	109A	4250	PCOS.SAV
02	2722	D6DB	55000	FREE BLOCK

Address (Hex) Segment	Offset	Block Size Hex	Decimal	Filename (Owner)
0A	8004	7620	30240	FREE BLOCK
0A	F628	0788	1928	dconfig.cmd
0A	FDB4	004C	76	FREE BLOCK
0A	FE04	0028	40	dconfig.cmd
0A	FE30	012C	300	FREE BLOCK
0A	FF60	004E	78	PCOS.SAV
0A	FFB2	004E	78	PCOS.SAV

Address (Hex) Segment	Offset	Block Size Hex	Decimal	Filename (Owner)
09	8004	3FFC	16380	FREE BLOCK

If both the %h and %m flags are specified then both the hardware and memory configurations are displayed.

If no flag is specified then a display such as the following is shown:

```
L1.M20 System Configuration.  
Total memory size: 192 KBytes.  
Free memory size: 141936 Bytes.  
Basic memory size: 55000 Bytes.  
Display Type: Black and White.  
Disk Drive(s): 2 ready.
```

The no-interaction (%n) flag cannot be used with this command.

Example

IF you enter..	THEN...
<code>dc %hm /CR/</code>	the hardware and memory configurations are displayed

This command invokes the Video File Editor for editing a specified file.

Within the Video File Editor it is possible to perform a range of text editing functions. These are described in the Chapter 12.

Lists any error, or errors specified by their number with a brief, one line, description of each.

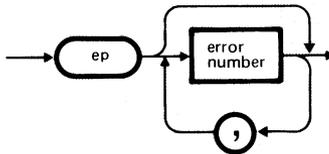


Fig. 13-11 EPRINT

Where

SYNTAX ELEMENT	MEANING
error number	any existing PCOS error number. If no error number is specified then no error is described, but the EPRINT command is loaded into memory

Characteristics

If an unprintable error or a BASIC error number is entered, no description is given.

Once the EPRINT command is resident in memory (via execution, PSAVE or PLOAD), then any errors returned from PCOS will not only be displayed with the error number, but will also have the associated descriptive label with it. Note that the EPRINT command has a SAV extension and therefore becomes resident on execution.

Example

IF you enter...	THEN...
ep 58,59 /CR/	the M20 will display the two errors as shown below: ERROR 58... file already exists ERROR 59... disk type mismatch

Displays a series of display frames that enable a user to display error messages in functional groups.

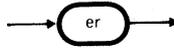


Fig. 13-12 ERROR

Remark

This command resides on one of the Help diskettes. The same information can be obtained via the HELP command.

- (1) - Copies the contents of one file into another.
- (2) - Copies one or more files (specified using wild cards) from one volume onto another

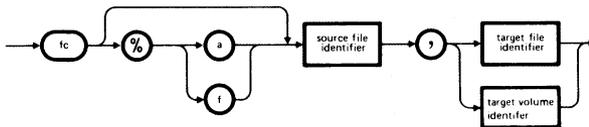


Fig. 13-13 FCOPY - (1)

Where

SYNTAX ELEMENT	MEANING
a	<p>the append flag.</p> <p>If the target file already exists and %a is specified, then the confirmation prompt will be suppressed and the contents of the source file will be appended to the end of the target file. If the target file does not exist, %a is ignored</p>
f	<p>the force copy flag.</p> <p>If the target file already exists and %f is specified, then the confirmation prompt will be suppressed and the contents of the target file will be overwritten with the contents of the source file. If the target file does not exist, %f is ignored</p>
source file identifier	<p>the file name, complete with any necessary password and volume identifier of the file to be copied</p>
target file identifier	<p>the name of a file on an unprotected volume, complete with any necessary password and volume identifier. If the file does not exist, it will be created and will have the same password as the source file</p>
target volume identifier	<p>the volume name or the drive number in which the target volume resides.</p> <p>If the volume is not enabled, then the password has to be specified</p>

Characteristics

If the target file exists and no flag is specified, FCOPY will prompt the user for confirmation before overwriting it.

File already exists. Do you wish to overwrite? (y/n)

If %n is specified in the command line then overwrite is automatically assumed.

If the target volume is specified the newly created file will have the same name and password as the source file.

Examples

IF you enter...	THEN...
fc %a dk1/dkpw1:FILE1, dk2:FILE2 /CR/	the contents of the file called "FILE1", on diskette "dk1", with password "dkpw1" are appended to the file called "FILE2", on diskette "dk2"
fc dk1:myfile, dk1:yourfile /CR/	a file may still be copied within the same volume. This example shows "myfile" is to be copied into "yourfile", both of which are on "dk1"
fc mydisk:*,10:	the entire contents of "mydisk" are copied onto the hard disk

Remarks

If there is sufficient space, a file can be copied or appended to another file on the same volume.

If the target file does not exist it will be created and given the same password (if any) as the source file.

If the target file already exists, it will maintain its password.

The target file must not be write-protected.

The copying process tidies-up the target file by gathering all the scattered data into one contiguous file - assuming there is enough space to do this. This saves on I/O operation time and is therefore worth doing on files that have become extensively fragmented.

At the end of an attempted FCOPY operation, error message 61 will result if there is insufficient space ("disk filled").

A file cannot be appended to itself.

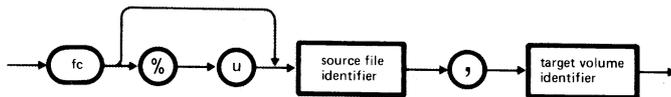


Fig. 13-14 FCOPY (2)

Where

SYNTAX ELEMENT	MEANING
u	the unprotected copy flag. If "%u" is specified, then the copy process will skip any copy-protected files specified by the source file identifier parameter (see below). If not specified, then any copy-protected files will still be allocated file space on the target volume even though the contents of the files are not copied
source file identifier	the volume identifier of the volume containing the files to be copied followed by a string of characters, including at least one wild card, that specifies one or more file names complete with any password, if it is common to all files specified Note: If the volume identifier is not specified, the M20 will search for the files specified on the volume inserted in the last drive accessed
target volume identifier	the volume name or the drive number in which the target volume is inserted. If the volume is not enabled, then the password has to be specified

Characteristics

The newly created files will have the same names and password as the source files.

If a file specified within the group already exists on the target file, then the prompt

File already exists. Do you wish to overwrite? (y or n)

is displayed. If %n is specified in the command line, then overwrite is assumed.

Example

IF you enter...	THEN...
fc %u 0:*.cmd,1: /CR/	all the files which have the extension CMD resident on the volume inserted in drive 0 are copied onto the volume inserted in drive 1. Any copy-protected files are skipped

Remarks

Any target files that do not exist will be created. Each newly created file will have the same password as the corresponding source file (if the source file has one). On the other hand, if the target file already exists, it will maintain its password.

A target file must not be write-protected.

The copying process tidies up the target file by gathering all the scattered data into one contiguous file, if there is enough space. This saves on I/O operation time and is therefore worth doing on files that have become extensively fragmented.

At the end of an attempted FCOPY operation, error message 61 will result if there is insufficient space ("disk filled").

Deletes a previously assigned file password on an unprotected and enabled volume.



Fig. 13-15 FDEPASS

Where

SYNTAX ELEMENT	MEANING
volume identifier	the volume name and/or drive number, plus any necessary volume password
file name	EITHER the name of an existing file OR a string of characters, including wild cards, that specifies a group of file names
old file password	the existing file password which is to be deleted

Examples

IF you enter...	THEN...
fd 0:*/secret /CR/	the user can optionally remove the password "secret" from any file on the diskette inserted in drive 0 by answering "y" (yes) or "n" (no) to the messages displayed
fd 0: myfile/newpass /CR/	the password "newpass" to the file called "myfile" on the diskette inserted in drive 0 is deleted
fd 1:FILX/PASSX /CR/	the password called "PASSX" on the file called "FILX" on the diskette inserted in drive 1 is deleted

Remarks

Since the password must be known before it can be deleted, this command cannot be used to gain access to a file for which the password has been forgotten.

Wild card characters cannot be used in the password portion of a file identifier. Thus, even though more than one file can be specified, you can only specify one password, which in turn can be the same for more than one file.

Frees any unused file sectors from a specified file or a group of files or an entire volume. The freed sectors are made available to the system for future use.

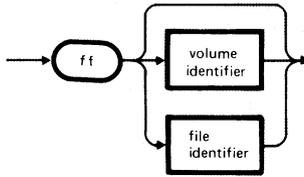


Fig. 13-16 FFREE

Where

SYNTAX ELEMENT	MEANING
volume identifier	the volume name or the drive number in which the volume resides, plus any necessary volume password
file identifier	<p>the name of the file plus any necessary password and volume identifier.</p> <p>The file name can be specified using wild card characters to specify a group of files.</p> <p>If the file identifier is left out, then FFREE will be executed on all the files resident on selected volume</p>

Characteristics

After entering the command the following prompt appears:

You may not change disks while FFREE in progress. Continue?

Replying "y /CR/" the operation continues. "n /CR/" aborts the operation.

If FFREE is used on the hard disk, any bad blocks that are allocated will not be freed.

Examples

IF you enter...	THEN...
ff 1:myfile /CR/	after replying "y" to the subsequent prompt, the file "myfile" resident on the diskette inserted in drive 1 is cleaned up of any unused sectors
ff myvol: /CR/	after replying "y" to the subsequent prompt, all the files on the volume "myvol" are cleaned-up of any unused sectors. That is, the entire volume is cleaned-up

Remarks

FFREE will not be executed if either the volume or the file is write-protected.

Deletes a specified unprotected file or group of files from an unprotected or enabled volume

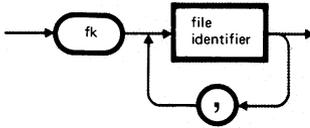


Fig. 13-17 FKILL

Where

SYNTAX ELEMENT	MEANING
file identifier	<p>the name of the file to be deleted, plus any necessary password and volume identifier.</p> <p>The file name can be expressed using wild cards to specify a group of files</p>

Characteristics

If the FKILL command is used on the hard disk, any bad blocks currently allocated to the specified file(s) will not be freed.

Examples

IF you enter...	THEN...
fk myvol: myfile/mypass /CR/	the file "myfile" with password "mypass" resident on the volume "myvol" is deleted
fk 1:myfile, yourfile /CR/	files "myfile" and "yourfile" are deleted from the diskette inserted in drive 1
fk mydisk:* /CR/	you can optionally delete any file resident on "mydisk" by answering "y" (yes) or "n" (no) to the interactive messages displayed

Remarks

FKILL will not function if either the volume or the file is write-protected.

Lists the contents of one or more files. The listing can be in either ASCII or hexadecimal form.

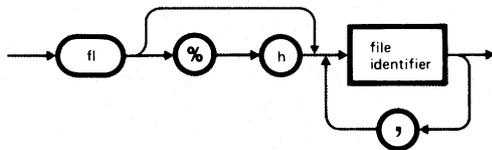


Fig. 13-18 FLIST

Where

SYNTAX ELEMENT	MEANING
file identifier	the name of the file, plus any necessary password and volume identifier. The file name can be specified using wild card characters to specify a group of files
h	the specified files are to be listed in hexadecimal form

Example

IF you enter...	THEN...
<code>fl dk2/diskpwd: myfile/flpwd /CR/</code>	the file "myfile" which has the password "flpwd", and resides on volume "dk2", which in turn has the password "diskpwd", is listed
<code>fl 0:F*/pass,1:G.1 /CR/</code>	all files beginning with the letter F and residing on the diskette inserted in drive 0 will be listed provided they all have the password "pass" File G.1 residing on the diskette inserted in drive 1 will also be listed
<code>fl %h 1:yourfile /CR/</code>	the file "yourfile" resident on the diskette inserted in drive 1 is listed in hexadecimal form

Copies a file from one diskette to another using only one drive.



Fig. 13-19 FMOVE

Where

SYNTAX ELEMENT	MEANING
source file identifier	the name of the file to be copied plus any necessary password
target file identifier	<p>the name of the file plus any necessary password.</p> <p>If the file does not exist it will be created.</p> <p>If the target file identifier is not specified then the source file is copied onto the target diskette into a newly created file of the same name as the source file</p>

Characteristics

When FMOVE is called the M20 displays a message asking the user to insert the source diskette in the drive. On hitting any key, the first part of the file is read into memory and consequently a message specifying the number of passes necessary to make the copy is displayed. At

this point the user is asked to insert the target diskette in the drive. On hitting any key the data that was read into memory will be transferred to the target diskette. If the copy operation takes n passes, the M20 will display messages asking the user to repeat the process described above n times. It is only after the last pass is completed that the PCOS prompt appears meaning that other commands can be entered.

FMOVE is only useful on M20 systems that have only one diskette drive. On dual diskette drive systems FCOPY would normally be used.

If the destination file already exists then the message

File already exists: overwrite, append or quit (o/a/q)

will be displayed.

If you enter "o", then the contents of the target file will be overwritten with the contents of the source file.

If you enter "a", then the contents of the source file will be appended to the target file.

If you enter "q", then the command is aborted, and the PCOS prompt appears on the screen.

Example

IF you enter...	THEN...
fm myfile,myfile /CR/	the FMOVE process is started, and if a diskette which has a file called "myfile" resident on it is inserted, then this file is read and stored in memory. If another diskette is now inserted, then the file "myfile" will be written on it

Remark

FMOVE can do all that FCOPY can do except:

- FMOVE does not allow wild cards whereas FCOPY does

- FCOPY deallocates blocks that are not used by the target file, while FMOVE does not. The number of blocks allocated to the target file after an FMOVE operation is always the same as that of the source file.

Creates an empty file on an unprotected or enabled volume, assigns a name to the file and optionally a password. The file system allocates only one extent (a series of contiguous blocks) to the file.

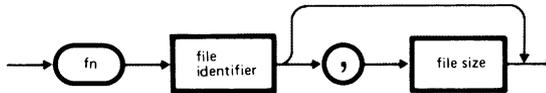


Fig. 13-20 FNEW

Where

SYNTAX ELEMENT	MEANING
file identifier	<p>the file name, optionally with a password and a volume identifier.</p> <p>If specified, the password is assigned to the file</p>

file size

the number of blocks to be made available to the file. The minimum value is 2 and the maximum value depends on the number of contiguous free blocks available on the volume.

There are two exceptions to the above:

(i) if this parameter is specified as 0 (or not specified at all) then the file system allocates the number of blocks specified in the "extent size" parameter (see the SSYS command), which by default is 8

(ii) if specified as 1 then the file system allocates two blocks to the file

Example

IF you enter...

THEN...

fn newfile/pass,12
/CR/

a 12 block file called "newfile" is created on the volume in the last drive to be selected and assigned the password "pass"

Remarks

If the volume has write-protection, the aluminised label must be removed before this utility can be used.

If there is not enough contiguous volume space to honour the request, a message

ERROR 61 ---- disk filled

will be displayed.

Assigns a password to a file, or a number of files, on an unprotected volume, or changes an already existing file password to a new one.

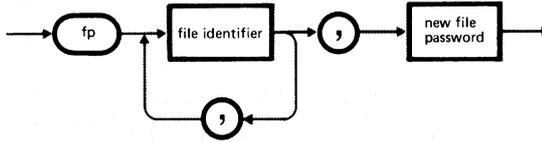


Fig. 13-21 FPASS

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the file name, plus any necessary password and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names plus any necessary password, if it already exists and is common to all the files specified
new file password	is an alphanumeric string to be assigned to the file(s) as a password

Examples

IF you enter...	THEN...
<pre>flist 0:myfile/pass /CR/ . . . fpass myfile/pass, newpass /CR/ . . . flist myfile /CR/ . . . flist myfile/newpass /CR/</pre>	<pre>file "myfile" is listed the password "pass" of the file called "myfile" is changed to newpass an error message ERROR 73 --- invalid password is displayed file "myfile" is again listed</pre>
<pre>fp 1:myfile,yourfile, ours /CR/</pre>	<pre>the password "ours" is assigned to both "myfile" and "yourfile"</pre>
<pre>fp 0:*,secret /CR/</pre>	<pre>all the file names on the diskette inserted in drive 0 are displayed one by one and an interactive message asks whether the passw- ord "secret" is to be set on the file. Entering "y" (yes) or "n" (no) for each file assigns or does not assign the password, respectively</pre>

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before issuing the command.

Like VPASS, the command FPASS can assign a new password directly to a file which already has one, but the user must specify the old password.

Once assigned, a password must be specified in all future references until it is deleted or changed. Otherwise, the file system will display error code 73 corresponding to the message "invalid password".

File (and volume) security is a user responsibility. It is important to remember passwords as neither PCOS nor BASIC will allow the user access to a file whose password has been forgotten.

To safeguard files from those who have access to them, use volume and file write-protection.

Changes the name of an existing file on an enabled or unprotected volume.

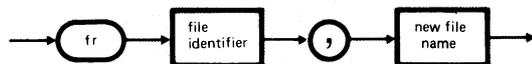


Fig. 13-22 FRENAME

Where

SYNTAX ELEMENT	MEANING
file identifier	the file name, plus any necessary password and volume identifier
new file name	any file name not currently used on the volume where the file being renamed resides

Example

IF you enter...	THEN...
fr 1:OLDname, NEWname /CR/	the file name "OLDname" of the file resident on the diskette inserted in drive 1 is changed to "NEWname"

Remark

The FRENAME command has no effect on the file password.

Removes write-protection from a file, or a number of files, on an unprotected or enabled volume.

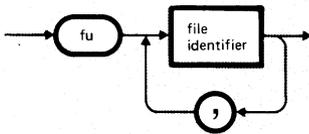


Fig. 13-23 FUNPROT

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the name of the file, including any necessary extension, password and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names complete with any necessary password, if one exists and is common to all the files specified

Examples

IF you enter...	THEN...
fu mydisk:myfile /CR/	the write-protection is removed from "myfile" which resides on the volume "mydisk"
fu DKI:* /CR/	you can optionally remove write-protection from any file resident on the volume "DKI", by answering "y" (yes) or "n" (no) to the interactive messages displayed

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before, and replaced after this operation.

Once write-protection is removed, the file can be deleted, overwritten, appended to, renamed, or in any other way modified (providing there is no write-protect label on the diskette).

If the file has no write-protection when this command is issued, this command will change nothing.

Write-protects a file, or a number of files, on an unprotected or enabled volume.

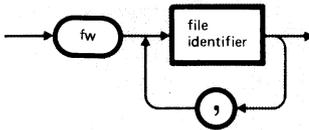


Fig. 13-24 FWPROT

Where

SYNTAX ELEMENT	MEANING
file identifier	EITHER the name of the file, plus any necessary password and volume identifier OR a volume identifier (if necessary) followed by a string of characters, including wild cards, that specifies one or more file names complete with any necessary password, if one exists and is common to all the files specified

Examples

IF you enter...	THEN...
fw myfile/diskps: myfile /CR/	the file "myfile" residing on volume "mydisk", which has a password of "diskps", is write-protected
fw 1:fileA, fileB /CR/	files "fileA" and "fileB" residing on the diskette inserted in drive 1 are both write-protected
fw 1:prog.??? /CR/	you can optionally apply write-protection to any file that has the file name "prog", with a three letter extension and which is resident on the diskette inserted in drive 1, by answering "y" (yes) or "n" (no) to the messages displayed

Remarks

If the file is on a write-protected diskette, the write-protect label must be removed before this operation and replaced afterwards.

Once write-protected, the file may not be modified until write-protection is removed with a FUNPROT command.

If the file already has write-protection assigned, this command will change nothing.

A write-protected file can be listed or copied.

Write-protection can be assigned to any file, whether it has a password or not.

If all the files of a particular diskette are only meant to be read, then the diskette should be write-protected.

Calls a series of display frames to guide the user about how to use PCOS and BASIC on the M20 system.



Fig. 13-25 HELP

Characteristics

The HELP command is an interactive utility that displays information about commands or errors.

Remarks

The information provided by the command and error sections is the same as that provided by the COMMANDS and ERROR commands.

Note: This utility is not supplied on the system diskette.

Loads and initialises the IEEE-488 extension package.

For more details about the IEEE-488 package see the "I/O with External Peripherals User Guide".

Displays a string with a specified colour, magnification and orientation at a given position, either within the entire screen or current window.

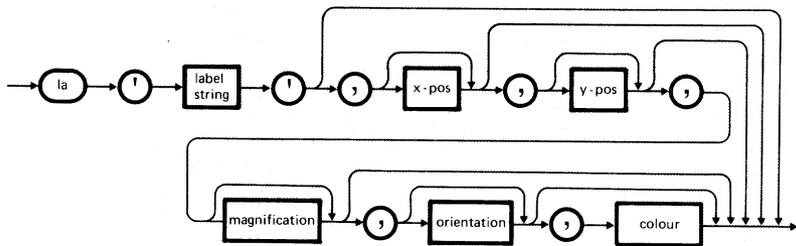


Fig. 13-26 LABEL

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
label string	<p>a valid string of printable characters that will be displayed at the specified position, magnification and orientation.</p> <p>The string must be entered with leading/trailing quotation marks</p>		

<p>x - pos</p>	<p>a legal value in pixels, used by the LABEL command to determine the distance of the left side of the first string character from the left hand edge of the screen or current window.</p> <p>The user can quote either:</p> <ul style="list-style-type: none"> - the pixel value, if you are in PCOS or BASIC and the hardware co-ordinate system has been assumed, or - the corresponding SCALEX function if you are in BASIC and a user co-ordinate system has been assumed (see "Remarks") 	<p>0</p>	<p>window width, less the character width, less 1 (all measurements in pixels)</p>
<p>y - pos</p>	<p>a legal value in pixels, used by the LABEL command to determine the distance of the string base from the bottom of the screen or current window.</p> <p>The user can quote either:</p> <ul style="list-style-type: none"> - the pixel value if you are in PCOS or BASIC and the hardware co-ordinate system has been assumed, or 	<p>0</p>	<p>window height, less the character height, less 1 (all measurements in pixels)</p>

	- the corresponding SCALEY function if you are in BASIC and a user co-ordinate system has been assumed (see "Remarks")		
magnification	a whole number n, where n is n times the normal character size	1	16
orientation	a value indicating the straight line aspect of the label string 0 - parallel to x-axis output left to right 1 - parallel to y-axis output bottom to top (+ 90 degrees) 2 - parallel to y-axis output top to bottom (- 90 degrees)	0	2
colour	the colour number in the range 0 to 7 for eight colour displays, 0 to 3 for four colour displays, or 0 or 1 for black and white displays. If omitted the foreground colour is assumed	0	7

Defaults

SYNTAX ELEMENT	DEFAULT
x-pos	0
y-pos	0
magnification	1
orientation	0
colour	current foreground colour set from BASIC

The parameters x-pos and y-pos assume their default values if they are not specified. If the command is PLOADed and x-pos and y-pos are not specified they will assume their default values if the LABEL command is being entered for the first time after PLOADing.

For successive LABEL commands these unspecified parameters will assume the value corresponding to the character position following the last character of the preceding label string displayed.

Orientation

The following table illustrates orientation.

ORIENTATION	DISPLAY
0	ABC →

1	QWVABC
2	QWVABC

Examples

IF you enter...	THEN...
la 'x-axis',,,8 /CR/ (in PCOS)	if the utility has not been PLOADED so that the given defaults apply, the "x axis" will be displayed starting at 0,0, with a magnification of 8 and orientation 0
SCALE 0,1,0,1 /CR/ CALL "la" ("title", SCALEX(.25), 1,0) /CR/ (in BASIC)	the label string "title" is displayed at magnification one and orientation 0 starting at a point on the screen one quarter of the way along the x-axis (0.25) and three quarters of the way along the y-axis (0.75) Note: In BASIC, strings must always be enclosed in double quotation marks
la 'name',,,1 /CR/	the string "name" will be displayed with normal orientation at co-ordinate (0,0) and without magnification. The last parameter indicates the colour in which the string is to be displayed

Remarks

LABEL is the only way that standard characters can be printed in graphics mode. It allows a string to be printed in various orientations and magnifications.

The string parameter must be specified.

If the parameters of a LABEL command are such that the label string does not entirely fit on the screen or within the specified window, the command will still be executed, and the label string will be displayed with the part that falls outside the screen or window clipped.

Windows cannot be opened under PCOS.

The hardware co-ordinate system is set only if one of the following conditions applies:

- the system is in the PCOS environment
- the system is in the BASIC environment and
 - . the video has not been split into windows
 - . 512 x 256 display mode is in use

Prints just the text displayed on the screen or specified window. Graphic elements are ignored.

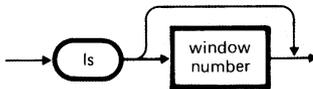


Fig. 13-27 LSCREEN.CMD

Where

SYNTAX ELEMENT	MEANING
window number	the number of the window to be printed. If this parameter is omitted then the current window will be printed

Characteristics

The specified window may contain text with any spacing. Any screen data within the normal 5 by 10 character dot matrix that is not recognisable as text will not be printed.

On colour systems, only screen plane 0 will be read. This means that if the background colour is even then the foreground colour must be odd, and vice versa.

The command does not distinguish highlighted displays on the screen, but both normal and inverse video characters will be recognised and printed.

Example

IF you enter...	THEN...
EXEC "ls 3" /CR/ (in BASIC)	the text included within window number 3 will be printed

Remark

Windows cannot be opened in PCOS.

Returns an integer (0, 1 or 2) corresponding to which of the three carriage returns (/↵/, /S1/ or /S2/) was last used.

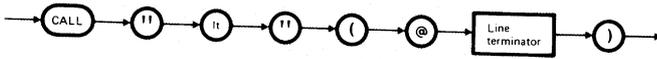


Fig. 13-28 LTERM

Where

SYNTAX ELEMENT	MEANING
line terminator	an integer variable which stores the value 0, 1 or 2, depending on whether the last line of input was terminated by /↵/, /S1/, or /S2/, respectively

Characteristics

The command is only usable in BASIC, using the CALL statement.

In a situation where a BASIC program prompts the user for an entry of some sort, the LTERM command can subsequently be CALLED to process the entry in one of three ways depending on which of the three line terminator keys was used to close the entry.

Note: The integer variable used in a CALL "lt" command must already exist in the program. For this reason the user has to define the name being used by setting it equal to some value in a preceding statement. Subsequently, CALL "lt" assigns the variable a value depending on which carriage return key was last pressed.

Example

DISPLAY	COMMENTS
<pre> 100 Input A\$ 110 Let I%=0 120 Call "lt" (@I%) 130 If I%= 0 then 1000 Goto 140 140 If I%=1 then 1100 or else Goto 1200 </pre>	<p>Line 110 defines the integer to be returned in line 120 which calls the LTERM command. Lines 130 and 140 determine whether the program is to continue at line 1000, 1100 or 1200 depending on whether /↵/, /S1/, or /S2/ was used to terminate the data element retrieved by statement 100</p>

Assigns a user-defined string to a particular key of the keyboard.

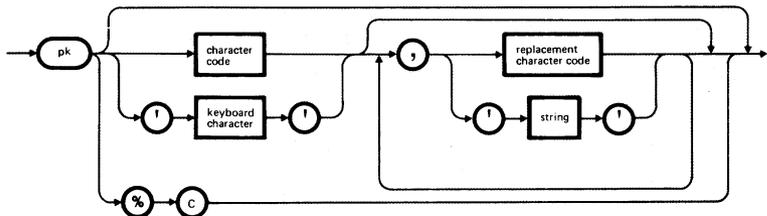


Fig. 13-29 PKEY

Where

SYNTAX ELEMENT	MEANING
character code	the decimal ASCII code, from 0 to 255, or hexadecimal code, from 0 to FF preceded by & corresponding to a key or combination of keys which are to be replaced
keyboard character	any of the characters on the keyboard, provided it has an ASCII equivalent (for example, A,a) excluding /RESET/, /CTRL/, /COMMAND/, /SHIFT/, /S1/, /S2/ and /↵/ to be assigned new representation
replacement character code	the ASCII code complying with the conditions described for "character code" which is to replace the original character
string	any string of characters the user finds useful to have available at a single key-stroke. The string can be up to 250 characters long The string must be entered with leading and trailing quotation marks
c	all strings previously assigned using the PKEY command are cancelled

Characteristics

Any PKEY command is cancelled by entering the character code parameter only. Since the entry of a character that has had a PKEY operation carried out on it causes the PKEYed string to be displayed, pk 'keyboard character' will not cancel the command. The character code itself must be used.

All assigned strings can be cancelled by specifying the %c program flag in the command line.

A key that already has a string assigned to it can be redefined using another PKEY command specifying the character code and the new string. It is not necessary to cancel the first string.

If the PKEY command is entered without a parameter then the defined keys are displayed along with their assigned strings. ANY VALUE ASSIGNED TO A KEY USING THE PKEY COMMAND WILL BE EFFECTIVE EVEN AFTER ENTERING ANOTHER ENVIRONMENT OR APPLICATION PROGRAM. THIS ENABLES THE USER TO, FOR INSTANCE, RE-ARRANGE FUNCTION KEYS AT WILL, BUT THE USER MUST TAKE CARE NOT TO DISABLE FUNCTIONS THAT WILL BE REQUIRED ON ENTERING THE NEW ENVIRONMENT/APPLICATION PROGRAM.

Examples

IF you enter...	THEN...
pk &41,&42 /CR/	each time "A" is pressed "B" appears. The ASCII character "A" is represented as hexadecimal &41, and "B" as &42

<p>pk '#','ba',13,10, 'files',13,10 /CR/</p>	<p>this time the keyboard character '#' is used, and is therefore enclosed within quotation marks. 13 is the decimal code for "carriage-return", 10 is the code for "line-feed". The command thus reads; when # is entered, display "ba", do a carriage-return/line-feed, display "files" and execute a further carriage-return/line-feed. The result is that the BASIC interpreter is entered and the command FILES is then executed on the volume inserted in the last selected drive</p>
<p>pk 237,'FILES "1:"', 13,10, /CR/</p>	<p>this command assigns the key combination /COMMAND/ /!1/ the string 'FILES "1:"',13,10</p> <p>Thus, when in the BASIC environment, the key combination /COMMAND/ /!1/ can be used to list the directory on the diskette inserted in drive 1</p>
<p>pk &41 /CR/</p>	<p>the effect set up in the first example above is cancelled</p>
<p>pk &23 /CR/</p>	<p>the effect set up in the second example above is cancelled.</p> <p>(# is represented by hex code &23)</p>
<p>pk /CR/</p>	<p>all defined keys are displayed along with their assigned strings</p>
<p>pk %c /CR/</p>	<p>all defined keys are cleared</p>

Loads a diskette-based or hard disk-based utility into memory.

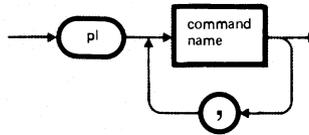


Fig. 13-30 PLOAD

Where

SYNTAX ELEMENT	MEANING
command name	the name of a transient command to be PLOADED

Characteristics

When a utility is PLOADED, it remains loaded in memory until either it is removed by the PUNLOAD command, or the current working session is terminated.

Once a utility has been PLOADED, the M20 displays the disk file name of the PLOADED utility (for example VCOPY.COMD), the program name (for example Volume Copy Rev.3.x), the operation mode (for example segmented/system), the entry points and memory allocated.

Example

IF you enter...	THEN...
pl vc /CR/	the VCOPY command is loaded into memory for the duration of the current working session.

Remarks

The user may wish to PLOAD a utility for the following reasons:

- to use the utility after the diskette on which it resides has been removed
- to save the time lost in having to load the utility from diskette (or hard disk) every time it is used

However, PLOADing utilities has the adverse effect of reducing user memory.

None of the utilities have to be PLOADed in order to be executed. If they are invoked without being PLOADed, they will be automatically sought, loaded and executed. Utilities having the CMD extension will then be removed from memory. But those having the SAV extension will remain until the end of the current working session.

The PLOAD command only works for transient commands. PLOAD, PUNLOAD and LTERM are all resident and thus cannot be PLOADed.

Loads and initialises alternative operating systems.

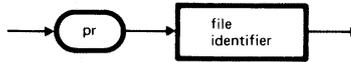


Fig. 13-31 PRUN

Where

SYNTAX ELEMENT	MEANING
file identifier	specifies the file containing the operating system to be loaded

Example

IF you enter...	THEN...
pr 0:alt.1 /CR/	the operating system previously saved in file "alt.1" on the diskette inserted in drive 0 is loaded and initialised

Characteristics

The command is used in conjunction with the PSAVE command. That is, an operating system must have been previously configured and then saved

using the PSAVE command before it can be reloaded and initialised by the PRUN command.

This command enables the user to configure different operating systems for different functions, then to alternate between the different operating systems. The following table shows an example of this feature:

STEP	OPERATION
1	Define an alternative operating system by setting global parameters, assigning functions to keys, and PLOADing selected commands
2	Save the new operating system in, for instance, file "alt.1" on the diskette inserted in, for instance, drive 1 by entering ps 1:alt.1 /CR/
3	Define another alternative operating system by setting global parameters, assigning functions to keys and PLOADing selected commands
4	Save it, for instance, on the same diskette by entering ps 1:alt.2 /CR/
5	Enter pr 1:alt.1 /CR/ to reload and initialise the operating system saved in "alt.1"
6	Enter pr 1:alt.2 /CR/ to reload and initialise the operating system saved in "alt.2"

Saves the current PCOS on an unprotected or enabled volume.

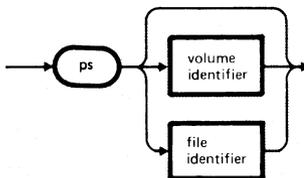


Fig. 13-32 PSAVE

Where

SYNTAX ELEMENT	MEANING
volume identifier	<p>a volume name or a drive number specifying the volume on which the current PCOS configuration is to be saved.</p> <p>If this parameter is omitted the PSAVE command will select drive 0 (even if a hard disk unit is configured)</p>
file identifier	<p>the name of the file which is to contain the current PCOS configuration, possibly preceded by a volume identifier.</p> <p>If the file identifier is omitted, the PSAVE command will assign the name PCOS.SAV</p>

Characteristics

The PSAVEd file will include any utilities which have been PLOADED, any programmed key definitions in effect, and the state of the system as set by the global commands (see Chapter 6). Moreover, any currently active user-defined font, and any globally specified device re-routing will also be included, as will control character display, if currently specified.

After entering a PSAVE command, the user is asked to confirm the intention to save PCOS on the specified file.

Save system on file drive number:file name ? (y / any other key)

A "y /CR/" response will complete the operation assuming the corresponding drive is ready, and that the specified volume is either unprotected or enabled. Entering any other key will abort the operation without error.

A PSAVEd file will follow other files that may already exist on the volume in question, but it is normal practice to PSAVE onto an empty diskette.

If a file with the same name as the PSAVEd file already exists on the specified volume, it will be overwritten. In any case, once the file is PSAVEd, the system is immediately re-booted via a standard boot file search (see Chapter 5). Furthermore, any future attempt to boot the system from this volume will cause the last PSAVEd file to be booted (unless the PRUN command is used).

Both write-protection and password protection can be applied to a PSAVEd file to protect it from being overwritten or modified by unauthorised users.

The no-interaction flag (%n) has no effect with the PSAVE command.

Examples

IF you enter...	THEN...
ps /CR/	the current PCOS configuration is saved on the diskette inserted in drive 0 in a file called PCOS.SAV as no file is specified
ps 1:alt1 /CR/	the current PCOS configuration is saved on the diskette inserted in drive 1 in a file called "alt1"

Remarks

You may find it is useful to use PSAVE in order to set the value of global parameters, for example, the default value for memory size, which may be set too low for some activities in BASIC.

Removes PLOADED commands from memory.

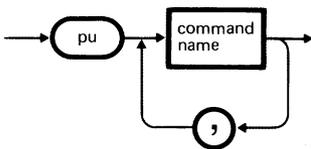


Fig. 13-33 PUNLOAD

Where

SYNTAX ELEMENT	MEANING
command name	the name of a PLOADED command to be made transient. Wild cards are not permitted

Characteristics

This command can be used to remove from memory commands that have either been PLOADED during the current working session or PLOADED and PSAVE during a previous working session. However, some commands change PCOS tables and therefore cannot be removed from memory. That is, the PUNLOAD command will not work on the following commands:

- CI
- RS232
- IEEE
- EPRINT
- VMOVE
- PDEBUG (not included on the system diskette)

Furthermore, the permanently resident commands PLOAD, PUNLOAD and LTERM cannot be removed from memory.

Example

IF you enter...	THEN...
<pre>pl vc,ps /CR/ . . . pu vc /CR/</pre>	<p>the transient commands VCOPY and PSAVE are loaded into memory</p> <p>the M20 responds</p> <p style="padding-left: 40px;">PUnloading vcopy.cmd</p> <p>and the VCOPY command is removed from memory and becomes transient once again</p>

Converts the system font data into a text file that describes each font character as a series of upper-case 'X's on a 10 by 8 matrix. This is the first step in creating a user-defined font.

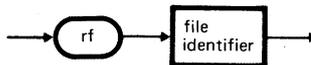


Fig. 13-34 RFONT

Where

SYNTAX ELEMENT	MEANING
file identifier	the name of the file in which the font matrices are to be created, plus any necessary password and volume identifier. If the file does not already exist it is created. If the file does already exist a prompt appears asking the user to confirm the intention. A "y" response overwrites the file

Characteristics

The file generated by the RFONT command comprises four lines of header information :

- the country identifier (USA, ITALY, etc.)
- the country number (see SLANG command)
- the font height (in scan lines). This must always be 10
- the number (decimal) of characters defined in the file (95 to 190)

This is followed by the font matrices. Each character is described in 11 lines, the first of which is the decimal code of the character, the remaining 10 describe the character, for example:

```
50
-----
----XXX-
---X---X
-----X
-----X-
-----X--
----X---
---XXXXX
-----
-----
```

where "50" is the decimal code of the number "2".

To change the character, invoke the Video File Editor on the file and place X's so that they show the intended appearance of the character. It is advised that the first two or three columns appear blank so that

characters do not run together. Once the changes are saved and the editor is exited, the WFONT command can then be entered to direct the system to display characters as they appear in the matrices.

The range of modified characters can be extended to twice the standard 95 character printable range. By adding 10 line font matrices (each preceded by the appropriate decimal code) to the end of the text font file, and by updating the "number of characters" that appears on the third line of the file, the desired number of characters can be defined. Confusing results may occur if matrix entries are not numbered consecutively from the decimal code 32, associated with "space" - the first printable ASCII character. Furthermore, the matrices must always be ordered in ascending ASCII sequence, otherwise the key/character correspondence will be destroyed. If the number of matrices exceeds the number of characters specified in the third line of the file, the extra fonts will not be defined.

Example

IF you enter...	THEN...
rf 1:italicfont /CR/	the system font data is converted into an ASCII font matrix file and saved on the disk inserted in drive 1 in the file named "italicfont"

Recovers (whenever possible) a recently deleted file.

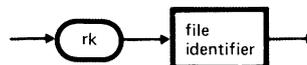


Fig. 13-35 RKILL

Where

SYNTAX ELEMENT	MEANING
file identifier	the name of the file to be recovered. It is not necessary to specify the file password

Characteristics

RKILL will only recover files that have not yet been overwritten. Once partly overwritten, a recently deleted file (either FKILLED from PCOS or KILLED from BASIC) cannot be recovered. Moreover, a file cannot be recovered on a volume that has been alphabetised since the file was deleted.

Examples

IF you enter...	THEN...
fn 1:myfile,20 /CR/	the file "myfile" is created on the diskette inserted in drive 1
vl 1: /CR/	a list of files resident on the diskette in drive 1 is displayed. "myfile" will feature at the end of this list
fk 1:myfile /CR/	the file "myfile" is deleted from the diskette in drive 1. (This can be checked using the VLIST command)
rk 1:myfile /CR/	the file "myfile" is recovered, and it will again feature in a volume list

Remarks

RKILL will not be executed if the volume is write-protected.

Loads the RS-232-C extension package.

When called this utility allows the user to communicate, via the RS-232-C interface, with devices that are compatible with this interface. The transmission environment is set using the SCOMM command. However it is necessary to execute the RS232 command before the SCOMM command.

For more details about the RS-232-C package refer to the "I/O with External Peripherals User Guide".

Sets the BASIC environment for programming.

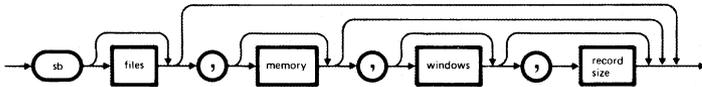


Fig. 13-36 SBASIC

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
files	the maximum number of files that can be opened concurrently under BASIC	0	15

memory	maximum number of bytes available for user memory in BASIC	0	The actual value can be determined using the DCONFIG command. (It can never be greater than 57K)
windows	the number of windows for which memory space is preallocated. Additional windows can always be opened if there is enough space	1	16
record size	maximum record size (in bytes) for random files	1	4096

Default Values

At initialisation from a standard PCOS these parameters have the default values shown in the table below. Alternatively, the user can establish other values using either an initialisation program (for example INIT.BAS) or by loading a user-configured PCOS (obtained using PSAVE).

In each case the value of these parameters can be changed at any moment using the SBASIC command. If this is done the values only become effective when the system goes into BASIC. If the SBASIC command is called from a BASIC program using an EXEC or a CALL statement, the newly set values will not be taken into account in the current program (otherwise the current program could be destroyed). These will become operative in subsequent programs.

A parameter, once changed, maintains its last assigned value until it is again modified or until the system is reinitialised (by switching the system ON, or by a reset). Thus nil parameters in an SBASIC command keep their last specified values in preceeding commands (or default values).

The following table gives the standard default values.

PARAMETER	DEFAULT VALUE	MEANING
files	3	a maximum of 3 files can be opened concurrently
memory	36000 bytes	PCOS has allocated 36000 bytes for BASIC programs, data which is being processed by BASIC programs, open file tables, etc.
windows	1	the first window allocated requires 38 bytes. Any additional windows for which space is allocated will require a further 108 bytes of user memory each
record size	256 bytes	the maximum record size for random files is 256 bytes

Table 13-1 BASIC Environment Default Parameters

Examples

IF you enter...	THEN...
sb /CR/	<p>the current value of the parameters will be displayed (as no parameters were specified in the command).</p> <p>For example</p> <pre>files = 3 memory = 40000 windows = 3 record size = 128</pre>

sb 5,38000,,80 /CR/

the command will establish that 5 files can be opened concurrently, that user memory is 38000 bytes and that the maximum record size for random files is 80 bytes. The omitted parameter (windows) will revert to the value specified in the preceeding SBASIC command since the last initialisation ,or the last SBASIC command followed by a PSAVE (if a user-configured PCOS is being used), or the default value (precedence in the order stated)

BASIC Memory

When the SBASIC command is used, and you enter BASIC, user memory is reduced by the following amount in bytes:

$$- 829 + (578 + R)F$$

where F is the number of files that can be opened, and R is the record size

- 108 bytes for each window (except the first) that space is allocated for

It must be kept in mind that utilities, auxiliary commands and functions assigned to keys reduce user memory and therefore reduce the maximum value of the "memory" parameter.

Sets the transmission environment of an RS-232-C communications port.

The RS-232-C communications driver must be loaded prior to the execution of SCOMM.

For more details refer to the "I/O with External Peripherals User Guide".

Displays the device name table and optionally renames a device.

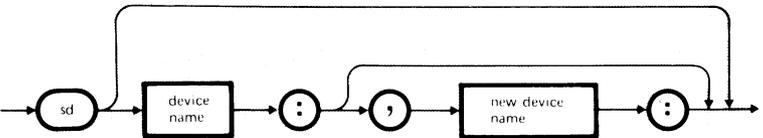


Fig. 13-37 SDEVICE

Where

SYNTAX ELEMENT	MEANING
device name	the current name of any device that exists on the device table consisting of at most 13 printable characters (see illegal characters in Chapter 4). This can be either the name assigned in the last SDEVICE command (in the current working session) or, in the absence of a preceding SDEVICE command, the default device name. Alternatively, if a user-configured PCOS is being used, the PSAVE device name
new device name	the name to be assigned to the device in question. The name must be at most 13 characters long (for illegal characters see Chapter 4)

Characteristics

The colon (:) must follow immediately after the device name.

Default Device Names

prt: - PCOS Printer Driver

cons: - PCOS Console Driver (video and keyboard)

com: - Standard RS-232-C communication port

com1: - First RS-232-C communication port on Twin Board

com2: - Second RS-232-C communication port on Twin Board

Examples

IF you enter...	THEN...
sd /CR/	the device table will be displayed on the VDU. The information displayed gives the device name and its type; W for a write-only type of device, R for a read-only type and R/W for a device that can be read from and written to
sd prt: /CR/	the M20 will display the device name "prt:" and "W" meaning that the printer is a write-only device
sd cons:, kboard.video: /CR/	the name "cons:" of the console is changed to "kboard.video:". The M20 will display the old name and device type first, followed by the message "changed to" and the new name and device type

This command specifies the type of printer which is being used and the printing format.

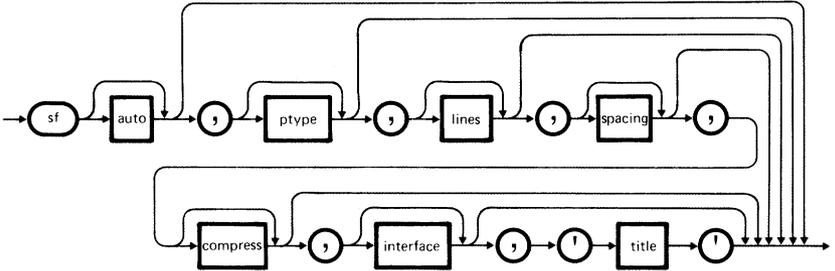


Fig. 13-38 SFORM

Where

SYNTAX ELEMENT	MEANING	DEFAULT VALUE
<p>auto</p>	<p>the status of the nil parameters</p> <p>'off' indicates they are to have their default values</p> <p>'on' indicates they are to have the values set by the most recent SFORM command since switch-on or, failing that, a previous SFORM command that was followed by a PSAVE command</p>	<p>off</p>

p <code>type</code>	<p>the type of printer connected: PR1450, PR1471, PR1481, PR2400, PR2300, PR2835, ET-121, ET-231, PR-320, PR-430, transp.</p> <p>"transp" sets transparent mode. That is, files are printed in free format, irrespective of the type of printer attached (The printer types must be entered exactly as shown)</p>	PR1450
l <code>ines</code>	the number of printed lines per page; 0 implies that no form feed will be issued	60
s <code>pac</code> ing	the number of interline spaces between printed lines. Single spacing = 1, double-spacing = 2, etc.	1
c <code>om</code> press	<p>a two letter parameter where the first letter indicates if the characters are bold (w) or normal (n), and the second letter the type of character. This can be:</p> <ul style="list-style-type: none"> - c (compressed) = 16.6 chars/in. - e (elite) = 12 chars/in. - p (pica) = 10 chars/in. 	ne (normal, elite)

interface	<p>whether the interface through which the M20 controls the printer is serial or parallel:</p> <p>se RS-232-C (serial)</p> <p>pa Centronics type (parallel)</p>	pa
title	<p>the title to be printed at the top of each page. It must not exceed 24 characters</p>	no title

Remarks

To print a title at the top of the first page of a listing, the user must issue a form feed from BASIC; for example, "lprint chr\$(12)".

For the title format, a string of "}" deletes the previous title, without replacement.

Example

IF you enter...	THEN...
sf on,,50 /CR/	<p>the command specifies that the nil parameters are to take the values specified in either the last SFORM command followed by a PSAVE command or the last SFORM command since switch on. The third parameter specifies the page length as 50 lines</p>

Reconfigures the keyboard to that of another country.

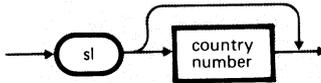


Fig. 13-39 SLANG

Where

SYNTAX ELEMENT	MEANING
country number	a number corresponding to the keyboard configuration of a particular country. See menu below

Characteristics

For the SLANG command to be executed the files "font.all" and "kb.all" must both be on an enabled volume.

The SLANG command can be executed in one of two ways:

- by entering the command without specifying the country number, in which case the following menu is displayed

Available Country Configurations :

Italy	0	Yugoslavia	10
West Germany	1	Norway	11
France	2	Greece	12
Great Britian	3	Switzerland/France ..	13
United States	4	Switzerland/Germany ..	14
Spain	5	German (original)	15
Portugal	6	Datev	16
Sweden/Finland	7	Delta	17
Denmark	8		

Enter Your Selection by Number (or q to quit) -->

The user must then select the country by entering the corresponding number followed by /CR/.

If the user enters a number that does not correspond to a supported language then the prompt re-appears.

Entering "q /CR/" quits the menu without changing the country number

- by entering the country number as a parameter in the command line. In this case an invalid country number is indicated by an error message

ERROR 76 --- error in parameter

The new keyboard configuration remains operative until either it is changed by another SLANG command, or the current working session is terminated.

The new keyboard configuration can be made permanent by means of the PSAVE command.

The keyboard layouts and the codes generated by the keys are shown for all supported national standards in Appendix B.

Examples

IF you enter...	THEN...
sl /CR/	the SLANG command displays the country number menu.
1 /CR/	The West Germany keyboard is simulated

sl 0 /CR/

the Italy keyboard is simulated

Loads and initialises the SPRINT utility to print the image of either the screen or a specified window.

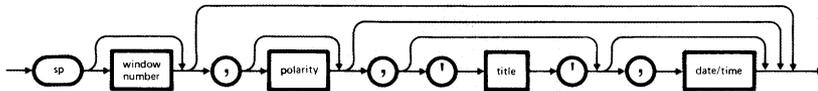


Fig. 13-40 SPRINT

Where

SYNTAX ELEMENT	MEANING	DEFAULT
window number	an integer ≥ 0 and ≤ 16 representing the window to be printed. 0 indicates the entire screen, independent as to whether or not the screen is divided into windows	0 (whole screen)

polarity	n or p, describing the paper image in monochromatic terms. Negative (n) gives black on paper for white on screen, while positive (p) gives black on paper for black on screen. For colour systems, all colours that are not the background colour are considered to be foreground	n (negative)
title	an alphanumeric title string, beginning with a non-numeric character, and no longer than 24 characters: to be printed above the graphic output	no title
date/time	dt or no determining whether (dt) or not (no) the current date and time are to be printed above the graphics output	no (not required)

Example

IF you enter...	THEN...
EXEC "sp 1,,'BAR CHART',dt" /CR/	the contents of window 1 are printed beneath the heading "BAR CHART" and the date and time

Remarks

Graphic elements can only be drawn on the PR 1450, PR 2300 and PR 2400 printers using the SPRINT command.

Windows cannot be opened in PCOS.

Sets system global parameters.

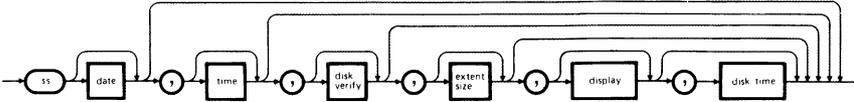


Fig. 13-41 SSYS

Where

SYNTAX ELEMENT	MEANING	RANGE	
		MIN	MAX
Date	any legal date after 1899. Date consists of month, day and year, (for USA ASCII and USA ASCII + BASIC keyboards) distinguished one from the other by a date separator. The order is day, month, year on other keyboard versions		

	month: is any number, representing January as 01... December as 12	01	12
	day: is any legal day according to the month of the year	01	31
	year: is any valid value in the range 1900 to 1999. (Note that only the last two digits need to be specified)	1900	1999
	date separator: is any legal name-printable character excluding digits		
Time	time consists of hour, minute and second, distinguished one from the other by a time separator, where:		
	hour: is any valid value	00	23
	minute: is any valid value	00	59
	second: is any valid value	00	59
	time separator: is any legal legal name-printable character excluding digits		

Disk Verify	whether or not verification is performed after each hard disk or diskette input/output operation. When verification is on, data that is written to the hard disk or diskette is subsequently read and checked. If an error is detected, "ERROR 57 --- disk i/o error" is displayed	0 (off)	1 (on)
Extent size	the number of sectors to be allocated to a file when an output operation requires additional space	1	1087
Display	whether the 64 x 16, or 80 x 25 display mode is in effect	0 (64x16)	1 (80x25)
Disk time	the number of seconds that the diskette drive motor will remain on after accessing that drive	1	30

Default Values

At initialisation from a standard PCOS, parameters attain default values shown in the following table. However, the user can assign these parameters different values at initialisation by using a user-configured PCOS, or by the use of an initialisation program (for example, INIT.BAS) which sets the parameters by calling the SSYS command.

In each case the parameter values can be changed by the user at any subsequent moment by using the SSYS command. Once modified, a parameter maintains the last value assigned to it until it is again modified by another SSYS command, or until the system is reinitialised. In an SSYS command, if a nil parameter is entered it will revert to the current value before the command was entered.

The following table gives the standard default values:

PARAMETER	DEFAULT VALUE	MEANING
Date	01/01/82	January 1 1982, and the date separator is a slash (/)
Time	00:00:01	system time starts from 1 second past midnight and the time separator is a colon (:)
Disk Verify	0 (off)	no verification after any disk or diskette read/write operation
Extent size	8	eight logically contiguous sectors will be allocated to a file when an output operation requires additional space
Display	0	16 rows of 64 characters each
Disk time	2	the diskette drive motor remains on for two seconds after the last access

Table 13-2 Set System Global Parameters Default Values

Example

IF you enter...	THEN...
ss 01/31/83, 00:00:01,0 /CR/	the command sets the date to January 31 1983, the time to one second after midnight and prescribes no verification after a read/write operation

The remaining unspecified parameters (extent size, display and disk time) will either take their default values or the values specified in the last SSYS command that was followed by a PSAVE command (if a user-configured PCOS is being used) or the last SSYS command entered during the current working session

Remark

The Disk time parameter cannot be PSAVEd.

Sorts a volume directory into alphabetic order and frees any unused directory blocks, thus improving access time whenever the directory is scanned

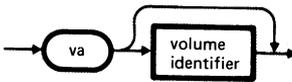


Fig. 13-42 VALPHA

Where

SYNTAX ELEMENT	MEANING
volume identifier	the volume name, or the drive number in which the volume is inserted and, if the volume is not enabled, its password. If the volume identifier is not specified the volume in the last selected drive is assumed

Example

IF you enter...	THEN...
va 1: /CR/	the directory of the diskette inserted in drive 1 will be reordered and all the files will be listed in alphabetical order

Remarks

Any volume write-protect label present must be removed before the VALPHA command can function.

The VALPHA command orders the directory such that files specified in upper case appear before those specified in lower case. On a standard system diskette the PCOS.SAV file will therefore always be placed first on the volume directory. But note that any file that starts with an upper case letter alphabetically higher than "P" will appear before the PCOS.SAV file after using the VALPHA command.

Copies a diskette from drive to drive

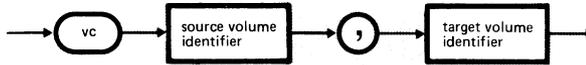


Fig. 13-43 VCOPIY

Where

SYNTAX ELEMENT	MEANING
source volume identifier	the volume name, or the drive number in which the diskette to be copied is inserted. It is not necessary to specify the password
target volume identifier	the volume name or the drive number in which the target volume is inserted. It is not necessary to specify the password. The target volume must not be write-protected

Characteristics

Before copying a diskette it is advisable to write-protect the source diskette to avoid accidentally overwriting it.

You must ensure that the copy process is being carried out in the correct direction - VCOPIY will overwrite your intended source volume with the target volume, if you instruct it to do so.

This command copies the first onto the second volume specified as long as the two volume identifiers do not select the same drive and the source volume is not of different capacity than the target volume. That is, it is only possible to use the VCOPY command when copying a 640 Kbyte diskette onto another 640 Kbyte diskette, a 320 Kbyte diskette to another 320 Kbyte diskette, or a 160 Kbyte diskette to another 160 Kbyte diskette. VCOPY does not work with the hard disk unit.

When the command is called, the screen will display the following message

Warning - VCopy deletes all files. Copy disk? (y/n)

Entering

y /CR/

starts the process.

After VCOPY, the two diskettes will be identical (except for the diskette's creation data which never changes throughout the life of a diskette).

It is not possible to copy a copy-protected diskette.

Examples

IF you enter...	THEN...
vc 1:,0: /CR/	the volume on drive 1 is copied onto the volume on drive 0
vc source/pass:, target: /CR/	the volume named "source" which has the password "pass" is copied to the volume named "target". Subsequently, they will both have the same name "source" and the same password "pass". It is not necessary, however, to specify the password "pass". If the volumes are to have different names the VRENAME command must be used. If the password is not required then VDEPASS must be used first

Deletes the password from a non write-protected volume. In order to do this the password must be known to the user.

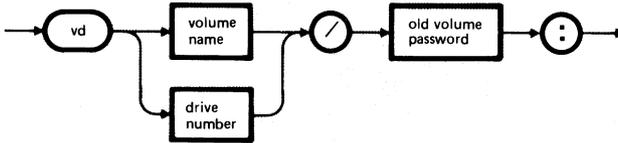


Fig. 13-44 VDEPASS

Where

SYNTAX ELEMENT	MEANING
volume name	the name of the volume whose password is to be deleted
drive number	0, 1 or 10 depending on the drive in which the diskette or hard disk resides
old volume password	the password that is to be deleted. The password must be known if it is to be deleted

Example

IF you enter...	THEN...
vd 0/mypassword: /CR/	the password "mypassword" is removed from the diskette inserted in drive 0

Remarks

VDEPASS will not operate if a volume write-protect label is present. This should be removed and replaced again after the operation.

If the password is not specified correctly, error code 72 "volume not enabled" will be displayed.

Once a volume password has been removed, accessing the volume will be the same as for a volume that never had a password.

Formats an unprotected volume and creates a blank file system on a diskette or hard disk, with the option to name the volume.

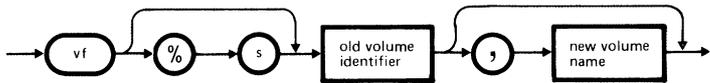


Fig. 13-45 VFORMAT

Where

SYNTAX ELEMENT	MEANING
s	the diskette is to be formatted 160 Kbytes double-density single-sided on a 320 Kbyte drive
old volume identifier	the volume name, or the drive number. It is not necessary to specify the password
new volume name	the new name which is to be assigned to the volume

Characteristics

This command is used to format 640 Kbyte diskettes, 320 Kbyte diskettes, 160 Kbyte diskettes, or the hard disk.

The formatting process puts an empty directory on the diskette or hard disk and places empty sectors on the tracks. Once a used volume has been formatted, any volume name that it had will no longer apply.

If a write-protect label is present on the diskette, it must be removed before VFORMAT can function.

When the command is called, the screen will display the following message

```
Warning VFormat deletes all files. Format disk? (y/n)
```

Entering

```
y /CR/
```

starts formatting and file system initialisation immediately if the volume has no password. If the volume has a password, and "y" is entered, the following message is displayed

```
Diskette appears password protected. Format disk? (y/n)
```

Entering

y /CR/

the diskette is formatted, and the password is deleted. Once begun, the VFORMAT process cannot be aborted.

If the diskette is a new OPE formatted one, then another message is displayed before formatting can start

Diskette is OPE formatted. Format disk anyway? (y/n)

Entering y /CR/

the diskette is formatted.

(OPE formatting is the Olivetti standard for diskettes and is common with other Olivetti machines.)

Examples

IF you enter...	THEN...
vf mydisk: /CR/	the volume called "mydisk" is formatted
vf 1:,newdisk /CR/	the diskette inserted in drive 1 is formatted and assigned the name "newdisk"
vf 10: /CR/	the hard disk (if present) is formatted
vf %s 1: /CR/	one side of the diskette inserted in drive 1 is formatted. That is, the diskette is formatted 160 Kbytes. The drive must, however, be a 320 Kbyte drive

Remarks

All M20 hard disks or diskettes must be formatted before files can be created on them. As for used M20 diskettes (or hard disk), the contents of which are no longer of interest, it is enough to re-initialise them via the VNEW command in order to use them again.

Olivetti-supplied diskettes are pre-formatted to a high precision but are not initialised. They therefore do not require the VFORMAT command, but must be initialised using the VNEW command. In this way, diskettes are guaranteed to be interchangeable between different M20s.

If the %n flag is specified the diskette (or hard disk) is formatted without any prompt being issued.

Lists the directory (or a specified part thereof) of a specified volume.

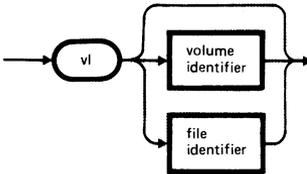


Fig. 13-46 VLIST

Where

SYNTAX ELEMENT	MEANING
volume identifier	the name of the volume to be listed or the drive number. It is not necessary to specify the volume password. If no volume identifier is specified the last selected drive is assumed

file identifier	<p>the name of a file, including any necessary volume identifier.</p> <p>Groups of files can be specified using the wild card facility, or a list of files may be specified. It is not necessary to specify the file password</p>
-----------------	---

Characteristics

In a volume list, the M20 displays the following information:

- the drive in which the volume is inserted
- the name of the volume (if it exists)
- the requested file list, specifying for each file:
 - . its name
 - . its length (in bytes)
 - . blocks used
 - . blocks allocated
 - . the number of extents
 - . whether write-protected or not (shown as WP)
 - . whether a password exists or not (shown as /PW)
- the total number of files, the total number of blocks allocated and used, and the total number of extents used by the files listed
- the total free space (in blocks) available on the volume.

If the number of files is too great to fit them all onto one screen of data then the list is displayed one screen at a time. A message is then displayed to prompt you to strike any key to examine the next screenful.

If the no-interaction (%n) flag is used then the list is scrolled until the last screen of data is displayed.

If the file identifier is not specified, then all the files resident on the volume in question are listed.

For a typical example of a volume list refer to Chapter 9.

Examples

IF you enter...	THEN...
vl /CR/	the M20 displays the directory of the volume inserted in the last drive selected
vl 1:b* /CR/	the M20 will display a volume list of all the files with a name that starts with "b", and are resident on the diskette inserted in drive 1
vl 10: /CR/	the M20 displays a volume list of all files on the hard disk

Copies a diskette using only one drive.

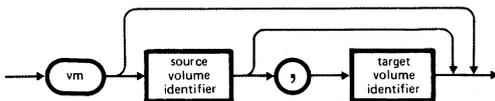


Fig. 13-47 VMOVE

Where

SYNTAX ELEMENT	MEANING
source volume identifier	the volume name or drive number (must be drive 0) of the diskette to be copied, plus any necessary volume password
target volume identifier	the volume name or drive number (must be drive 0) of the diskette to which the source volume is to be copied, plus any necessary volume password

Characteristics

Before copying a diskette it is advisable to write-protect the source diskette to prevent accidentally overwriting it.

As VMOVE can do all that VCOPY does using only one drive, VMOVE is therefore only useful on single drive systems, or for copying diskettes on hard disk systems. It cannot be used to copy the hard disk.

If neither the source nor target volume is password protected then the command name alone is sufficient. If the source volume is password protected then the full source volume identifier - including the password - must be specified in the command line. If the target volume is password protected, then both the source and target volume identifiers must be specified.

When VMOVE is called the M20 displays a message warning the user that VMOVE deletes all files and PCOS, and asks the user to confirm the intention. Following a "y /CR/" (yes) reply, the M20 displays another message asking the user to insert the source diskette in drive 0, and to hit any key.

At this point the M20 will fill all its memory space with data from the source diskette. When the memory is full, a message will be displayed asking the user to insert the target diskette in drive 0 and to hit any key. The M20 will then transfer all data from its memory onto the target diskette.

This process will be repeated (typically) two more times. At the end of the copy operation you can either continue to make more copies, or re-boot PCOS.

Note: After VMOVE, PCOS has to be re-booted because VMOVE uses all memory space as a buffer thus overwriting PCOS.

The target diskette must not have a write-protection label applied.

Any name and/or password assigned to the source volume will be copied to the target volume.

Initialises an unprotected or enabled volume with the option to name the volume.



Fig. 13-48 VNEW

Where

SYNTAX ELEMENT	MEANING
s	a 160 Kbyte double-density single-sided diskette is to be initialised in a 320 Kbyte drive
old volume identifier	the volume name, or the drive number, and, if the volume is not enabled, its password
new volume name	is the new name which is to be assigned to the volume. If omitted, the new volume will have no name

Characteristics

This command can be used to initialise any diskette or the hard disk.

If a volume write-protect label is present on the diskette, it must be removed before VNEW can function.

When this command is called, the screen will display the following message:

```
Warning -vnew deletes all files. Initialize disk? y/n
```

Entering

```
y /CR/
```

starts the file system initialisation immediately. Once started, the VNEW process cannot be aborted.

Once initialised, any old volume name will no longer apply.

Examples

IF you enter...	THEN...
vn mydisk:,newdisk /CR/	the diskette called "mydisk" is initialised and renamed "newdisk"
vn 10: /CR/	the hard disk is initialised
vn %s 0: /CR/	the 160 Kbyte diskette inserted in 320 Kbyte drive 0 is initialised

Remarks

A volume can only be initialised using VNEW if it has previously been formatted. Since VFORMAT performs everything VNEW does, VNEW is normally used to re-initialise volumes which have unwanted files on them.

If the volume is not already formatted, or if there is some error at initialisation, the M20 will display an error message.

If the volume password has been forgotten VNEW cannot be used. Use VFORMAT instead.

After performing a VVERIFY destructive or non-destructive test (%d option) VNEW cannot be used. Use VFORMAT instead.



Assigns a password to an unprotected volume or changes the password of a volume that already has one.

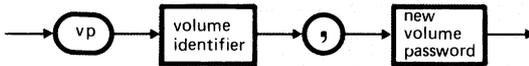


Fig. 13-49 VPASS

Where

SYNTAX ELEMENT	MEANING
volume identifier	a volume name, or the number of the drive in which the volume resides, and, if the volume has one, its password
new volume password	the password to be assigned to the volume. If the volume has a password this will be replaced by the new one

Characteristics

If a volume write-protection label is present on the diskette, it must be removed before VPASS can operate.

When a volume has a password assigned, the user will not be able to use the volume without specifying the password as part of the volume identifier. Once this has been done a first time, it need not be specified again until the volume is removed and another one is referenced on the drive unit on which the volume was inserted. This point is illustrated in the following sequence.

STEP	OPERATION
1	Insert a diskette having the password "mypass" into drive 0 and rename it by entering vr 0/mypass:,newname /CR/
2	Rename it again by entering vr 0:,testdisk /CR/ Note that the password need not be specified as the volume is already enabled
3	Remove "testdisk" from drive 0 and insert another diskette that has a password
4	Attempt to rename it by entering vr 0:,newdisk /CR/ The message ERROR 72 volume not enabled is displayed because the new diskette has not had its password specified

5	Remove "newdisk" and reinsert "testdisk"
6	<p>Attempt to rename "testdisk" by entering</p> <pre style="margin-left: 40px;">vr testdisk:,mydisk /CR/</pre> <p>and ERROR 72 is again displayed because another diskette (see step 4) has been referred to in drive zero. The original diskette must have its password restated to re-enable it</p>

Example

IF you enter...	THEN...
vp myvol:,psswd /CR/	the volume "myvol" is given the password "psswd"

Lists all the file names, or a desired subset of the file names resident on a volume, and displays the number of free blocks on the volume.

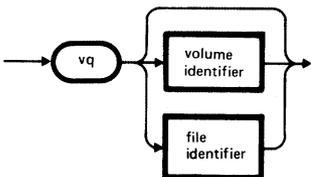


Fig. 13-50 VQUICK

Where

SYNTAX ELEMENT	MEANING
volume identifier	<p>the volume name or the drive number. If omitted, the last selected drive is assumed.</p> <p>It is not necessary to specify the volume password</p>
file identifier	<p>the name of the file plus any necessary volume identifier.</p> <p>The file name can be specified using wild card characters to specify a group of files.</p> <p>If the file identifier is not specified, then all the file names resident on the volume in question are displayed.</p> <p>It is not necessary to specify the file password</p>

Characteristics

In a volume quick list the M20 displays the drive number in which the volume in question is inserted, the volume name (if it exists) and the number of free blocks on the volume. This information is followed by a list of requested file names in four columns (for a 64 x 16 display) or five columns (for an 80 x 25 display).

The no-interaction flag (%n), if used, is ignored.

A typical volume quick list is given in Chapter 9.

Examples

IF you enter...	THEN...
vq /CR/	the M20 will display a volume quick list. The last selected drive is assumed
vq 0:*.cmd /CR/	only the files with the file name extension CMD and resident on the diskette inserted in drive 0 will be displayed in the list
vq 1:?s* /CR/	the only files listed are those resident on the diskette inserted in drive 1 and which have "s" as the second letter of their file name

Names or renames an unprotected or enabled volume.

If the new volume name is not specified, the old name will be deleted.



Fig. 13-51 VRENAME

Where

SYNTAX ELEMENT	MEANING
old volume identifier	the old volume name, or the drive number and, if the volume is not enabled, its password
new volume name	<p>the name the volume is to be given.</p> <p>Leaving this parameter out results in the volume being given no name and it can then only be referenced by the drive number (and its password if it has one)</p>

Examples

IF you enter...	THEN...
vr 0: /CR/	the name is removed from the diskette in drive 0
vr 0:,mydisk /CR/	the name "mydisk" is assigned to the diskette inserted in drive 0
vr mydisk:,d1 /CR/	the name is changed from "mydisk" to "d1"

Remarks

Any volume write-protect label present must be removed before VRENAME can function.

A volume can be named (again, or for the first time) by referencing either the drive number, or by specifying a new name in the second parameter.

VRENAME changes only the volume name. The password is unaffected.

If a new volume name is not specified, the old one will be deleted and in future it will only be possible to refer to the volume by the drive number (followed by its password if it exists).

Checks the hard disk for faulty blocks.

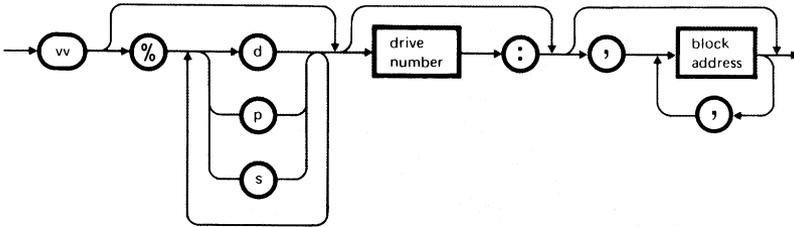


Fig. 13-52 VVERIFY

Where

SYNTAX ELEMENT	MEANING
d	destructive test. The hard disk is checked for faulty blocks. Because the destructive test destroys all the information held on the hard disk, the user is prompted to confirm the intention. Throughout the test the user is kept informed of the test's progress by a series of messages that tell the user which side of which cylinder is currently being tested. On completion of the test, a further message displays the number of bad blocks that there are on the hard disk, but not

	<p>their addresses.</p> <p>Following a destructive test, the hard disk cannot be re-used until it has been re-formatted using the VFORMAT command. Re-formatting also prevents listed bad blocks from being re-allocated</p>
p	<p>displays on the VDU the physical addresses of the faulty blocks. No test is performed. Each block address is displayed as two words, each comprising four hexadecimal digits as follows:</p> <pre> word 1 word 2 XXXX XXXX ~~~~~ ~~~~~ cylinder number _____ side number _____ physical block number _____ </pre>
s	<p>block(s) specified by the block address parameter(s) are defined as being faulty. This is particularly useful if the user suspects some bad block(s) as it allows the block(s) to be added to the bad block list without performing the test. The specified blocks, however, will not be deallocated until the hard disk is reformatted</p>
drive number	<p>the drive number of the hard disk is always 10. If this parameter is omitted then the last selected drive is assumed</p>
block address	<p>the physical block address on the disk. It must be entered as an eight-digit hexadecimal number preceded by the letter "h". The format is as follows:</p> <pre> hXXXXXXXX ~~~~~ cylinder number _____ side number _____ block number _____ </pre>

Characteristics

If VVERIFY is entered without parameters (other than the optional drive number parameter), then a non-destructive test is performed. Faulty blocks are listed on cylinder 0 side 0 block 1, but the information held on the hard disk remains intact. The non-destructive test, however, takes longer than the destructive test and is less exhaustive. Throughout the test, the cylinder and side number currently being tested are displayed on the VDU, and the final message displays the total number of bad blocks found. Following the test, bad blocks will continue to be allocated by the system until the VFORMAT command is used to re-format the hard disk. Before re-formatting, it is the user's responsibility to copy onto diskette the files that need to be saved.

If the contents of the hard disk are no longer required, the %d option may be specified, thereby reducing execution time. Furthermore, if the user already suspects which blocks are faulty, these blocks may be added to the bad block list using the %s option, thereby not performing any test at all.

The following are the possible program flag combinations:

- %dp - performs the destructive test then displays a list of the updated bad block list
- %pd - displays the bad block list then performs the destructive test
- %sp - adds the specified block addresses to the bad block list, then displays the updated list
- %ps - displays the bad block list then adds the specified block addresses to it

The no-interaction flag (%n) can be used to suppress interactive messages and the display of the command title. It does not suppress the display of the bad block list with the %p option, neither will it suppress the display of error messages.

The VVERIFY command can only be used on a formatted drive. Any attempt to do otherwise will result in the message

```
ERROR 111 --- invalid device
```

Examples

IF you enter...	THEN ...
vv 10: /CR/	the hard disk is checked for faulty blocks. The bad block list on side 0, cylinder 0, block 1 is updated
vv %p 10: /CR/	the bad block list held on cylinder 0, side 0, block 1 is displayed
vv %np 10: /CR/	the bad block list is displayed, but display of the command name is suppressed
vv %nd 10: /CR/	the hard disk is checked for faulty blocks. The entire contents of the hard disk are destroyed. Only error messages are displayed
vv %s h01010001 /CR/	cylinder 1, side 1, block 1 is added to the bad block list

Makes an active font from an ASCII file which was first created by the RFONT command, then subsequently modified using the Video File Editor.

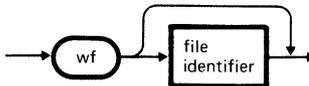


Fig. 13-53 WFONT

Where

SYNTAX ELEMENT	MEANING
file identifier	the file containing the character font matrices. This must be complete with any necessary volume identifier and file password. The file must be structured like the output from an RFONT command. If the volume is not enabled the volume password must also be specified. If the file identifier is omitted the system will return to the font that was active at initialization

Characteristics

At the beginning of the file is a four line header. All but the character count line can be modified without affecting the program.

- line 1: Any text that describes the file. (The content is ignored by the WFONT command. It is there for reference only.)
- line 2: The country number that was active when the file was created. (The content is ignored by the WFONT command. It is for reference only.)
- line 3: The height (in lines) of a valid font matrix - 10 lines. (The content is ignored by the WFONT command.)
- line 4: the character count followed by at least one other word (like "characters"). The count must match the number of font matrices that follow. The minimum is 95 characters and the maximum is 190 characters

Example

```
USA
4
10
95 ch
```

Each matrix is defined as follows:

- line 1: the character number. This is not read by the WFONT command. Changing this number will have no effect, as long as the line is not deleted altogether
- line 2 - 11: Any combination of X's (upper case only) and '-'s are acceptable

Example of a Matrix

```

50
-----
----XXX-
---X---X
-----X
-----X-
----X--
----X---
---XXXXX
-----
-----

```

When the command is entered, assuming the specified font matrix file is on an active volume and that enough memory is available, the character code will be read, converted to binary, and written into memory.

Once execution is completed, the new fonts will replace those known to the system from initialisation. The system will return to the font that was active at initialisation when re-initialisation occurs, or when the WFONT command is invoked with no parameter.

User-defined fonts can be made permanent by means of the PSAVE command.

Examples

IF you enter...	THEN...
wf italicfont /CR/	the font matrix file named "italicfont" is converted to binary and made active
wf /CR/	the original system font is re-activated

Remarks

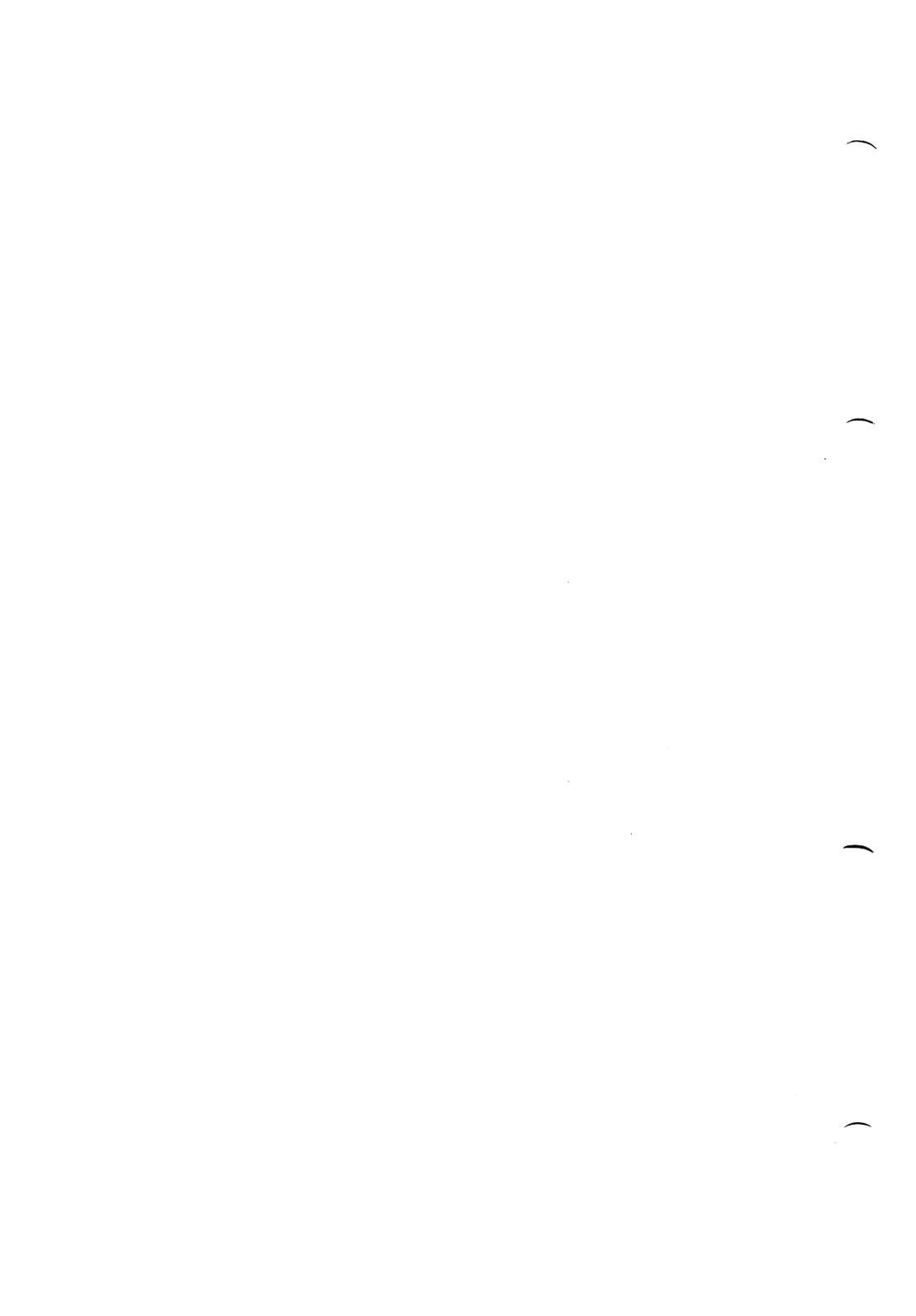
1. In 64 x 16 display mode, the leftmost 3 columns are generally reserved for spacing between characters and should not be used except in special cases where regular spacing between characters is not desired
2. In 80 x 25 display mode, the leftmost two columns are not displayed at all. The third column should be left empty for spacing, unless connected characters are desired
3. For fonts that will work well in either display mode, the user is urged to confine X's to the right-most 5 columns
4. If a numeric character is placed after the 8th character but not by means of the editor's "insert mode", it is ignored and no error will be issued. The same is not true however, for non-numeric characters
5. More font matrices can be added to a font file by adding a new font matrix at the end of the file (with a character number) and by updating the character count (4th header line). If a font matrix is inserted between two existing matrices, this serves only to increment the character numbers of all matrices beyond the one inserted after Write Font is invoked with this file. Any character numbers that appear before each matrix are not interpreted and are only for reference. The added characters are read by position. That is, the character following the one referenced to ASCII code 127, will correspond to code 128, whatever the character number is
6. Within the range of character codes 127 through 222, many character codes do not have keys that correspond to them. If it is desired to have a key or a key combination that corresponds to one of the character codes in this range, use the CKEY command

Memory Usage

The WFONT command allocates user memory each time it is invoked with a valid font file. This memory space is released either by re-initialising PCOS, or by invoking the WFONT command with no parameter. In order to save memory, it is advisable to release space allocated by the WFONT command before activating another user-defined font set.

Example

IF you enter...	THEN...
wf 1:script.font /CR/	script is made active
.	
.	
wf /CR/	memory for script is freed
.	
.	
wf 1:bold.font /CR/	bold print is made active
.	
.	
wf 1:script.font /CR/	script is made active, but data for bold print
	cannot be released from memory without re-init-
	ialising PCOS



A. ASCII CODE

ABOUT THIS APPENDIX

This appendix provides a table of ASCII codes.

CONTENTS

ASCII CODE

A-1