MSX--DOS version 2

The advanced disk operating system

for MSX 2 computers

CONTENTS Page

This manual describes the user interface and commands provided by MSX-DOS 2 version 2.20.

1. INTRODUCTION

=================

MSX-DOS 2, like its predecessor MSX-DOS 1, is provided in a cartridge and in some disk files. The disk files are MSXDOS2.SYS, COMMAND2.COM, help files and transient commands.

MSXDOS2.SYS has the ability to load and execute programs in an enhanced CP/M-compatible environment. COMMAND2.COM is a special program which, when loaded and executed, provides the user with many sophisticated commands and features generally compatible with and in many cases better than those found in MS-DOS and MSX-DOS 1, such as extended memory management.

It also has the ability to load and execute specially written MSX-DOS 1 programs and most standard CP/M programs, and can execute batch files with parameter substitution and other features similar to those found in MS-DOS.

An APPEND facility is provided to increase the ease of use of directories with CP/M programs which were not written to handle them.

Throughout the rest of this manual the term MSX-DOS is used to mean MSX-DOS version 2.xx unless otherwise stated.

2. EDITING COMMAND LINES

========================

When typing in a command line to MSX-DOS, a simple editing facility is available for correction of mistakes or the re-entering and editing of previous commands.

Typing ordinary characters at the keyboard cause the characters to appear on the screen as would be expected. Typing most control characters cause them to be represented by a '^' symbol followed by the control letter. Exceptions are carriage return (RET or CTRL-M), back space (BS or CTRL-H), tab (TAB or CTRL-I), insert (INS or CTRL-R), escape (ESC or CTRL-[), home (HOME or CTRL-K), CTRL-C, CTRL-J, CTRL-N, CTRL-P, CTRL-S, CTRL-U and CTRL-X (SELECT). These perform the following functions:

```
          CTRL-C  - this  acts as a 'break' key. A more drastic and preferred
                    'break' key is CTRL-STOP.
          CTRL-J  - line  feed;  nothing  happens  if this  was given  in the
                    command line.
          CTRL-K  - home cursor (HOME).
          CTRL-N  - this  turns  the  printer  off after  being turned  on by
                    CTRL-P.
```

CTRL-P - this turns the printer on. When on, all characters printed on the screen are also printed on the printer.
CTRL-S - this suspends all character output until another key is pressed.
CTRL-U - this erases the line currently being entered.
CTRL-X - this erases the line currently being entered (SELECT).

The line is entered when the 'enter' key is pressed.

At any point whilst typing in a command line, the backspace key (marked BS or BACK SPACE on most MSX machines) can be used to delete the character immediately to the left of the cursor in the normal way.

The cursor left and right keys will move the cursor left and right along the line. Typing a character at this point will overwrite the character currently underneath the cursor.

Pressing the insert key (marked INS on most MSX machines) will toggle to 'insert mode', and the cursor will change to an underline cursor to indicate this. Instead of the characters being typed overwriting the characters under the cursor, they will instead be inserted before the cursor character, the remaining characters to the end of the line being moved one position to the right.

The delete key (marked DEL on most MSX machines) will delete the character under the cursor and move the remaining characters to the end of the line one position to the left.

The home key (marked 'HOME' on most MSX machines) will move the cursor to the start of the line.

Pressing ESC, CTRL-U or CTRL-X will clear the line to allow a new one to be entered.

The command editor also keeps a list of previous commands entered, up to a limit of 256 characters. Pressing the cursor up key will move up the list and display the previous command line entered, allowing this old command line to be edited and re-entered. Pressing the cursor down key will similarly move to the next old command line that was entered.

If a previous command line is changed, then it will be used as the new command line and added to the bottom of the list. If it was not changed, then it will not be added to the list and the current command line will be the next one which was originally entered. This allows a whole sequence of previous commands to be entered easily.

The list of previous commands is in fact circular and moving off the top or bottom will move to the last or first command in the list respectively. The previous command can be called to be re-entered or edited from this command history list.

The features described here are in fact available to many programs that MSX-DOS can execute. In any program that does 'line inputs', each line can be edited as described above. Previous lines can be recalled for re-entering and editing, although the list of previous lines will of course include previous commands.

3. NOTATION

===========

The syntax of the commands available from MSX-DOS are described in chapter 4 using the following notation:

Words in upper case

These are keywords and must be entered as shown in any mixture of upper or lower case.

Items in lower case

These are parameters which must be supplied to the command at this point in the command line.

Items in square brackets ('[' and ']')

These are optional items. The brackets themselves should not be included in the command line.

Items separated by a vertical bar ('|')

This indicates that only one of the items is required. The vertical bar itself should not be included in the command line.

The following is a list of items which can appear on a command line:

d:

This indicates that a drive name is required (A:, B: etc.).

If d: is shown as optional and is not specified, then the currently logged-on drive, as indicated by the command prompt, is assumed.

path

This indicates that a directory path is required, the syntax of which is similar to MS-DOS. Each directory in the path is separated by a backslash '\'. A backslash at the start of the path indicates that the path starts at the root directory, otherwise the path starts at the current directory as indicated by the CHDIR command. Frequently a filename follows a path, in which case the two must be separated by a backslash.

Two consecutive dots '..' signify the immediate parent directory in the path. A single dot '.' signifies the current directory in the path and therefore usually has no value in a path specification.

On non-English MSX machines, the backslash character '\' may be replaced by some other character. In particular, on Japanese MSX machines the Yen character is used.

If a path is shown as optional and is not specified, then the current directory as indicated by the CHDIR command is assumed.

The syntax of the directory names that make up a path name follows that for filenames given below.

filename

This indicates that the name of a file is required, the syntax of which is similar to MS-DOS and MSX-DOS 1. An ambiguous filename is one that contains '*' and '?' characters and may match more that one file on disk, whilst one that does not contain these is an unambiguous filename.

A filename has the following syntax:-

```
mainname[.suf]
```

where mainname is a sequence of up to 8 characters and suf is a sequence of up to 3 characters inclusive. Any characters beyond these fields are ignored. A '*' in the main name or suffix is equivalent to filling from that character position to the end of the field with '?'. If the suffix is given then it must be separated from the main part of the filename by a single dot '.'.

The following characters cannot be used in filenames:-
Control codes and SPACE (in the range 0 to 20h, and 7FH and FFH)
: ; . , = + \ < > ( ) | / " [ ]

All characters are converted to upper case where appropriate and therefore lower and upper case characters have the same meaning. Note that extended two-character Japanese characters(SHIFT-JIS code) are allowed.

If a filename is shown as optional and is not specified, then a filename of *.* is assumed.

filespec

This is used to identify a file or several files in the same directory on a disk. It's syntax is:-

```
[d:][path][filename]
```

where at least one of the three optional items must be given. Where this is used to specify existing files, /H may be given to allow hidden files to be found.

Generally, d: if not given defaults to the currently logged on drive, path if not given defaults to the current directory of that drive and filename if not given defaults to a filename of *.*.

compound-filespec

This is used in many commands to specify the files or directories to which the command is applied. It's syntax is:-

```
filespec [+ filespec [+ filespec ...]]
```

Thus several filespecs (see above) can be given, separated by '+' symbols, with spaces etc. allowed either side of the +. The effect of this in commands is exactly the same as if all the matched files could have been matched by a single filespec.

Where a compound-filespec is used to specify existing files, /H may be given after each filespec (see above), in which case it will take effect only for the files matched by that single filespec. If a /H is given before the compound-filespec, then it will apply over all the filespecs.

volname

This indicates that a volume name is required. A volume name is a sequence of up to 11 characters, which can include the characters not valid for filenames with the exception of control codes and '/', although leading spaces will be deleted.

device

This indicates one of the five standard MSX-DOS devices is required. These and their meaning are:

```
CON  - screen/keyboard I/O
NUL  - 'null' device, does nothing
AUX  - auxiliary I/O (eg. RS232 serial)
LST  - printer output
PRN  - printer output
```

Unlike on some other systems, a colon is not required after the device name.

Device names can generally be used wherever filenames can be used. For example, the command COPY MYFILE PRN will read the file MYFILE and write it to the printer.

When using the CON device as an input filename, lines can be typed in and edited in the same way as command lines (see chapter 2 on Editing Command Lines). To end the operation, control-Z (^Z) must be typed at the start of a line. For example, a small text file called MYFILE can be created with the command COPY CON MYFILE:

```
A>COPY CON MYFILE
All work and no play makes Jack a dull boy.
Can you hear me?
^Z
A>
```

Lines of text can then be typed in, and they will be written to the file MYFILE. The command will then complete when a line containing a single control-Z is entered.

If the NUL device is written to by the command COPY CON NUL, then the characters written are simply ignored. If read from, then an end-of-file condition is returned straight away (which is equivalent to typing the control-Z in the example above).

For most commands, it is not sensible to specify a device (the CON device cannot be deleted using the ERASE command, for example). The commands that devices are likely to be used with are those that read and write data from and to files, such as CONCAT, COPY and TYPE.

number

This indicates that a number is required. This may be in the range 0 to 255 or 0 to 65535 depending on the command.

4. COMMANDS

===========

This chapter describes in detail all the commands available from the MSX-DOS CLI. Each command is described using the notation described in chapter 3. Where two or more parameters are described using this notation, they must be separated by separators. Separators consist of zero or more leading spaces, a separator character, and zero or more trailing spaces. Valid separator characters are:

```
space tab ; , =
```

Option letters introduced by '/' characters are an exception to this and need not be preceded by a separator.

A transient MSX-DOS or CP/M-80 program can be loaded and executed by typing the main name of the filename plus an optional extension of .COM. Batch files can similarly be executed except that the extension is .BAT. Where COM and BAT files exist in the same directory and with the same name, the COM file is found and executed in preference to the BAT file. The exact location on disk of the command can be specified by including its drive and/or path with its name.

When looking for a COM or BAT file, the specified directory of the specified drive is searched. If not found and a drive or path was given with the command, then an 'unrecognized command' error results.

If just the filename and optional extension were given, the current directory is searched first. If not found, then a list of directories is searched. This list can be specified and changed using the PATH command. If still not found then an 'unrecognized command' error again results.

No CP/M program will be able to specify directories or path names since these do not exist in CP/M, only the current directory of the appropriate drive being accessible from these programs. An APPEND environment item is available which increases the usability of these programs by allowing an alternative directory to be searched by the program as well as the current one (see chapter 7 on Environment Items).

Many commands and programs perform input or output using the 'standard input' and 'standard output'. The standard input normally refers to the keyboard, and the standard output normally refers to the screen. These can be changed, however, to refer to other devices or to disk files for the duration of the command by including the redirection symbols <, > and >> on the command line, followed by a device or file name. The standard output of one command can also be sent to the standard input of the next command by including the piping symbol | on the command line between the two commands. See chapter 5 on Redirection and Piping for more details of these facilities.

When a transient command is executed, that command may overwrite some of the memory that COMMAND2.COM was using. Thus when the command terminates, COMMAND2.COM may need to re-load itself from disk into memory from the COMMAND2.COM file that it was originally loaded from. This file is located by looking at the SHELL environment item (see chapter 7 on Environment Items), or the root directory of the boot drive if it is not found there. If it is still not found, then a prompt is issued. For example, if MSX-DOS was booted from drive A:, then the prompt will be:

```
          Insert COMMAND2.COM disk in drive A:
          Press any key to continue...
```

After inserting into drive A: a disk containing COMMAND2.COM in the root directory and pressing a key, COMMAND2.COM will be re-loaded and the system will continue as normal.

Although not a command as such, the currently logged on drive can be changed by giving the command:

```
     d:
```

which causes the drive d: to become the current drive. This should be shown by the prompt letter.

In the command examples that follow, underlined text is an example response to a command, and the other text consists of the example command given by the user. In most examples a single space is shown as the parameter separator, although other separator characters can be used as specified above.

ASSIGN

Format

```
          ASSIGN [d: [d:]]
```

Purpose

Sets up the logical to physical translation of drives.

Use

If no drives are given, then all current drive assignments are cancelled.

If only one drive is given, then the physical drive to which this refers is printed.

If both drives are given, then the subsequent access to the first drive (logical drive) will be done to the second drive (physical drive) by MSX-DOS.

Examples

```
ASSIGN
```

Un-assigns all previous drive assignments.

```
ASSIGN A: B:
```

Assigns drive A: to drive B:, so that all accesses that previously would go to drive A: go instead to drive B:

```
ASSIGN A:
A:=B:
```

Displays the drive to which A: is currently assigned, which in this case is B:.

ATDIR

Format

```
ATDIR +|-H  [/H] [/P] compound-filespec
```

Purpose

Changes the attributes of directories to make them hidden/not hidden.

Use

The compound-filespec specifies the directories whose attributes are to be changed.

If +H is given, then the selected directories are marked as hidden, and will not be affected by other directory commands or shown by a DIR command unless a /H option is given with those commands. The -H option marks the selected directories as not hidden, and will not have any effect unless the /H option is given.

Unlike files, directories cannot be made read only.

When an error occurs, the erroneous directory name is printed followed by an error message, and the command will continue with the next directory. If many errors occur, then the /P option can be used to cause the output to pause at the end of the screen.

The DIR /H command can be used to indicate the current attributes of directories.

Examples

```
ATDIR +H DIR1
```

Marks the directory called DIR1 as hidden.

```
ATDIR -H DIR1/H
```

Marks the hidden directory DIR1 as not hidden.

```
ATDIR +H DIR?
```

Marks all directories matching DIR? as hidden (for example DIR1, DIR2 and DIR3).

```
ATDIR +H \DIR1+\DIR2
```

Marks the DIR1 directory and the DIR2 directory as hidden.

ATTRIB

Format

```
ATTRIB +|-R|H [/H] [/P] compound-filespec
```

Purpose

Changes the attributes of files to make them hidden/not hidden and read only/not read only.

Use

The compound-filespec specifies the files whose attributes are to be changed, and /H allows hidden files to also have their attributes changed.

If +H is given, then the selected files are marked as hidden, and will not be affected by most commands or be shown by the DIR command unless a /H option is given with those commands. -H marks the selected files as not hidden, and will not have any effect unless the files are hidden.

If +R is given, then the selected files are marked as read only. -R marks the selected files as not read only (read/write). Read only files cannot be written to or changed.

When an error occurs, the erroneous filename is printed followed by an error message, and the command will continue with the next file. If many errors occur, then the /P option can be used to cause the output to pause at the end of the screen.

The DIR command can be used to indicate the current attributes of files.

Examples

```
     ATTRIB +R FILE1
```

The file FILE1 is marked as read only, and will not subsequently be modifiable or deletable.

```
     ATTRIB +HB:\DIR1\*.COM
```

Marks all *.COM files in the directory B:DIR1 as hidden, and will not be displayed by the DIR command.

```
     ATTRIB -R -H \DIR1/H/P
```

All the files in DIR1 are marked as not read only and not hidden. The output, if any, will pause at the bottom of the screen.

```
ATTRIB +R \DIR1 + \DIR2 + FILE1
```

All files in the directories DIR1 and DIR2 and the file FILE1 are marked as read only.

BASIC

Format

```
BASIC [program]
```

Purpose

Transfers control to MSX disk BASIC.

Use

[Program] is the name of a BASIC program on disk.

Control is passed to the built-in MSX BASIC, which will load and execute the BASIC program if specified. If a RAM disk has been set up, then it can still be used from BASIC.

The BASIC command CALL SYSTEM("command") can be used to return to MSX-DOS, and the optional command, which can be any command executable on MSX-DOS, will be executed. If the command is not given, then a batch file called REBOOT.BAT will be searched for and executed if found (see chapter 6 on Batch Files).

Examples

```
BASIC
```

MSX disk BASIC is entered.

```
BASIC MYPROG.BAS
```

MSX disk BASIC is entered, and the BASIC program MYPROG.BAS loaded and RUN.

BUFFERS

Format

```
BUFFERS [number]
```

Purpose

Displays or changes the number of disk buffers in the system.

Use

If the number is not given, then the number of disk buffers currently in the system will be displayed, otherwise the number of buffers will be changed to the specified number, the now unused memory being freed for other purposes if the new number is less than the previous. If there is not enough memory for the specified number of buffers, then as many as possible are created and no error is given.

Increasing the number of disk buffers may speed up some applications, particularly those that perform random accesses to files. Setting the number above 10 is unlikely to improve performance much, and unnecessarily uses up memory.

The memory area used for disk buffers is also used for environment items and for opening files. Thus keeping buffers set to the maximum possible may prevent some commands from working, particularly SET, COPY and CONCAT. If any of these commands give a 'not enough memory' error then it may help to reduce the number of buffers. Reducing them below about three however will impair performance considerably.

The default number of buffers in the system is 5, which will be adequate for most purposes.

Examples

BUFFERS

```
     BUFFERS=5
```

The current number of disk buffers is printed, which in this case is 5.

```
BUFFERS 10

(or BUFFERS=10)
```

The number of buffers is increased to as many as possible up to a limit of 10.

```
BUFFERS = 5

(or BUFFERS 5)
```

The number of buffers is reduced again to 5.

CD

See CHDIR

CHDIR

Format

```
     CHDIR [d:] [path]
          or
     CD [d:] [path]
```

Purpose

Displays or changes the current directory.

Use

If no path is specified, then the current directory path for the current or specified drive is printed. This is the directory path from the root directory to the current directory.

If a path is specified, then the current directory for the current or specified drive is changed to the directory specified by the path.

Each drive has it's own current directory. This remains at the directory specified by the last CHDIR command for that drive (or at the root directory initially) until another CHDIR command is given or it cannot be found on the disk when it is accessed (because the disk has been changed, for example). It is then returned to the root directory.

The CD command is an abbreviated form of the CHDIR command provided for convenience and MS-DOS compatibility.

Note that the command prompt can be changed to display the current directory with the command SET PROMPT ON (see chapter 7 on Environment Items).

Examples

CHDIR \DIR1

The current directory of the current drive is changed to DIR1.

CHDIR A:DIR2

The current directory of drive A: is changed to DIR2.

CD

E:\DIR1

The current directory of the current drive is displayed, which in this case is DIR1.

CHDIR A:

A:\DIR2

The current directory for drive A: is displayed, which is also DIR2.

CHKDSK

Format

CHKDSK [d:] [/F]

Purpose

Checks the integrity of the files on the disk.

Use

The integrity of the data structures on disk in the specified or current drive is checked and lost disk space is checked for. When errors are found on the disk, corrective action is taken. If lost clusters are found, a prompt is issued allowing the lost disk space to be either converted into usable disk space or into files. If the latter is chosen, then files of the form FILE0000.CHK, FILE0001.CHK etc. will be created.

If the /F option is not given, then CHKDSK will not actually write any corrections it makes to disk, but will behave as though it has. This allows CHKDSK to be executed to see what would be done to the disk if /F was given.

Disk space can become lost (ie. lost clusters created) when some programs are aborted. This applies particularly to CP/M programs. It is recommended that CHKDSK is used occasionally on all disks.

Note that this is a transient command, and must therefore be loaded off disk.

Examples

CHKDSK

The disk in the current drive is checked. A 'status report' will be printed. Any errors found will not be written to disk.

CHKDSK B:

The disk in drive B: is checked. Any errors found will not be written to disk.

```
CHKDSK/F

    20 lost clusters found in 1 chain
    Convert lost chains to files (Y/N) ?
```

The disk in the current drive was checked, and some lost disk space found. Since /F was given, the corrections will be written to disk and the lost space recovered.

CLS

Format

```
          CLS
```

Purpose

Clears the screen.

Use

Simply clears the screen and homes the cursor.

Examples

```
CLS
```

The screen is cleared, and another command can be typed.

COMMAND2

Format

```
     COMMAND2 [command]
```

Purpose

Invokes the command interpreter.

Use

command is any command that can normally be typed at the prompt (such as the commands in this manual).

COMMAND2 is simply the name of the command interpreter on disk, and can be executed as a normal transient program. In normal use it gets loaded and executed once by MSXDOS2.SYS at the boot time, and this provides the ability to perform all the commands in this manual.

Advanced users may, however, wish to invoke another command interpreter for a variety of reasons. The second COMMAND2.COM may, for example, be a later version and provide more facilities. If a transient program has the ability to load and execute programs, as some sophisticated programs do, then they can load the COMMAND2.COM program and any MSX-DOS command can then be given. When COMMAND2.COM exits by EXIT command, the original program will be returned to.

If no command is given as a parameter, then the second COMMAND2.COM will simply issue the normal prompt (without executing AUTOEXEC.BAT or REBOOT.BAT) and wait for commands in the normal way. It will terminate and exit back to the original command interpreter or program when the EXIT command is given (see the EXIT command). If an error code is given to this EXIT command, then the original command interpreter or program will receive it and, in the case of COMMAND2.COM and MSXDOS2.SYS, print an appropriate error message (see chapter 8 on Errors).

If a command is given as a parameter to COMMAND2.COM however, then it will be executed as though it had been typed in the normal way. The command may be an internal command or an external COM or BAT command. After executing the command, COMMAND2.COM will immediately exit back to the original command interpreter or program.

In this way, invoking a second COMMAND2.COM from the normal command interpreter with a batch file name as a command can be used to 'nest' batch files (see chapter 6 on Batch Files), instead of 'chaining'.

When COMMAND2.COM is executed, it saves the whole environment, and then restores it again when it exits. It only sets up the default environment items however if they are not already defined. Thus the second COMMAND2.COM inherits the environment of the first. Any changes made whilst the second COMMAND2.COM is executing will only last as long it does, and will be lost when it exits.

Each incarnation of COMMAND2.COM uses up some memory which is freed again when it exits. This depends partly on the number of environment items, and is typically about 1.5K.

When COMMAND2.COM executes a transient program, it may have to re-load itself off disk since the program is allowed to use the memory occupied by COMMAND2.COM. In this case, it uses the SHELL environment item to locate the file that it must use to load itself (see chapter 7 on Environment Items). When first loaded from the COMMAND2.COM file on disk, SHELL is set to refer to that file.

Examples

```
COMMAND2

A>
```

Another copy of COMMAND2 is loaded, and prints it's normal prompt. EXIT will exit back to the original prompt.

```
COMMAND2 FILE.BAT
```

Normally in a batch file. The file FILE.BAT is executed, and when it ends the current batch file will be resumed with the command after this one.

CONCAT

Format

```
     CONCAT [/H] [/P] [/B] [/V] compound-filespec filespec
```

Purpose

Concatenates (joins together) files.

Use

The compound-filespec specifies the files that are to be joined together, and /H allows hidden files to be joined.

The second parameter is a filespec that must be unambiguous and is created before the source files are read. Each source file is then read, joined onto the end of the previous one and written out to the destination.

As each source file is read, it's filename is printed. If for some reason the file cannot be read (eg. it is the file that has been created as the destination) then the filename is followed by an error message and the CONCAT operation continues with the next source file. If many files are being concatenated, then /P will cause the output to pause at the end of the screen until a key is pressed.

Normally, the concatenation is performed on ASCII files. Source files are read up to the first end-of-file character (CTRL-Z) and a single end-of-file character is appended to the destination after all data has been written out. If, however, /B (binary mode) is given, then no interpretation is given to the data read and no additional data is added.

It is also possible to give the /B to the destination or to any of the filespecs in the compound-filespec, and it will then refer only to those files. /A may also be given to reverse the effect of /B.

The /V option can be given to turn write verification on for the duration of the CONCAT command (see the VERIFY command). This will ensure that data is written correctly to disks if the device driver being used has the feature, but will slow the operation down for the verification.

If CONCAT gives a 'Not enough memory' error then probably reducing the number of buffers (see the BUFFERS command) or removing some environment items (see chapter 7 on Environment Items) will free up sufficient memory.

Examples

```
CONCAT *.DOC ALL.PRN
```

A new file called ALL.PRN is created, and all files matching *.DOC (for example FILE1.DOC, FILE2.DOC and FILE3.DOC) are joined together and written to the new file in the order that they are found on disk. Any existing file called ALL.PRN will be overwritten.

```
CONCAT /H /P *.DOC ALL.DOC
     FILE1.DOC
     FILE2.DOC
     FILE3.DOC
     ALL.DOC -- Destination file cannot be concatenated
```

A new file called ALL.DOC is created, and all files matching *.DOC are joined together and written to the new file in the order that they are found on disk. Since the destination file ALL.DOC also matched the source filename *.DOC, the message is printed and it is not included in the concatenation. Since /H was given, hidden files are also concatenated and, since /P was given, the key input is waited after each screenful output if the number of the lines of the file list is larger than that of the screen.

```
CONCAT /B  FILE2.DOC + FILE3.DOC + FILE1.DOC  ALL.DOC
```

A new file ALL.DOC is created, and the files FILE2.DOC, FILE3.DOC and FILE1.DOC and joined together in that order and written to the new file. They are joined together in binary mode.

COPY

Format

```
     COPY [/A] [/H] [/T] [/V] [/P] [/B] source dest
```

Purpose

Copies data from files or devices to other files or devices.

Use

The definition of the source is:
compound-filespec | device

The compound-filespec specifies the files that are to be copied. It may be the device specification. If /H is given then hidden files may be copied.

The definition of the dest is:
[d:] [path] [filename] | device

Where d: and path default to the current drive and directory respectively. If any part of the filename is ambiguous then the appropriate character from the source filename is substituted, thus allowing the files to be renamed in the process. If the filename is not given, then the entire source filename is used. If the dest is an unambiguous directory, then the files are copied into that directory with a filename of *.*.

COPY will read as many source files as possible into memory before writing any out. When it can read no more into memory (eg. when it has used all available memory) it will write out each file in the order that it read them. When it creates each destination file, it prints the source filename. Then if it is unable to create the destination file, an error message is printed and the copy operation continues with the next file. /P can be given to make the output pause at the end of the screen.

Many reasons exist for COPY to be unable to create the destination, such as a read-only file already existing with the same name. Sometimes COPY will refuse to create the destination because the user may have made a mistake. For example, a file cannot be copied onto itself, or several files cannot be copied onto one file. A 'Cannot create destination' error may be given if the destination of one file would delete a previous source file or a file already being used for something else (eg. the currently executing batch file). A 'Cannot overwrite previous destination file' error results if an attempt is made to copy many files to one file. This usually means that the intended destination was a directory, but that the name has been misspelled.

If /A is specified, then an ASCII copy is performed. This means that source files will only be read as far as the first end-of-file (EOF) character (CTRL-Z) and then each destination will have a single end-of-file character appended to it.

It is also possible to give a /A to the destination or to any of the filespecs in the compound-filespec separately, in which case it applies only to that source or dest specification.

The /B option can be given to copy in binary mode, that is, the file being read will be copied as it is and no data will be added.

The /V option can be given to turn write verification on for the duration of the COPY command (see the VERIFY command). This will ensure that data is written correctly to disks if the device driver being used has the feature, but will slow the operation down.

Normally, the destination files are given the same date and time as the source files. However, the /T option can be given to cause the destination files to have the current date and time. The destination files will not be hidden or read-only, regardless of the attributes of the source files. The ATTRIB command can be used to change these.

If COPY gives a 'Not enough memory' error then probably reducing the number of buffers (see the BUFFERS command) or removing some environment items (see chapter 7 on Environment Items) will free up sufficient work area.

Note that the COPY command is simpler than that in MS-DOS and MSX-DOS 1 because it cannot concatenate (join together) files. To do this, a CONCAT command is available (see the CONCAT command).

Examples

```
COPY FILE1 B:
```

The file FILE1 is copied from the current directory of the current drive to the current directory of drive B:.

```
COPY /H  MSXDOS2.SYS + COMMAND2.COM  B:
```

The two hidden files MSXDOS2.SYS and COMMAND2.COM are copied to drive B:, thus making it a booting disk.

```
COPY A:\DIR1 B:\DIR1 /V
```

All files in the directory DIR1 from the root of drive A: are copied to a similar directory on drive B: with verify on to ensure that the files were written correctly.

```
COPY B:
```

All files in the current directory of drive B: are copied to the current directory of the current drive.

```
COPY /A AUX CON
```

Characters are read from the device AUX (which may be used for RS232 serial for example) to the device CON, which is the screen. The copy is done as far as the first end-of-file character. If /A was not given, then there may have been no way of stopping the COPY operation without pressing the CTRL-STOP key.

```
COPY A:*.DOC B:/T
```

All files matching *.DOC (for example FILE1.DOC, FILE2.DOC and FILE3.DOC) are copied to the current directory of drive B: and are given the current date and time instead of the dates and times of the *.DOC files.

```
 COPY *.BAT
 AUTOEXEC.BAT -- File cannot be copied onto itself
 REBOOT.BAT -- File cannot be copied onto itself
```

This command told COPY to copy all files matching *.BAT (in this case AUTOEXEC.BAT and REBOOT.BAT) from the current directory of the current drive to the same place, and COPY printed the messages to warn of this. No data in this case was actually copied.

```
COPY *.BAT DIR2
AUTOEXEC.BAT
REBOOT.BAT -- Cannot overwrite previous destination file
```

This command told COPY to copy all files matching *.BAT (in this case AUTOEXEC.BAT and REBOOT.BAT) to a directory called DIR2. DIR2, however, did not exist so AUTOEXEC.BAT was copied to a file called DIR2, and then an attempt was made to copy REBOOT.BAT also to a file called DIR2. The message as printed as a warning that a
mistake was probably made in this case DIR2 not existing). REBOOT.BAT was not actually copied anywhere.

DATE

Format

```
DATE [date]
```

Purpose

Displays or sets the current date.

Use

If the date is given after the command, then the date is set to this value (for the format see below). If the date is not given after the command, then the current day and date is printed and the new date is prompted for and input. If no input is given (ie. if the 'enter' key alone is pressed) then the current date is not altered. Otherwise the input is assumed to be a new date, and is interpreted as described below. If the date is invalid then an error message is displayed and the new date again prompted for and input.

The date is expected to consist of up to three numbers, separated by one of the following characters:

```
space tab , - . / :
```

with spaces allowed either side of the character. Any missing numbers will default to the current setting. The year may either be a full century and year, or may be just the year in which case the century defaults to 19 if the year is greater than 80 or 20 otherwise. The date and the year specifications may be substituted by '-' to be omitted.

The format in which the date is printed and input is flexible and can be changed. An environment item called DATE is set up by default to a format that is appropriate for the country of origin of the MSX machine (see chapter 7 on Environment Items). For example, on Japanese machines the default setting is YY-MM-DD. The command SET DATE DD-MM-YY will change the date format to the European format. The format also affects the dates printed by the DIR command.

If the DATE environment item is defined, then it will be printed by the DATE command to indicate the format in which the date is required to be input.

Examples

```
DATE 86-6-18
```

The current date is set to the 18th June 1986.

```
DATE
     Current date is Wed 1986-06-18
     Enter new date (yy-mm-dd): - -19
```

No parameter was given, so the current date of 18th June 1986 was printed and a new date prompted for. In the reply to the prompt, the date was updated to the next day by only specifying the 19th. Since the year and month were not given, they remained the same.

```
SET DATE = DD/MM/YY
```

The date format has been changed to the European format.

```
DATE
Current date is Thu 19-06-1986
Enter new date (DD/MM/YY):
```

No parameter was given, so the current date of 19th June 1986 was printed in the European format, and the prompt printed. The reply is expected in the European format.

Formats are:

ISO	YY/MM/DD
American	MM/DD/YY
European	DD/MM/YY

DEL

See ERASE.

DIR

Format

```
DIR [/H] [/W] [/P] [compound-filespec]
```

Purpose

Displays the names of files on disk.

Use

The compound-filespec specifies which files are to be listed. If the /H option is given, then hidden files will also be listed.

In the DIR command, unlike all other commands, it is permissible to not give the main filename or the filename extension, and both will default to '*'. Thus a filename of 'FRED' is equivalent to 'FRED.*' and a filename of '.COM' is equivalent to '*.COM'. Note that if the '.' at the end of a main filename is given, then the extension is also assumed to have been given, so that the filename 'FRED.' is not equivalent to 'FRED.*', unlike the example above.

There are two formats of the listing. If the /W option is given, then a 'wide' listing is printed, with several filenames output per line. Sub-directory names, file attributes, and the date and time each file was created are not displayed.

If the /W option is not given, then the filenames are printed with one filename per line, together with the attributes, the file size and the date and time at which the file was last modified. The attributes are printed as an 'r' if the file is read-only and an 'h' if the file is hidden (and /H is given). If the time of a file is zero (ie. the file does not have an associated time) then the time field will not be printed. If the date of a file is zero, then neither the date nor the time fields will be printed. The formats in which the dates and times are printed can be changed (see the DATE and TIME commands).

The non-/W display is designed to fit within a 40 column screen, but if fewer columns are available then some fields of the listing will not be shown so that the display will always fit on one line. The number of files per line that are printed when /W is specified is also adjusted according to the screen width. If the width of the display is less than 13 characters however, then in both cases the filenames will wrap to the next line.

At the top of the list of files, the volume name of the disk and the name of the directory being listed is displayed. At the bottom, the number of files listed, the total number of bytes in the files and the amount of remaining disk space is printed.

When the directory of a sub-directory is printed, the first two items listed will always be two special sub-directories called '.' and '..'. These are automatically created when a new directory is created, and it is these that allow '.' and '..' to be given in path names to signify the current and parent directories respectively (see chapter 3 on Notation for a description of paths).

When printing a number of bytes, the number is truncated and printed as the number of kilobytes if 1K or greater.

If the /P option is given, then the output will pause at the bottom of the screen until a key is pressed.

Examples

DIR

All filenames and directory names in the current directory of the current drive will be printed. This might be as follows:

```
 Volume in drive A: is MSX-DOS 2
 Directory of A:\

MSXDOS2  SYS r    4096 86-06-19  2:45p
COMMAND2 COM r   10496 86-06-19  2:46p
UTILS           <dir>  86-06-19  2:50p
HELP            <dir>  86-06-19  2:50p
 14K in 2 files   222K free
```

The disk thus contains the two MSX-DOS system files MSXDOS2.SYS and COMMAND2.COM, which are read only, and two directories called UTILS and HELP.

DIR B:\HELP/W

A 'wide' directory format has been requested of the HELP directory on drive B:. This might be as follows:

```
 Volume in drive B: is MSX-DOS 2
 Directory of B:\HELP

BUFFERS .HLP    ATTRIB  .HLP    ASSIGN  .HLP
ATDIR   .HLP    CHDIR   .HLP    CD      .HLP
SYNTAX  .HLP    ENV     .HLP    BATCH   .HLP
EDITING .HLP
 25K in 10 files   222K free
```

DIR UTILS + HELP/P

This will list all the files in the UTILS directory and all the files in the HELP directory, and will pause at the end of every screen full.

DIR .COM

No main filename was given, and so defaults to *. Thus this command is equivalent to the command DIR *.COM.

DIR COMMAND2

No extension was given, so this defaults to .*. Thus this command is equivalent to the command DIR COMMAND2.*.

DISKCOPY

Format

```
DISKCOPY [d: [d:]] [/X]
```

Purpose

Copies one disk to another.

Use

The first drive is the source drive and the second the destination, which defaults to the current drive. If no drives are given, then DISKCOPY will prompt for both the source and the destination.

Before DISKCOPY is used, the destination disk must be formatted with the same format as the source disk, and an error will be given if this is not the case.

If /X is given, then various messages printed during the disk copy operation will be suppressed.

Note that this is a transient command, and must therefore be loaded from disk.

Examples

```
     DISKCOPY A: B:

Insert source disk in drive A:

Insert target disk in drive B:

Press any key to continue...
```

The command was given to copy the disk in drive A: to the disk in drive B:, thus destroying all existing data on the disk in drive B: The prompt is printed first.

```
DISKCOPY B:
```

The disk in drive B: is copied to the disk in the current drive.

```
DISKCOPY
Enter source drive:
Enter target drive:
```

The DISKCOPY command was given with no parameters, so the source and destination disks were prompted for. The reply to the prompts consists of just a single drive letter.

ECHO

Format

```
ECHO [text]
```

Purpose

Prints text.

Use

The text is simply displayed on the screen. If no text is given, then just a blank line is output.

This command should not be confused with the 'echo' state in batch files, which is controlled by an environment item called ECHO (see chapter 7 on Environment Items).

Examples

```
ECHO AUTOEXEC batch file executed
AUTOEXEC batch file executed
```

The specified text ('AUTOEXEC batch file executed') was printed on the screen.

```
ECHO
```

No parameters were given, so just a blank line was printed.

ERA

See ERASE.

ERASE

Format

```
     ERASE [/H] [/P] compound-filespec
              or
     DEL [/H] [/P] compound-filespec
              or
     ERA [/H] [/P] compound-filespec
```

Purpose

Deletes one or more files.

Use

The compound-filespec specifies which files are to be deleted. The /H option allows hidden files to also be deleted.

During the delete operation, if a file cannot be deleted for some reason (eg. it is set to 'read only') then the offending filename is printed along with an error message, and the delete operation continues with the next file. If many such errors occur, then the /P option will cause the output to pause at the end of the screen.

If the filename is *.*, then the prompt:

```
          Erase all files (Y/N) ?
```

is printed, and a reply is waited for. If the reply is anything other than 'Y' or 'y', then the file deletion does not take place. This is a safety feature designed to prevent accidental loss of all files in a directory.

If files are deleted unintentionally on a disk that was formatted under MSX-DOS 2, then the UNDEL command may be used immediately afterwards to restore them again.

Examples

```
ERASE FILE1.BAK
```

The file FILE1.BAK is deleted from the current directory of the current drive.

```
DEL *.COM/H
```

All files matching *.COM, both hidden and not hidden, are deleted.

```
DEL B:\UTIL\*.COM + B:\UTIL\*.BAT
```

All files matching *.COM or *.BAT are deleted from the directory called UTIL on drive B:.

```
DEL B:\UTIL
     Erase all files (Y/N) ?
```

All files in the directory called UTIL on drive B: are deleted. Since so many files are being deleted, a prompt is printed first to prevent a catastrophe.

```
DEL *.BAT
AUTOEXEC.BAT -- Read only file
REBOOT.BAT -- Read only file
```

All files matching *.BAT are deleted except for AUTOEXEC.BAT and REBOOT.BAT which have been marked as read only.

EXIT

Format

```
EXIT [number]
```

Purpose

Exits COMMAND2.COM to the invoking program.

Use

The number is an error code and defaults to 0, which in MSX-DOS indicates no error (see chapter 8 on Errors).

EXIT exits the command interpreter (COMMAND2.COM) and returns the error code to the program that originally loaded and executed it (see the COMMAND2 command). This may be another COMMAND2.COM, another program or, normally, MSXDOS2.SYS. In the latter case, an appropriate error message will be printed and COMMAND2.COM simply reloaded and executed.

COMMAND2.COM when loaded saves the current environment (see chapter 7 on Environment Items), and EXIT restores it. Thus, when EXIT exits back to

MSXDOS2.SYS (that is, EXIT is executed at the primary level), the environment will be cleared. COMMAND2.COM will then be reloaded and will set up the default environment again, providing a method of resetting the environment to it's default values.

Examples

```
EXIT
```

The command interpreter is exited. What happens next depends on what loaded it.

```
EXIT 40
*** User error 40
```

The command interpreter is exited with an error code of 40. Since this does not correspond to an error that is known to the system, the error message is printed by whatever loaded the command interpreter in the first place (see chapter 8 on Errors).

FIXDISK

Format

```
     FIXDISK [d:] [/S]
```

Purpose

Updates a disk to the full MSX-DOS 2 format.

Use

d: specifies the drive on which FIXDISK is to operate. If d: is not specified, the current drive will be assumed. /S option causes the disk to be updated to full MSX-DOS 2 disk.

This command is mainly used to update MSX-DOS 1 disks to full MSX-DOS 2 compatibility, but may also be useful for updating other disks of a similar format or reparing incorrect boot sector.

Although the disk format used by MSX-DOS 1 and MSX-DOS 2 is standardized, MSX-DOS 1 does not use the information stored on certain parts of the disk (the boot sector) and so this information is not necessarily correct on MSX-DOS 1 disks. This can cause problems when MSX-DOS 2 is used with these disks. Additionally, the MSX-DOS 2 UNDEL command will only work with disks that were formatted using MSX-DOS 2 (ie. disks that have a 'volume id' in the boot sector) and so will not work with MSX-DOS 1 disks or disks formatted on other systems.

The FIXDISK command will update a disk so that it is fully MSX-DOS 2 compatible, and it's use will allow full use of MSX-DOS 2 disk features. If /S option is specified, the boot program will be updated for MSX-DOS 2 so that the features of MSX-DOS 2 disk can fully work. When a disk has been updated in this way, however, it may no longer be fully compatible with the original system. For example, if /S option is specified against the disk of the application which uses non-standard boot program, such as some games, the application will no longer be booted, although it will still be able to start MSX-DOS 1 or MSX-DOS 2.

To help prevent accidental updates of boot disks from other systems, a prompt is issued before updating a disk.

Examples

```
FIXDISK B: /S
Disk in drive B: will only be able to boot MSX-DOS 2
Press any key to continue...
```

Drive B: will be updated to be fully MSX-DOS 2 compatible. Since the disk may have been a boot disk from another system, a prompt is issued before the disk is actually updated.

FORMAT

Format

```
     FORMAT [d:]
```

Purpose

Formats (initializes) a disk.

Use

The specified or current drive is formatted, and all data on the disk will be destroyed.

After giving a FORMAT command, an option may be prompted for, allowing the required format of the disk (such as 1DD or 2DD) to be selected. The exact nature of these prompts depends on the manufacturer of the MSX machine, so obey the descriptions of the manual of the machine when you want to format the disk.

After formatting, there will be no files or directories on the disk, and the maximum amount of disk space will be free. The disk will not have a volume name, but can be given one with the VOL command. To turn the disk into a boot disk so that MSX-DOS can be started up from it, the files MSXDOS2.SYS and COMMAND2.COM must be copied onto it with the COPY command.

Examples

```
FORMAT B:
1 - Single sided
2 - Double sided
? 2
All data on drive B: will be destroyed
Press any key to continue...
```

The command was given to format the disk in drive B:. In this case, the options available were to select either double sided or single sided, and double sided was selected. The standard warning prompt was then printed.

```
FORMAT
```

This will format the current drive after the prompts given above.

HELP

Format

```
HELP [subject]
```

Purpose

Provides on-line help for an MSX-DOS feature.

Use

If no parameter is given, then a list of standard subjects on which help is available is printed. This includes all the commands and the major system features.

If a subject is specified, then help text on this subject is printed on the screen from a 'help file'. This will pause at the end of every screen until a key is pressed.

Help files have a filename of:

```
subject.HLP
```

and are located by default on the standard MSX-DOS boot disk in a directory called HELP.

An environment item called HELP is set up initially to refer to the HELP directory (see chapter 7 on Environment Items). This can be changed with the SET command to refer to any other directory or disk if required.

Any HELP subject can be added by the user for his own use simply by adding the appropriate .HLP file in the HELP directory. The help file will be displayed very much like the TYPE command would display it.

Examples

```
HELP
```

A general help screen is printed. This lists all the subjects on which help is available, including the standard commands and main features of MSX-DOS. The user-supplied subjects are not available here.

```
HELP XCOPY
```

Help information on the XCOPY command is printed. This includes a description of how to use the command and what options are available.

```
HELP ME
*** File for HELP not found
```

This command caused HELP to look for a file called ME.HLP from which to get the help text, but did not find it so printed the error message. The files containing the help text are normally found in a directory called \HELP on the drive from which MSX-DOS was booted, and any other help files may be added if required. If ME.HLP was added then HELP ME would print it on the screen.

See MKDIR

MKDIR

Format

```
MKDIR [d:] path
      or
MD [d:] path
```

Purpose

Creates a new sub-directory.

Use

The last item in the path is the name of the new sub-directory which is to be created on the current or specified drive. Thus if this is the only item in the path, the new directory is created in the current directory. If the new directory is to be hidden, then it must be separately made hidden with the ATDIR command.

When a new directory is created, it is empty except for two special sub-directories called '.' and '..'. These are automatically created in the directory and it is these that allow '.' and '..' to be given in path names to signify the current and parent directories respectively (see chapter 3 on Notation for a description of paths).

The MD command is an abbreviated form of the MKDIR command provided for convenience and MS-DOS compatibility.

Examples

```
MKDIR UTIL
```

A directory called UTIL is created in the current directory of the current drive.

```
MKDIR A:\UTIL\RAM
```

A directory called RAM is created in the UTIL directory in the root directory of drive A:.

MODE

Format

```
MODE number
```

Purpose

Changes the number of characters/line on the screen.

Use

The number must be in the range 1 to 80 inclusive, and the number of characters per line on the screen will be set to this. The screen will be cleared and the cursor moved to the top left corner in the process.

Examples

```
MODE 80
(or MODE=80)
```

The screen is set to 80 column mode and is cleared in the process.

```
MODE 25
(or MODE=25)
```

The screen is set to 25 columns.

MOVE

Format

```
MOVE [/H] [/P] compound-filespec [path]
```

Purpose

Moves files from one place to another on a disk.

Use

The compound-filespec specifies which files are to be moved, and /H allows hidden files to be included in the move.

The path specifies the directory to which the files are to be moved, the current directory being used if this is not given. The path must exist on each drive referenced by the filespecs in the compound-filespec.

If a particular file cannot be moved into the specified or current directory (eg. if a file of the same name already exists) then the offending filename is printed along with an error message, and the move operation continues with the next file. If many errors occur, then the /P option will cause the output to pause at the bottom of the screen.

Examples

```
MOVE FILE1 \
```

The file FILE1 is moved from the current directory of the current drive to the root directory of the current drive.

```
MOVE /H /P E:*.COM \
COMMAND2.COM -- File exists
```

All files matching *.COM, both hidden and not hidden, in the current directory of drive E: are moved to the root directory of that drive. The file COMMAND2.COM already existed in the root directory, so the error was printed. Neither of the COMMAND2.COM files were moved or altered. If many such errors had occurred then a prompt would have been printed after a screen full.

```
MOVE \UTIL\*.COM + \UTIL\*.BAT
```

All files matching *.COM or *.BAT in a directory called UTIL on the current drive are moved to the current directory of that drive.

MVDIR

Format

```
     MVDIR [/H] [/P] compound-filespec [path]
```

Purpose

Moves directories from one place to another on a disk.

Use

The compound-filespec specifies which directories are to be moved, and /H allows hidden directories to be included in the move.

The second parameter specifies the directory into which the directories are to be moved, the current directory being used if this is not given. The path must exist on each drive referenced by the filespecs in the compound filepecs.

If a particular directory cannot be moved into the specified or current directory (eg. if a directory of the same name already exists) then the offending directory name is printed along with an error message, and the move operation continues with the next directory. If many errors occur, then the /P option will cause the output to pause at the bottom of the screen.

Note that it is not possible to move a directory into one of it's own descendant directories, as this would produce an invalid sub-directory tree. An error is given if this is attempted.

Examples

```
MVDIR COM UTIL
```

A directory called COM and all descendant directories and files are moved into a directory called UTIL, both directories being in the current directory of the current drive.

```
MVDIR \COM + \BAT \UTIL
```

A directory called COM and a directory called BAT, and both their contents, are moved into a directory called UTIL.

```
MVDIR E:DIR?/H/P ALL
DIR2 -- Duplicate filename
```

All directories in drive E matching DIR? (eg. DIR1, DIR2 and DIR3), which may be hidden, and the contents of the directories, are moved into a directory called ALL. A directory called DIR2 already existed in ALL so the error was printed. Neither of the DIR2 directories were affected at all.

PATH

Format

```
PATH [ [+|-] [d:]path [ [d:]path [ [d:]path ...]] ]
```

Purpose

Displays or sets the COM and BAT command search path.

Use

If no parameters are specified, then the search path currently set will be displayed, separated by semi-colons (';').

If + or - is not given, then the search path will be set to the list of path names given and any existing search path will be deleted.

If - is given before the list of paths, then each path in the list will be deleted from the currently set search path, and an error will be given if any of the given paths do not already exist.

If + is given before the list of paths, then each path specified will first be deleted from the currently set search path if it exists, and will then be added onto the end. This allows the order of the paths in the search path to be changed and allows new paths to be appended to the end of the current search path. The + syntax can also be used to set a search path longer than can be given in one command, the maximum length of the search path being 255 characters and the maximum length of a command 127 characters.

When searching for a COM or BAT file, the paths in the current search path will be used in order from left to right. It is recommended that the paths in the search path are specified as full paths starting at the root directory and with the drive specified. If this is not the case, then the meaning of the search path could change when the current drive or directory is changed.

The search path is stored as an environment item (see chapter 7 on Environment Items), and so can also be accessed with the SET command.

Examples

```
PATH E:\COM E:\BAT
```

When a COM or BAT command is next searched for, the directories searched will be the current directory of the current drive, the COM directory in the root directory of drive E: and the BAT directory in the root directory of drive E:, in that order.

```
PATH
; E:\COM; E:\BAT
```

No parameters were given so the current search path was printed.

```
PATH +A:\COM; A:\BAT
```

The directories A:\COM and A:\BAT are added to the end of the search path.

```
PATH
; E:\COM; E:\BAT; A:\COM; A:\BAT
```

The new search path is printed.

```
PATH -E:\COM, E:\BAT
```

The directories E:\COM and E:\BAT are deleted from the current search path.

```
PATH
 ; A:\COM; A:\BAT
```

The new search path is again printed.

PAUSE

Format

```
PAUSE [comment]
```

Purpose

Prompts and waits for a key press in a batch file.

Use

The comment consists of an arbitrary sequence of characters.

The comment, if given, is printed followed by the prompt 'Press any key to continue... ' on the next line. The system will then wait for a key to be pressed and will echo the key pressed if it is a printable character. If no comment is given as a parameter, then just the prompt will be printed.

The main use of this command is to issue prompts from within a batch file.

Examples

```
PAUSE
Press any key to continue...
```

No comment was given, so just the prompt was printed.

```
PAUSE Insert document disk in drive B:
Insert document disk in drive B:
Press any key to continue...
```

The comment given was 'Insert document disk in drive B:' so this was printed followed by the prompt.

RAMDISK

Format

```
     RAMDISK [number[K]] [/D]
```

Purpose

Displays or sets the RAM disk size.

Use

If no parameters are given, then the current RAMDISK size is displayed as the number of kilobytes.

The number, if given, specifies the maximum size for the new RAM disk, and is specified in kilobytes. The range is 0 to 4064. If the number is 0 or only /D is specified, the RAM disk will be deleted. This number will be rounded up to the nearest multiple of 16K since the RAM disk is always a multiple of 16K. A RAM disk smaller then the specified maximum size may be created if there is not enough free memory for the full size, although a 'not enough memory' error will be given if there is no memory at all available for the RAM disk. Note that the number specified is the maximum amount of RAM to use for the RAM disk, which is not the same as the maximum amount of free space available on the newly-created RAM disk since the system needs to use some for FAT or directories.

On MSX machines with 128K RAM, the maximum amount of RAM disk is 32K.

If a RAM disk already exists before a new one is created, then a 'Destroy all data on RAM disk (Y/N)?' prompt is printed to avoid accidental loss of data. /D can be given which will automatically delete any existing RAM disk first, thus suppressing the prompt.

Having created a RAM disk, it can be referred to as drive H:.

The RAMDISK command is normally only used in an AUTOEXEC.BAT batch file, with a large number specified so that as large a RAMDISK as possible is created. It is not advisable to keep any data on a RAM disk except for a short length of time that is not also kept on a floppy disk, since it will be lost if, for example, the power to the computer fails.

Examples

```
RAMDISK
RAMDISK=160K
```

No parameters were given, so the current size is printed, in this case 160K.

```
     RAMDISK
*** RAM disk does not exist
```

No parameters were given but no RAM disk has been created, so the error is given.

```
RAMDISK = 300
Destroy all data on RAM disk (Y/N)? y
```

A RAM disk already existed, so the prompt was printed. In this case, the reply was 'y' so the current RAM disk was deleted and the new one set up with a maximum size of 300K.

RD

See RMDIR

REM

Format

```
REM [comment]
```

Purpose

Introduces a comment in a batch file.

Use

The comment is simply ignored, and the next command executed. The comment consists of a sequence of any characters up to the maximum length of a command line (127 characters).

Examples

```
REM This is my AUTOEXEC batch file
```

This command, either in a batch file or typed in, does nothing at all with it's parameters.

REN

See RENAME

AME

Format

```
     RENAME [/H] [/P] compound-filespec filename
                    or
     REN [/H] [/P] compound-filespec filename
```

Purpose

Renames one or more files.

Use

The compound-filespec specifies the files that are to be renamed, and /H allows hidden files to be included in the rename operation.

The second filename specifies the new name for the files. A '?' in the new name indicates that the corresponding character from the filename being renamed will be used, thus allowing an ambiguous rename. Thus '*' in the second filename, which is just equivalent to a series of '?'s, indicates that the whole of the filename or extension will remain unchanged.

If for some reason a particular file cannot be renamed (eg. if a file or directory with the new name already exists or they are read-only) then the offending filename will be printed along with an error message and the rename operation will continue with the next file. If many errors occur, then /P will cause the output to pause at the end of the screen.

Examples

```
RENAME FILE1 FILE2
```

The file FILE1 in the current directory of the current drive is renamed to FILE2.

```
REN B:\DIR1\*.DOC/H/P *.OLD
FILE2.DOC -- Duplicate filename
```

All files matching *.DOC in the directory called DIR1 in the root directory of drive B:, including hidden files, are renamed with the same main name but with an extension of .OLD. The file FILE2.DOC could not be renamed because there was already a file called FILE2.OLD in the directory, so the error was printed. Neither FILE2.DOC nor FILE2.OLD was altered at all. If many such errors had been printed, then a prompt would have been printed at the bottom of every screen full, since /P was given.

```
     REN DOC + FILE1 *.OLD
```

All files in the directory called DOC and the file FILE1, both in the current directory of the current drive, and renamed with an extension of .OLD.

RMDIR

Format

```
     RMDIR [/H] [/P] compound-filespec
                    or
     RD [/H] [/P] compound-filespec
```

Purpose

Removes one or more sub-directories.

Use

The compound-filespec specifies which directories are to be deleted, and /H allows hidden directories to be included in the delete operation.

In order to delete a directory, it must contain no other files or other directories except for the special '.' and '..' directories which are always contained in a directory. These are put in a new directory when it is created and cannot be removed. It is these that allow '.' and '..' to be used in path names to specify the current and parent directories respectively (see chapter 3 on Notation for a description of paths).

If a directory cannot be deleted for some reason (eg. it is not empty) then the name of the offending directory is printed along with an error message, and the delete operation continues with the next directory. If many errors occur then /P will cause the output to pause at the end of the screen.

Examples

```
RMDIR DIR1
```

The directory called DIR1 in the current directory of the current drive is removed.

```
RD B:\COM + B:\BAT
```

The directories COM and BAT are removed from the root directory of drive B:.

```
RD \*.*
UTIL -- Directory not empty
```

An attempt was made to remove all directories from the root directory of the current drive, but a directory called UTIL was not empty and so the error was printed. UTIL and it's contents are not affected at all.

RNDIR

Format

```
RNDIR [/H] [/P] compound-filespec filename
```

Purpose

Renames one or more sub-directories.

Use

The compound-filespec specifies the directories that are to be renamed, and /H allows hidden directories to be included in the rename operation. The contents of the directories remain unchanged.

The second filename specifies the new name for the directories. A '?' in the new name indicates that the corresponding character from the name of the directory being renamed will be used, thus allowing an ambiguous rename. Thus '*' in the second filename, which is just equivalent to a series of '?'s, indicates that the whole of the filename or extension of the directory name will remain unchanged.

If for some reason a particular directory cannot be renamed (eg. if a file or directory of the new name already exists) then the offending directory name will be printed along with an error message and the rename operation will continue with the next directory. If many errors occur, then /P will cause the output to pause at the end of the screen.

Examples

```
RNDIR UTIL COM
```

The directory called UTIL in the current directory of the current drive is renamed COM.

```
RNDIR A:\*.*/H/P *.OLD
UTIL -- Duplicate filename
```

All directories, hidden and not hidden, in the root directory of drive A: are renamed with an extension of .OLD. The directory UTIL could not be renamed because a directory called UTIL.OLD already existed, so the error was printed. If many such errors were printed then /P would cause a prompt to be printed at the end of every screen full.

```
RNDIR COM + BAT *.OLD
```

The directories COM and BAT are renamed to COM.OLD and BAT.OLD respectively.

SET

Format

```
SET [name] [separator] [value]
```

Purpose

Displays/sets environment items.

Use

If no parameters are given, then all currently defined environment items and their current values are displayed. Initially there are several items set up to default values (see chapter 7 on Environment Items).

If just a name is given as the parameter, then the current value of the specified environment item is printed.

If the name is followed by a separator, then the separator is ignored and the name is set to the following value. If the value is blank (ie. not given) then the environment item is deleted from the environment space.

The area of memory used for environment items is also used for disk buffers. Thus if a 'not enough memory' error occurs when using the SET command, then it may help to reduce the number of disk buffers (see the BUFFERS command).

Chapter 7 contains more information about environment items and the items and values that are set up by default.

Examples

```
     SET
     ECHO=OFF
     PROMPT=OFF
     PATH=;
     TIME=12
     DATE=yy-mm-dd
     HELP=A:\HELP
     SHELL=A:\COMMAND2.COM
```

No parameters were given, so all the currently set environment items were printed, in this case typical default values.

```
SET HELP=A:\HELP
```

An item called HELP is set to the value A:\HELP.

```
SET HELP
A:\HELP
```

The current value of HELP is printed.

```
SET HELP=
```

The item HELP is set to a null value, thus removing it from the environment item list.

TIME

Format

```
     TIME [time]
```

Purpose

Displays or sets the current time.

Use

If the time is given after the command, then the time is set to this value (for the format see below). If the time is not given after the command, then the current time is printed and the new time is prompted for and input. If no input is given (ie. if the 'enter' key alone is pressed) then the current time is not altered. Otherwise the input is assumed to be a new time, and is interpreted as described below. If the time is invalid then an error message is displayed and the new time again prompted for and input.

The time is expected to consist of up to four numbers, separated by one of the following characters:

space tab , - . / :

with spaces allowed either side of the character. Any missing numbers will default to the current setting. The first number is the hour, the second is the minutes, the third is the seconds and the forth is the centi-seconds. The centi-seconds are not printed however since it is not very useful to know the current value, or indeed to enter a new one.

The format in which the time is printed is flexible and can be changed. An environment item (see chapter 7 on Environment Items) called TIME is set up by default to the value '12', which indicates that the time will be printed in 12 hour format with a following 'a' or 'p' for am and pm. The command SET TIME 24 will cause the time to be printed in 24 hour mode. The time can be input unambiguously in either format. The time format also affects the times printed by the DIR command.

Examples

```
TIME 16:45
```

The current time is set to 4:45 pm.

```
TIME
     Current time is 10:45:00a
     Enter new time:
```

No parameters were given, so the current time is printed (in this case in 12 hour mode) and the new time prompted for.

```
TIME 10-50-30-23
```

The time is set to 30.23 seconds after 10:50 am.

TYPE

Format

```
TYPE [/H] [/P] [/B] compound-filespec | device
```

Purpose

Displays data from a file or device.

Use

The compound-filespec specifies the files that are to be displayed, and /H allows hidden files to be typed. If the compound-filespec is ambiguous, then the filename is printed before each one is typed.

If /B is specified, then data is read from each file and displayed without modification on the screen, until the end of file is reached. This may have strange effect on the screen if the file contains control characters.

If /B is not given, then TYPE will look for the end-of-file character (CTRL-Z) and stop when it finds it. Also control characters except carriage return, line feed and tab will be converted into characters that can be printed, A for ^A, W for ^W, etc.

If /P is given, then the output will pause at the end of the screen until a key is pressed.

Examples

```
     TYPE FILE1
```

Data is read from the file and printed on the screen, up to the first end-of-file character.

```
TYPE *.BAT/H/P
```

All batch files, including hidden ones, are read in and displayed. A prompt is printed at the end of every screen full.

```
TYPE AUTOEXEC.BAT + REBOOT.BAT
```

The files AUTOEXEC.BAT and REBOOT.BAT are displayed.

```
TYPE /B DIR1
```

All files in the directory DIR1 are printed on the screen and no interpretation is put on the data in the files.

UNDEL

Format

```
UNDEL [filespec]
```

Purpose

Recovers a previously deleted file.

Use

The filespec specifies which files are to be undeleted if possible, and defaults to *.*.

Files can only be undeleted if they have been deleted using MSX-DOS 2 on an MSX-DOS 2 formatted disk and if no disk allocation has taken place since the file was originally deleted, which usually means that they have to be undeleted immediately after they have been deleted.

Each deleted file and directory reference that is found in the directory specified by the filespec will be undeleted if it's name is matched by the filename in the filespec, and if undeletion is possible. UNDEL can therefore be used to restore a directory removed with the 'RD' or 'RMDIR' commands; to restore the contents of the directory a further UNDEL command is required specifying the now undeleted directory.

Note that UNDEL is a transient command, and therefore must be loaded from disk.

Examples

```
UNDEL B:HELP.MAC
```

Attempts to undelete the file HELP.MAC from the current directory of drive B:.

```
UNDEL A:\DIR1
```

All undeletable files and directories in DIR1 are undeleted.

VER

Format

```
VER
```

Purpose

Displays the system's version numbers.

Use

The version numbers of the three main components of the MSX-DOS disk system are displayed. Each version number consists of three digits. The first digit is the main MSX-DOS version number and for MSX-DOS 2 will always be 2. The second digit is the version number and will change for future versions that have, for example, had extra major features added. The last digit is the release number and will change with different releases of the same version of the system which have had minor changes, improvements and corrections made.

Examples

```
VER
MSX-DOS kernel version 2.20
MSXDOS2.SYS version 2.20
COMMAND2.COM version 2.20
Copyright 1988 ASCII Corporation
```

The version numbers of all the components of the MSX-DOS disk system are printed out.

VERIFY

Format

```
VERIFY [ON | OFF]
```

Purpose

Displays/sets the current disk write verify state.

Use

If no parameters are given, then the current verify state is displayed on the screen.

If ON or OFF is given, then the verify state is changed appropriately.

The verify state affects all writes to disk. If OFF, the default state, then data is simply written. If ON, then after writing the data it is read back and compared with the original to ensure that it was written correctly. The extra overhead of this means that writing is slower when verify is on.

This feature depends on the device driver, so this will have no effect if the driver does not have the feature.

Examples

```
VERIFY
     VERIFY=OFF
```

No parameters were given, so the current verify state is printed, which in this case is off.

```
VERIFY ON
```

Disk write verification is turned on.

VOL

Format

```
     VOL [d:] [volname]
```

Purpose

Displays or changes the volume name on a disk.

Use

If no parameters are given, or if only a drive name is given, then the volume name of the current or specified drive is printed.

If a volname is given, then the volume name of the specified or current drive is changed to the specified volume name.

Examples

```
VOL B:
     Volume in drive B: has no name
```

Just a drive was given, so the volume name for the disk in that drive is printed. In this case there was no volume name defined.

```
VOL B:BACKUP
```

The volume name of the disk in drive B: is changed to BACKUP.

XCOPY

Format

```
     XCOPY [filespec [filespec]] [options]
```

Purpose

Copies files and directories from one disk to another.

Use

The options available are:

[/A] [/E] [/H] [/M] [/P] [/S] [/T] [/W] [/V]

XCOPY is an extended file copying command (compare with the COPY command) that can selectively copy both files and directories. The first filespec specifies the source filenames, and if /H is given then hidden files will also be copied. The second filespec is the destination filename. Thus files can be renamed during the copy (as in the standard COPY command).

/T (Time) will cause the copied files to have the current date and time rather than the source file's date and time.

If /A (Archive) is specified, then only files with the 'archive' attribute set are copied. A file has an archive attribute in the same way as a 'hidden' attribute and a 'read only' attribute. It is set whenever a file is updated (written to).

/M is similar to /A, but resets the archive bit after copying the file. Thus, using this option, files can be regularly copied onto another disk only if they have been updated, providing a file backup facility.

/S causes XCOPY to copy directories as well as files. Within each directory, all files are copied and then any matching files within each directory are copied, with the directory being created on the destination if it does not already exist. Normally, these directories will not be created if no files are to be copied into them.

/E can be given to cause the /S option to create all directories, even if they are empty.

The /P (Pause) option will cause XCOPY to pause and prompt before copying each file, which allows files to be selectively copied.

/W (Wait) causes XCOPY to pause and prompt before copying any files, so that disks can be changed.

/V option can be given to turn write verification on for the duration of the XCOPY command (see the VERIFY command). This will ensure that data is written correctly to disks if the device driver being used has the feature, but will slow the operation down.

Note that XCOPY is a transient command, and so must be loaded off disk.

Examples

XCOPY B:\

All files in the root directory of drive B: are copied to the current directory of the current drive. There is no advantage in this case over using the standard built-in COPY command.

XCOPY *.* B: /H/S/M

All files, including hidden files, are copied to drive B: only if they have been modified since a similar command was last given. The archive attributes are then reset so that the files are marked as unmodified. Not only are all the files in the current directory copied, but so are directories and all their descendant directories and files.

XDIR

Format

```
XDIR [filespec] [/H]
```

Purpose

Lists all files within directories.

Use

The filespec specifies which files are to be listed, and /H allows hidden files to be included.

XDIR is similar to the DIR command, but does not print the files' dates and times.

After all files in the specified directory have been listed, then files within descendant directories are also listed, and are shown indented. This allows a DIR of a complete directory tree or disk to be obtained.

Note that XDIR is a transient command, and so must be loaded off disk.

Examples

```
XDIR
```

The directory of the entire disk in or descending from the current directory of the current drive is printed.

```
Volume in drive A: is MSX-DOS 2
X-Directory of A:\

MSXDOS2.SYS     r     4480
COMMAND2.COM    r    14976
AUTOEXEC.BAT            57
REBOOT.BAT              57

\UTILS
        CHKDSK.COM       7680
        DISKCOPY.COM     7168
        FIXDISK.COM       768
        UNDEL.COM        3968
        XCOPY.COM       10112
        XDIR.COM         7168
        MKSYS.BAT         569
        AUTOEXEC.BAT       47
        REBOOT.BAT         90
\HELP
        ASSIGN.HLP        819
        ATDIR.HLP        1527
        ATTRIB.HLP       1828
        .
        .
        .
292K in 117 files   530K free
```

```
XDIR B:\DIR1
```

All the files and directories and their contents are printed from the directory DIR1.

```
XDIR \*.COM/H
```

The names of all files, including hidden files, matching *.COM are displayed.

5. Redirection and Piping

COMMAND2.COM offers the redirection and piping features as described below. They may be bypassed by setting the environment item "REDIR" to "OFF" ("SET REDIR=OFF"), so the compatibility to MSX-DOS 1 or CP/M can be achieved.

Redirection

Most commands, CP/M programs and MSX-DOS programs output text to the screen by writing to the 'standard output', and read from the keyboard by reading from the 'standard input'. COMMAND2.COM, however, provides facilities for changing the standard input and standard output for the duration of the command to refer to other MSX-DOS devices or to files on disk by including one or more of the redirection symbols <, > and >> on the command line, followed by a filename.

For example, the ECHO command normally just outputs it's parameters to the screen by writing the characters to the standard output. It can be made to output to the printer instead by redirecting it's output, as follows:

```
          ECHO text >PRN
```

which changes the standard output to refer to the device PRN for the duration of the ECHO command. Similarly, the command:

```
          ECHO text >file1
```

will cause the specified file ("file1") to be created, and the output of the ECHO command written to the file. To append the output of a command to the end of an existing file, the >> symbol can be used instead of the > symbol, and the file will only be created if it does not already exist.

To change the standard input, the < symbol is used in a similar manner to the > symbol. In this case, the file must already exist, and must contain adequate input for the command. If the command attempts to read input beyond the end of the file, then it will be aborted since it cannot continue.

When redirection information is given on the command line, it is used by COMMAND2.COM to set up the redirection and then removed from the command line. Thus in the examples above, the ECHO command will not echo the redirection symbols or the filename.

If the input or output of a batch file is redirected, then that redirection is applied to all commands within the batch file. Individual commands within the batch file may still be redirected, however, which will override the batch file redirection. See chapter 6 on Batch Files for more information on commands in batch files.

Piping

As well as redirecting the input and output of a command or program to another device or a disk file, it is possible to redirect or 'pipe' the standard output of one command into the standard input of another. Typically the second command will be a program which reads from it's standard input, modifies the data, and writes it to it's standard output. Such a program is called a 'filter'. For example, a filter could be produced which read data from it's standard input, sorted it into alphabetical order, and wrote it to it's standard output. Thus the output of the DIR command could be sorted.

Piping is indicated on the command line by separating the two commands by the | symbol. The command to the left of the | symbol will be performed first, and it's output will be redirected to a temporary file created by COMMAND2.COM. Then the second command will be performed, with it's standard input redirected from the same temporary file. When the second command ends, the temporary file will be deleted. The standard output of the second command may of course have been piped into the standard input of a third command, and so on.

If any input redirection occurs on a command line involving a pipe, then the redirection is applied to the first command in the pipe, as all the other commands receive their standard input from the standard output of the previous command in the pipe. Similarly, if any output redirection occurs on a command line involving a pipe, then the redirection will apply to the last command on the command line.

It is not possible to use pipes on either the input or the output of batch files directly. It is, however, possible to use piping with batch files if they are executed with the COMMAND2 command (see chapter 4) since it is then the COMMAND2 command that is being redirected and not the batch file.

As mentioned above, in order to pipe the output of one command into the input of another, temporary files will be created and deleted by COMMAND2.COM. The location of these temporary files is specified by the TEMP environment item (see chapter 7 on Environment Items), and this may be changed to refer to any drive and directory (for example piping will be speeded up considerably if TEMP refers to a directory on a RAM disk). By default, TEMP refers to the root directory of the boot disk. The filename used for the temporary file is created by COMMAND2.COM, so TEMP should specify just the drive and directory. The filename is of the form:

```
     %PIPExxx.$$$
```

where xxx is a three digit number chosen by COMMAND2.COM to avoid clashes with any other files in the TEMP directory.

6. Batch Files

When a command is given to MSX-DOS and it is not one of the internal commands, a file of that name is searched for with an extension of COM or BAT. If not found in the current directory then the current search path is searched (see the PATH command). If a COM file is found, then it is loaded and executed. If a BAT file is found, then MSX-DOS starts execution of the batch file.

A batch file is a text file that contains a list of commands, and these commands are read from the file a line at a time and executed as though they were typed at the keyboard. Several of the commands described in chapter 4 are in fact provided mainly for use in batch files, such as ECHO and PAUSE.

As each command is read, normally it is executed immediately. An environment item ECHO exists, however, that can be set to ON (with the command SET ECHO ON) to cause each command line to be printed on the screen before it is executed (see chapter 7 for Environment Items). The command line is echoed in this way after % parameter substitution (see below) has taken place. The command SET ECHO OFF will restore the normal state.

In the command line that invoked the batch file, parameters may follow the name of the batch file just like any other command or transient program name. These parameters may be accessed anywhere in the batch file by specifying %0 to %9. %1 is the first parameter specified in the command line, %2 is the second and so on. %0 is the name of the batch file itself. The % number will be replaced by the parameter on the original command line, and may appear anywhere on a batch file command line. To actually use a % symbol on a command line a double % must be given ('%%') which will then be replaced by a single one.

If the execution of any command in a batch file is terminated prematurely for some reason (typically the control-STOP or control-C key being pressed) then the following prompt is issued:

```
          Terminate batch file (Y/N)?
```

If the response to this is 'Y', then execution of the whole batch file is stopped. If the response is 'N', then batch file execution continues with the next command in the batch file.

After MSX-DOS has executed a command in batch file, it may need to read the next command in the batch file off disk. If the correct disk is not in the drive when it comes to do this, then a prompt is issued. For example, the following prompt will be issued if the batch file was originally executed from drive A::

```
          Insert disk for batch file in drive A:
```

```
Press any key to continue...
```

When the correct disk has been inserted and a key pressed, batch file execution will continue normally.

Below is a very simple batch file, which just prints the first few parameters.

```
ECHO Parameter 0 = %0
ECHO Parameter 1 = %1
ECHO Parameter 2 = %2
ECHO Parameter 3 = %3
```

If this is called MYBAT.BAT then the command MYBAT a b c will output

```
Parameter 0 = MYBAT
Parameter 1 = a
Parameter 2 = b
Parameter 3 = c
```

When MSX-DOS starts up for the first time, a special batch file called AUTOEXEC.BAT is looked for and is executed if found. This may contain any MSX-DOS command, and usually contains once-only initialization commands, such as a RAMDISK command to set up a RAM disk.

One % parameter is passed to AUTOEXEC.BAT as %1. This the drive that MSX-DOS booted from and is in the form of a normal drive letter followed by a colon.

Another special batch file is REBOOT.BAT. This is executed when MSX-DOS is re-booted after using DISK-BASIC. As with AUTOEXEC.BAT files, the single %1 parameter passed is the drive from which MSX-DOS was re-booted.

Usually some commands need to be performed whenever MSX-DOS is booted, whether for the first time or sometime later, and these are put in the REBOOT batch file. They can then be executed from the AUTOEXEC batch file by ending it with the command REBOOT %1. An example of a command that might be put in the REBOOT batch file is the PATH command to set up the transient command search path. When setting up the search path using the command, %1 can be used to set up the path on whatever drive was booted from.

When a command in a batch file is the name of another batch file, then that second batch file is executed in the normal way. When it ends, control passes back to the interactive command interpreter, and not to the first batch file. Batch file commands thus 'chain' together.

If it is desired to 'nest' batch files ie. to pass control back to the first batch file above, then this can be done with the COMMAND2 command (see chapter 4), passing the name of the second batch file as the parameter. Then when the second batch file ends, the first one will be continued with the command after the COMMAND2 command.

A typical AUTOEXEC batch file is as follows:

```
          ECHO AUTOEXEC executing
          RAMDISK 100
          RAMDISK
               COPY COMMAND2.COM H:\
          REBOOT %1
```

A typical REBOOT batch file is as follows:

```
          ECHO REBOOT executing
          PATH H:\, %1\UTILS, %1\BATCH
               SET SHELL=H:\COMMAND2.COM
               SET TEMP=H:\
          SET PROMPT ON
          H:
```

When the AUTOEXEC batch file executes, the message "AUTOEXEC executing" is printed, and then a RAM disk is set up with a maximum size of 100K. Another RAMDISK command is then given which will print out the actual size of RAM disk created. The COPY command then copies COMMAND2.COM onto the RAM disk so that it can load and re-load quickly. Finally the REBOOT batch file is executed, with the %1 parameter (the boot drive) passed to it.

The REBOOT batch file prints a message and then sets a PATH. The first item in the path refers to the RAM disk that was created by the AUTOEXEC batch file, and the other items refer to directories on the disk from which MSX-DOS was booted (ie. %1). The SHELL environment item is set up so that COMMAND2.COM can re-load quickly off the RAM disk, and the TEMP environment item is set up so that temporary piping files are created on the RAM disk. The prompt is set ON so that the current directory is printed as the prompt and, finally, the RAM disk is made the default drive.

7. Environment Items

MSX-DOS maintains a list of 'environment items' in its work area. An environment item is a named item that has a value associated with it.

An environment item can have any name chosen by the user, and can consist of the same characters that are valid in a filename. The maximum length of an environment item name is 255 characters. MSX-DOS provides several environment items set up by default.

The value of an environment item is simply a string of arbitrary characters up to a maximum length of 255. No processing is performed on the characters and so the casing of characters is preserved. Any environment item that does not exist has a null value (ie. no characters).

An environment item can be changed or set up by the SET command, which can also display currently set environment items.

The environment items set up by default and the manner in which their value is interpreted are as follows:

ECHO

This controls the echoing of lines read from a batch file (see chapter 6 on Batch Files). Any value except 'ON' (lowercase also allowed) is interpreted as 'OFF'.

PROMPT

This controls the displaying of the prompt at command level. Any value except 'ON' is interpreted as 'OFF'.

When PROMPT is OFF, as it is by default, then the prompt consists of the current drive followed by '>' eg. A>.

When PROMPT is ON, then the prompt consists of the current drive and the current directory of that drive followed by '>' eg. A:\COM>. In order to do this, the current drive must be accessed to read the current directory and so may take a little longer to appear.

PATH

The current search path by which COMMAND2.COM searches the command given is maintained as an environment item PATH, and it is this that the PATH command manipulates.

SHELL

The SHELL environment item indicates where the command interpreter (COMMAND2.COM) exists, and is set up by default to where it was loaded from. When the command interpreter needs to re-load itself from disk (after running a transient command) it looks at the SHELL environment item and attempts to load itself from the file that it specifies. If this gives an error then it attempts to load itself from the root directory of the drive that it was originally loaded from.

To cause the command interpreter to re-load itself from another drive or directory, COMMAND2.COM can be copied there and SHELL set to refer to it. For example, it might be copied to the RAMDISK with the command COPY COMMAND2.COM H:\ and then SHELL set with the command SET SHELL=H:\COMMAND2.COM.

TIME

TIME specifies the format the time is displayed by MSX-DOS. If not '24', which indicates that it is displayed as a 24-hour time, then '12' is assumed, which indicates that it is displayed as a 12-hour time with an am. or pm. indication. The TIME environment item does not apply when the time is input because it can be input in either format unambiguously.

DATE

DATE specifies the format the date is displayed and input by MSX-DOS. It defaults to a format appropriate for the country of origin of the MSX machine. It takes the form of three letters or three letter pairs separated by date/time separators (see the DATE command). To set the American format, for example, the command SET DATE=MM/DD/YY could be given.

HELP

When the HELP command is given the name of command for which help is required, then it reads the information displayed from a file on disk. This file is in the directory specified by the HELP environment item. It defaults to a directory called HELP in the root directory of the drive that MSX-DOS was booted from.

APPEND

APPEND is not actually defined by default, but when set up is an environment item that has a special meaning to the system. It is used only with standard CP/M programs.

CP/M programs do not know how to use sub-directories because CP/M does not have sub-directories, but instead just has the equivalent of the current directory. When such a program opens a file, it searches for it only within this single directory and thus only has drives and filenames, not paths.

When a CP/M program is run under MSX-DOS and attempts to open a file, it only searches for the filename in the current directory of the appropriate drive. Similarly, when the user types in a filename to a CP/M program it may only contain a drive and filename, and thus also refers only to files in the current directory.

When this search is performed through MSX-DOS, if the file is not found in the current directory, then the APPEND environment item is looked at. If it is not set up then the file has not been found. If set up, then it is assumed to be a path name, and specifies a single alternative directory in which the search for the file continues.

This will only be of use if the CP/M program opens a file and then reads or writes to it. If it attempts to, for example, delete a file or create a file, then APPEND will not be used. Indeed, it may have undesirable effects and for this reason it is recommended that APPEND is used normally only in a batch file which sets it up, executes the CP/M program, and then unsets it again.

Typical uses for APPEND include specifying the directory in which large programs (such as word processors and database programs) find 'overlay' or messages files, and specifying the directory in which compilers, assemblers and linkers find their source and working files. Typical cases for which APPEND is not suitable and may have undesirable effects include editing a file with a wordprocessor when the file to be edited probably cannot be in a directory other than the current one, even if APPEND is set up.

PROGRAM and PARAMETERS

These special environment items are set up by COMMAND2.COM when a transient command is executed and removed when it finishes, and should thus be avoided for general use.

TEMP

When piping is performed (see chapter 5 on Redirection and Piping) it is necessary for COMMAND2.COM to create one or more temporary files, and the TEMP environment item indicates the drive and directory in which these temporary files are to be created. By default, it refers to the root directory of the boot drive, and typically may be changed to refer to a RAM disk since this increases the speed of piping.

Although the standard MSX-DOS system only uses TEMP for piping, any other programs and utilities that need to create temporary files may also use the TEMP environment item.

UPPER

UPPER controls whether the command line from 80h to be passed to the transient program be converted to uppercase. Any value except 'ON' is interpreted as 'OFF'.

When UPPER is 'OFF' (default), no conversion will be done and the values will be passed to the transient program as they are typed.

When UPPER is 'ON', each character in the command line will be converted to its associated uppercase character and then passed to the transient program. This is compatible to CP/M environment.

REDIR

REDIR controls whether the redirection or piping characters in the command line be processed by COMMAND2.COM. Any value except 'ON' is interpreted as 'OFF'.

When REDIR is 'OFF', the redirection or piping characters will be passed to the transient program as they are typed, and the transient program may process them.

When REDIR is 'ON' (default), the redirection or piping characters will be interpreted and executed by COMMAND2.COM, so they will not be passed to the transient program.

8. ERRORS AND MESSAGES

Disk errors

Disk errors occur when a command or program is attempting to access a disk and fails for some reason, such as a disk not being in the drive. When this happens, message and prompt appears which allows the user the either retry the operation which may now work (eg. if a disk has been inserted into the drive), to ignore the operation or to abort the entire command.

An example disk error message and prompt is as follows, and may be given if the disk was taken out whilst drive A: was being accessed:

```
          Not ready reading drive A:
          Abort, Retry or Ignore (A/R/I)?
```

The 'not ready' part of the message indicates why the disk operation failed, and other possibilities exist (see below). 'reading' indicates that the command was trying to read the disk, and may be replaced by 'writing' if that is what it was doing. 'drive A:' is the drive in which the disk was attempted to be accessed.

The 'Abort, Retry or Ignore' part indicates the possible actions that can be taken by the user, and these are selected by pressing the 'A', 'R' or 'I' key.

If Abort is selected, then the entire command is aborted and the message 'Disk operation aborted' is printed before another command can be typed.

If Retry is selected then the failed disk operation will simply be retried and may fail again or may work, particularly if some corrective action has been taken such as inserting a missing disk.

Ignore causes the failed disk operation to be ignored by the command. In many cases, ignoring an error may not be recommended and in these cases the Ignore option will not even be displayed, although it may still be selected. Doing so may however cause serious system malfunction and could destroy data on the disk. Even if the Ignore option is displayed, it should be used with extreme caution, and only when all else fails. Normally Ignore is only used when the data on a disk has got corrupted and ignoring the disk error offers the only possibility of recovering all or part of the data.

A few serious errors which generally mean the disk has been corrupted beyond possible use are automatically aborted, and just the appropriate error message is printed (eg. 'Bad file allocation table').

The possible errors that can occur as disk errors and their meanings are, in alphabetical order, as follows:

Bad file allocation table

The disk contains an invalid file allocation table (FAT). The FAT is an area on disk in which the system keeps information to tell it where on the disk the data in each file is stored. Thus if the FAT is invalid, it cannot read any data at all. This message usually means that the disk has been corrupted beyond possible use.

Cannot format this drive

An attempt was made to format a disk in a drive that does not support disk formatting. This probably means that a FORMAT command was given specifying the RAM disk.

Data error

The data was read or written without error, but the CRC check failed. This usually means the disk has been corrupted.

Disk error

The data could not be read or written to the disk.

Incompatible disk

An attempt was made to access 2D or 1D disk or a double sided disk in a single sided drive.

Not a DOS disk

The disk is not a format that MSX-DOS can read. For example, although MSX-DOS can execute CP/M programs it cannot access CP/M disks.

Not ready

The disk is not in the drive being accessed. The disk should be inserted into the drive and 'Retry' selected.

Sector not found

MSX-DOS tried to read or write to a non-existent sector. May indicate that the disk has been corrupted.

Seek error

The requested track on the disk could not be found. Could mean a corrupted disk or faulty disk drive.

Unformatted disk

The disk has not been formatted. Use the FORMAT command on the disk before accessing it.

Verify error

Only occurs when verify is on, and means that data appeared to be written to disk successfully but when read back was found to be different to that written.

Write error

Data was not written correctly.

Write protected disk

The disk is write protected and an attempt was made to write data to it. The disk should be made unprotected and 'Retry' selected.

Wrong disk and Wrong disk for file

MSX-DOS accessed a disk once and subsequently needed to access it again, but found that the drive contained a different one. The correct disk should be inserted and 'Retry' selected.

Command Errors

Command errors occur when a command cannot perform it's intended function for some reason.

If an error occurs in a command and it is unlikely to be able to continue, then an appropriate error message is printed, and the next command is read at the prompt.

An example error message is:

```
*** File not found
```

The three asterisks *** are printed first to indicate that an error has occurred. The message is then printed, followed by the normal command prompt on the next line. The possible errors that can occur are listed below.

When a command error occurs in a specific situation, an 'error type' message may also be printed. For example, generally when a required file cannot be found on disk, the 'File not found' message is printed as in the above example. However, if the file required is a file specified by the redirection symbol < (see chapter 5 on Redirection and Piping) then the message printed will be:

```
*** Redirection error: File not found
```

The possible error types are:

Batch file error:

An error occurred whilst attempting to read from a batch file, for example a disk error occurred and 'abort' was selected.

Piping error:

The error occurred during a piping operation, probably in connection with the temporary files that COMMAND2.COM creates (see chapter 5 on Redirection and Piping). For example, the TEMP environment item did not refer to a valid drive or directory (see chapter 7 on Environment Items).

Redirection error:

The error occurred during a redirection operation. For example, an invalid filename was specified after a redirection symbol <, > or >>, or the specified input file was not found (see chapter 5 on Redirection and Piping).

Standard input error:

An error occurred on the standard input to a command or program after redirection or piping has been set up, for example the standard input has been redirected from a file and the end of the file has been reached.

Standard output error:

An error occurred on the standard output of a command or program after redirection or piping has been set up, for example the standard output has been redirected to a file and the disk is full.

Many commands operate on files or directories, and if an ambiguous filename is given then the command operates on several files or directories (for example the RENAME command or the COPY command). Often an error occurs when it is trying to perform the command on one of the files, but which may be successful on one of the other files. In this case, the filename is printed followed by the error message and the command then continues. For example:

```
          COMMAND2.COM -- File cannot be copied onto itself
```

The possible command errors that can be given are, in alphabetical order, as follows:

Cannot concatenate destination file

This error is given by CONCAT and means that one of the filenames matched by the source file specification is the destination file. This is not necessarily wrong but may indicate a mistake in the command.

Cannot create destination file

This is given by COPY, and usually means that the destination file for the file it is copying would, if it was created, overwrite a file that was already in use. This is likely to be a previously copied source file but may be some other file such as the currently executing batch file.

Cannot overwrite previous destination file

This is given by COPY, and means that the destination file for the file it is copying would, if it was created, overwrite the destination file of the file that was previously copied. This usually means that the intended destination was a directory but that it's name was misspelled.

Cannot transfer above 64K

This should not normally occur from commands.

Command too long

A command that was given is too long. This will not occur when typing commands from the keyboard, but may occur from a batch file. The maximum length of a command is 127 characters after % parameter substitution.

Ctrl-C pressed

The command was interrupted by pressing CTRL-C.

Ctrl-STOP pressed

The command was interrupted by pressing CTRL-STOP.

Directory exists

A command attempted to create a new file or directory on disk with the same name as an existing directory.

Directory not empty

The RMDIR (RD) command tried to remove a directory that contains files or other directories. These must be deleted first with the ERASE and RMDIR commands since directories must be empty before they can be removed.

Directory not found

A directory command (eg. RNDIR) could not find the specified directory.

Disk full

There is no more room on the disk and files will have to be deleted and the command given again.

Disk operation aborted

A disk error occurred and the 'Abort' option was chosen, thus aborting the whole command.

Duplicate filename

RENAME (REN) or RNDIR cannot perform the specified rename because the new filename is the same as a filename that already exists. Also occurs from MOVE or MVDIR because a filename already exists in the destination directory with the same name as the file or directory being moved.

End of file

This should not normally occur from commands.

Environment string too long

This should not normally occur from commands.

Error on standard input

This should not normally occur from commands, and means that an error occurred while a command was attempting to read from the keyboard.

Error on standard output

This should not normally occur from commands, and means that an error occurred while a command was attempting to write to the screen.

File access violation

This should not normally occur from commands.

File allocation error

This should not normally occur from commands.

File cannot be copied onto itself

The destination file when trying to do a COPY is the same file as the source.

File exists

MKDIR (MD) tried to create a new directory but a file with the same name already exists in the specified directory.

File for HELP not found

The HELP command looked for a file to get the help text from but could not find it. Help files are usually kept in a directory called \HELP on the boot disk.

File handle not open

This should not normally occur from commands.

File is already in use

A command tried to modify a file that is already being used for some other purpose, such as the currently executing batch file.

File not found

A command could not find the specified file or files.

Internal error

This should not normally occur from commands.

Invalid MSX-DOS call

This should not normally occur from commands.

Invalid attributes

Usually means an invalid +/- attribute was specified in ATTRIB or ATDIR.

Invalid date

The date typed into the DATE command is not a valid date or was typed in in an invalid format.

Invalid device operation

A command cannot perform it's function on one of the built-in system devices eg. a file cannot be renamed CON.

Invalid directory move

MVDIR attempted to move a directory into one of it's own descendant directories, which cannot be done.

Invalid drive

A drive that does not exist was specified.

Invalid environment string

The name of an environment item contains invalid characters. Only those characters valid in filenames are valid in environment item names.

Invalid file handle

This should not normally occur from commands.

Invalid filename

A filename contains invalid characters. This may be a filename explicitly given, or may be the result of attempting to rename a file with an ambiguous new name.

Invalid number

A number given in a command contained non-digit characters.

Invalid option

An invalid letter was given after a / on a command line.

Invalid . or .. operation

A command cannot perform it's function on the special '.' and '..' directories that are present at the start of all sub-directories.

Invalid parameter

The parameter to a command is generally not valid for that command in some way.

Invalid pathname

A path specified on a command line does not exist or is syntactically incorrect.

Invalid process id

This should not normally occur from commands.

Invalid time

The time typed into the TIME command is not a valid time or was typed in in an invalid format.

Missing parameter

A command expected a parameter but did not find one.

No spare file handles

Should not normally occur from commands.

Not enough memory

Not enough memory is available for the given command. For example, a large program to large to fit into memory or not enough memory for a new environment string.

Not enough memory, system halted

This special error message is printed when MSX-DOS attempts to start up and finds that there is not enough memory to continue. As the message suggests, the computer must then be reset. This should normally occur.

Pathname too long

A path is too long. Either the length of the pathname given exceeds 100 characters, or the total length of a path from the root directory to a file is more than 63 characters.

RAM disk already exists

Should not normally occur from commands.

RAM disk does not exist

The RAMDISK command was used to display the current size of the RAM disk, but no RAM disk exists.

Read only file

An attempt was made to modify or overwrite a file marked as read only. The DIR command shows this, and the ATTRIB command can make it not read only.

Root directory full

The fixed maximum number of files in the root directory (often 64 or 112) has been reached. Directories do not have this limitation.

System file exists

An attempt was made to create a file which would, if it was created, overwrite a file that is marked as a system file. System files are not used in MSX-DOS, and are not shown by the DIR command or accessible from any other commands, and so this error should not normally occur with commands.

Too many parameters

All the parameters a command expected were found on a command line, but there were still more parameters left on the end of the line.

Unrecognized command

A given command was not an internal command or an external COM or BAT command found along the current search path as set by the PATH command.

Wrong version of command

After executing a program, COMMAND2.COM tried to re-load itself from the

COMMAND2.COM file on disk, and found it was not the same version. A prompt is then printed and COMMAND2.COM will attempt to re-load itself again.

Wrong version of MSX-DOS, system halted

This special error message is printed when MSX-DOS attempts to starts up and finds that some other part of the MSX-DOS system has a version number earlier than required. As the message suggests, the computer must then be reset. This should not normally occur.

Internally, errors are represented as error numbers. The numbers corresponding to the errors above start at 255 and decrease. If an error number is received for which there is no message, then it is printed. Numbers above 64 are reserved for future version of MSX-DOS and so are called 'system errors' and numbers below 63 can be used by external commercially available programs and are called 'user errors'. User errors below 32 never print a message. The two default error messages (which will not normally occur from commands) are thus:

System error 64

and

User error 63

where the 64 and 63 are example error numbers. The only command which uses error numbers is the EXIT command. A list of the actual numbers for the above messages is available in the Program Interface specification.

Prompt Messages

There are several situations in which user interaction is required before the system can continue with what it was doing, typically inserting a disk. Also many potentially dangerous commands require confirmation prompts to be answered before they perform their operation. These various system prompts are given below.

All data on drive A: will be destroyed
Press any key to continue...

This prompt is given by the FORMAT command, and is issued to reduce the risk of accidentally formatting the wrong disk. To abort the FORMAT command, control-STOP or control-C can be pressed.

```
Destroy all data on RAM disk (Y/N)?
```

A RAMDISK command was given to set up a RAM disk, but a RAM disk already existed. If the response to the prompt is 'Y', then any files on this existing RAM disk will be destroyed. A response of 'N' or control-STOP or control-C will abort the command.

```
Disk in drive A: will only be able to boot MSX-DOS
Press any key to continue...
```

This prompt is given by the FIXDISK command, and is issued to reduce the risk of accidentally updating a non-MSX-DOS 2 disk. To abort the FIXDISK command, control-STOP or control-C can be pressed.

```
Erase all files (Y/N)?
```

This prompt occurs when a DEL (or ERA or ERASE) command is given specifying all the files in a directory, and is issued to reduce the risk of accidentally deleting a lot of files.

```
Insert COMMAND2.COM disk in drive A:
Press any key to continue...
```

This may occur after running a program, and requires a disk containing COMMAND2.COM in the root directory to be present in the specified drive. After inserting the disk in the drive (which is the drive from which MSX-DOS was originally booted) and pressing a key, the system will continue as normal. If COMMAND2.COM has been copied somewhere else (eg. a RAM disk) then the SHELL environment item can be set up to make COMMAND2.COM re-load from there instead (see chapter 7 on Environment Items).

```
Insert batch file disk in drive A:
Press any key to continue...
```

This may occur during the execution of a batch file, and means that the system needed to read the next command from the batch file but found that the wrong disk was in the drive. After inserting the disk in the specified drive (which will be the drive from which the batch file was originally started) and pressing a key, execution of the batch file will continue as normal.

```
Press any key to continue...
```

This prompt is generally issued when some user interaction is required, and is normally printed after some other message which describes the action required. It is also printed by the PAUSE command. To abort the command that issued the prompt, control-STOP or control-C can be pressed.

Terminate batch file (Y/N)?

When MSX-DOS aborts a command prematurely (such as when the control-STOP or control-C key is pressed) and the command was executing in a batch file, this prompt is issued. If the response is 'Y' then the batch file will also be aborted. If 'N' is the response then the batch file will continue with the ommand that follows
the aborted command.

9. COMMAND SUMMARY

The following is a list of all the standard commands available in MSX-DOS, together with their syntax and purpose.

ASSIGN [d: [d:]]

Sets up the logical to physical translation of drives.

ATDIR +|-H [/H] [/P] compound-filespec

Changes the attributes of directories to make them hidden/not hidden.

ATTRIB +|- R|H [/H] [/P] compound-filespec

Changes the attributes of files to make them hidden/not hidden and read only/not read only.

BASIC [program]

Transfers control to MSX disk BASIC.

BUFFERS [number]

Displays or changes the number of disk buffers in the system.

CD [d:] [path]

Displays or changes the current directory.

CHDIR [d:] [path]

Displays or changes the current directory.

CHKDSK [d:] [/F]

Checks the integrity of the files on the disk.

CLS

Clears the screen.

COMMAND2 [command]

Invokes the command interpreter.

CONCAT [/H] [/P] [/B] [/V] compound-filespec filespec

Concatenates (joins together) files.

COPY [/A] [/H] [/T] [/V] [/P] compound-filespec [filespec]

Copies data from files or devices to other files or devices.

DATE [date]

Displays or sets the current date.

DEL [/H] [/P] compound-filespec

Deletes one or more files.

DIR [/H] [/W] [/P] [compound-filespec]

Displays the names of files on disk.

DISKCOPY [d: [d:]] [/X]

Copies one disk to another.

ECHO [text]

Prints text in a batch file.

ERA [/H] [/P] compound-filespec

Deletes one or more files.

ERASE [/H] [/P] compound-filespec

Deletes one or more files.

EXIT [number]

Exits COMMAND2.COM to the invoking program.

FIXDISK [d:] [/S]

Updates a disk to the full MSX-DOS 2 format.

FORMAT [d:]

Formats (initializes) a disk.

HELP [subject]

Provides on-line help for an MSX-DOS feature.

MD [d:] path

Creates a new sub-directory.

MKDIR [d:] path

Creates a new sub-directory.

MODE number

Changes the number of characters/line on the screen.

MOVE [/H] [/P] compound-filespec [path]

Moves files from one place to another on a disk.

MVDIR [/H] [/P] compound-filespec [path]

Moves directories from one place to another on a disk.

PATH [[+|-] [d:]path [[d:]path [[d:]path ...]]]

Displays or sets the COM and BAT command search path.

PAUSE [comment]

Prompts and waits for a key press in a batch file.

```
RAMDISK [number[K]] [/D]

          Displays or sets the RAM disk size.


RD [/H] [/P] compound-filespec

          Removes one or more sub-directories.


REM [comment]

          Introduces a comment in a batch file.


REN [/H] [/P] compound-filespec filename

          Renames one or more files.



RENAME [/H] [/P] compound-filespec filename

          Renames one or more files.


RMDIR [/H] [/P] compound-filespec

          Removes one or more sub-directories.


RNDIR [/H] [/P] compound-filespec filename

          Renames one or more sub-directories.


SET [name] [separator] [value]

          Displays or sets environment items.


TIME [time]

          Displays or sets the current time.


TYPE [/H] [/P] [/B] compound-filespec | device

          Displays data from a file or device.
```

UNDEL filespec

Recovers a previously deleted file.

VER

Displays the system's version numbers.

VERIFY [ON | OFF]

Displays/sets the current disk write verify state.

VOL [d:] [volname]

Displays or changes the volume name on a disk.

XCOPY [filespec [filespec]] [/A][/E][/H][/M][/P][/S][/T][/V][/W]

Copies files and directories from one disk to another.

XDIR [filespec] [/H]

Lists all files within directories.

10. DISK-BASIC 2.0

Overview

When the system disk (the one which contains MSXDOS2.SYS and COMMAND2.COM) does not exist at the system startup, or when MSX-DOS BASIC command is executed, DISK-BASIC 2.0 will be started.

DISK-BASIC 2.0 is the extended version of previous DISK-BASIC 1.0. Instructions to operate with the RAM disk or the directory has been added or extended.

Description of Commands

CALL CHDIR

Format	CALL CHDIR("[d:][path]")
Purpose	Set or display current directory
Example	CALL CHDIR("WORK")
Description	The function is the same as MSX-DOS CHDIR. See CHDIR.

CALL CHDRV

Format	CALL CHDRV("[d:]")
Purpose	Switch default drive
Example	CALL CHDRV("H:")
Description	The same action is taken as when the drive name is given at MSX-DOS prompt.

CALL MKDIR

Format CALL MKDIR("[d:][path]")

Purpose Create new subdirectory

Example CALL MKDIR("WORK")

Description The function is the same as MSX-DOS MKDIR. See MKDIR.

CALL RMDIR

Format CALL RMDIR("[d:][path]")

Purpose Remove one or more subdirectories

Example CALL RMDIR("WORK")

Description The function is the same as MSX-DOS RMDIR. See RMDIR.

CALL RAMDISK

Format CALL RAMDISK[([number][, variable name])]

Purpose Set size of RAM disk or assign it into a variable

Example CALL RAMDISK(32)
CALL RAMDISK(1000, A)

Description The function is the same as MSX-DOS RAMDISK. See RAMDISK.

CALL SYSTEM

Format CALL SYSTEM
or
CALL SYSTEM[("DOS command name")]

Purpose Pass control back to MSX-DOS

Example CALL SYSTEM("WORK")

Description CALL SYSTEM passes the control back to MSX-DOS. The command name may be given to specify the operation to be executed after the control returns to DOS. If no command name is specified, REBOOT.BAT in the root directory on the boot drive, if any, will be executed.

Format FILES["filename"][, L]

Purpose Display names of files and directories on disk

Example FILES"W*.*"

Description The function is the same as MSX-DOS DIR. See DIR. "FILES, L" displays the names in the long format.

* * * * *