MS-DOS User's Reference

HEWLETT
PACKARD

HP Part No. D1151-90001
Printed in Singapore   July, 1989

Edition 1
E0789

Notice

The information contained in this document is subject to change without notice.

Hewlett-Packard makes no warranty of any kind with regard to this material, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

MS-DOS^R and Microsoft^R are registered trademarks of Microsoft Corporation.

IBM^R is a registered trademark of International Business Machines Coporation.

Personal Computer Group
974 E. Arques Avenue
P.O. Box 486
Sunnyvale, CA 94086, USA

Printing History

July 1989 Printed in Singapore

Contents

3. MS-DOS Command Descriptions

A. How to Use Code Pages

B. The HP Vectra Disk Cache Program

C. MS-DOS Message Directory

Index

MS-DOS: Introduction and New Features

Introduction

MS-DOS stands for "Microsoft^R Disk Operating System." MS-DOS allows you to use literally thousands of applications designed to operate in the MS-DOS environment. MS-DOS provides two vital functions for you:

- **Application Program Support.** MS-DOS is used by applications for basic data file operations and character input.
- **Disk, File, and System Management.** MS-DOS has a full range of commands and utilities to perform the disk and file management operations for your computer.

To install MS-DOS, refer to the installation card that is included in the package.

About This Manual

This is a reference manual written for people who have some experience with personal computers. It is designed for users looking for specific information. If you are a new user, read the "Glossary of MS-DOS Terms and Concepts" chapter, or purchase a book on DOS fundamentals from a book store.

New Features of This MS-DOS Version

MS-DOS 4.0 has several new features over previous versions of MS-DOS, including:

- The **SELECT** command is now a full-screen installation program that takes you through installation and system configuration. SELECT also creates corresponding CONFIG.SYS and AUTOEXEC.BAT files for you.

- A new command called **MEM** has been added to display the amount of used and free memory, a list of allocated and free memory areas, and a list of the programs that are currently loaded.
- The **TREE** command now displays the tree structure of a disk, indenting each directory and using block graphics to make the structure more obvious.
- Confirmation on multiple file deletes. A new parameter for the **DEL** command, **/P**, prompts you before a file is actually deleted.
- Various enhancements to existing commands including: **APPEND, ATTRIB, BACKUP, COUNTRY, DEL, FASTOPEN, FDISK, GRAFTABL, GRAPHICS, MODE, NLSFUNC, REPLACE,** and **SELECT.**
- A new disk cache program called **HP Vectra Disk Cache** has been added to allow your computer to access frequently used data more quickly, without having to read a disk.
- Device driver and file system support for hard disks larger than 32 megabytes.
- A new installable driver called **INDSKBIO.SYS** has been added to provide compatibility with 3.5-inch disks used in a wide variety of early and late model HP personal computers.
- A new installable driver called **HIMEM.SYS** has been added which allows programs to utilize the High Memory Area above the 1 megabyte boundary of base memory.
- The **RAMDRIVE.SYS** driver (formerly called **VDISK.SYS**) now supports virtual drives in expanded (EMS) memory.
- **MODE** and **GRAPHICS** command support for enhanced video adapter modes. MS-DOS 4.0 supports a text mode display interface as well as EGA and VGA display modes. **GRAPHICS** also supports a wider variety of graphics printers including HP PCL printers.
- Improved file system performance and **Expanded Memory** support for **BUFFERS** and **FASTOPEN** system configuration commands. Expanded memory support (EMS) requires the use of a Lotus^{R}/Intel^{R}/Microsoft^{R} (LIM) 4.0 driver.

2

Glossary of MS-DOS Terms and Concepts

This glossary defines MS-DOS terms and concepts for the new MS-DOS user.

Active Drive

The active drive is the drive you are currently working on. The MS-DOS prompt tells you which drive you are currently working on (unless you have changed the MS-DOS prompt with the PROMPT command). For example, if the MS-DOS prompt is

```
C>
```

the active drive is drive C:. If the MS-DOS prompt is

```
A>
```

the active drive is A:.

When you start your computer, the active drive is the drive from which you started your computer. You can change the active drive by typing the name of another drive at the MS-DOS prompt. For example, to change the active drive from drive C: to drive A:, type

```
A:
```

at the C> prompt and press Enter. MS-DOS changes the active drive and the MS-DOS prompt becomes A>.

Note If you do not specify a drive before you specify a directory or a file in an MS-DOS command, MS-DOS assumes the directory or file is on the active drive.

Commands

MS-DOS commands are programs which help you manage your files and computer. In order to use an MS-DOS command, the MS-DOS prompt must be displayed to indicate that MS-DOS is ready to accept and execute a command. The "MS-DOS Command Descriptions" chapter contains detailed information on how to use each command. After you type the command and any additional information it requires, press (Enter) and MS-DOS will execute the command.

Internal Commands

The most commonly used commands reside in system memory. This means that they are always available for use. These commands are part of the file called COMMAND.COM.

External Commands

Other commands are separate programs that are stored on a disk. These commands have one of two file name extensions: .COM or .EXE. Before you can use an external command, MS-DOS must know where the command file is. If you used the SELECT command to install MS-DOS 4.0 and did not change the installation procedure, MS-DOS already knows where the command files are.

If you changed the default installation procedure and installed your MS-DOS 4.0 files on a drive or directory of your choice, add the PATH command to where these files are located to your AUTOEXEC.BAT file. (Refer to the "Batch Processing" chapter for more information about the AUTOEXEC.BAT file.) **Or,** enter the PATH command (with the path to your MS-DOS command files) before you use an external command. **Or,** type the drive and path to the directory in which your MS-DOS command files are stored each time you use an external command. For example, to tell MS-DOS that the command file for the FORMAT command is found in the directory MYDOSDIR on drive D:, type

```
D:\MYDOSDIR\FORMAT A:
```

Directories

Directories allow you to organize the files on a disk, much as you organize papers into folders in a file cabinet. For example, you can organize all your accounting files in one directory, all your BASIC program files in another directory, and so on. A directory can contain a number of files. It can also contain other directories.

You can create a new directory with the MKDIR command, using the same naming conventions for directories as you do for files. (See *files, naming conventions* in this glossary.) You can make a different directory the current directory with the CHDIR command. And you can delete a directory with the RMDIR command.

Root Directory

When you format a disk, MS-DOS automatically creates one directory. This is the **root directory.** MS-DOS represents the root directory with a backslash (\). For example, C:\ is the root directory on drive C: and A:\ is the root directory on drive A:.

There is a limit to the number of files and/or directories that the root directory can contain. This limit is determined by the type of disk and the capacity of the disk. (Refer to the table below.)

Maximum Number of Entries (Files and/or Directories) in Root Directory

Disk Type	Capacity	# of Files and/or Directories
5.25-inch flexible disk	360 Kilobytes	112
3.5-inch flexible disk	720 kilobytes	112
5.25-inch flexible disk	1.2 Megabytes	224
3.5-inch flexible disk	1.44 Megabytes	224
Hard disk	20 Megabytes	512
Hard disk	40 Megabytes	512

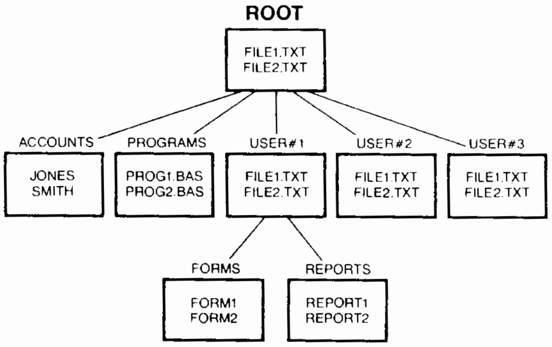
Note The root directory can hold a limited number of entries (files and/or directories). However, there is no limit to the number of files and/or directories that other directories can hold. To maximize the number of files your disk can hold, store your files in directories other than the root directory.

Subdirectories

On disks that contain many files, the root directory may not be sufficient to adequately organize your files. You may create directories within the root directory. If you need to further organize your files, you may create directories, or **subdirectories**, within directories. Unlike the root directory, there is no limit to the number of files and/or subdirectories that a directory can have. You can maximize the number of files that your disk will hold by creating subdirectories. Then store your applications and other files in these directories.

Hierarchical Directory Structure

The root directory may be thought of as the topmost level of directories on your disk. If you create a directory in the root directory, the new directory may be thought of as the second level. If you create subdirectories within these directories, they may be thought of as the third level, and so on. This is what is called a hierarchical directory structure.



Hierarchical Directory Structure

Current Directory

The directory that you are currently working in is the current directory. To work in another directory, use the CHDIR command.

Files

A file on a disk is very much like a file in a filing cabinet. A file contains information and it has a name.

Each time you use your computer to write a letter or create a spreadsheet, your computer holds the text or data in internal memory while you are working on it. When you turn your computer off, the information in internal memory is lost. To save your information, you must save it as a file on a disk. Disks hold information whether your computer's power is on or not.

If you save a file on a disk, you can later have your computer copy the file back into its internal memory. Then you can add information to the file, change the information that is there, or delete information. When you're finished, you can save the new version of the file. If you want to keep both the old version and the new version of the file, you can give the new version a new file name.

Data Files

Data files store information. Your word processing document is a data file. So is your spreadsheet.

Program Files

A program file stores a set of instructions that tells the computer how to perform a certain task. Your word processing software and spreadsheet application are examples of program files.

System Configuration Files

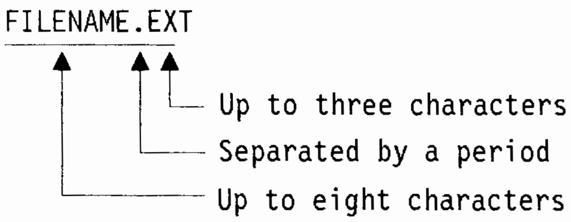
There two files which you can use to set up your computer's configurations. These files are called CONFIG.SYS and AUTOEXEC.BAT and they normally run each time you start or restart your computer.

The CONFIG.SYS file sets specific configurations for your computer to enhance its operation. It allows you to adjust the size of your MS-DOS command environment, add support for new devices, and speed up access to files used by MS-DOS or your application programs, and more. To find out more about this file and the commands you can include in it, refer to the "System Configuration" chapter.

The AUTOEXEC.BAT is a batch file that can run several commands automatically. This is helpful because it means that you won't have to manually run these commands when you need them. To find out more about the commands which can be placed in this file, refer to the "MS-DOS Command Descriptions" chapter. To find out more about the AUTOEXEC.BAT file, refer to the "Batch Processing" chapter.

File Names

All files must have names. A name consists of two parts: a file name and an optional file name extension. The file name can be up to eight characters in length, the file name extension can be up to three characters. The file name and extension must be separated by a period (.). A file name can't consist of an extension only (such as .BAK): If you use an extension, you must also use a file name.



MS-DOS File Names

The following table lists the **valid** (those you may use) and **invalid** (those you may not use) characters for MS-DOS file names and extensions.

Valid and Invalid Characters for File Names and Extensions

Valid Characters	Description	Invalid Characters	Description
A - Z	Alpha characters	.	Period
a - z	Alpha characters	[ ]	Brackets
0 - 9	Numeric characters	?	Question mark
$	Dollar sign	\	Backslash
&	Ampersand	/	Forward slash
#	Pound sign	=	Equal sign
%	Percent sign	"	Quote
’	Apostrophe	,	Comma
!	Exclamation point	+	Plus Sign
( )	Parentheses	*	Asterisk
-	Hyphen	:	Colon
_	Underscore	;	Semicolon
@	“At” sign	⊔	Space
^	Carat	< >	Angle Brackets
{ }	Braces	|	Vertical Bar
~	Tilde		
’	Single quotation		

In addition, there are “reserved” file names and extensions that MS-DOS uses. You **cannot** use these file names and extensions unless you have a specific purpose for doing so. If you use a reserved file name or extension improperly, MS-DOS will ignore it or display an error message.

Note

Some application programs also have file names and extensions that you should not use. Be sure to check the manuals that come with your application programs for additional reserved file names and extensions.

The following table lists the file names you **cannot** use. These are called device names and MS-DOS uses these names for determining input and output. You can use device names with some MS-DOS commands to direct where you want input and/or output to go.

MS-DOS Reserved File Names

File Name	Function
AUX	Auxiliary device.
CLOCK$	This is a special purpose device MS-DOS uses to read or set the system time and date. Unlike the other device names, CLOCK$ cannot be used in an MS-DOS command.
COM1	Serial Port 1.
COM2	Serial Port 2.
COM3	Serial Port 3.
COM4	Serial Port 4.
CON	Console. This device name represents the keyboard for input and the screen for output. If used as an input file name in an MS-DOS command, press F6, then Enter to signify the end of input.
LPT1	Parallel Port 1. This device supports output only.
LPT2	Parallel Port 2. This device supports output only.
LPT3	Parallel Port 3. This device supports output only.
NUL	Dummy device. When used as an input device, it immediately returns an end-of-file condition. As an output device, it accepts data, but immediately discards it (that is, no data is written).
PRN	Printing Device.

The following table lists the extensions you **should not** use unless you have a specific reason for doing so. (For example, if you are creating a batch file you will want its file extension to be .BAT.)

MS-DOS Reserved Extensions

Extension	Function
.BAK	Backup file created by Edlin (the MS-DOS editor) and several word processing programs.
.BAT	MS-DOS uses this extension for all batch files.
.CHK	The extension assigned files that are recovered with the CHKDSK command.
.COM	MS-DOS program file.
.EXE	MS-DOS program file.
.MAP	The default extension for list files created by the Microsoft 8086 Object Linker.
.MSG	Applications from Hewlett-Packard use this extension for message files.
.OVL	MS-DOS uses this extension for overlay files.
.REC	MS-DOS assigns this extension for files that are recovered with the RECOVER command.
.SYS	This extension is typically used for device drivers.
.$$$	MS-DOS uses this extension for temporary files.

File Allocation Tables (FATs)

MS-DOS creates and maintains File Allocation Tables (FATs) to locate the files on a disk. MS-DOS allocates disk space to files in units called **clusters.** The size of a cluster is dependent on the size of the disk; however, a cluster is usually 1 to 8 kilobytes. If a file uses all the space in a cluster, MS-DOS allocates another cluster to the file. The clusters allocated to a single file may not be adjacent on the disk. The FAT keeps track of all the clusters allocated to each file. Without the FAT, MS-DOS cannot find all the information in each file.

The FAT is critically important to the files on a disk. Therefore, MS-DOS creates and maintains two copies of the FAT for each disk. MS-DOS also provides a command to check the FATs on a disk: This is the CHKDSK command. When you use the CHKDSK command, MS-DOS reports any problems with the FATs on the disk and attempts to fix them. As a general rule, it's a good idea to use the CHKDSK command periodically to make sure that the FATs are intact.

MS-DOS Prompt (Command Prompt)

You can enter MS-DOS commands only when the MS-DOS prompt (sometimes called the **command prompt**) is displayed on the screen. The MS-DOS prompt tells you which drive you are currently working on (unless you have changed the MS-DOS prompt with the PROMPT command). For example, if the MS-DOS prompt is

```
C>
```

the active drive is drive C:.

You can use the PROMPT command to modify the MS-DOS prompt. You might change the prompt to include the time, date, current directory, or the MS-DOS version number, for example.

Path

When you use a hierarchical directory structure, you must tell MS-DOS where the directories and files are located in the directory structure. You use a **path** to do this. The path is the route MS-DOS must travel to reach a specific directory or file. Usually, you begin a path with the name of the drive. Next, you designate the root directory with a backslash (\). Then list each directory on the route, in order, and separate each directory with a backslash. For example, the path

```
C:\USER#1\REPORTS\REPORT1
```

tells MS-DOS to go to the root directory of drive C: (C:\), then to the USER#1 directory, then to the REPORTS subdirectory in order to find the file called REPORT1.

If the directory or file is located in a directory at a level below your current position in the hierarchical directory structure, you need only specify the route from your current position. For example, if you are currently working in the USER#1 directory of drive C:, the path to the file called REPORT1 is

```
REPORTS\REPORT1
```

There is no limit to the number of levels of directories you can create in a hierarchical directory structure. However, a path has a limit of 64 characters, counted from the root directory. This limit exists even if you do not need to specify the entire path from the root directory. If you use a path which is more than 64 characters, MS-DOS will display this error message:

```
Invalid Directory
```

Wildcard Characters

There are two wildcard characters: the question mark (?) and the asterisk (*). Wildcard characters allow you to specify more than one file in an MS-DOS command without typing the name of each file.

You can use the ? wildcard character in an MS-DOS command as a substitute for any single valid character in a file name or extension. For example, you can use CH?.TXT to specify the files CH1.TXT, CH2.TXT, and CH3.TXT.

You can use the * wildcard character in an MS-DOS command as a substitute one or more of the characters in a file name or extension. For example, you can use AUG*.MEM to specify the files AUG8.MEM and AUG12.MEM. You can use *.* to specify all the files in a directory.

3

MS-DOS Command Descriptions

Introduction

This chapter contains descriptions of all MS-DOS commands, listed in alphabetical order. Use this chapter as a reference. Each command description contains the following information:

Syntax The way you must type the command, including all of the required and optional parameters.

Operation A complete explanation of what the command will do.

Examples Examples of the more common uses of the command.

Notes Additional information about the command.

Command Syntax Conventions

Many commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the command and its parameters.

This manual uses the following conventions for command syntax:

CAPS Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that you should type the name of your file.

... An ellipsis indicates that you can repeat the parameters as many times as necessary.

Enter all punctuation (except the ellipsis), such as commas, colons and equal signs, exactly as shown in the syntax.

Information Common to All MS-DOS Commands

The following information is common to all MS-DOS commands:

- MS-DOS executes the command when you press Enter.
- The backspace key Backspace is a delete key. If you make an error while entering a command, simply backspace over the incorrect character(s). Always read the syntax of a command carefully before pressing Enter.
- Commands can be followed by one or more parameters.
- Commands and parameters must be separated by delimiters. Unless otherwise specified, you must use spaces to separate commands from their parameters, for example:

  ```
  COPY FILE1.OLD FILE1.NEW
  ```

  With some commands, you may use a semicolon (;), an equal sign (=), or a tab (Tab) as a delimiter. The space is the delimiter used in this manual.
- When MS-DOS commands instruct you to `Press any key when ready ...` , you can press any alphanumeric key (A-Z or 0-9).
- When referring to a file, if its file name includes an extension, you must specify the file name and extension with the command.
- You cannot use wildcard characters in command names.
- You can abort a command while it is executing by holding down Ctrl and then pressing C.

APPEND

The APPEND command sets search path(s) to one or more directories for data files. It also displays and deletes appended search paths. APPEND is an **external** command.

Syntax

First use only:

```
APPEND /X /E
```

To specify directories to be searched:

```
APPEND drive1:\path1;drive2:\path2 ...
```

To delete appended paths:

```
APPEND;
```

To turn ON or OFF /X processing with the ATTRIB, BACKUP, DIR, REPLACE, RESTORE and XCOPY commands for a specified path:

```
APPEND path /X:ON /PATH:ON
```

or

```
APPEND path /X:OFF /PATH:OFF
```

Parameter **Description**

`/X` or `/X:ON` extends the number of MS-DOS functions that use the appended search paths. It allows the executable files that use Exec, Search First, and Find First operations to use appended search paths. Without /X, MS-DOS will only search for data files in appended search paths if Open File, Open Handle, or Get File Size is used. /X is only valid the first time you use the APPEND command after starting your system.

`/X:OFF` turns off the ability for executable files that use Exec, Search First and Find First operations to use appended search paths.

`/E` causes appended search paths to be stored in the MS-DOS environment. Search paths stored in the MS-DOS environment can be displayed and changed by using either the APPEND or SET command. Without /E, you can only display and change search paths by using the APPEND command. /E is only valid the first time you use the APPEND command after starting your system.

drive1: drive that contains the first search path.

path1 first directory MS-DOS searches for a file.

drive2: drive that contains the second search path.

path2 second directory MS-DOS searches for a file.

path directory MS-DOS searches.

`/PATH:ON` tells MS-DOS to process files at specified drive and/or path.

`/PATH:OFF` tell MS-DOS **not** to process files at specified drive and/or path.

Operation

The APPEND command is similar in operation to the PATH command. The PATH command allows you to specify a list of directories to be searched for external MS-DOS commands, program files, and batch files. The APPEND command allows you to specify a list of directories to be searched for data files. Then, MS-DOS automatically searches this list when it is unable to find a data file in the current directory. Each directory in the list is an **appended search path.** You can set one or more search paths with the APPEND command.

MS-DOS only searches one directory at a time. It always starts by searching the current directory. Then, it searches the directories specified by the APPEND command in the order they appeared when you typed the command. Once MS-DOS finds the data file it's looking for, it stops the search. It doesn't search the remaining directories specified in the command.

The APPEND command should be used at the beginning of a work session (i.e., just after you start your system). This sets the search paths for your entire work session. MS-DOS uses these search paths until you end your work session (turn your system off or restart it) or issue another APPEND command.

Examples

To do this:	Type this:
Display appended search paths	`APPEND`
Set the root directory and one subdirectory (USER1) on drive C: as search paths	`APPEND C:\;C:\USER1`
Process executable files (those that use Exec, Find First and Search First) in appended directory (USER1)	`APPEND C:\USER1 /X`
Suspend /X processing with ATTRIB, BACKUP, DIR, REPLACE, RESTORE and XCOPY commands on subdirectory (USER1) of drive C:	`APPEND C:\USER1 /X:OFF PATH:OFF`
Cancel existing appended search paths	`APPEND;`

Note that the search path to the root directory and each subdirectory must be individually specified.

Notes

1. You can use the APPEND command across a network to locate remote data files.
2. If you want to use the APPEND command and the ASSIGN command, you must use the APPEND command first.
3. If you specify the /X parameter and you want to use the BACKUP or RESTORE command, you must cancel the existing appended paths first.

ASSIGN

The ASSIGN command reassigns all disk read and write requests from one drive to another drive. ASSIGN is an **external** command.

Syntax

```
ASSIGN [illegible] ve2 . . .
```

Parameter	Description
drive1	drive to which read and write requests are currently sent. These requests are to be reassigned to another drive.
drive2	drive to which read and write requests are to be sent.

Operation

The ASSIGN command is used to instruct MS-DOS to reassign all disk operations from one disk drive to another. This is useful if you have an application program that was originally designed to run only on specific drives (for example, A: and B:), yet you want to run the application program on another drive (for example, C:).

Examples

To do this:	Type this:
Redirect all disk read/write requests for drive A: to drive C:	`ASSIGN A=C`
Redirect the disk operations from both drives A: and B: to drive C:	`ASSIGN A=C B=C`
Cancel all drive reassignments	`ASSIGN`

Notice that the above examples do not use a colon after the drive name. If you enter the colon, the following error message is displayed:

```
Invalid parameter
```

Notes

1. Use the ASSIGN command for application programs only. Also, cancel all drive reassignments before you use the following commands: BACKUP, JOIN, LABEL, PRINT, RESTORE and SUBST.
2. The DISKCOMP and DISKCOPY commands ignore any drive reassignments.
3. The FORMAT command issues an error message if you attempt to format an assigned drive.
4. Only resident drives can be assigned. For example, on a system with two flexible drives (A: and B:), a hard drive (C:), and a virtual drive created by RAMDRIVE (D:), you can assign drives A: through D:. However, if you attempt to assign drive E:, ASSIGN issues an error message.

ATTRIB (Attribute)

The ATTRIB command sets or clears the read-only and archive attributes for one or more files. You can also use the ATTRIB command to determine the current setting of the read-only and archive attributes for a specific file or set of files. ATTRIB is an **external** command.

Syntax

ATTRIB [illegible] *file* /S

Parameter	**Description**
+R	sets the read-only attribute of a file.
-R	clears the read-only attribute.
+A	sets the archive attribute of a file.
-A	clears the archive attribute.
drive:	drive that contains the file to be changed.
path	path to the file to be changed.
file	the file to be changed.
/S	processes all files in the specified directory, plus its subdirectories.

Operation

The ATTRIB command allows you to mark a file as **read-only.** A read-only file can't be changed (modified) or deleted. It also allows you to mark a file for archiving by other commands, such as BACKUP and XCOPY.

The ATTRIB command without the A and R parameters displays the current state of the read-only and archive attributes for the specified file. To determine the state of a group of files, use wildcards. For example:

```
ATTRIB *.TXT
```

results in the following listing:

```
R A   C:\FILE1.TXT
      C:\FILE2.TXT
R     C:\FILE3.TXT
  A   C:\FILE4.TXT
```

In the above example, R = Read-only, and A = Archive. To mark or clear attributes on a group of files, use wildcards and the /S parameter.

Examples

To do this:	Type this:
Display the attributes of a file called FILE1.TXT in the current directory	`ATTRIB FILE1.TXT`
Protect file FILE1.TXT	`ATTRIB +R FILE1.TXT`
Return the file to its original state so it can be changed or deleted	`ATTRIB -R FILE1.TXT`
Protect the file and mark it as archived	`ATTRIB +R +A FILE1.TXT`
Mark all the files on drive C: as archived	`ATTRIB +A C:\*.* /S`

Notes

1. If you use the COPY command to copy read-only files, the read-only attribute isn't passed on to the new files.
2. When using the BACKUP and XCOPY commands with the /M parameter, if the archive attribute is set for a specified file, that file is copied; if it isn't set, that file isn't copied.
3. XCOPY automatically turns off the archive bits of the files on drive C: as it copies them.
4. You can share (access simultaneously) a file marked read-only across a network, even if an application opens the file with read/write access.

BACKUP

The BACKUP command makes backup copies of the files on a disk and stores the backup copies on another disk. BACKUP is an **external** command.

Syntax

BACKUP *drive1:\path1\file1 drive2:* /S /M /A /D:*date* /T:*time*
/L:*drive:\path\file* /F:*size*

Parameter	**Description**
drive1:	source drive. Contains the file(s) to be backed up.
path1	path to the file(s) to be backed up.
file1	name(s) of the file(s) to be backed up. If you don't specify a file name, MS-DOS backs up all the files in the specified directory.
drive2:	target drive to contain the backed up files. The source drive and the target drive cannot be the same.
/S	backs up the files in all subdirectories of the specified directory, in addition to the files in the specified directory.
/M	backs up only those files that have been modified since the last BACKUP occurred.
/A	adds the files to be backed up to those files already present on the last backup disk in a series. It does not erase any old files that are already present on the backup disk. Whether you specify /A alone or with another parameter, you must insert the **last** backup disk in the series of back up disks. If you do not specify /A, MS-DOS erases all files in the root directory of the target (flexible) disk.

/D:*date* backs up only those files that have been modified on or after the specified date. The date format is dependent on the country code you select using the COUNTRY command in the CONFIG.SYS file. *mm-dd-yy* is the date format for the U.S.

/T:*time* backs up only those files that have been modified on or after the specified time. The time format is dependent on the country code you select using the COUNTRY command in the CONFIG.SYS file. *hh:mm:ss* is the time format for the U.S.

/L or
/L:*drive\path\file* creates a log file with backup log entries. If you don't specify a file name with this parameter, BACKUP creates a file called BACKUP.LOG in the root directory of the source drive. If the log file already exists, MS-DOS appends the log entries to it.

/F:*size* formats the flexible disk in the target drive if it hasn't been formatted. To use this parameter, the FORMAT command must be located in the current directory (or accessible via the PATH command), and the capacities of the target diskette and drive must be identical. Possible values for *size* are listed below:

Size	***size* (Select One)**
160 KB single-sided 5.25-inch disk	160, 160K, 160KB
180 KB single-sided 5.25-inch disk	180, 180K, 180KB
320 KB double-sided 5.25-inch disk	320, 320K, 320KB
360 KB double-sided 5.25-inch disk	360, 360K, 360KB
720 KB double-sided 3.5-inch disk	720, 720K, 720KB
1.2 MB double-sided 5.25-inch disk	1200, 1200K, 1200KB, 1.2, 1.2M, 1.2MB
1.44 MB double-sided 3.5-inch disk	1440, 1440K, 1440KB, 1.44, 1.44M, 1.44MB

Operation

The BACKUP command backs up files from one disk to another. It displays the name of each file as it is being backed up. If you have a lot of files to back up, BACKUP will prompt you for a series of disks. Each backup disk should be labeled and numbered consecutively to help you to properly restore the backed-up files later.

If the target drive is a hard disk, the backup files are automatically placed in a directory called \BACKUP. If the target drive is a flexible disk, the backup files are automatically placed in the root directory: this means that **all files currently in the root directory of the flexible disk will be erased.**

BACKUP creates a BACKUP.LOG file which you can use to find out:

- the date time of the last backup,
- the names of each file and the number of the backup disk on which each file resides.

This means that you can restore a particular file on a particular disk.

The BACKUP command is not the same as the COPY command. The COPY command makes an exact duplicate of a file. The BACKUP command includes control data in each file. The RESTORE command uses this control data. Therefore, you cannot use backed-up files as data files until they have been restored. Refer to the RESTORE command description in this chapter for further information.

The BACKUP command returns several exit codes. The following table lists these codes and their descriptions.

Exit Code	Description
0	Normal Completion
1	No files were found to back up
2	Some files not backed up due to file-sharing conflicts
3	Terminated by user (Ctrl) (Break)
4	Terminated due to error

These error codes can be used with the ERRORLEVEL parameter of the batch processing IF command. Refer to the "Batch Processing" chapter for additional information.

Examples

To do this:	Type this:
To back up the entire contents of hard disk drive C: (including subdirectories) to flexible disks in drive A:	`BACKUP C:\ A: /S`
Back up files from the current directory of the hard disk to flexible disk(s) in drive A:	`BACKUP C: A:`
Back up all files with the file name extension of .TXT in the root directory of drive C: to drive A:	`BACKUP C:\*.TXT A:`
Back up a single file (LETTERS.TXT)	`BACKUP C:\LETTERS.TXT A:`
Back up **only** files that have been modified since the last BACKUP (also called **incremental** backup)	`BACKUP C: A: /M`
Add files being backed up to the files already on the **last** disk in the series of backup disks (so that new files will be added to, and not erase, old files)	`BACKUP C: A: /A`
Back up only files that have been modified on or after the specified date *mm-dd-yy* (another form of an incremental backup)	`BACKUP C: A: /D:05/12/89`
Back up **and** format the disk in the target drive if it has never been formatted before (you must match disk and drive capacities, example given is for a 1.2 MB drive)	`BACKUP C: A: /F:1.2MB`

Notes

1. If the source drive is a flexible disk, the flexible disk cannot be write-protected because BACKUP needs to mark the files as being backed up (/M).
2. The archive attribute, which can be set with the ATTRIB command, affects the /M parameter of the BACKUP command. If the archive attribute is set for a specified file, that file is backed up; if it isn't set, that file isn't backed up.
3. You cannot use an old version of the RESTORE command (MS-DOS 3.xx or earlier) with files backed up with the MS-DOS 4.0 version of the BACKUP command.
4. **Do not** use the BACKUP command on a drive that has been ASSIGNed, JOINed, or SUBSTituted.

BREAK

The BREAK command extends Ctrl Break and Ctrl C checking to include all MS-DOS operations. (Normally, MS-DOS only checks for a Ctrl Break or Ctrl C during keyboard input, screen output, printer output, and auxiliary input and output.) BREAK is an **internal** command.

Syntax

```
BREAK ON
```

or

```
BREAK OFF
```

Parameter	**Description**
ON	causes MS-DOS to check for Ctrl Break and Ctrl C during all MS-DOS operations.
OFF	causes MS-DOS to check for Ctrl Break and Ctrl C only when it is performing keyboard, screen, printer, or auxiliary input and output operations. OFF is the default setting.

Operation

When BREAK is set to ON, Ctrl Break and Ctrl C can be used to stop the execution of an MS-DOS command or an application program that performs few or no keyboard, screen, printer, or auxiliary input and output operations.

Examples

To do this:	Type this:
Set extended checking	`BREAK ON`
Turn off extended checking	`BREAK OFF`
Determine current status	`BREAK`

Notes

1. When BREAK is set to ON, system performance may be slower due to frequent checking. Therefore, a setting of OFF is recommended.
2. The BREAK command can be included anywhere in the system configuration file (CONFIG.SYS).
3. Some programs bypass MS-DOS for character and disk operations. When these programs are running, a Ctrl Break or Ctrl C might not interrupt them, even though you have issued the BREAK ON command.

CHCP (Change Code Page)

The CHCP command displays or changes the active system code page. This is the code page to be used by MS-DOS for all devices prepared for code page switching. To change the code page for a specific device, use the MODE command. CHCP is an **internal** command.

Syntax

```
CHCP
```

Parameter	Description
nnn	the new active system code page. Possible values are: 437 (United States) 850 (Multilingual) 860 (Portuguese) 863 (French-Canadian) 865 (Nordic) If you omit this parameter, MS-DOS displays the current active system code page.

Operation

The CHCP command allows you to change the **active** system code page to one of the **prepared** system code pages defined by the COUNTRY command (for additional information on COUNTRY, refer to the "System Configuration" chapter).

If you select a system code page that hasn't been prepared, an error message similar to the following is displayed:

```
Code page 865 not prepared for system
```

If a device (keyboard, display screen, or printer) hasn't been prepared for the active system code page, a message similar to the following is displayed:

```
Code page 850 not prepared for device xxx
```

Any program that you run after starting a new code page will use the new code page. Programs that started before the new code page will use the original code page.

You must execute the NLSFUNC command before you execute CHCP.

For additional information on using code pages, refer to the "How to Use Code Pages" appendix.

Examples

To do this:	Type this:
Select code page 850 (Multilingual) as the active system code page	`CHCP 850`
Display the active system code page	`CHCP`

CHDIR (Change Directory)

The CHDIR (or CD) command changes the current directory on the active or specified drive. You can also use the CHDIR command to display the path to the current directory. CHDIR is an **internal** command.

Syntax

CHDIR *path*

or

CD [illegible]

Parameter	Description
path	is the path to the new directory.

Operation

The CHDIR command is used to display or change the current directory on the active drive. Including the optional path with the command changes you from one directory on the active drive to another directory. CHDIR without parameters displays the current directory.

CHDIR has two shortcuts. You can type `CD` instead of `CHDIR` to change directories. And, you can type `CD ..` to move up **one directory at a time** from any directory but the root directory.

Examples

To do this:	Type this:

Find out what the current directory is	`CHDIR`
	or
	`CD`

Change the current directory from the root directory to REPORTS (which is a subdirectory of USER1)	`CD \USER1\REPORTS`

From the REPORTS directory, move up **one** directory to USER1	`CD \USER1`
	or
	`CD ..`

Change from the USER1 directory back to REPORTS	`CD REPORTS`
Return to the root directory from any directory or subdirectory	`CD \`

CHKDSK (Check Disk)

The CHKDSK command tests the integrity of a disk or certain file(s) on a disk. Specifically, it scans the directory of the specified disk, checks it for consistency, and displays a status report. CHKDSK also reports information about files on the disk and system memory. Finally, if errors are encountered, CHKDSK can be used to correct some, but not all, of them. CHKDSK is an **external** command.

Syntax

```
CHKDSK
```

Parameter	**Description**
drive:	drive that contains the disk or file(s) to be checked.
path	path to the file(s) to be checked.
file	file(s) to be checked.
/F	instructs MS-DOS to fix any errors, if possible.
/V	instructs MS-DOS to display the names of all files and subdirectories.

Operation

Use the CHKDSK command to determine the status of a disk or file(s). If you specify a drive, CHKDSK cross-checks the root directory, all subdirectories and the two copies of the File Allocation Table (FAT). If any errors are found, MS-DOS displays an error message. The error messages, along with the recommended steps to correct them, are listed in the appendix "MS-DOS Message Directory."

When CHKDSK completes its inspection, it displays several statistics about the disk, its files, and system memory. An example of this listing is shown below. (This is only a representative listing of a CHKDSK display. The entries for volume, hidden files, directories, user files, and bad sectors won't be listed if they don't exist on the disk.)

```
Volume HARDDISK      Created AUG 12, 1988 10:00

10592256 bytes total disk space
   40960 bytes in 2 hidden files
   45056 bytes in 11 directories
 4435968 bytes in 391 user files
   24576 bytes in bad sectors
 6045696 bytes available on disk

  655360 bytes total memory
  220000 bytes free
```

When CHKDSK encounters lost clusters, it asks you if you want them recovered. If you entered CHKDSK with the /F parameter, entering `Y` (Yes) instructs CHKDSK to place each recovered allocation chain (one or more clusters) in a file with the name FILE*nnnn*.CHK. (If you did not enter CHKDSK with the /F parameter, CHKDSK will not recover lost clusters.)

The *nnnn* portion of the file name is a number starting with 0000. MS-DOS places these files in the root directory. You can look at them to see if the information recovered is of any value, and erase them if it is not.

If you specify a file name instead of a disk drive, CHKDSK determines how the file is stored on disk. In addition to the information shown above, it reports the number of non-contiguous blocks contained in the file. You'll see the message:

```
All specified file(s) are contiguous
```

or

file `contains` *n* `non-contiguous blocks`

This information is important because system performance is degraded if the number of fragmented (non-contiguous) files on a disk (particularly a hard disk) grows too large.

You can use wildcards in the file name to determine the status of more than one file at a time. However, CHKDSK only examines the files in one directory—either the current directory or the one specified with the optional path.

Examples

To do this:	Type this:
Check drive C: and correct any errors found, if possible	`CHKDSK C: /F`
Instruct CHKDSK to display the names of all the files in the root directory and subdirectories of drive C:	`CHKDSK C: /V`
Check the current drive for fragmented files	`CHKDSK`

Notes

1. CHKDSK won't allow you to specify a path without a file name in the command because it attempts to interpret the path as a file. For example, if you want to determine the amount of fragmentation in the files contained in the subdirectory USER1, do not enter the command:

   ```
   CHKDSK C:\USER1
   ```

   CHKDSK incorrectly interprets this command to mean, "report on the file USER1 in the root directory" and displays the error message:

   ```
   File not found
   ```

   Instead, enter the command line:

   ```
   CHKDSK C:\USER1\*.*
   ```

   and CHKDSK executes correctly.

2. You can redirect CHKDSK's output to a device or file for later use. However, you can't redirect the output and use the /F parameter simultaneously. See the chapter "Redirecting Input and Output" for more information on redirecting command output.

CLS (Clear Screen)

The CLS command moves the cursor to the upper left corner of the display screen and then clears the entire screen. CLS is an **internal** command.

Syntax

```
CLS
```

COMMAND

This command loads the MS-DOS command processor. COMMAND.COM can also be used to load an additional command processor, into system memory. COMMAND is an **external** command.

Syntax

```
COMMAND                  [illegible] /C [illegible]
```

Parameter	Description
drive:	is the drive that contains the COMMAND.COM file if MS-DOS needs to reload the transient part into memory.
path	is the path to the COMMAND.COM file if MS-DOS needs to reload the transient part into memory.
ctty-dev	allows you to specify a different device (such as AUX) for input and output.
/P	instructs MS-DOS to make the new version of COMMAND.COM permanent. This version can only be removed by restarting your system.
/E:*nnnnn*	instructs MS-DOS to set the size in bytes of the MS-DOS environment. *nnnnn* is a base 10 integer from 160 to 32768 that is rounded up to the nearest paragraph boundary. The default value is 160.
/C *string*	tells the new version of the command processor to execute the command or commands specified by *string* following the /C parameter, then return to the primary version of the command processor. If you enter both the /C and /P parameters, MS-DOS ignores the /P parameter.

Operation

COMMAND.COM is the top-level command processor for MS-DOS. If sufficient memory is available, you can use the COMMAND command to load an additional (or second version) of the command processor if you want to create a customized environment.

When you start a new command processor, you also create a new command environment. This new environment is a copy of the old, parent environment. However, you can change the new environment without affecting the old one. If the SET command is used to change the MS-DOS environment, only the MS-DOS environment used by the new command processor is affected. The MS-DOS environment used by the old command processor remains unchanged. Then, when you EXIT the new command processor and return to the old, the environment used by the old command processor resumes operation.

The command processor is loaded into memory in two parts: a transient part and a resident part. Some applications write over the transient memory part of COMMAND.COM when they run. When this happens, the resident part of the command processor looks for the COMMAND.COM file on disk so that it can reload the transient part. *Drive:* and *path* tell the command processor where to look for the COMMAND.COM file if it needs to reload the transient part into memory.

Examples

To do this:	Type this:
Load a second version of COMMAND.COM	`COMMAND`
Load a second version of COMMAND.COM and make it permanent.	`COMMAND /P`
Load a second command processor, execute the DIR command on drive C: and then exit back to primary command processor	`COMMAND /C DIR C:`

To do this:	Type this:
With COMMAND.COM located in a directory called DOS, put a line into CONFIG.SYS to increase the command environment space to 256 bytes	`SHELL=C:\DOS\COMMAND.COM /P /E:256`

Notes

1. If you use the /C parameter, it must be the last parameter specified in the command (after *ctty-dev,* /P or /E:*nnnnn*).

2. COMMAND can be used in a batch file to avoid **chaining** (refer to the chapter "Batch Processing" for further information on chaining batch files). For example, if you create a batch file called BATCH1.BAT that contains the following information:

   ```
   \COMMAND /C BATCH2.BAT
   \COMMAND /C BATCH3.BAT
   ```

   a number of things happen when you execute this batch file. First, MS-DOS loads the new command processor. The new command processor executes BATCH2.BAT and exits to the old command processor. Then, MS-DOS loads the new command processor again. It executes BATCH3.BAT and exits to the old command processor for the final time.

3. If you use the /P parameter, you cannot use the EXIT command to return to the old command processor. The /P parameter also increases the resident size of MS-DOS in memory.

COMP (Compare)

The COMP command compares the contents of two files and reports any differences. COMP is an **external** command.

Syntax

```
COMP [illegible] drive2:\path2\file
```

Parameter **Description**

drive1: is the drive that contains the first of two files to be compared.

path1 is the path to the first of two files to be compared.

file1 is the first of two files to be compared. If the first file name is omitted, COMP assumes *.* (all files in the directory).

drive2: is the drive that contains the second of two files to be compared.

path2 is the path to the second of two files to be compared.

file2 is the second of two files to be compared. If the second file name is omitted, COMP assumes it is the same as *file1*.

Operation

The COMP command compares the files specified in the command, and reports any differences. Wildcards may be used. If you do not specify any drives, paths, or file names, you will be prompted to enter them.

If the files are identical, MS-DOS displays

```
Files compare OK
```

If differences are discovered, MS-DOS displays them in the following format:

```
Compare error at OFFSET xxxxx
File1 = yy
File2 = zz
```

where *xxxxx* is the byte offset from the start of the files, *yy* is the hex value of the byte in the first file, and *zz* is the hex value of the byte in the second file.

After ten differences are found, COMP stops comparing the files, displays the following message:

```
10 Mismatches - ending compare
```

and then displays the prompt:

```
Compare more files (Y/N)?
```

If you enter N, you are returned to the MS-DOS prompt. If you enter Y, COMP prompts you for the names of the new files as follows:

```
Enter primary file name
Enter 2nd file name or drive id
```

After you type a response to these prompts, COMP compares the files.

If the two files do not contain the same number of bytes (as indicated in their respective directory entries), COMP doesn't compare them. Instead, the following error message is displayed:

```
Files are different sizes
```

Examples

To do this:	Type this:
Compare FILE1.TXT in the USER directory of drive C: with FILE2.TXT in the root directory of drive B:	`COMP C:\USER\FILE1.TXT B:\FILE2.TXT`
Compare every file in the root directory of drive A: with the file FILE1.TXT in the directory USER1 on drive C:	`COMP A: C:\USER1\FILE1.TXT`
Compare all files in the current directory with the extension .TXT with all files with the extension .BAK	`COMP *.TXT *.BAK`

Notes

1. If a file isn't terminated with an end-of-file marker (Ctrl Z), MS-DOS displays the following message:

   ```
   EOF mark not found
   ```

   However, COMP still compares the two files.

COPY

The COPY command copies one or more files from one disk or directory to another, or makes a copy of a file in the same directory, but with a different name. The COPY command can also combine two or more files into one file. COPY is an **internal** command.

Syntax

To copy files:

COPY *drive1:\path1\file1 drive2:\path2\file2* /A /B /V

or

COPY *drive1:\path1\file1* /A /B /V *drive2:\path2\file2*

To combine (concatenate) files:

COPY *drive1:\path1\file1* + *drive2:\path2\file2* ... *driveN\pathN\fileN*

Parameter	Description
drive1:	**when copying files,** this is drive that contains the source file. **When combining files,** this is the drive containing the first file to be combined.
path1	**when copying files,** it is the path to the source file. **When combining files,** it is the path to the first file to be combined.
file1	**when copying files,** it is the name of the source file. **When combining files,** it is the name of the first file to be combined.
drive2:	**when copying files,** it is the drive that contains the target file (the file to be copied to). **When combining files,** it is the drive containing the second file to be combined.
path2	**when copying files,** it is the path to the target file. **When combining files,** it is the path to the second file to be combined.
file2	**when copying files,** it is the name of the target file. **When combining files,** it is the name of the second file to be combined.

/A — treats a file like an ASCII file. When you specify /A for a source file, MS-DOS copies the file up to the first end-of-file marker (Ctrl Z). It does not copy the remainder of the file, if any. When you specify /A for a target file, MS-DOS adds an end-of-file marker to the file.

/B — treats a file like a binary file. When you specify /B for a source file, MS-DOS copies the entire file, (including the end-of-file marker) based on the directory file size. When you specify /B for a target file, MS-DOS does not add an end-of-file marker to the file.

/V — verifies the target file as it is written. This parameter performs the same function as the VERIFY command, except that it only works while the COPY command is executing.

driveN — drive to contain the target file.

pathN — path to the target file.

fileN — name of the target file. If the target file name is omitted, the source files are combined under the name of the first file to be combined.

Note

The position of the /A and /B parameters in the command determines the files affected by them. The /A and /B parameters affect the file immediately preceding them and any files following them until MS-DOS encounters another /A or /B.

Operation

Copying a File

The COPY command copies the source file(s) to the desired target (or destination). If the target file name isn't included in the command, COPY assumes the target file has the same name as the source file. However, if you try to make a duplicate of a file with the same file name in the same directory MS-DOS displays the following error message:

```
File cannot be copied onto itself
0 File(s) copied
```

To make a duplicate of a file in the same directory, you must give the target file a new name.

You can use wildcards to copy groups of files.

You can use COPY to transfer data between system devices. You can copy information from a device, such as your console (display), to another device, such as a file on a disk. You can use a valid device name as the source and/or the target file name. For example:

```
COPY CON FILE1.TXT
```

allows you to enter data directly into a file called FILE1.TXT from your console. Type lines one at a time and move to the next line by pressing the Enter. When you are finished typing the data to your display, hold down Ctrl and press Z once, then press Enter to write the data to a file to be called FILE1.TXT on your disk.

Note

You can use the COPY CON form of the COPY command to create batch files. See the "Batch Processing" chapter for an example using COPY CON to create a batch file.

Combining Two or More Files

You can also use COPY to combine (or append) two or more files into one file. For example, you can combine the individual chapters (files) of a book into a new file that contains the entire book.

If you don't want to combine the files into a new file, omit the target file name. Then, the files are appended to the end of the first source file. COPY compares the file name of the source file with the file name of the target file. If they are the same, COPY will skip that one input file and proceed to combining the rest of the files. For example, the following command appends all *.TXT files (except ALL.TXT) to ALL.TXT:

```
COPY ALL.TXT + *.TXT
```

Examples

To do this:	Type this:
Copy the file USER1.TXT in the USER directory of drive C: to the REPORTS directory of drive A:	`COPY C:\USER\USER1.TXT A:\REPORTS`
Copy all files from the current drive and directory to root directory of drive A:	`COPY *.* A:\`
Copy (ASCII) text in FILE1 on the active drive, up to but not including the first end-of-file mark, to drive A: as FILE1.TXT	`COPY FILE1 /A A:FILE1.TXT`
Make a copy of binary file BILLING.ASM, name the new file BILLING2.ASM and do not add an end-of-file marker	`COPY BILLING.ASM /B BILLING2.ASM`
Combine the files INTRO, BODY and SUM to create a single file to be called REPORT	`COPY INTRO+BODY+SUM REPORT`
Append the files LETTER2 and LETTER3 to LETTER1	`COPY LETTER1+LETTER2+LETTER3`
Copy all chapter files (CHAP1.TXT, CHAP2.TXT, and CHAP3.TXT) to the printer for printing	`COPY CHAP?.TXT PRN`

Notes

1. When you use COPY to copy a file, MS-DOS treats the source and target files as binary if no parameter is specified. When you use COPY to combine files, MS-DOS treats the files as ASCII if no parameter is specified.

2. The attributes of the source file (for example, read-only) are not transferred to the target file.

CTTY (Change I/O Device)

The CTTY command allows you to change the standard input and output device (CON) to another valid I/O (input and output) device. CTTY is an **internal** command.

Syntax

CTTY *device*

Parameter	Description
device	is the device you want as the standard I/O device.

Operation

The CTTY command allows any valid character I/O device to be used as the MS-DOS console. The device can be any standard MS-DOS I/O device, such as AUX, COM1, COM2, COM3, and COM4. Or, it can be an optional character device driver installed with the DEVICE command in the CONFIG.SYS file.

Examples

To do this:	Type this:
Move all command input and output from the current device to the device (let's say another terminal) configured as AUX	`CTTY AUX`
Return command input and output to the console	`CTTY CON`

Notes

1. Many programs do not use MS-DOS for standard input and output routines. These programs bypass the software and access (communicate with) the hardware directly, or they use ROM BIOS I/O routines. The CTTY command has no effect on these programs.
2. The assigned device must be a character I/O device. In addition, it must be capable of receiving input and sending output. Therefore, a device such as PRN (a printer) cannot be used as the assigned device because it can't send output to MS-DOS.

DATE

The DATE command sets or displays the system date. DATE is an **internal** command.

Syntax

```
DATE
```

Parameter	**Description**
date	format determined by the COUNTRY command. In the U.S., the date format is *mm-dd-yy,* where: *mm* is a 1- or 2-digit number from 1 to 12 representing the month. *dd* is a 1- or 2-digit number from 1 to 31 representing the day of the month. *yy* is a 2-digit number from 80 to 99 representing the year (century 19 is assumed), or a 4-digit number from 1980 to 2099 representing the year. You can use a hyphen (-), slash (/), or period (.) as a separator.

Operation

The DATE command allows you to set or display the MS-DOS system date. MS-DOS uses the date and time values to record the exact time files are created or last updated. As you build your library of files, you will find this information both necessary and useful.

MS-DOS checks for invalid dates and separators. When using the DATE command, if you enter an invalid date or use an incorrect separator, MS-DOS responds with:

```
Invalid date
Enter new date:
```

It repeats this response until you enter a valid date in the correct format.

The date format may vary if you are using a code page other than the one for the U.S. For more information about international date formats, see the appendix "How to Use Code Pages."

The DATE command also changes the date kept by your computer's internal real-time clock.

Examples

To do this:	Type this:
Set the system date	`DATE 10-14-90`
Display the current system date	`DATE`

Notes

1. If you start your system without PAM, the DOS Shell, or an AUTOEXEC.BAT file, you are prompted to enter the date and time every time you start it. However, if you have an AUTOEXEC.BAT file, you aren't prompted for the date and time unless your AUTOEXEC.BAT file includes the DATE and TIME commands.

DEL (or ERASE)

The DEL (or ERASE) command removes unwanted files from a disk. DEL is an **internal** command.

Syntax

DEL *drive:\path\file* /P

or

ERASE *drive:\path\file* /P

Parameter	**Description**
drive:	drive that contains the file(s) to be deleted.
path	path to the file(s) to be deleted.
file	file(s) to be deleted.
/P	prompts you with "`Delete (Y/N)`" before deletion actually occurs.

Operation

The DEL command deletes one or more files from a disk. You can use wildcards in the file name to delete more than one file. If you want to delete all files in the current directory, MS-DOS prompts you with "`Are you sure?`" If you enter `Y` (for Yes), all the files in the current directory are deleted.

The DEL command removes unwanted files from a directory, but it doesn't remove the directory, itself. To remove a directory, use the RMDIR (Remove Directory) command described in this chapter. Note that a directory must be empty before it can be removed. Thus, if you want to remove a directory, use the DEL command first to delete all of the files in it; then, use the RMDIR command.

Examples

To do this:	Type this:

Delete the file FILE1.TXT in the current directory	`DEL FILE1.TXT`
	or
	`ERASE FILE1.TXT`

Delete all files with the extension of .TXT in the directory named REPORTS on drive C:	`DEL C:\REPORTS\*.TXT`
Delete all files in the current directory	`DEL *.*`
Review all the files in the current directory and tell MS-DOS whether to save or delete each file	`DEL *.* /P`

Notes

1. The DEL command can't be used to delete files with hidden or read-only attributes.

DIR (Show Directory)

The DIR command lists the names of the files in a directory. DIR is an **internal** command.

Syntax

```
DIR
```

Parameter	**Description**
drive:	drive that contains the file name(s) to be displayed.
path	path to the file name(s) to be displayed.
file	file name(s) to be displayed.
`/P`	instructs MS-DOS to pause when the screen is filled. It is also referred to as Page mode.
`/W`	instructs MS-DOS to list file names only, five file names per line. It is also referred to as Wide Display mode.

Operation

Use the DIR command to determine what files are in a certain directory. DIR lists the size of each file in bytes, and the date and time the file was last updated. DIR also lists any subdirectories, (the designation <DIR> appears after the directory name). The listing concludes with the total number of files and directories displayed, and the number of free bytes remaining on the disk. For example, the following command entered from the root directory of drive C:

```
DIR
```

results in a listing similar to the following:

```
Volume in drive C is HARDDISK
Directory of C:\

FILE1    TXT   23210  10-07-88  1:43p
ACCOUNTS    <DIR>     10-12-88  8:03a
PROGRAMS    <DIR>     10-12-88  8:05a
USER#1      <DIR>     10-12-88  8:10a
USER#2      <DIR>     10-12-88  8:11a
USER#3      <DIR>     10-12-88  8:13a
FILE2    TXT    4329  10-13-88  3:45p
      7 File(s)   6348800 bytes free
```

DIR displays files and directories initially in the order they were created or modified. Then as you delete files, new files fill the vacated directory locations, and the directory is no longer in its original date/time order. You can use the SORT command, described later in this chapter, to display a directory in alphabetical order.

The directory listing for a subdirectory has special entries. The following example shows the directory listing for the directory USER1, which is a subdirectory of the root directory. The . entry indicates the current directory, which is directory USER1, and the .. entry indicates its parent directory, which, in this case, is the root directory:

```
Volume in drive C is HARDDISK
Directory of C:\USER1

.           <DIR>      10-12-88  8:10a
..          <DIR>      10-07-88  1:43p
FORMS       <DIR>      10-12-88  8:17a
REPORTS     <DIR>      10-12-88  8:18a
FILE1    TXT     206   10-14-88  3:45p
FILE2    TXT     310   10-17-88  9:28a
 6 file(s)  7518208 bytes free
```

The /P parameter is useful for directories that contain a large number of files. It instructs MS-DOS to pause when a full screen of data have been displayed. MS-DOS displays the next full screen when you press any key.

The /W parameter displays the file names left to right, five per line. Information about file size and date/time of last update is omitted. As with other directory listings, the list concludes with the total number of files and the number of free bytes.

To display particularly large directories, you can use the /P and /W parameters together to produce a listing in wide format that pauses at the end of every page.

Examples

To do this:	Type this:
List all files and subdirectories in the directory USER1 on drive C:	`DIR C:\USER1`

List all files in the current directory with the extension .TXT	`DIR *.TXT`
	or
	`DIR .TXT`

List all files in the current directory with the name LETTER and any extension	`DIR LETTER.*`
	or
	`DIR LETTER`

List all files in the current directory in wide screen format	`DIR /W`

Notes

1. The COUNTRY command in the CONFIG.SYS file determines the date and time display format.
2. MS-DOS does not display hidden files in any directory listing.

DISKCOMP (Disk Compare)

The DISKCOMP command compares the contents of two flexible disks, sector by sector. DISCOMP is an **external** command.

Syntax

```
DISKCOMP [illegible] /8
```

Parameter	Description
drive1:	the first of two flexible disks to be compared.
drive2:	the second of two flexible disks to be compared. If you omit this parameter, DISKCOMP uses the active drive as the second drive.
/1	compares only the first side of the flexible disks, even if they are double-sided disks.
/8	compares only 8 sectors per track, even if the flexible disks have more sectors per track.

Operation

The DISKCOMP command compares the contents of the flexible disk in the first drive with the flexible disk in the second drive on a track-by-track basis. It automatically determines the number of sides and sectors per track based on the format of the source disk. If MS-DOS discovers a compare error, it displays the track and sector containing the mismatch in the following format:

```
Compare error side xx, sector yy
```

where *xx* is the side number (0 or 1) and *yy* is the sector number.

After the compare operation is complete, DISKCOMP asks if you want to compare more disks with the message:

```
Compare more diskettes (Y/N)?
```

If you enter `Y`, you are prompted to insert the new disks. If you enter `N`, you are returned to the MS-DOS prompt.

If the target and source disks have been formatted differently, DISKCOMP displays the following error message:

```
Drive types or diskette types
are not compatible
```

The DISKCOMP command returns the following exit codes:

Exit Code	Description
0	Compared OK The disks compared exactly.
1	Did not compare The disks were not the same.
2	CONTROL+C error The user terminated with Ctrl C.
3	Hard error An unrecoverable read or write error occurred—did not compare.
4	Initialization error There is not enough memory—invalid drives or command syntax.

You can use the batch processing IF command to perform error processing based on the ERRORLEVEL displayed by DISKCOMP.

Examples

To do this:	Type this:
Compare the flexible disk in drive A: with the flexible disk in drive B:	`DISKCOMP A: B:`
Compare two diskettes using a computer that only has one flexible disk drive	`DISKCOMP A:`

Notes

1. DISKCOMP cannot be used on hard or virtual disks.
2. To compare information between 5.25-inch and 3.5-inch disks, use the COMP command.
3. DISKCOMP ignores the volume serial numbers in its comparison.
4. Two disks are usually identical only if one has been made from the other using the DISKCOPY command. If the COPY command has been used to transfer files from one disk to another, the disks probably won't compare. The reason is that COPY doesn't necessarily put data in the same place on the target drive as it is on the source drive. To compare disks made with the COPY command, refer to the FC and COMP commands.
5. DISKCOMP does not work on network drives and cannot be used with ASSIGNed, JOINed, or SUBSTed drives.

DISKCOPY

The DISKCOPY command copies the contents from one flexible disk to another flexible disk. DISKCOPY is an **external** command.

Syntax

```
DISKCOPY
```

Parameter **Description**

drive1: source drive. It contains the source disk (the disk to be copied from).

drive2: target drive. It contains the target disk (the disk to be copied to). If you omit this parameter, DISKCOPY uses the active drive as the target.

`/1` only copies the first side of the disk, whether the disk is single- or double-sided.

Operation

The DISKCOPY command **makes an exact copy of one flexible disk on another flexible disk.** You cannot do a DISKCOPY from a 5.25-inch to a 3.5-inch, or visa versa. If the target disk is not formatted, DISKCOPY formats it to be the same as the source disk (same number of sides and sectors per track as the source disk). If you have only one disk drive, DISKCOPY prompts you to insert the source and target disks at the appropriate times and waits for you to press any key before continuing. When the copy is complete, DISKCOPY asks:

```
Copy another (Y/N)?
```

If you enter `Y`, DISKCOPY instructs you to insert the next set of disks. If you enter `N`, you are returned to the MS-DOS prompt.

If you omit both the *drive* parameters, DISKCOPY performs a single-drive copy on the active drive.

DISKCOPY returns the following exit codes:

Exit Code	Description
0	Copied successfully
1	Non-fatal read/write error An unrecoverable but non-fatal read or write error occurred.
2	CONTROL+C error The user entered Ctrl C to terminate DISKCOPY.
3	Fatal hard error DISKCOPY was unable to read the source disk or format the target disk.
4	Initialization error There is not enough memory - invalid drives or command syntax.

You can use the batch processing IF command to perform error processing based on the ERRORLEVEL displayed by DISKCOPY.

If the source disk has a volume serial number, DISKCOPY create a new volume serial number for the target disk.

Examples

To do this:	Type this:
Make the disk in drive B: an exact copy of the disk in drive A:	`DISKCOPY A: B:`
Make an exact copy of two flexible disks using a computer that has only one flexible disk drive	`DISKCOPY A:`

Notes

1. DISKCOPY cannot be used on hard or virtual disks.
2. DISKCOPY cannot be used with ASSIGNed, JOINed, or SUBSTed drives.
3. If MS-DOS encounters disk errors on either the source or target disk, it reports the track, sector, and size of the error; then, it proceeds with the copy process. The target disk may or may not be usable in this situation. Use DISKCOMP or COMP to verify the state of the target disk.

EXE2BIN (Executable to Binary)

The EXE2BIN command converts files from executable format (.EXE) to binary format (.BIN or .COM). EXE2BIN is an **external** command.

Syntax

EXE2BIN *drive1:\path1\**file1** drive2:\path2\**file2***

Parameter	Description
drive1:	source drive. It contains the source file (the file to be converted).
path1	path to the source file.
file1	name of the source, or input file. You must specify an input file name; however, the file name extension is optional. If you don't specify a file name extension, MS-DOS uses .EXE.
drive2:	target drive. It contains the target file (the converted file).
path2	path to the target file.
file2	name of the target, or output file. If you don't specify an output file name, the input file name is used. If you don't specify a file name extension, MS-DOS uses .BIN.

Operation

Use the EXE2BIN command to convert executable files to binary format. This conversion may save disk space and allow some programs to load more quickly. However, not all .EXE files can be converted.

If your executable program cannot be converted, MS-DOS displays the following message:

```
   File cannot be converted
```

Examples

To do this:	Type this:
Convert the file MYPROG.EXE in the current directory from .EXE format to a file called MYPROG.COM in .COM format	`EXE2BIN MYPROG MYPROG.COM`
Convert MYPROG.EXE in the PROGRAMS directory on drive C: from .EXE format to binary format, then copy it to the root directory of drive A: to a file called NEWPROG with the extension .BIN	`EXE2BIN C:\PROGRAMS\MYPROG A:\NEWPROG`

Notes

1. For more detailed information on creating .EXE programs, refer to Mircosoft's *MS-DOS Programmer's Reference.*

EXIT

The EXIT command exits the MS-DOS command processor (COMMAND.COM) and returns you to the previous command processor, if one exists. EXIT is an **internal** command.

Syntax

```
EXIT
```

Operation

The MS-DOS command processor, COMMAND.COM, can be loaded by PAM or the MS-DOS Shell, an application program, or COMMAND.COM itself. For example, when it's loaded by PAM or the MS-DOS Shell, the MS-DOS command processor appears as the application label "MS-DOS Commands" or "Command Prompt." In order to return to PAM or the MS-DOS Shell, enter:

```
EXIT
```

Notes

1. EXIT doesn't perform any operation if a previous command processor doesn't exist.

FASTOPEN

The FASTOPEN command decreases the amount of time needed to open frequently used files and directories on one or more hard disk drives. FASTOPEN is an **external** command.

Syntax

```
FASTOPEN ( [illegible] n,m) [illegible] /X
```

Parameter	Description
drive1:	first drive on which you want FASTOPEN to track files and directories. You can use FASTOPEN to track files and directories on up to 4 hard disk drives at one time.
n	number of files and directories tracked per drive. Possible values are 10 to 999 (default is 10). When used **without** *m,* do not include the parenthesis and comma, for example: `FASTOPEN C:=25`. When used **with** *m,* include the parentheses and comma, for example: `FASTOPEN C:=(25,10)`
m	number of file extent entries for files on the drive specified. The range is 1 to 999. If omitted, this feature is not provided. When used **with or without** *n,* include the parenthesis and comma, for example: `FASTOPEN C:=(25,10)` or, `FASTOPEN C:=(,10)`
/X	indicates that the FASTOPEN cache will be in (LIM 4.0) expanded memory. For best results with this parameter, use the SELECT program's default values.

Operation

Accessing a file in a complex directory structure can be time consuming. In addition, if you run applications that access multiple files (like data base applications), the time to open and close files can noticeably degrade your system's performance. The FASTOPEN command tracks the location of files and directories on hard disk drive(s) for fast access.

FASTOPEN

Every time you access a file or directory, FASTOPEN records its name and location in a name cache. Then, if a file or directory recorded by FASTOPEN is accessed again, the access time is greatly reduced.

In addition, with the *m* parameter a **file extent cache** can be set up for open files. This cache maintains information about open files which reduce the need for MS-DOS to reference the FAT (File Allocation Table) to locate clusters of open files.

FASTOPEN will be automatically executed whenever you start your computer if you insert the command in your CONFIG.SYS file. For more information, refer to the section on the INSTALL command in the "System Configuration" chapter.

Examples

To do this:	Type this:
Track the location of up to 100 files and directories on drive C:	`FASTOPEN C:=100`
Disable the name cache for drive C: and set up an extent cache having 10 entries	`FASTOPEN C:=(,10)`
On drive C:, track the location of 50 files and directories, and set up an extent cache having 25 entries in expanded memory	`FASTOPEN C:=(50,25) /X`

Notes

1. You can execute the FASTOPEN command only once per work session. If you want to change the FASTOPEN settings, you must restart your computer and then re-issue the command with the new settings.
2. FASTOPEN uses approximately 48 bytes of memory for each file (*n*) and directory location it tracks, and 16 bytes of memory for each file extent entry (*m*).

FC (File Compare)

The FC command compares the contents of two files. FC is an **external** command.

Syntax

FC /A /B /C /L /LB*n* /W /T /N /*nnnn* *drive1:\path1\file1* *drive2:\path2\file2*

Parameter **Description**

/A abbreviates the output of an ASCII (text) comparison. Instead of displaying all the lines that are different, it displays only the first and last lines of each set of differences. It represents the lines in between with ellipses (...).

/B does a binary comparison of both files, byte-for-byte, with no attempt to re-synchronize after a mismatch. This is the default for files with file name extensions of .EXE, .COM, .SYS, .OBJ, .LIB, and .BIN.

/C instructs MS-DOS to ignore the case of letters and consider all letters as uppercase. Use this parameter only for ASCII comparisons.

/L does an ASCII comparison of both files. This is the default for files with file name extensions other than .EXE, .COM, .SYS, .OBJ, .LIB, and .BIN.

/LB*n* sets the internal buffer to *n* during an ASCII comparison. The default length of the internal buffer is 100 lines. *n* has to exceed the number of differing lines by 2. If not, FC aborts and the results of the file comparison are invalid.

/W compresses the white spaces created by tabs and spaces during the comparison. Thus, multiple contiguous white spaces on a line are considered one white space. Note that although FC compresses white spaces, it doesn't ignore them, except the beginning and ending white spaces on a line. This parameter should only be used for ASCII comparisons.

/T doesn't expand tabs to spaces during the comparison. The default is to treat tabs as spaces to 8-column positions.

/N displays line numbers during an ASCII comparison.

/*nnnn* specifies the number of lines that must match after a difference is found for the files to be considered as matching again. That is, FC will continue to list lines that are different between the two files until it encounters *nnnn* number of lines that are the same. The default is 2.

drive1: drive that contains the first file to be compared.

path1 path to the first file to be compared.

file1 name of the first file to be compared.

drive2: drive that contains the second file to be compared.

path2 path to the second file to be compared.

file2 name of the second file to be compared.

Operation

The FC command allows you to compare the contents of two files. Two types of comparisons can be made:

- ASCII or text comparison (line-by-line)
- Binary comparison (byte-by-byte)

FC compares the first file to the second and reports any differences between them.

FC displays the following information on a file comparison:

- the file name
- the last matching line preceding a difference
- the different lines
- the next matching line

Thus, if FILE1.TXT and FILE2.TXT contain the following information:

FILE1.TXT	FILE2.TXT
all	all
good	good
things	Things
must	come
eventually	To
come	some
to	kind
some	of
kind	end
of	
an	
end	

the ASCII comparison of the two files shows the following:

```
***** C:\USER1\FILE1.TXT
things
must
eventually
come
***** B:FILE2.TXT
things
come
*****

***** C:\USER1\FILE1.TXT
of
an
end
***** B:FILE2.TXT
of
end
*****
```

MS-DOS displays the results of a binary comparison in three columns of hexadecimal numbers:

First Column	Second Column	Third Column
The position (in the files) of each pair of mismatched bytes. The first byte in each file is byte number 0.	The values of the mismatched bytes in *file1*.	The values of the mismatched bytes ın *file2*.

A portion of a binary comparison of two sample files is shown below. Note that if one file contains more data than the other, MS-DOS displays a message at the end of the comparison.

```
00000013: 6d 63
00000014: 75 6f
00000015: 73 6d
⋮
fc: C:\USER1\FILE1.TXT longer than B:FILE2.TXT
```

Examples

To do this:	Type this:
Make a line by line (ASCII) comparison of FILE1.TXT in the REPORTS directory on the active drive and FILE2.TXT in the SALES directory on the active drive	`FC \REPORTS\FILE1.TXT \SALES\FILE2.TXT`
Make a binary comparison of FILE1.TXT in the USER1 directory of drive C:, and FILE2.TXT in the root directory of drive B:	`FC /B C:\USER1\FILE1.TXT B:\FILE2.TXT`

To do this:	Type this:
Make an ASCII comparison of FILE1.TXT and FILE2.TXT in the current directory and stop the comparison when six consecutive lines are found to be the same after finding a difference	`FC /6 FILE1.TXT FILE2.TXT`

Notes

1. Options placed after file names are ignored and the default values are used instead. Incorrect or unrecognized parameters are also ignored and do not prevent the FC command from executing.
2. If you use the line buffer (/LB*n*) parameter, and the files you compare have more consecutive differing lines than the buffer length you specified, the comparison aborts and the following message is displayed:

   ```
   Resync failed. Files are too different.
   ```

FDISK

Hard disks can be divided into separate sections, called **partitions.** You can use partitions for organizational purposes (for example, each containing different files, or used by different people within a company), or to enable you to use different operating systems (such as Unix) on the same disk.

Syntax

```
FDISK
```

Operation

To prepare your hard disk for the MS-DOS operating system, you must use FDISK to create a partition for MS-DOS, called a **DOS partition.** You can use FDISK to perform the following:

- Create a primary MS-DOS partition
- Create an extended DOS partition
- Create a logical DOS drive
- Set the active partition
- Delete a DOS partition or logical drive
- Display partition information
- Review or modify the configuration of another hard disk on your computer

Caution

Reconfiguring your disk with FDISK destroys all existing files. If you are planning on running FDISK on a hard disk that already has data on it, be sure to make a backup of all files on your disk before creating or modifying partitions with FDISK.

Notes

1. FDISK can be used to create a single primary DOS partition on any hard disk of up to 2 gigabytes.
2. When FDISK formats a partition, if it finds any defective tracks at the start of the partition, it adjusts the partition boundaries to avoid those bad tracks.

3. If you have created a partition on your disk with another operating system (let's say Unix), you must set its partition to "Active" in order to use it when you turn on or reset your computer. Only one partition can be active at a time.
4. Once you delete a partition, you cannot recover the data that was on it.
5. You cannot use FDISK to delete a non-DOS partition. Instead, delete non-DOS partitions using the utilities originally used to create them.
6. If you choose to delete the primary DOS Partition, you must first delete all the extended DOS partitions on the drive.

   If you choose to delete an extended partition, you must first delete the logical drives associated with that partition.

FIND

The FIND command searches one or more files for a specified text string. FIND is an **external** command.

Syntax

FIND /V /C /N "*string*" *drive1:\path1\file1* ..

Parameter	Description
/V	displays each line that **does not** contain the specified text string.
/C	counts the number of lines containing the specified text string, instead of displaying the lines.
/N	displays each line containing the specified text string, preceded by its line number in the file.
"*string*"	string (words) to be located. It must be enclosed by quotation marks.
drive1:	drive that contains the first file to be searched.
path1	path to the first file to be searched.
file1	name of the first file to be searched.

Operation

FIND is one of three MS-DOS filters. For a detailed explanation of filters, refer to the chapter "Redirecting Input and Output."

FIND looks through one or more specified files and attempts to locate the specified text *string.* The text string must be exact as to case (upper-case and lower-case letters) and spacing. In addition, it must be enclosed by quotation marks. Quotation marks should be used even if the string to be located is already enclosed in quotation marks.

The /C parameter takes precedence over the /N parameter. If you use both /C and /N, FIND ignores the /N parameter. If you use both /C and /V, FIND counts the number of lines **not** containing the string.

If you don't specify a file name, input from the standard input device (the console) is used as the source, and its output is used as input to the FIND filter. For example:

```
DIR | FIND "<DIR>"
```

displays all of the subdirectories within the current directory. In this example, the DIR command is input from the standard input device, and its output is piped to the FIND filter. As a result, MS-DOS displays the directory entries containing the string <DIR>.

Examples

To do this:	Type this:
Locate the word Hello in FILE1.TXT in the BOOK directory of drive C:	FIND "Hello" C:\BOOK\FILE.TXT
Locate the string "just in case" in FILE1.TXT in the current directory	FIND ""just in case"" FILE1.TXT
Locate the string He said "Hi" but only count the number of lines containing the string	FIND /C "He said "Hi"" FILE1.TXT
Locate the string My name is John in FILE1.TXT and FILE2.TXT in the current directory	FIND "My name is John" FILE1.TXT FILE2.TXT

Notes

1. You **cannot** use wildcards with the FIND command to reference a group of files. Instead, you must specify each file.
2. FIND searches the specified file (or input from the standard input device) up to the first end-of-file marker ((Ctrl) (Z)).

FORMAT

The FORMAT command prepares a disk to accept MS-DOS files. FORMAT is an **external** command.

Caution The FORMAT command destroys all of the data on a disk. Therefore, make sure the disk doesn't contain valuable data before you format it.

Syntax

`FORMAT` *drive:* `/1 /4 /8 /N:`*sectors* `/T:`*tracks* `/V:`*label* `/S`

or

`FORMAT` *drive:* `/1 /B /N:`*sectors* `/T:`*tracks*

or

`FORMAT` *drive:* `/V:`*label* `/F:`*size* `/S`

Parameter	**Description**
drive:	drive that contains the disk to be formatted.
/1	formats a 5.25-inch disk for single-sided use. **Only** single-sided (160/180 KB) and double-sided flexible disks (320/360 KB) can be used with this parameter.
/4	formats 5.25-inch single-sided and double-sided flexible disks (160/180 KB and 320/360 KB) in a high capacity (1.2 MB) disk drive.
/8	formats a disk with only 8 sectors per track. Only 5.25-inch single-sided flexible disks (160/180 KB) and double-sided flexible disks (320/360 KB) can be used with this parameter.
/N:*sectors*	specifies the number of sectors per track. This parameter must be used in conjunction with /T. To format a 3.5-inch 720 KB disk, specify /N with a value of 9.

/T:*tracks* specifies the number of tracks per side. This parameter must be used in conjunction with /N. To format a 3.5-inch 720 KB disk, specify /T with a value of 80.

/V or /V:*label* specifies the volume label to use. A volume label can be up to 11 characters in length (no tabs allowed). If you do not specify the /V switch, or specify it without giving the volume label, you will be prompted for one when formatting is complete.

/S copies the operating system files (IBMBIO.COM and IBMDOS.COM), plus the MS-DOS command processor (COMMAND.COM) onto the disk. Do not use the /S and /B parameters together.

/B formats a 5.25-inch flexible disk for backward compatibility. This parameter formats a disk with 8 sectors per track and reserves space for MS-DOS system files (including earlier versions) to be placed on it using the SYS command. Do not use the /B and /S parameters together.

/F:*size* specifies the size of the diskette to format. Do not use the /N and /T parameters with /F. *Size* can be one of the following:

Disk Type	***size* (Select One)**
160 KB single-sided 5.25-inch disk	160, 160K, 160KB
180 KB single-sided 5.25-inch disk	180, 180K, 180KB
320 KB double-sided 5.25-inch disk	320, 320K, 320KB
360 KB double-sided 5.25-inch disk	360, 360K, 360KB
720 KB double-sided 3.5-inch disk	720, 720K, 720KB
1.2 MB double-sided 5.25-inch disk	1200, 1200K, 1200KB, 1.2, 1.2M, 1.2MB
1.44 MB double-sided 3.5-inch disk	1440, 1440K, 1440KB, 1.44, 1.44M, 1.44MB

Operation

The FORMAT command allows you to prepare a disk for use by MS-DOS. You must format a disk before you can use it.

If you're formatting a hard disk that's been formatted before, MS-DOS prompts you with the following message before it formats the disk:

```
Enter current Volume Label for drive x
```

If your disk doesn't have a volume label, press Enter. Otherwise, type the volume label and press Enter. If you type the wrong volume label, FORMAT displays the following error message:

```
Invalid Volume ID Format Failure
```

and you must enter the FORMAT command again. Otherwise, FORMAT displays this message:

```
WARNING: ALL DATA ON NON-REMOVABLE
DISK DRIVE x WILL BE LOST!
Proceed with Format (Y/N)?
```

To format the hard disk, type "Y" and press Enter. To cancel the FORMAT command and return to the MS-DOS prompt, type "N" and press Enter.

FORMAT performs several tasks during the format process:

- It places tracks and sectors on the disk.
- It marks defective tracks to prevent MS-DOS from placing data in them.
- It builds the root directory and FATs (File Allocation Tables).
- It prints a status report indicating:
  - □ Total disk space
  - □ Defective tracks (if any)
  - □ Space allocated to MS-DOS system files (if any)
  - □ Available disk space

Not all of the parameters are compatible with each type of disk. The following table shows which parameters are compatible with each disk type:

Disk Type	Valid Options
160/180 KB	/1 /4 /8 /B /N /T /V /S /F
320/360 KB	/1 /4 /8 /B /N /T /V /S /F
720 KB	/B /N /T /V /S /F
1.2 MB	/B /N /T /V /S /F
1.44 MB	/B /N /T /V /S /F
Hard Disk	/B /V /S

Parameters can be entered in any order in the command. However, if two or more parameters conflict, MS-DOS uses the last parameter specified in the command.

The FORMAT command returns several exit codes. The exit codes and their descriptions are listed below.

Exit Code	Description
0	Successful completion
3	Terminated by user Ctrl Break
4	Terminated due to error
5	Terminated due to "N" response on a hard disk

These error codes can be used with the ERRORLEVEL parameter of the batch processing IF command. For additional information, refer to the "Batch Processing" chapter.

Examples

To do this:	Type this:
Format the flexible disk in drive A:	`FORMAT A:`

Format a 360 KB flexible disk in a 1.2 MB flexible disk drive	`FORMAT A: /4`
	or
	`FORMAT A: /F:360`

Format a 720 KB flexible disk in a 1.44 MB flexible disk drive	`FORMAT A: /T:80 /N:9`
	or
	`FORMAT A: /F:720`

Create an application work disk in drive A: so that the computer will be able to start from it	`FORMAT A: /S`

Notes

1. A hard disk must be reformatted if the size of the MS-DOS partition is changed with the FDISK command.
2. MS-DOS formats a flexible disk according to disk drive type, **not** disk type, unless the /1, /4, /8, or the /T and /N parameters are specified. Note that once you format a lower capacity disk in a higher capacity drive, that disk can only be used in the higher capacity drive. Don't try to use it in any other drive unless you reformat it first.
3. **Do not** use the FORMAT command on a drive that has been ASSIGNed, JOINed, or SUBSTituted. Also, don't use it on a network drive.
4. If you're formatting an application work disk and you want to use it to start your system, use the /S parameter.

GRAFTABL (Graphics Tables)

The GRAFTABL command loads additional ASCII characters into memory for use by a color/graphics adapter (CGA) in graphics mode. GRAFTABL is an **external** command.

Syntax

```
GRAFTABL
```

or

```
GRAFTABL /STATUS
```

or

```
GRAFTABL
```

Parameter	**Description**
nnn	is the code page associated with the additional ASCII characters. Possible values are: 437 (United States) 850 (Multilingual) 860 (Portuguese) 863 (French-Canadian) 865 (Nordic) The default value is 437. For additional information on code pages, refer to the "How to Use Code Pages" appendix.
/STATUS	displays the active code page.
?	displays instructions for using the GRAFTABL command.

Operation

The GRAFTABL command loads additional ASCII characters into memory. Specifically, it loads character codes 128 through 255. This command is useful for displaying extended characters (non-U.S., math, and line drawing) while in graphics mode. For example, if you enter

```
    GRAFTABL 865
```

MS-DOS loads the ASCII characters associated with the Nordic code page into memory and displays the following message:

```
Graphics characters loaded
```

The GRAFTABL command returns several exit codes. The exit codes and their descriptions are listed below.

Exit Code	Description
0	Successful completion
1	Characters already loaded
2	File error
3	Incorrect parameter; no action taken
4	Incorrect version of MS-DOS; version 4.0 required

These error codes can be used with the ERRORLEVEL parameter of the batch processing IF command. For additional information, refer to the "Batch Processing" chapter.

Notes

1. The resident size of MS-DOS in memory increases by approximately 1 KB when you use this command.

GRAPHICS

The GRAPHICS command (an **external** command) allows you to print the contents of a graphics screen when you're using an adapter such as the Color Graphics Adapter (CGA), Enhanced Graphics Adapter (EGA), or Video Graphics Array (VGA).

Syntax

```
GRAPHICS type profile /B /R /PRINTBOX:id
```

Parameter **Description**

type type of printer. The possible values (listed below) are explained in more detail in the next section.

For HP Printers:	For IBM Printers:
DESKJET	COLOR1
LASERJET	COLOR4
LASERJETII	COLOR8
PAINTJET	GRAPHICS
QUIETJET	GRAPHICSWIDE
QUIETJETPLUS	THERMAL
RUGGEDWRITER	
RUGGEDWRITERWIDE	
THINKJET	

profile the name of the file that contains information on all supported printers. Graphic **profiles** are text files which contain a list of supported printers and the commands for each video mode for each of the printers listed. If you do not specify a profile, MS-DOS uses the GRAPHICS.PRO file.

/B instructs GRAPHICS to print the background color. If you don't specify this parameter, the background color **isn't** printed. This parameter is only valid with two types of printers: COLOR4 and COLOR8.

/R prints the colors exactly as they appear on the display screen. If you don't specify this parameter, black is printed as white and white is printed as black.

/PRINTBOX:*id* or /PB:*id* selects the printbox size. The *id* should match the first operand of a PRINTBOX statement in the printer profile. Valid value for *id* is STD.

Operation

The GRAPHICS command allows you to send the contents of a graphics screen to a printer. After you issue the GRAPHICS command, press (Print Screen) (or (Shift) (Prt Sc) on some computers) to print the contents of the screen. The following types of printers can be used with the GRAPHICS command:

Parameter Value	Description
DESKJET	HP DeskJet Printer.
LASERJET	HP LaserJet Printer.
LASERJETII	HP LaserJet Series II Printer.
PAINTJET	HP PaintJet Color Printer.
QUIETJET	HP QuietJet Printer.
QUIETJETPLUS	HP QuietJet Plus Printer using wide paper.
RUGGEDWRITER	HP RuggedWriter Printer.
RUGGEDWRITERWIDE	HP RuggedWriter Printer using wide paper.
THINKJET	HP ThinkJet Printer.
COLOR1	IBM Personal Computer Color Printer with a black ribbon.
COLOR4	IBM Personal Computer Color Printer with an RGB (red, green, blue, and black) ribbon.
COLOR8	IBM Personal Computer Color Printer with a CMY (cyan, magenta, yellow, and black) ribbon.

Parameter Value	Description
GRAPHICS	IBM Personal Graphics Printer, IBM Proprinter, or IBM Quietwrite Printer.
GRAPHICSWIDE	IBM Personal Graphics Printer with an 11-inch cartridge.
THERMAL	An IBM PC Convertible Thermal Printer.

If you don't specify a printer type, GRAPHICS will still provide limited print quality with your printer. For best results, we recommend that you specify your printer's *type* in the command.

Examples

To do this:	Type this:
Set graphics screen dump support for the HP LaserJet Series II Printer	`GRAPHICS LASERJETII`
Set graphics screen dump support for the HP QuietJet Printer	`GRAPHICS QUIETJET`
Place a GRAPHICS command in the AUTOEXEC.BAT file to set graphics screen dump support for the HP LaserJet Printer every time the computer is started or reset. Let's say that the GRAPHICS command is located in the DOS directory on drive C:	`C:\DOS\GRAPHICS LASERJET`

Notes

1. The resident size of MS-DOS in memory increases when you use this command.
2. If you are using an HP QuietJet Plus with narrow (letter-size) paper, specify `QUIETJET` instead of `QUIETJETPLUS` for your printer *type*.

JOIN

The JOIN command allows a disk drive to be logically connected to another disk drive via a directory. JOIN is an **external** command.

Syntax

```
JOIN
```

or

JOIN *drive1:* /D

Parameter **Description**

drive1: source drive. This is the drive to be joined to a directory on the target drive. You can't specify the active drive as the source drive.

drive2:\path directory on the target drive. The source drive is to be joined to the target drive via this directory. It must be a directory of the root directory.

/D separates (or unjoins) two drives that have been joined. When this parameter is used, the combined drive becomes two physically addressable drives again.

Operation

The JOIN command allows you to combine two physical disk drives into one logical disk drive. This means that you can reference more than one disk drive with a single drive name. This could be helpful if you have an application that is set up to only work with, let's say, drive C:. This is done by making a directory name that will represent another drive. For example:

```
   JOIN A: C:\DRIVEA
```

connects drive A: to drive C: via a directory on drive C: called DRIVEA. As a result, all the files on the disk in the drive A: now "logically" reside on drive C:, in the directory DRIVEA. Any attempt to reference drive A: at this point, results in the following error message:

```
Invalid drive specification
```

The directory used to join the source drive to the target drive *must* be a directory of the root directory. It cannot be a subdirectory of a directory. For example, if you enter:

```
JOIN A: C:\DRIVEA\SUB1
```

MS-DOS displays the error message:

```
Invalid parameter
```

You can specify a new or existing directory in the command. If you specify a new directory, the JOIN command automatically creates the directory before it joins the two drives. If you specify an existing directory, it **must** be empty or MS-DOS displays the "`Directory not empty`" message.

Examples

To do this:	Type this:
Join drive A: to drive C: so that the files on drive A: will appear as a directory of drive C: called REPORTS	`JOIN A: C:\REPORTS`
Show all drives currently joined	`JOIN`
Separate drive A: from drive C:	`JOIN A: /D`

Notes

1. A JOIN command is in effect until you enter another JOIN command with the /D parameter, or until you reset the computer.
2. When you unjoin two drives, MS-DOS doesn't delete the directory used to join them. As a result, we recommend that you use the same directory each time you join two drives. This prevents the root directory from becoming cluttered with empty directories.
3. **Do not** use the JOIN command on a drive that has been ASSIGNed or SUBSTituted. Also, don't use it on a network drive.

4. **Do not** use the BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FDISK, FORMAT, LABEL, RECOVER, RESTORE, or SYS commands on a drive that has been JOINed.

KEYB (Keyboard)

The KEYB command provides support for non-U.S. keyboards. KEYB is an **external** command.

Caution The keyboard routines associated with previous versions of MS-DOS are **not** compatible with this version.

Syntax

```
KEYB [illegible] ID:nnn
```

Parameter **Description**

xx keyboard code. Most of the possible values for *xx* are shown in the table on the next page, for a complete list of the valid keyboard codes, refer to the appendix "How to Use Code Pages."

yyy code page associated with the keyboard routine. Valid values:
437 (United States)
850 (Multilingual)
860 (Portuguese)
863 (French-Canadian)
865 (Nordic).
If you specify this parameter you must also specify *xx* and include the comma separator. The default value is country dependent.

drive:\path\file name of the keyboard definition file. If this file is not in the same directory as KEYB, you must supply the drive and path to where it can be found. The default file name is KEYBOARD.SYS. If you do not also specify *yyy,* use a comma separator in its place. For example,

```
KEYB BE,,C:\DOS\KEYBOARD.SYS
```

ID:*nnn* specifies an enhanced keyboard.

Possible values for *xx, yyy* and *nnn* are:

Keyboard Code	**Country**	**KEYB ID**	**Code Page**
BE	Belgium	120	437,850
CF	Canadian-French	058	863,850
DK	Denmark	159	865,850
FR	France	189/120	437,850
GR	Germany	129	437,850
IT	Italy	141/142	437,850
LA	Latin America	171	437,850
NL	Netherlands	143	437,850
NO	Norway	155	865,850
PO	Portugal	163	860,850
SF	Swiss-French	150	437,850
SG	Swiss-German	000 *	437,850
SP	Spain	172	437,850
SU	Finland	153	437,850
SV	Sweden	153	437,850
UK	United Kingdom	166/168	437,850
US	United States (default)	103	437,850

Localized hardware and DOS are required for these languages:

JA	Japan	081	932,437
KO	Korea	082	934,437
CH	Peoples Republic of China	086	936,437
TN	Taiwan	088	938,437

* 000 is used for Swiss-German because the actual keyboard ID for Swiss-French and Swiss-German is 150.

Operation

The KEYB command lets you use characters that are not part of the normal (QWERTY) keyboard format. Using the KEYB command with one of the two-letter codes above, you can type commands or text to MS-DOS using either the standard keyboard or a special keyboard. (The KEYB code remains in memory until you turn off or restart your computer.)

Notice that the characters that appear on your screen when you type on a standard keyboard do not necessarily match the label on the key. With the various language keyboard layouts provided with your computer hardware documentation, you can locate the keys to produce the characters you want.

To produce accented characters, you must use non-escape key combinations called "dead keys." Dead keys are keys that do not display a character when used alone. To display an accented letter, press and release the accent key, then press the appropriate letter. To display an accent alone, press the accent key, then press the spacebar.

If you type KEYB without parameters, MS-DOS displays a message showing the current keyboard code, its related code page, and the current code page used by your console (display). For example:

```
Current keyboard code: FR code page: 437
Current CON code page: 437
```

Pressing Ctrl Alt F1 returns you to the default keyboard format (United States). Pressing Ctrl Alt F2 returns you to the memory-resident keyboard program.

You can include the KEYB command in your AUTOEXEC.BAT file (see the "Batch Processing" chapter), or include an INSTALL command in your CONFIG.SYS file (see the "System Configuration" chapter).

For more information on using code pages, refer to the appendix "How to Use Code Pages."

The KEYB command displays the following exit codes:

Exit Code	Description
0	Successful completion
1	Invalid syntax
2	Bad or missing keyboard definition file
3	Could not create keyboard table in resident memory
4	Error with CON device
5	Code page requested not prepared
6	Table for selected code page cannot be found in the resident keyboard table
7	Incorrect MS-DOS version

Examples

To do this:	Type this:
Show current keyboard code and code page	`KEYB`
Use a German keyboard	`KEYB GR`
Use a Canadian-French keyboard and set the proper code page	`KEYB CF,863`
Use a Belgian keyboard and specify KEYBOARD.SYS, located in the DOS directory on drive C:, as the definition file	`KEYB BE,,C:\DOS\KEYBOARD.SYS`

LABEL

The LABEL command adds, changes, or deletes the volume label on a disk. LABEL is an **external** command.

Syntax

```
LABEL
```

Parameter **Description**

drive: drive that contains the disk you want to label. If a drive is not specified, the active drive is assumed.

label volume label used to identify the disk. It can be up to 11 characters in length; additional characters are truncated. You **cannot** use the following characters in a volume label:

```
* ? / \ | . , ; : + = < > [ ] ( ) & ^
```

You also cannot include spaces and tabs.

Operation

The LABEL command allows you to add, change, or delete the volume label on a disk. In many cases, the volume label is placed on the disk at the time it is formatted (see the /V parameter of the FORMAT command for additional information). If a disk isn't labeled at the time it is formatted, you can use the LABEL command to add one. It can also be used to change or delete an existing volume label on a disk.

If you enter the LABEL command and drive name, but not the new volume label, LABEL prompts you for it with the following message:

```
Volume label (11 characters, ENTER for none)?
```

Enter a new volume label and press Enter.

To delete the existing volume label on a disk, enter the LABEL command followed by the drive name. Then, when LABEL prompts you for a new volume label, simply press (Enter). As a precaution, the following message is displayed:

```
Delete current volume label (Y/N)?
```

Enter **N** to cancel the command and keep the current volume label intact. Enter Y to delete the current volume label.

Examples

To do this:	Type this:
Add or change the volume label of hard disk drive C: to HARDDISK	`LABEL C:HARDDISK`
Display and/or change the existing volume label of the disk in the active drive	`LABEL`

Notes

1. Do not use the LABEL command on a drive that has been ASSIGNed, JOINed, or SUBSTituted. Also, don't use it on a network drive.

MEM (Memory)

Displays the amount of used and free memory, lists allocated and free memory areas, and programs that are loaded. MEM is an **external** command.

Syntax

```
MEM /PROGRAM
```

or

```
MEM /DEBUG
```

Parameter	**Description**
/PROGRAM	displays programs loaded in memory.
/DEBUG	displays programs, internal drivers, and other programming information.

You **cannot** specify /PROGRAM and /DEBUG at the same time.

Operation

When MEM is used without parameters, conventional, expanded (EMS), and extended memory usage is displayed.

Notes

1. MS-DOS displays extended memory only if memory above 1 MB is installed in the system. Expanded memory is displayed only if a LIM 4.0 driver is installed. All DOS 4.0 components that allocate Expanded Memory Specification (EMS) memory define a name for the EMS handle returned. This name is set to the component name (for example, BUFFERS, FASTOPEN, or RAMDRIVE) which is used in the /DEBUG parameter to display and identify EMS memory usage.

MKDIR (Make Directory)

The MKDIR (or MD) command creates a new subdirectory on a disk. MKDIR is an **internal** command.

Syntax

MKDIR *drive:*\path

or

MD *drive:*\path

Parameter	**Description**
drive:	drive to contain the new directory. If a drive isn't specified, MS-DOS assumes the active drive.
path	name of the new directory.

Operation

The MKDIR command allows you to build a hierarchical directory structure on your disk by creating new directories and subdirectories.

You create directories one level at a time, beginning at the root directory. In addition, there are two ways to create directories. You can specify the path from the root directory to the new directory when you enter the MKDIR command. Or, you can use the CHDIR (change directory) command to move to a directory and enter the MKDIR command from there.

Examples

To do this:	Type this:

Create a new subdirectory called USER1 within the root directory of drive C:	`MKDIR \USER1`
	or
	`MD \USER1`

Create a new subdirectory called REPORTS within the subdirectory USER1	`MKDIR \USER1\REPORTS`
While **in** the directory USER1, create a subdirectory called REPORTS	`MKDIR REPORTS`

Notes

1. The root directory has a predetermined maximum number of entries (files and directories) based on the type and capacity of your disk drive. For additional information, refer to the chapter "Glossary of MS-DOS Terms and Concepts."
2. Unlike the root directory, there is no limit to the number of directory entries that a subdirectory can have.
3. Subdirectories can be any level deep. However, the path from the root directory to a subdirectory, plus the subdirectory name, is limited to a maximum of 64 characters.

MODE

The MODE command (an **external** command) prepares MS-DOS for communication with devices, such as displays, printers, and modems. It also prepares displays and printers for code page switching. MODE has the following syntax variations:

- Device Status Mode
- Display Mode
- Serial Communication Mode
- Parallel Printer Mode
- Redirecting Parallel Printer Output Mode
- Set Keyboard Mode
- Code Page Switching Modes

Each variation is described in the following sections.

Device Status Mode

```
MODE [illegible]S
```

Parameter	**Description**
device	name of device for MODE to check status of.
`/STATUS` or `/STA`	parameter required for requesting status for a parallel (or redirected) printer.

Operation

Displays device status. Typing MODE by itself displays the status of all devices installed in your system. /STATUS is required only when requesting status for (redirected) parallel printers.

To display the status of a specific device, type MODE followed by the device name. For example, type:

```
MODE CON
```

Display Mode

MODE *display,n*

or

MODE *display,shift,*T

or

MODE CON:COLS=*m* LINES=*n*

Parameter	Description
display	display mode. Valid display modes are 40, 80, BW40, BW80, CO40, CO80, and MONO (see table in "Operation" section for more detail).
n	specifies the number of lines on the display. Possible values are 25, 43, and 50. Not all adapters support all sizes.
shift	direction that a Color Graphics Adapter (CGA) display is to be shifted. Valid directions are R (right) or L (left). This parameter increases the resident size of MS-DOS in memory.
T	produces a test pattern that can be used to align the display.
m	specifies the number of characters per line. Possible values are 40 or 80.

Operation

MODE performs the following two operations on a display:

- It selects the display mode.
- It aligns the display on the screen.

To select the display mode, enter:

```
MODE display
```

where *display* is one of the valid display modes below.

Display Mode	Description
40	sets the number of characters per line to 40 for a CGA-compatible video adapter.
80	sets the number of characters per line to 80 for a CGA-compatible video adapter.
BW40	uses the CGA-compatible video adapter as the active display adapter, disables color or gray scale, and sets the number of characters per line to 40.
BW80	uses the CGA-compatible video adapter as the active display adapter, disables color or gray scale, and sets the number of characters per line to 80.
CO40	uses the CGA-compatible video adapter as the active display adapter, enables color or gray scale, and sets the number of characters per line to 40.
CO80	uses the CGA-compatible video adapter as the active display adapter, enables color or gray scale, and sets the number of characters per line to 80.
MONO	selects the monochrome display adapter as the active display adapter if two adapters are attached. Monochrome display adapters only support 80 characters per line.

In addition to selecting the display mode, the MODE command also provides a means of aligning the display on the screen and generating a test pattern. The display can be adjusted to the left or right on the screen (one character at a time in 40 column mode and two in 80 column mode).

When the T parameter is specified, MODE asks if you want to continue adjusting the display. Enter `Y` to continue shifting the display in the same direction or `N` to stop.

Examples

To do this:	Type this:
Display the current MODE settings	MODE
Set the display to CO80 mode and shift the display 2 characters to the right	MODE CO80,R
Set the display to CO80 mode, shift the display 2 characters to the left, and display a test pattern	MODE CO80,L,T

Serial Communication Mode

MODE COM*n*:*b*

or

MODE COM*n*:BAUD=*b* PARITY=*p* DATA=*d* STOP=*s* RETRY=*ra*

Parameter	Description
COM*n*:	serial communications port (COM1, COM2, COM3, or COM4). The default is COM1.
b	specifies baud (transmission) rate. Valid rates are 110, 150, 300, 600, 1200, 2400, 4800, 9600, and 19200. Only the first two digits need to be specified; the remaining digits are ignored. When specifying *p, d, s,* or *ra,* baud must also be specified.
p	specifies parity. Parity is N (none), O (odd) or E (even). The default parity is E. With or without *p,* comma separator must be included when specifying *d, s,* or *ra.*
d	specifies number of databits in each character transmitted (5, 6, 7, or 8). The default number of databits is 7. With or without *d,* comma separator must be included when specifying *s,* or *ra.*

s specifies number of stopbits (`1`, `1.5`, or `2`). The default number of stopbits is 2 if the baud rate is 110, and 1 for all other baud rates. With or without *s*, comma separator must be included when specifying *ra*.

ra specifies what type of retry action to take. If you are using the MODE command over a network, do not use *ra*. Valid entries for *ra* are:

`E` (returns an error from a status check of the busy port. Default).

`B` (returns busy from a status check of the busy port. Provides the same support as the P parameter in previous MS-DOS releases).

`R` (returns ready from a status check of the busy port.)

`NONE` (means that no retry action is taken.)

Operation

The MODE command is used to change the parameters and protocol for the serial (also called RS-232) port(s) on your interface cards.

The RETRY parameter indicates how MODE will respond to a status check or status request for the printer ports. To indicate that the port will be used as the system printer port (and therefore the computer should retry sending data after each printer time-out) you should specify B.

Examples

To do this:	Type this:
Display the current MODE settings	`MODE`
Set the serial port COM1 to 9600 baud, no parity, 8 databits, 1 stopbit and specify B for RETRY	`MODE COM1:12,N,8,1,B`
Set the serial port COM1 to 9600 baud, designate the port as the printer port with the B retry parameter, and set everything else to the default	`MODE COM1:96,,,,B`

Parallel Printer Mode

MODE LPT*n*:

or

MODE LPT*n*: COLS=*c* LINES=*l* RETRY=*ra*

Parameter **Description**

LPT*n*: parallel printer port (LPT1, LPT2, or LPT3).

c horizontal spacing (either 80 or 132 characters per line). With or without *c*, include the comma separator when specifying *l* or *ra*.

l vertical spacing (either 6 or 8 lines per inch). With or without *l*, include comma separator when specifying *ra*.

ra specifies what type of retry action to take. If you are using the MODE command over a network, do not use *ra*. Valid entries for *ra* are:

E (returns an error from a status check of the busy port. Default.)

B (returns busy from a status check of the busy port. Provides the same support as the P option in previous MS-DOS releases.)

R (returns ready from a status check of the busy port.)

NONE (means that no retry action is taken.)

Operation

MS-DOS supports three parallel printer devices: LPT1, LPT2, and LPT3. You can use the MODE command to set the number of characters per line (*c*) and the number of lines per inch (*l*) for these devices.

The RETRY parameter indicates how MODE will respond to a status check or status request for the printer ports. To indicate that the port will be used as the system printer port (and therefore the computer should retry sending data after each printer time-out) you should specify B.

Examples

To do this:	Type this:
Display the current MODE settings	`MODE`
Set the horizontal spacing of LPT2 to 132 characters per line and the vertical spacing of LPT2 to 6 lines per inch	`MODE LPT2:132,6`
Set the same parameters as the previous example, plus specify B as the RETRY parameter to use the parallel port as the printer port	`MODE LPT2:132,6,B`
Set B as the retry parameter to use parallel port 1 as the printer port and leave the other settings at their default	`MODE LPT1:,,B`

Notes

1. If you enter a value other than 80 or 132 for the number of characters per line, MS-DOS ignores the value and preserves the current value.
2. If you enter a value other than 6 or 8 for the number of characters per inch, MS-DOS ignores the value and preserves the current value.

Redirecting Parallel Printer Output Mode

MODE LPT*n:*=COM*n:*

Parameter	**Description**
LPT*n:*	parallel printer port (LPT1, LPT2, or LPT3).
COM*n:*	serial communications port (COM1, COM2, COM3, or COM4).

Operation

MODE allows you to redirect data originally intended for the parallel printer port to one of the serial ports (COM1, COM2, COM3, or COM4).

Examples

To do this:	Type this:
Display the current MODE settings	`MODE`
Redirects data from LPT1 to COM1	`MODE LPT1:=COM1:`
Cancel redirection from LPT1 to COM1	`MODE LPT1`

Notes

1. Before you can redirect parallel printer output to a serial port, you must set the serial communication mode for that port (refer to the "Serial Communications Mode" section). Also, if the serial device is a printer, we recommend you specify B for RETRY.
2. Redirecting parallel printer output increases the resident size of MS-DOS in memory.

Set Keyboard Mode

`MODE CON:RATE=`*r* `DELAY=`*d*

Parameter	Description
r	specifies the typematic interval time, the rate of approximate repetitions per second. The range for possible values is 1 to 32.
d	specifies the auto-repeat start delay time. The range of possible values is 1 to 4 (which correspond to delays of .25, .5, .75 and 1 second, respectively).

The **typematic** rate of your keyboard is the speed at which characters are automatically repeated when a key is held down (or, how fast you see characters repeated on you display). The **auto-repeat start delay time** refers to the amount of time it takes to start automatic repeating when a key is held down. To set the **typematic** rate, you must specify both RATE and DELAY in your command.

Code Page Switching Modes

MODE *device* CODEPAGE PREPARE=((*yyy*)*drive:\path\file*)

and

MODE *device* CODEPAGE SELECT=*yyy*
MODE *device* CODEPAGE REFRESH
MODE *device* CODEPAGE

Parameter	**Description**
device	device to support code page switching. Valid device names are CON, LPT1 (or PRN), LPT2, and LPT3.
CODEPAGE or CP	specifies the CODEPAGE switching mode.
PREPARE or PREP	tells MS-DOS to prepare code pages for a specific device. Before you can prepare code pages for a specific device, you must identify those code pages by using the DEVICE configuration command.
yyy	code page(s) you want to use with this device. Valid values: 437 (United States) 850 (Multilingual) 860 (Portuguese) 863 (French-Canadian) 865 (Nordic).
drive:\path\file	name of the Code Page Information (.CPI) file for this device. If the file is not in the root directory, you must supply a drive and path to the location of the file. Possible values are: 4201.CPI (IBM Proprinter) 5202.CPI (IBM Quietwriter III Printer) EGA.CPI (EGA or VGA display adapter)
SELECT or SEL	tells MS-DOS which prepared code page you want to use with the device. Before you can select a code page, you must prepare it.

REFRESH or REF — tells MS-DOS to reinstate the prepared code pages for a device. This is useful if prepared code pages are "lost" due to a hardware error.

/STATUS or /STA — tells MS-DOS to display the code pages prepared and selected for a device. Typing MODE *device* CODEPAGE (without /STATUS) produces the same result.

Operation

The PREPARE, SELECT, REFRESH, and /STATUS parameters associated with the CODEPAGE variation of the MODE command allow you to set up and control code page switching on your display and printers.

For additional information on code page switching, refer to the appendix "How to Use Code Pages."

Examples

To do this:	**Type this:**

Display code page information selected for the CON (console or display) device	`MODE CON CODEPAGE /STATUS`
	or
	`MODE CON CP /STA`
	or
	`MODE CON CP`

Prepare two code pages for the CON device. The CON device is an EGA display and adapter, and the code page information file is EGA.CPI located in the root directory	`MODE CON CP PREPARE=((437,850) C:\EGA.CPI)`
Selects code page 850 as the active code page for the CON device	`MODE CON CP SELECT=850`

Notes

1. To automatically prepare a device for code page switching every time you start your system, include the appropriate DEVICE configuration command in your CONFIG.SYS file and the appropriate **MODE** *device* **CODEPAGE PREPARE** command in your AUTOEXEC.BAT file.
2. If you want to include both a **MODE CON CODEPAGE PREPARE** command and a KEYB command in your AUTOEXEC.BAT file, make sure the MODE command occurs first.
3. The **MODE** *device* **CODEPAGE SELECT** command selects a code page for an individual device. To select a code page for all devices prepared for code page switching, use the CHCP command.

MORE

The MORE command displays output on your display, one screen at a time. MORE is an **external** command.

Syntax

MORE < *source*

or

source | MORE

Parameter	Description
drive:	drive that contains the *source.*
path	path to the *source.*
< and |	these symbols redirect data through MORE from a source. For more information on redirecting, refer to the "Redirecting Input and Output" chapter.
source	file name and/or command.

Operation

MORE is one of three filters provided with the MS-DOS operating system. For a detailed explanation of filters, refer to the "Redirecting Input and Output" chapter.

MORE displays the first full screen of data, displays the message "`--MORE--`", and pauses. It waits until you press any key, and then displays the next full screen of data. This process continues until all the data has been displayed.

Examples

To do this:	Type this:
Display the contents of the current directory, one screen at a time	`DIR | MORE`
Display the contents of FILE1.TXT, one screen at a time	`TYPE FILE1.TXT | MORE`
Display the contents of FILE1.TXT a screen at a time, by redirecting it to the MORE filter	`MORE < FILE1.TXT`

For a detailed explanation on redirection, refer to the "Redirecting Input and Output" chapter.

Notes

1. To hold input until it is displayed, the MORE command creates a temporary file on the disk. If the disk is full or write-protected, MORE will not work.

NLSFUNC (National Language Support)

The NLSFUNC command loads support for extended country information and code page switching. NLSFUNC is an **external** command.

Syntax

```
NLSFUNC
```

Parameter	**Description**
drive:	drive that contains the file that contains extended country information.
path	path to the file that contains extended country information.
file	file that contains extended country information. The default file name is defined by the COUNTRY command in the CONFIG.SYS file. If there isn't a COUNTRY command in the CONFIG.SYS file, MS-DOS uses the COUNTRY.SYS file in the root directory as the default.

Operation

The NLSFUNC command provides support for extended country-specific information and code page switching. For additional information on both topics, refer to the appendix "How to Use Code Pages."

To use the default country-specific information found in the COUNTRY.SYS file and provide support for code page switching, enter the following command:

```
NLSFUNC
```

Notes

1. You must execute this command before you execute the CHCP (Change Code Page) command.

PATH

The PATH command sets search path(s) to one or more directories for external MS-DOS commands, program files, and batch files. It also displays and deletes existing search paths. PATH is an **internal** command.

Syntax

```
PATH
```

or

```
PATH ;
```

Parameter	**Description**
drive1:	drive that contains the first search path.
path1	the first search path.
drive2:	drive that contains the second search path.
path2	the second search path.

Operation

The PATH command allows you to specify a list of directories. MS-DOS automatically searches this list when it is unable to find an external command, a program file, or a batch file in the current directory. Each directory in the list is referred to as a **search path.** One or more search paths can be set with the PATH command. The PATH command also displays or cancel existing search paths. The maximum length for a list of directories in the PATH command is 127 characters.

MS-DOS only searches one directory at a time. It always starts by searching the current directory. Then, it searches the directories specified by the PATH command (in the order they appear in the command). Once MS-DOS finds the command or file it's looking for, it stops the search. It doesn't search the remaining directories.

The PATH command should be used at the beginning of a work session (that is, just after you turn your system on). This sets the search paths for your entire work session. MS-DOS uses these search paths until you end your work session (turn your system off or restart it) or issue another PATH command.

If you want to establish a permanent search path, include the PATH command along with the search path in your AUTOEXEC.BAT file. For additional information on the AUTOEXEC.BAT file, refer to the "Batch Processing" chapter.

Examples

To do this:	Type this:
Display existing search paths	`PATH`
Set the root directory and the DOS directory (that contains all MS-DOS commands) on drive C: as search paths	`PATH C:\;C:\DOS;`
Set the same search paths as the previous example, plus add the subdirectory REPORTS (which is a subdirectory of USER1)	`PATH C:\;C:\USER1;C:\USER1\REPORTS;`
Cancel existing search paths	`PATH ;`

Notes

1. The search path to each subdirectory must be individually specified.
2. MS-DOS only searches for program files and batch files when you execute them, not when you perform any other tasks on them (such as editing).
3. MS-DOS can't detect an invalid search path until it attempts to use it. If MS-DOS detects an invalid search path (due to incorrect syntax or a non-existent subdirectory), MS-DOS ignores it and continues on with the next search path.

4. When you start your computer, MS-DOS executes commands in the following order: .COM, .EXE, .BAT. If you execute commands that load resident code (such as GRAPHICS, KEYB, MODE, and PRINT) before you execute the PATH command, there may not be enough room in the MS-DOS environment to store the list of directories you specified. If there isn't enough room, MS-DOS displays:

   ```
   Out of environment space
   ```

   To correct this situation, execute the PATH command first, or increase the MS-DOS environment specified in your CONFIG.SYS file using the /E:*nnnnn* parameter of the COMMAND command.

PRINT

The PRINT command prints the contents of a file (or a list of files) on your printer while you're performing other tasks on your system. This process is frequently referred to as **background printing.** PRINT is an **external** command.

Syntax

```
PRINT [illegible] size /U:value1 /M:valu [illegible]
[illegible]
```

Parameter **Description**

/D:*device* specifies the list (output) device. A valid list device is any character device driver supported by your computer. For example, AUX, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, or PRN. The default list device is PRN. You must specify /D:*device* before any other parameter.

/B:*size* sets the size of the internal buffer in bytes. The default internal buffer size is 512 bytes. Increasing the size of the internal buffer may improve the performance of the PRINT command.

/U:*value1* specifies the number of computer clock ticks that PRINT will wait until the printer is available. If PRINT has to wait a longer period of time, it gives up its time slice. The default value is 1.

/M:*value2* specifies the number of computer clock ticks that PRINT can have to send characters to the printer. The number of computer clock ticks can range from 1 to 255. The default value is 2.

/S:*timeslice* specifies the time slice (the amount of time) that PRINT can use for printing in a multitasking environment. The time slice can range from 1 to 255. The default value is 8.

/Q:*qsize* specifies the number of files allowed in the print queue. The number of files can range from 4 to 32. The default value is 10.

/T — cancels or "terminates" all files in the print queue. If a file is in the process of being printed, the printing stops and MS-DOS displays a cancellation message.

drive: — drive that contains the file(s) to be printed.

path — path to the file(s) to be printed.

file — name(s) of the file(s) to be printed.

/C — turns on the cancel mode and removes the preceding file name and all following file names from the print queue. You must specify a file name before the /C parameter.

/P — turns on the print mode and adds the preceding file name and all of the following file names to the print queue until a /C is found in the command, or until you press Enter. You must specify a file name before the /P parameter.

Operation

The PRINT command creates and maintains a print queue. A print queue is a list of files to be printed as a background task (while you use your system to perform some other task).

Files can be added to the print queue by entering the names of the files to be printed in the command. The files are added to the print queue are printed in the order you entered them in the command. After each file is printed, the printer paper is advanced to the next page.

Caution

The print queue only holds the names of the files to be printed, and not the actual contents of those files. Therefore, the disk(s) containing the files to be printed *must* remain in its drive(s) until the print queue is empty. Also, the files to be printed *must not* be changed or erased while their names are in the print queue.

If you use the /D, /B, /U, /M, /S and /Q parameters, you *must* specify them the first time you execute the PRINT command after you start your computer. Otherwise, the default parameters are automatically set. Also, once the print queue's system parameters are set, they remain in effect until you turn off or restart your computer.

If the `/D:`*device* option isn't specified the first time the PRINT command is executed, PRINT prompts for the list (print) device with the message:

```
Name of list device (PRN):
```

To select `PRN` as the list device, simply press (Enter). To select another character device as the list device, type the device name and press (Enter).

Examples

To do this:	Type this:
Add all the files with the file name extension of .TXT to the print queue	`PRINT *.TXT`
Queue the files FILE1.TXT and FILE2.TXT for printing	`PRINT FILE1.TXT FILE2.TXT`
Remove the files FILE1.TXT and FILE2.TXT from the print queue and add the file FILE3.TXT to the print queue	`PRINT FILE1.TXT /C FILE2.TXT /P FILE3.TXT`

To do this:	Type this:
Remove all of the files currently in the print queue	`PRINT /T`
Determine which files are currently in the print queue	`PRINT`

Notes

1. The printer cannot be used by any other MS-DOS command or application program while PRINT is queuing and printing files. If another MS-DOS command or application program attempts to use the printer, the following error message is displayed:

   ```
   Out of paper
   ```

2. If you want to permanently set the print queue's system parameters, include the PRINT command along with the appropriate options in your AUTOEXEC.BAT file. For additional information on the AUTOEXEC.BAT file, refer to the chapter "Batch Processing."
3. Tab characters are expanded to the next 8-column boundary with blanks.
4. The path to each file in the print queue can contain a maximum of 64 characters, including the drive name and the file name. Therefore, you may need to print the file from the directory that contains the file.
5. The PRINT command increases the resident size of MS-DOS in memory the first time it is executed.
6. If PRINT encounters a disk error while printing a file, printing stops, a message prints on the printer, and then PRINT continues on to the next file in the queue.

PROMPT

The PROMPT command changes the MS-DOS prompt. PROMPT is an **internal** command.

Syntax

```
PROMPT [illegible]ter ...
```

Parameter **Description**

text text and special characters for the new MS-DOS prompt.

$*character* PROMPT command characters used to create special prompts as shown below (both upper- and lower-case characters can be used).

Characters	Prompt to be displayed
$$	The dollar sign ($) character
$B	The vertical bar (|) character
$D	The current date
$E	The ASCII escape code (1BH)
$G	The greater than (>) character
$H	The backspace character
$L	The less than (<) character
$N	The active drive
$P	The current directory of the active drive
$Q	The equal sign (=) character
$T	The current time
$V	The MS-DOS version number
$_ (underline)	A carriage return/line feed sequence (go to the next line)

Operation

The PROMPT command allows you to change the MS-DOS prompt to any prompt of your choice. The default prompt is the active drive name (the active drive letter followed by the greater than (>) character). The new MS-DOS prompt can consist of any combination of text and special characters.

The text characters can consist of any of the characters permissible in MS-DOS file names. See the "Glossary of MS-DOS Terms and Concepts" chapter.

In addition to text characters, there are special characters that may be inserted in the MS-DOS prompt. These characters are all preceded by the dollar sign (**$**) character to differentiate them from text strings (see table on previous page).

If ANSI.SYS support is installed, the ASCII escape code special character (**$**E) can be used to produce inverse video, cursor positioning, and other video attributes in the MS-DOS prompt.

Examples

To do this:	Type this:
Create an MS-DOS prompt that displays the word `HELLO`	`PROMPT HELLO`
Set the MS-DOS prompt to display the current time	`PROMPT $T`
Set a two-line prompt that prints `TIME =` followed by the current time, `DATE =` followed by the current date, and the greater than (>) sign	`PROMPT TIME = $T$_DATE = $D$G`
Set the MS-DOS prompt to the current directory of the active drive followed by the greater than (>) sign	`PROMPT $P$G`
Set the prompt in inverse video mode and return to video mode for other text (ANSI.SYS support must be installed)	`PROMPT $e[7m$n:$e[m`

Notes

1. MS-DOS stores the PROMPT command as a string in the MS-DOS environment. Refer to the SET command for additional information.
2. If you execute commands that load resident code (such as GRAPHICS, KEYB, MODE, and PRINT) before you execute the PROMPT command, there may not be enough room in the MS-DOS environment to store the prompt you specified. If there isn't enough room, the following message is displayed:

   ```
   Out of environment space
   ```

   To correct this situation, execute the PROMPT command first, or increase the MS-DOS environment specified in your CONFIG.SYS file using the /E:*nnnnn* parameter of the COMMAND command.

RECOVER

The RECOVER command recovers readable information from a file or files when one or more sectors of a disk become bad or defective. The data in the bad sector(s) cannot be recovered. RECOVER is an **external** command.

Syntax

RECOVER [illegible]\file

or

RECOVER *drive:*

Parameter	Description
drive:	drive that contains the file(s) to be recovered.
path	path to the file(s) to be recovered.
file	name(s) of the file(s) to be recovered.

Operation

If a sector on a disk is causing read/write errors, and the CHKDSK command shows that a sector on your disk is bad, you can use the RECOVER command to recover each file on the disk or just the file containing the bad sector.

The RECOVER command causes MS-DOS to read the file sector by sector and to skip the bad sectors. When MS-DOS finds a bad sector, it will no longer use that sector to store data.

Specifying an entire subdirectory in the RECOVER command causes the subdirectory to be recovered to the root directory, rather than to the original subdirectory where the files were stored.

Specifying the entire disk (or drive) causes all files on the disk to be recovered to the root directory. Each recovered file is placed on the root directory in a *filennnn*.REC file, where *nnnn* is a sequential number starting with 0001. Use this form of the RECOVER command if the directory of the disk is not usable.

Since all data in the bad sectors are lost after recovery is run on a file, recovering files singly allows you to edit the files once they are recovered and to re-enter any missing information.

Caution

Since the root directory can only hold a finite number of entries, some of the recovered files may be lost. If all files on a disk need to be recovered, recover the files one at a time.

Examples

To do this:	Type this:
Recover a file called FILE1.TXT located in the root directory of drive B:	`RECOVER B:FILE1.TXT`
Recover a file called FILE2.TXT located in the REPORTS directory of drive C:	`RECOVER C:\REPORTS\FILE2.TXT`
Recover all the files located on the disk in drive B:	`RECOVER B:`

Notes

1. The RECOVER command does not work on a network from a remote workstation.
2. RECOVER does not work on drives used in the SUBST or JOIN commands.
3. RECOVER command does not work with the backup or restore commands. You must use the RESTORE command with files backed up with the BACKUP command.
4. If there isn't enough room in the root directory for the recovered files, delete some unnecessary files from the root directory and run RECOVER again.

RENAME

The RENAME (or REN) command changes the name of a file. RENAME is an **internal** command.

Syntax

```
RENAME        file1 file2
```

or

```
REN        file1 file2
```

Parameter	**Description**
drive:	drive that contains the file to be renamed.
path	path to the file to be renamed.
file1	file to be renamed.
file2	new file name.

Operation

RENAME changes the name of a file (*file1*) to a new name (*file2*). You can specify a drive and/or path for *file1,* but not for *file2.* Thus, the file remains on the same drive and in the same directory after its name is changed.

If you attempt to rename a file that doesn't exist or to rename a file to a name already present in the directory, MS-DOS displays the following error message:

```
Duplicate file name or File not found
```

Examples

To do this:	Type this:

Rename file FILE1.TXT to MEMO1.TXT	`RENAME FILE1.TXT MEMO1.TXT`
	or
	`REN FILE1.TXT MEMO1.TXT`
	or
	`REN FILE1.TXT MEMO?.???`

Rename all the files with the extension of .TXT to .DOC	`RENAME *.TXT *.DOC`

REPLACE

The REPLACE command selectively replaces target files with source files of the same name. It also selectively adds files from the source to the target. REPLACE is an **external** command.

Syntax

REPLACE [illegible] *file* [illegible] *ath2* /A /P /R /S /W /U

Parameter	**Description**
drive1:	source drive. It contains the source file(s) (the files to be used to replace the files on the target drive or the files to be added to the target drive).
path1	path to the source file(s).
file	name(s) of the source file(s).
drive2:	target drive. It contains the target file(s) (the files to be replaced or added to).
path2	path to the target file(s).
/A	adds source files to the target instead of replacing existing files on the target. Only source files that don't already exist on the target are added. You cannot use /A with /S.
/P	prompts you before replacing a target file or adding a source file to the target. This allows you to selectively replace and add files.
/R	replaces read-only files as well as unprotected files. If you don't specify /R, any attempt to replace a read-only file creates an error which stops the REPLACE command.
/S	searches all subdirectories on the target while replacing matching files. Only subdirectories on the target are searched (subdirectories on the source are not searched). You cannot use /S with /A.

/W waits for you to press any key before replacing or adding files. This allows you to switch flexible disks, if necessary.

/U replaces only those files on the target directory that are older than those in the source directory. Compares the time/date of the source files with that of the target files and replaces only those files that need updating.

Operation

The REPLACE command allows you to selectively replace the files on a disk or add files to it. As MS-DOS replaces or adds each file, it displays the file name on the screen. When all files are replaced or added, MS-DOS displays one of the following messages:

```
XXX File(s) replaced/added
```

or

```
No files replaced/added
```

The REPLACE command returns several exit codes. The exit codes and their descriptions are listed below.

Exit Code	Description
1	Command line error
2	File not found
3	Path not found
5	Access denied
8	Insufficient memory
15	Invalid drive
Other	Standard MS-DOS error

These error codes can be used with the ERRORLEVEL parameter of the batch processing IF command. See the "Batch Processing" chapter for additional information.

Examples

To do this:	Type this:
Replace Files:	
Replace FILE1.TXT in the root directory of drive C: with FILE1.TXT in the root directory of drive A:	`REPLACE A:\FILE1.TXT C:\`
Replace FILE1.TXT where ever it can be found (root directory or other directory) on drive C: with the file FILE1.TXT in the root directory of drive A:	`REPLACE A:\FILE1.TXT C: /S`
Replace all of the files with the file name extension of .TXT in the root directory of drive C: with all of the files with the same file name extension in the root directory of drive A:	`REPLACE A:\*.TXT C:\`
Have REPLACE prompt you with `Yes` or `No` before replacing each file with the extension .TXT on the root directory of drive C:	`REPLACE A:\*.TXT C:\ /P`
Add Files:	
Add the file FILE1.TXT from the SALES directory of drive A: to the root directory of drive C:	`REPLACE A:\SALES\FILE1.TXT C:\ /A`
Add all files with the extension .TXT in the current directory to the directory SALES of drive A:	`REPLACE *.TXT A:\SALES /A`

Notes

1. You can't use the REPLACE command to replace or add hidden or system files.

RESTORE

The RESTORE command restores one or more files that have been backed up with the BACKUP command. RESTORE is an **external** command.

Syntax

RESTORE *drive1:* *drive2:\path\file* /S /P /B:*date* /A:*date* /E [illegible] /M
/ [illegible]

Parameter **Description**

drive1: source drive. It contains the file(s) to be restored.

drive2: target drive. The drive that you want to contain the restored file(s). This is usually the drive from which the files were backed up originally.

path path to the subdirectory that you want to contain the restored file(s).

file name(s) of the file(s) to be restored.

/S restores the files in all subdirectories of the specified directory, in addition to the files in the specified directory.

/P prompts you before restoring a file that (1) has been changed since it was backed up with the BACKUP command or (2) has a hidden or read-only attribute set. This prevents the loss of the most recent version of the file by inadvertently restoring an older version. If you do not use this parameter, all files on the source disk will automatically overwrite those on the target disk.

/B:*date* restores only those files modified on or before the specified date. The default format is *mm-dd-yy* (U.S. format). The date format can vary depending on the COUNTRY command in the CONFIG.SYS file.

/A:*date* restores only those files modified on or after the specified date. The default format is *mm-dd-yy* (U.S. format). The date format can vary depending on the COUNTRY command in the CONFIG.SYS file.

`/E:`*time* restores only those files modified at or earlier than the specified time. The default format is *hh-mm-ss.* The time format can vary depending on the COUNTRY command in the CONFIG.SYS file.

`/L:`*time* restores only those files modified at or later than the specified time. The default format is *hh-mm-ss.* The time format can vary depending on the COUNTRY command in the CONFIG.SYS file.

`/M` restores only those files modified since the last backup.

`/N` restores only those files that no longer exist on the target drive.

Operation

The RESTORE command restores files from one disk to another. You can restore an entire disk or a group of files, depending on which files were originally backed up with the BACKUP command.

You need not restore to the same disk you backed up from. However, when restoring files, you can't change the directory structure used when the files were backed up. This means that files must be restored into the **same directory** from which they were originally backed up. If you don't specify the name of the old directory a file was backed up from, RESTORE will recreate the old directory for you.

If more than one source disk is required to restore the contents of a drive, RESTORE prompts you to insert the other disks. The disks must be inserted into drive A: in the order they were originally created (starting with backup disk 01).

The RESTORE command returns the following exit codes:

Exit Code	Description
0	Normal completion
1	No files were found to restore
2	Some files not restored due to sharing conflicts
3	Restore terminated by user
4	Restore terminated due to error

These error codes can be used with the ERRORLEVEL parameter of the batch processing IF command. See the "Batch Processing" chapter for additional information.

Examples

To do this:	Type this:
Restore all files from the disk in drive A:, including subdirectories, to drive C:	`RESTORE A: C: /S`
Restore all files with the extension of .TXT on drive A: that were originally backed up from the subdirectory USER1 on drive C:	`RESTORE A: C:\USER1\*.TXT`
Restore all files on drive A: modified on or after January 10, 1989 to the root directory of drive C:	`RESTORE A: C:\ /A:01-10-89`

Notes

1. You can't restore the system files (IBMBIO.COM and IBMDOS.COM) or COMMAND.COM. Thus, you can't use RESTORE to create an MS-DOS (bootable) disk. To create an MS-DOS disk:

   a. Use the SYS command to copy the system files.
   b. Use the COPY command to copy COMMAND.COM.
   c. Use the RESTORE command to copy the rest of your files.

2. RESTORE can't be used if a drive was ASSIGNed, JOINed, or SUBSTituted during the backup. Also, don't use RESTORE while APPEND is in effect.

3. You can't restore shared files that you don't have access to. If you attempt to do so, RESTORE displays an error message.

4. You can use this version (4.0) of the RESTORE command to restore files backed up with an earlier version (3.xx) of the BACKUP command.

RMDIR (Remove Directory)

The RMDIR (or RD) command is used to remove a directory from a disk. RMDIR is an **internal** command.

Syntax

`RMDIR` *path*

or

`RD` *path*

Parameter	Description
drive:	drive that contains the directory to be removed.
path	path and name of the directory to be removed.

Operation

The RMDIR command allows you to remove an unwanted directory from a disk. Before you remove a directory, it must be empty. That means it can't contain any subdirectories or files. If you attempt to remove a directory that isn't empty, MS-DOS displays the following error message:

```
Invalid path, not directory,
or directory not empty
```

(A directory is empty if, when you enter the DIR command, the . and .. entries are the only two entries listed. You can't remove these two entries.)

Examples

To do this:	Type this:

Remove the directory REPORTS, which is a subdirectory of USER1	`RMDIR \USER1\REPORTS`
	or
	`RD \USER1\REPORTS`

Remove the directory USER1 from drive C:	`RD C:\USER1`

Notes

1. You can't remove the root directory or the current directory.
2. A directory isn't empty if it contains hidden files.

SELECT

SELECT is a utility program that installs MS-DOS from your master disks onto a hard disk or flexible disks, and creates AUTOEXEC.BAT and CONFIG.SYS files containing the desired configuration information.

Syntax

If you have a brand new system, or are upgrading from a previous version of MS-DOS, insert the MS-DOS 4.0 *Install* disk in drive A: and either reset the computer, or turn the computer and display on to start the SELECT program (see note).

If you have already installed MS-DOS 4.0 on the hard disk and you need to update the specifications made previously, insert the *Install* disk in drive A:, and enter the following at the MS-DOS prompt:

```
SELECT MENU
```

Once the desired changes are made, SELECT allows you to install the updated files on the hard disk. New versions of the CONFIG.SYS and AUTOEXEC.BAT files are copied over to the root directory (see note).

Note

If CONFIG.SYS and AUTOEXEC.BAT do not exist, they are created. If CONFIG.SYS and AUTOEXEC.BAT do exist, your old versions of these files are renamed to CONFIG.OLD and AUTOEXEC.OLD. Compare your new CONFIG.SYS and AUTOEXEC.BAT files with your CONFIG.OLD and AUTOEXEC.OLD files for information you wish to save. Then, update your new CONFIG.SYS and AUTOEXEC.BAT files with Edlin, or a text processor. When done, press Ctrl Alt Del to restart your system.

Operation

SELECT lets you install MS-DOS and perform the following functions:

- Specify how the available system memory will be shared between MS-DOS and application programs
- Specify the country and keyboard. (This determines date and time formats, and currency symbols)
- Specify printer type and printer port. (You may also designate a serial port as a parallel port if necessary)
- Specify various installation options, including: code-page switching, expanded memory support, extended display support (ANSI.SYS), file performance enhancements (FASTOPEN), GRAFTABL display support, GRAPHICS (print screen) support, SHARE support, PAM or MS-DOS Shell support, and RAM disk support (RAMDRIVE.SYS).
- Create CONFIG.SYS and AUTOEXEC.BAT files containing the specifications made while running SELECT.

The default for each entry screen of SELECT is highlighted. Use the ▲ and ▼ keys to choose an item other than the screen default selection. In addition, each entry screen has extensive help information that can be displayed by pressing the F1 key. Refer to these files for information that will help you to make the selection that is right for your system.

Before you begin SELECT, make certain the you know the following:

- how much memory is available with your computer,
- what type of printer you have (serial or parallel),
- whether or not your computer has expanded memory.

Having this ready will allow you to respond more quickly to many of the prompts displayed when you run SELECT.

SET

The SET command adds, changes, displays, and deletes strings of characters from the MS-DOS environment during a work session. SET is an **internal** command.

Syntax

```
SET
```

Parameter	**Description**
name	name of a value in the MS-DOS environment to be set for the current work session.
value	value for *name.* To delete a previously SET *name* string, re-issue the SET command without the *value* parameter.

Operation

The SET command adds, changes, displays, and deletes values that affect the environment used by the MS-DOS command processor and application programs during a work session. The values set are stored in the system's RAM; thus, they're removed each time you restart your system.

The MS-DOS environment consists of a series of character strings. Each string consists of a *name* and a *value,* separated by an equal (=) sign. To insert a string into the MS-DOS environment, enter the SET command followed by a name, the equal sign, and a value.

Let's say that you use the SET command to insert the string FILE=FILE1.TXT into the MS-DOS environment. Now, if you enter the SET command without parameters, MS-DOS displays the current environment variables and their values:

```
COMSPEC=drive:\COMMAND.COM
FILE=FILE1.TXT
```

Notice that the MS-DOS environment contains the string `COMSPEC=`*drive:*`\COMMAND.COM` (*drive:* is the drive from which you started your system). This is a permanent string; the MS-DOS environment always contains it. It describes what and where the command processor is, and is used when all or part of the command processor needs to be reloaded.

Examples

To do this:	Type this:
Display the current environment variables and their values	`SET`
Insert the string FILE=FILE1.TXT into the MS-DOS environment	`SET FILE=FILE1.TXT`
Change the value of the existing string for FILE from FILE1.TXT to LETTERS	`SET FILE=LETTERS`
Delete string created in previous example	`SET FILE=`

Notes

1. When you execute the PATH and PROMPT commands, their environment variables are added to the MS-DOS environment. As a result, `SET PATH=` has the same effect as issuing the PATH command, and `SET PROMPT=` has the same effect as issuing the PROMPT command.
2. Environment variables added with the SET command can be reprogrammed in a batch file. Refer to the section on "Replaceable Parameters" in the chapter "Batch Processing" for additional information.
3. If you execute commands that load resident code (such as GRAPHICS, KEYB, MODE, and PRINT) before you execute the SET command, there may not be enough room in the MS-DOS environment to store the strings you specified. If there isn't enough room, MS-DOS displays the following message:

   ```
   Out of environment space
   ```

   To correct this situation, execute the SET command first, or increase the MS-DOS environment specified in your CONFIG.SYS file using the /E:*nnnnn* parameter of the COMMAND command.

SHARE

The SHARE command loads support for file sharing in a network environment. SHARE is an **external** command.

Syntax

```
SHARE [illegible]:locks
```

Parameter	Description
/F:*space*	reserves space (in bytes) for the MS-DOS storage area used to record file sharing information. Each file opened in share mode requires 11 bytes, plus the number of bytes (characters) in the path and file name. The default value is 2048 bytes.
/L:*locks*	reserves space for the number of locks allowed. The default value is 20 locks.

Operation

SHARE is normally used in a network environment where files will be shared between processes. If SHARE is installed, all read and write requests of an application are validated against the file sharing code in which the file was opened.

Caution Before attempting to use the SHARE command, check the documentation that came with your network to make sure this command is allowed. If it isn't allowed, the use of this command may cause you to lose data.

If a read or write operation takes place when a disk has been changed in the middle of the operation, SHARE looks at the volume label to determine if it is the same disk that was previously in the drive. If it is not, MS-DOS sends an invalid disk error message and requests that you insert the proper disk in the drive. When the proper disk is inserted, share allows the read/write operations to be completed successfully.

Support for file sharing can only be loaded once. If you attempt to load it more than once, the following error message is displayed:

```
SHARE already installed
```

The SELECT utility will load SHARE for you automatically if you have a hard disk which is greater than 32 MB in a single partition.

SHARE can be executed by using the INSTALL command. Refer to the chapter "System Configuration" for details on the INSTALL command.

Examples

To do this:	Type this:
Load support for file sharing using the default values for the number of shared files and locks allowed	`SHARE`
Load support for file sharing and increase the number of MS-DOS storage area to record file sharing information to 4096 bytes	`SHARE /F:4096`

Notes

1. When you execute the SHARE command, MS-DOS checks the FCB (File Control Block) control table. If you specified FCBS=4,0 (the default value) in the CONFIG.SYS file, the table is adjusted to 16,8. For additional information, refer to the FCBS command in the "System Configuration" chapter.

SORT

The SORT command reads data from the standard input device, sorts it, and writes the sorted data out to the standard output device. SORT is an **external** command.

Syntax

SORT /R /+*n* < *drive1:\path1\source* > *drive2:\path2\target*

or

source | SORT /R /+*n*

Parameter	**Description**
/R	sorts data in reverse order (that is, from Z to A, and then from 9 to 0).
/+*n*	specifies the column in which the sort is to begin. *n* must be a positive integer; the default for *n* is column 1.
< > and |	these symbols redirect data through SORT from a *source* to a *target*. For more information on redirecting, refer to the "Redirecting Input and Output" chapter.
drive1:	drive that contains the *source.*
path1	path to *source.*
source	name of an input command and/or file.
drive2:	drive that contains the *target.*
path2	path to the *target.*
target	name of an output file or device.

Operation

The SORT command is a filter program that lets you alphabetize a file according to the character in a certain column. For more on filters, refer to the "Redirecting Input and Output" chapter.

The SORT program uses the collating sequence table, based on the country code and code page settings.

SORT reads the data from the standard input device (such as output from a command), sorts it, and then writes it to the standard output device (usually, a file or the display screen). SORT can read up to 63 KB of input data at one time.

SORT is designed to sort text. Therefore, it works with lines of data (all of the characters on a line, including the CR/LF sequence). SORT arranges the lines of data such that the characters in column *n* ascend (or descend if the /R parameter is specified) in alphabetical order. If the *n*th character in two or more lines is the same, SORT compares the *n*th+1 character. SORT continues this compare process until the characters are different or the end of the line of data (CR/LF) is encountered.

The following example sends the output from the DIR command to the SORT filter. This filter sorts the directory listing starting with column 14 (the column in the directory listing that contains the file size) and displays it on your screen.

```
DIR | SORT /+14
```

The output might look like this (with the sorted file size in bytes shown across from the file name):

```
1313195D            0   12-09-88    3:44p
1313963            50   12-09-88    3:44p
FILE3     TXT     140   10-04-88    1:37p
FILE1     TXT     230   11-16-88   10:53a
MEMO1             261   11-16-88    2:17a
FILE2     TXT     676   03-03-88    1:39p
```

Examples

To do this:	Type this:
Add the MORE filter to the SORT command so you can read a large sorted directory one screen at a time	`DIR | SORT | MORE`
Alphabetically SORT a list of names in a file called LIST1.TXT and then place them in a new file called LIST2.TXT	`SORT < LIST1.TXT > LIST2.TXT`
SORT, in reverse alphabetical order, a list of names stored in a file called LIST1.TXT and then place them in a new file called LIST2.TXT	`SORT /R < LIST1.TXT > LIST2.TXT`
SORT the files in the current directory by the month they were created	`DIR | SORT /+23`
SORT the files in the current directory by the year they were created (earliest to latest)	`DIR | SORT /+29`

Notes

1. ASCII character codes 128 through 255 are sorted based on the country code in the CONFIG.SYS file. For additional information on country codes, refer to the COUNTRY command in the "System Configuration" chapter.
2. SORT doesn't distinguish between uppercase and lowercase letters.

SUBST (Substitute)

The SUBST command allows a new drive, called a **virtual drive,** to be substituted for an existing drive and path. SUBST is an **external** command.

Syntax

SUBST

or

SUBST *drive1:* /D

Parameter	**Description**
drive1:	name for virtual drive (E:, F: etc.). This is the new drive name to be substituted for the existing drive and path. This drive cannot be the active drive.
drive2:\path	existing drive and path. The virtual drive is to be substituted for the directory specified in *path.*
/D	deletes a substitution (a virtual drive).

Operation

The SUBST command allows you to substitute a virtual drive for an existing drive and path. This is useful if you have subdirectories that you use frequently or application programs that don't recognize paths.

The SUBST command creates a new drive, called a virtual drive, and substitutes it for an existing disk drive and path specified. As a result, the existing subdirectory becomes the root directory of the virtual drive.

To create a virtual drive, enter the SUBST command followed by the virtual drive, the existing drive, and the existing path. For example, the following command line:

```
SUBST E: C:\USER1\REPORTS
```

creates the virtual drive E:. The subdirectory REPORTS on drive C: is now the root directory of drive E:. To see the contents of the file REPORT1.TXT in the subdirectory REPORTS, all you have to enter is:

```
TYPE E:\REPORT1.TXT
```

instead of:

```
TYPE C:\USER1\REPORTS\REPORT1.TXT
```

Thus, you can think of a virtual drive as a "nickname" for an existing drive and path.

When a drive and path are substituted, the subdirectory specified by *path2* becomes the root directory of the virtual drive. However, the relationship between the specified subdirectory and its subdirectories remains the same. Therefore, in the above example, if the subdirectory REPORTS had a subdirectory called PAYROLL, you could reference PAYROLL with the path `E:\PAYROLL`.

The value assigned to the virtual drive (*drive1:*) must be less than or equal to the value assigned to the LASTDRIVE command in your CONFIG.SYS file. If you don't have a LASTDRIVE command in your CONFIG.SYS file, the default value for LASTDRIVE is drive E:. This means that you can specify drive A:, B:, C:, D:, or E: as the virtual drive. If you want to specify a drive greater than E:, you need to change the value assigned to LASTDRIVE first.

Examples

To do this:	Type this
Display the current substitutions	`SUBST`
Substitute the drive name E: for the subdirectory SALES (which is a subdirectory of the FORCAST) directory of drive C:	`SUBST E: C:\FORCAST\SALES`
Cancel a substitution that creates virtual drive E: from a subdirectory	`SUBST E: /D`

Notes

1. *drive1:* and *drive2:* cannot be the same; they must be different. Also, *drive1:* cannot be the same as the active drive.
2. Even though a drive and path are substituted, you can still reference them by specifying the old drive and path.
3. If you specify an existing physical drive as the virtual drive (*drive1:*), that physical drive is inaccessible while the substitution is in effect. For example, if you specify drive B: as the virtual drive, it is inaccessible while the substitution is in effect.
4. You can't use the SUBST command with a drive and/or path that has been redirected over a network. If you attempt to do this, MS-DOS displays the following error message:

   ```
   Cannot SUBST a network drive.
   ```
5. Do not use ASSIGN, BACKUP, CHKDSK, DISKCOMP, DISKCOPY FDISK, FORMAT, JOIN, LABEL, RECOVER, RESTORE, or SYS on a substituted drive.

SYS (System)

The SYS command transfers the hidden MS-DOS files to the disk in the specified target drive. SYS is an **external** command.

Syntax

SYS *drive:*

Parameter	**Description**
drive:	target drive. The drive to which the system files are to be transferred.

Operation

The SYS command transfers the two hidden MS-DOS system files to the disk in the target drive. This command allows you to:

- Put the MS-DOS operating system on an application work disk that was formatted with the /B parameter. (This means that your computer can start from this disk.)
- Upgrade or replace the MS-DOS operating system on a disk.

The first two allocatable contiguous sectors of the disk in the target drive must be unused (they're automatically left unused when you FORMAT a disk with the /B parameter).

If you attempt to transfer the files to a disk whose first two sectors are **not** contiguous, MS-DOS displays the following error message:

```
Destination disk is too fragmented
```

If you attempt to transfer the files to a disk that doesn't have enough room to hold the two files, MS-DOS displays the following error message:

```
No room for system on destination disk
```

Example

To do this:	Type this:
Transfer the two MS-DOS system files, IBMBIO.COM and IBMDOS.COM, from the active drive to drive C:	`SYS C:`

Notes

1. The two MS-DOS system files (IBMBIO.COM and IBMDOS.COM) must be the first two directory entries in the root directory of the active drive (the source drive).
2. The SYS command doesn't transfer the COMMAND.COM file. To copy this file to the disk in the target drive, use the COPY command.

TIME

The TIME command sets or displays the system time. TIME is an **internal** command.

Syntax

```
TIME [illegible]
```

Parameter	**Description**
hh:	1- or 2-digit number from 0 to 23 representing the hour.
mm:	1- or 2-digit number from 0 to 59 representing minutes.
ss:	1- or 2-digit number from 0 to 59 representing seconds.
.xx	1- or 2-digit number from 0 to 99 representing hundredths of a second.

Operation

The TIME command allows the system time to be set or displayed. MS-DOS uses the TIME and DATE values to record the exact time files are created or last updated. As you build your library of files, you will find this information both necessary and useful.

In order to set the time, you must at least enter the hours (*hh:*). The minutes, seconds and hundredths of a second (*mm:ss.xx*) are optional, but MS-DOS sets them to 00:00.00 if they aren't entered. When entering the time, remember that MS-DOS uses a 24-hour clock.

When you enter TIME without any parameters, MS-DOS responds by displaying the current system time and asking for a new time as follows:

```
Current time is 13:30:00.00
Enter new time:
```

You can enter a new system time or press (Enter) to retain the current system time.

MS-DOS checks for invalid times and separators. If you enter an invalid time or use an incorrect separator, the system responds with:

```
Invalid time
Enter new time:
```

MS-DOS repeats this response until you enter a valid time in the correct format.

Examples

To do this:	Type this
Set the system time to 1:30 p.m.	`TIME 13:30:00.00`
Reset the system time in the previous example forward to 2:00 p.m.	`TIME 14:`
Display the current system time	`TIME`

Notes

1. The TIME display format is determined by the COUNTRY command in the CONFIG.SYS file. To see the TIME display format for your computer, enter the TIME command without parameters.
2. The TIME command also changes the time kept by your computer's internal real-time clock.
3. If you start your system without PAM, the MS-DOS Shell, or an AUTOEXEC.BAT file, you are prompted to enter the time and date every time you start it. However, if you have an AUTOEXEC.BAT file, you aren't prompted for the time and date unless your AUTOEXEC.BAT file includes the TIME and DATE commands.

TREE

The TREE command displays the directory structure of a disk (the root directory and all of the subdirectories). It can also be used to display all of the files in the root directory and the subdirectories. TREE is an **external** command.

Syntax

```
TREE [illegible] /F /A
```

Parameter	**Description**
drive:	drive that contains the directory structure you want to display.
path	path to directory structure you want to display. If you don't specify a path, the directory structure of the entire disk is displayed.
/F	displays the names of the files in the root directory and the subdirectories.
/A	instructs TREE to use the graphics characters available on all code pages. Allows for faster printing.

Operation

The TREE command allows you to view the directory structure of a disk. For each directory found, MS-DOS displays the path to the directory, the names of its subdirectories, and (if you use the /F parameter) the names of its files.

Let's say you use the TREE command with the /F parameter to view a directory called USER1. MS-DOS would display information in the following format:

```
Directory PATH listing for Volume DOS4.0
Volume Serial Number is xxxx-xxxx

C:\USER1
│   FILE1.TXT
│   FILE2.TXT
│   FILE3.TXT
├───FORMS
│       FILE1.TXT
│       FILE2.TXT
└───REPORTS
        FILE1.TXT
```

In this example, the root directory has a directory called USER1. The directory USER1 has three files (FILE1.TXT, FILE2.TXT and FILE3.TXT) and two subdirectories of its own (FORMS and REPORTS). In addition, the subdirectory FORMS contains two files (FILE1.TXT and FILE2.TXT) and the subdirectory REPORTS contains one file (FILE1.TXT).

Examples

To do this:	Type this
Display the directory structure of drive C: starting at the root directory	`TREE C:\`
Display the directory structure and the file names of the directory USER1 on drive C:	`TREE C:\USER1 /F`
Redirect the output from the TREE command so that it goes to a printer instead of the display screen	`TREE C:\ /F > PRN`
Show the directory structure and all files on drive C:, but use the MORE filter to pause after each full screen of data is displayed	`TREE C:\ /F | MORE`

TYPE

The TYPE command displays the contents of a file to the standard output device (typically, this is your display). TYPE is an **internal** command.

Syntax

```
TYPE          file
```

Parameter	**Description**
drive:	drive that contains the file to be displayed.
path	path to the file to be displayed.
file	name of the file to be displayed.

Operation

The TYPE command allows you to view the contents of a file.

The TYPE command displays a file exactly as it was entered, with one exception: TAB characters are expanded to the current settings for tabs (generally eight spaces wide).

The TYPE command is primarily intended to display ASCII text files. As a result, binary files (program files) or data file that contains non-ASCII characters will display unusual things like bells, formfeeds, and escape sequence symbols.

TYPE cannot be used to alter the contents of a file, use Edlin or a word processing application, instead.

Examples

To do this:	Type this
Displays the contents of the file FILE1.TXT on the screen	`TYPE FILE1.TXT`
Redirect output from the TYPE command to send the contents of a file to a printer	`TYPE FILE1.TXT > PRN`
Display the contents of a file, using the MORE filter to pause after each full screen of data	`TYPE FILE1.TXT ¦ MORE`

Notes

1. Wildcards can't be used in the file name because TYPE can only display the contents of one file at a time. If you attempt to use wildcards, MS-DOS displays an error message.

VER (Version)

The VER command displays the MS-DOS version number on the screen. VER is an **internal** command.

Syntax

```
VER
```

VERIFY

The VERIFY command turns the disk write verify switch on and off. VERIFY is an **internal** command.

Syntax

```
VERIFY
```

or

```
VERIFY
```

Parameter	Description
`ON`	turns the verify switch ON. When the verify switch is ON, MS-DOS verifies that data is correctly written to disk.
`OFF`	turns the verify switch OFF. OFF is the default setting.

Operation

The VERIFY command allows you to turn the disk write verify switch ON and OFF, and to display the current status of the switch.

When the verify switch is ON, every time MS-DOS writes data to a disk during a work session, it verifies that the data is intact.

If MS-DOS is unable to successfully write data to disk when the verify switch is ON, MS-DOS displays an error message.

The VERIFY ON setting remains in effect until you turn off or restart your computer, you enter the VERIFY command with the OFF parameter, or an application program resets it (see the "Notes" section below for additional information).

To check the current status of the verify switch, enter the VERIFY command without parameters. For example, if the verify switch is ON and you enter the following command line:

```
    VERIFY
```

MS-DOS responds with:

```
VERIFY is on
```

Examples

To do this:	Type this
Turn the disk write verify switch OFF	`VERIFY OFF`
Turn the disk write verify switch ON	`VERIFY ON`
Display verify status	`VERIFY`

Notes

1. The verify switch can be turned ON and OFF through a system call from an application program. Thus, it's possible for VERIFY to be ON, even though you haven't issued the VERIFY command with the ON parameter.
2. Writing to disk takes longer when the verify switch is ON.

VOL (Volume)

The VOL command displays the volume label of a disk. VOL is an **internal** command.

Syntax

```
VOL
```

Parameter	Description
drive:	drive containing the disk whose label you want to display.

Operation

The VOL command displays the volume label and volume serial number of the disk in the specified drive. If you don't specify a drive, MS-DOS displays the volume label of the disk in the active drive. For example, if you enter

```
VOL
```

MS-DOS displays a message similar to the following:

```
Volume in drive C is HARDDISK
Volume serial number is xxxx-xxxx
```

If the disk doesn't have a label, MS-DOS displays the following message:

```
Volume in drive C has no label
Volume serial number is xxxx-xxxx
```

Notes

1. You can create a volume label with the FORMAT and LABEL commands. For additional information, see the FORMAT and LABEL command descriptions in this chapter.

XCOPY

The XCOPY command copies groups of files, including subdirectories and files in subdirectories. XCOPY is an **external** command.

Syntax

XCOPY [illegible] \file1 *drive2:\path2\file2* [illegible] D:*date* /E [illegible]

or

XCOPY *drive1:* [illegible] \file1 *drive2:\path2\file2* [illegible] D:*date* /E /M [illegible]

Parameter	Description
drive1:	source drive. This is the drive that contains the source files to be copied.
path1	path to the source file(s). If a path isn't specified, source files in the current directory are copied.
file1	name(s) of the source file(s). If a file name isn't specified, *.* (all files in the directory) is assumed.
drive2:	target drive. This is the drive to which the source file(s) are to be copied.
path2	path to the target file(s).
file2	name(s) of the target file(s). If a file name isn't specified, *.* (all files in the directory) is assumed.
/A	only copies source files whose archive bits are set to one. /A, unlike /M, does *not* modify the archive bits of the source files.
/D:*date*	only copies source files that have been modified on or after the date. The date syntax can vary depending upon the COUNTRY command in the CONFIG.SYS file. The default syntax is the U.S. format *mm-dd-yy*
/E	copies subdirectories even if they don't contain any files (empty subdirectories). /E can only be used with /S.

/M only copies source files whose archive bits are set to one. /M, unlike /A, clears (sets to zero) the archive bits of the source files after they are copied. Thus, /M allows XCOPY to be used to backup files.

/P prompts you with (Y/N?) before copying each source file. This allows you to confirm whether or not you want to create each target file.

/S copies source files in all the subdirectories of the specified directory, in addition to the files in the specified directory. The subdirectory structure (with the exception of empty subdirectories) is copied along with the files. To copy empty subdirectories, use /E in addition to /S. If you omit /S, only the files in the specified directory are copied.

/V verifies data as it is written to the disk in the target drive.

/W causes XCOPY to wait before starting to copy source files. XCOPY displays the following message:

```
Press any key to begin copying file(s)
```

You can insert a source and/or target disk and then press any key to continue. Or, you can press Ctrl C to cancel the XCOPY command and return to the MS-DOS prompt.

Operation

The XCOPY command allows you to copy files and subdirectories from one disk to another.

To copy a portion of the directory structure of a disk, enter the XCOPY command followed by the path to the portion of the directory structure you want to copy.

The XCOPY command doesn't disturb the existing directory structure on the target disk. In other words, the new files and subdirectories are added to any existing files and subdirectories.

XCOPY may prompt you to specify whether the target is a file or a directory. Type F to indicate a file or D to indicate a directory, and then press Enter to continue the XCOPY process.

Examples

To do this:	Type this
Copy all of the files in the current directory of the active drive to the current directory of drive A:	`XCOPY *.* a:`
Copy all the files in the root directory of drive A: to the root directory of drive C:	`XCOPY A:\ C:\`
Copy all of the files and directories from the disk in drive A: to the disk in drive B:, including empty directories	`XCOPY A:\ B:\ /S /E`
Copy everything in the directory USER1 to the disk in drive B:	`XCOPY A:\USER1 B:\USER1 /E /S`
Copy all files with the extension .TXT in the root directory of drive A: to drive B:, and rename all .TXT extensions to .LST	`XCOPY A:\*.TXT B:\*.LST`

Notes

1. If you want to copy files between disks of **different** formats, you should use XCOPY instead of DISKCOPY. Since DISKCOPY copies disks track-by-track, it requires your source and target disks to have the same format.
2. You must specify either the source drive (*drive1:*), the source path (*path1*), or the source file name (*file1*) with the command. You cannot omit all three.
3. If the path to the target files doesn't exist, XCOPY automatically creates it before it copies any files.
4. Unlike the COPY command, you can't use the XCOPY command to copy to or from system devices (for example, CON or LPT1).
5. The XCOPY command doesn't copy hidden source files or copy over read-only target files.

4

Batch Processing

As you use MS-DOS, you may find yourself executing certain command sequences frequently. You can automate these sequences with batch processing.

What Is Batch Processing?

Batch processing is a two-step process. The first step involves putting MS-DOS commands and/or the names of application programs into a single file, called a **batch file.** The second step involves executing the batch file as a command. When you execute the batch file, the batch processor in MS-DOS executes the commands and programs in the batch file one at a time; thus, saving you from individually executing them yourself.

In addition to MS-DOS commands and the names of application programs, the batch processor has its own set of special commands that can be included in batch files. These special commands provide a rudimentary **program language** which further enhances the capabilities of MS-DOS batch processing. You will find descriptions of these commands in this chapter.

There is one limitation to using batch processing: if an MS-DOS command prompts you for a response, you must type that response at the keyboard. You can't include the response in your batch file. This prevents you from using batch files for unattended system operations which require responses to command prompts.

Valid Batch File Names

A batch file name can be any valid MS-DOS file name. However, it must have a file name extension of .BAT.

In addition, the batch file name shouldn't be the same as an MS-DOS command or an application program. MS-DOS executes files in the following order: first .COM files, then .EXE files, and finally .BAT files. If you had two files in the same directory with the same name, but one had the extension of .COM (or .EXE) and the other .BAT, MS-DOS would execute .COM file unless you specified the .BAT extension in your command. Thus, you can create a batch file with the same file name as an MS-DOS command or application program; however, MS-DOS won't allow you to execute the batch file unless you specify both the batch file name and its .BAT extension.

Creating a Batch File

There are a number of ways to create a batch file. You can use:

- Any word processor or editor that stores text in ASCII format
- Edlin, the MS-DOS editor
- The MS-DOS COPY command

We'll use the COPY command in our examples, though you may also create a batch file with Edlin, or any word processor that creates unformatted files. You can use a form of the COPY command to create a file directly from the keyboard. To do this, enter the COPY command followed by the CON (console device) parameter and the batch file name (with a file name extension of .BAT).

As an example, let's create a sample batch file with the MS-DOS commands DIR and CHKDSK in it. We'll name the file BATCH1.BAT and use COPY to create it.

1. First type:

```
COPY CON BATCH1.BAT (Enter)
```

2. Then type the commands you want executed as part of the batch file:

   ```
   DIR Enter
   CHKDSK Enter
   ```

3. To save this as a file, after the last line, hold down the Ctrl key and press the Z key once. Then press Enter to copy these lines into the file called BATCH1.BAT.

Executing a Batch File

Executing a batch file is similar to executing an MS-DOS command. You simply enter the file name (with or without the .BAT extension) at the MS-DOS prompt. For example, to execute the batch file we just created (BATCH1.BAT), enter:

```
BATCH1
```

This batch file will then list the names of the files in the current directory on drive C:, then run a CHKDSK on drive C: to determine the amount of disk space available.

You can terminate the execution of a batch file by pressing either of the MS-DOS abort key sequences, Ctrl Break or Ctrl C. However, you are not returned directly to the MS-DOS prompt. Instead, the prompt:

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

appears on the screen. If you enter "Y", you are returned to the MS-DOS prompt. If you enter "N", execution of the batch file resumes with the next command in the batch file. The command that was executing when Ctrl Break or Ctrl C was pressed isn't resumed.

You can change the current directory and/or active drive during the execution of a batch file. MS-DOS "remembers" the directory and drive that contain the batch file. However, you can't remove the disk that contains the batch file while the batch file is executing. If you do remove the disk, you will receive a "`Not ready reading drive`" error, which means that you will either have to retry or abort the execution of your batch file.

Chaining Batch Files

Batch files can be **chained** for execution; that is, a batch file can contain the name of another batch file in it as a command to be executed. This allows a long string of commands to be assembled from several smaller batch files. There is no limit to the number of batch files that you can chain together.

The line containing the name of the second batch file you want to chain to must be the *last* command to be executed in the first batch file. That's because the batch processor doesn't return to the first batch file after it finishes executing the second batch file. If you want to return to the first batch file, use the CALL command.

The AUTOEXEC.BAT File

MS-DOS allows you to create a special batch file called AUTOEXEC.BAT. This file contains MS-DOS commands you want executed every time you start (or restart) your system. This file must reside in the root directory of the drive from which you start your computer. If you used SELECT to install MS-DOS on your hard disk drive C:, here's an example of what you may find in this file:

```
@ECHO OFF
SET COMSPEC=C:\DOS\COMMAND.COM
VERIFY OFF
PATH C:\DOS
APPEND /E
APPEND C:\DOS
PROMPT $P$G
C:\DOS\GRAPHICS
VER
DOSSHELL
```

These are commonly used commands which enhance your computer's ability to find command files, print screens, and run the menu-based Microsoft^{R} DOS Shell. Note that if you are using DOS Shell, the DOSSHELL command must be the last command in your AUTOEXEC.BAT file.

Replaceable Parameters

You can create batch files with replaceable parameters (or variables). These replaceable parameters allow you to alter the data used by the batch file each time it is executed. When you enter the batch file name, you specify the data you want to use. Then, as the batch file is executed, the data is substituted for the replaceable parameters.

You can specify replaceable parameters in two ways:

- %0 - %9 Positional Parameters
- Named Parameters

Using %0 - %9 Positional Parameters

You can insert ten **positional parameters** (%0 - %9) into batch files to specify replaceable parameters. %0 is always replaced by the drive name (optional) and the name of the batch file. Note, however, that the batch file name that replaces %0 does not include the file name extension (.BAT). %1 is replaced by the first data value specified, %2 is replaced by the second data value, and so on.

To illustrate the use of replaceable parameters, here is the contents of a batch file you can create that we will call BATCH2.BAT:

```
TYPE %0.BAT
DIR %1
CHKDSK %2
COPY %1*.* %2
```

To execute this batch file, you must specify the batch file name and two data values. The batch file name will replace %0 and the two data values will replace %1 and %2. For example, if you enter:

```
BATCH2 A: B:
```

BATCH2 replaces %0, A: replaces %1 and B: replaces %2. As a result, the following commands are executed:

```
TYPE BATCH2.BAT
DIR A:
CHKDSK B:
```

```
COPY A:*.* B:
```

The batch file itself isn't modified.

If the % character is contained in a name in the batch file, it must be entered as a double % (%%) to differentiate it from a replaceable parameter. For example, to specify the file FILE1%.EXE in a batch file, enter FILE1%%.EXE.

If there are more replaceable parameters in the batch file than data values with the command, the unspecified parameters are replaced with blanks. This may or may not cause problems as the batch file executes. For example, if the batch file BATCH2.BAT is invoked with the following:

```
BATCH2 A:
```

the second data value is omitted. Therefore, the batch file executes the following commands:

```
TYPE BATCH2.BAT
DIR A:
CHKDSK
COPY A:*.*
```

In this instance, drive B: (the second data value) is omitted. As a result, CHKDSK checks the active drive instead of drive B:. If drive C: happened to be the active drive, then all the files in the current directory of drive A: would also be copied to drive C:.

The number of positional parameters can be increased beyond ten by using the batch command SHIFT. This command is discussed later in this chapter.

Using Named Parameters

In addition to the ten positional parameters, you can use named parameters in batch files. **Named parameters** allow you to define replaceable parameters by name instead of by number.

Named parameters differ from positional parameters in that you don't specify their values with the command. Instead, MS-DOS retrieves the value of each named parameter from the MS-DOS environment.

Specifically, you use the MS-DOS SET command to put named parameters and their associated values in the MS-DOS environment. Then, you put the named parameter enclosed by percent signs (%) in a batch file. When you execute the batch file, the batch processor searches the MS-DOS environment for the named parameter. When it finds the named parameter, it substitutes the corresponding value.

For example, suppose we assign the value EXAMPLE.TXT to the name parameter FILE. The following command accomplishes this.

```
SET FILE=EXAMPLE.TXT
```

Then, the named parameter FILE is inserted in the batch file that contains the single line:

```
TYPE %FILE%
```

When this batch file is executed, the TYPE command displays the file EXAMPLE.TXT.

Note

The MS-DOS environment is lost when the system is turned off or restarted. Thus, after you start or restart your system *and* before you execute a batch file, you must use the SET command to reassign values to any named parameters contained in that batch file.

Batch Commands

This section discusses the batch commands. Batch commands are **internal** commands you can include in batch files along with regular MS-DOS commands and application program names to expand the capabilities of the batch processor. Use these commands only within batch files, unless otherwise noted.

Batch Command Syntax Conventions

Many batch commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the command and its parameters. This manual uses the following conventions for command syntax:

CAPS — Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Blue — Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics — Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that you should type the name of your file.

Enter all punctuation, such as commas, colons and equal signs, exactly as shown in the syntax.

CALL

The CALL command calls one batch file from within another batch file without ending the first (calling) batch file.

Syntax

CALL *drive:\path\file arguments*

Parameter	**Description**
file	name of the batch file you want to call. It must have a file name extension of .BAT.
arguments	the information passed to the called batch file.

Example

To do this:	Type this:
Call the batch file BATCH3.BAT with the argument FILE1.TXT	`CALL BATCH3 FILE1.TXT`

Notes

1. Don't use piping and redirection with the CALL command.
2. A batch file can call itself (recursion) as long as you make sure it eventually terminates.

ECHO

The ECHO command turns the screen display on or off during the execution of batch commands, or displays a message.

Syntax

```
ECHO ON
```

or

```
ECHO OFF
```

or

ECHO *message*

Parameter	Description
ON	instructs the batch processor to display all commands as they are executed. This is the default state.
OFF	suppresses the display of commands as they are executed. However, displays resulting from the execution of commands and application programs, themselves, still appear.
message	displays the message regardless of the current ECHO state (ON or OFF).

Operation

If you enter the ECHO command without parameters, MS-DOS displays the current state (ECHO ON or ECHO OFF).

Examples

To do this:	Type this:
In the first line of a batch file, enter the ECHO command so that when other commands in the batch file are executed MS-DOS only displays the result of each command	`ECHO OFF`
Enter a line in a batch file which displays the word "Processing" when the batch file is executed	`ECHO Processing`
Display the present state of the ECHO command	`ECHO`

Notes

1. The ECHO command can be executed at the MS-DOS prompt.
2. To prevent a command from echoing, put an "at" sign (@) in front of it. For example, to prevent the displaying of DIR C: or ECHO OFF, enter `@DIR C:` and `@ECHO OFF`.
3. To prevent MS-DOS from interpreting a character, put double quotes around it. For example, to echo the piping symbol (|), enter `ECHO "|"`.

FOR

The FOR command performs a command for a set of files.

Syntax

When used for batch processing:

`FOR %%`*variable* `IN` (*set*) `DO` *command*

When used for interactive processing:

`FOR %`*variable* `IN` (*set*) `DO` *command*

Parameter **Description**

variable a positional variable consisting of any single character. (To avoid a conflict with the positional parameters %0 - %9, don't use digits 0 - 9.) The variable is made equal to each value in the set.

set consists of a series of files separated by spaces, or a file name with one or more wildcards (* or ?). File names must be enclosed in parenthesis.

command a command that invokes an MS-DOS command or an application program. Each time the variable is made equal to a value in the set, the command after DO is executed. In most cases, the variable is included with the command.

Examples

To do this:	Type this:

Assign the variable %%F to all letter files (LET1, LET2, and LET3) in the current directory, then delete all of those files	`FOR %%F IN (LET1 LET2 LET3) DO DEL %%F`
	or
	`FOR %%F IN (LET?) DO DEL %%F`

Assign the variable %%B to all files with the extension .TXT in the current directory, then run them sequentially through an program called INDEX	`FOR %%B IN (*.TXT) DO INDEX %%B`

Notes

1. Only one FOR command can be included in each command.
2. Paths are allowed in the set of variables.
3. When you execute the FOR command from a batch file there must be two percent signs (%%) before the variable. When you execute the FOR command from the MS-DOS command prompt, you can only use one percent sign (%) before the variable.

GOTO

The GOTO command processes commands starting with the line **after** the specified label. You can use GOTO with IF for batch files with the IF-THEN branching capabilities.

Syntax

GOTO :*label*

Parameter	Description
label	a string of up to eight characters following GOTO which may not include spaces or other separators (such as semicolons or equal signs). If your batch file does not contain the *label,* the batch process terminates. Any line in a batch file that starts with a colon (:) is ignored during batch processing.

Examples

To do this:	Type this:

Send the program processor from the line GOTO TWO to the line following the *label* (:TWO)	`GOTO TWO`
	`:TWO`
	`REM Processing is complete.`

Send the program processor to the *label* (:END)—only if no errors occur when you format the disk in drive A:	`ECHO OFF`
	`:BEGIN`
	`FORMAT A:`
	`IF ERRORLEVEL 0 GOTO END`
	`ECHO An error occurred during formatting.`
	`:END`
	`ECHO End of batch file.`

IF

The IF command performs a command based on the result of a condition.

Syntax

IF NOT ERRORLEVEL *number command*

or

IF NOT *string1* == *string2 command*

or

IF NOT EXIST *drive:\path\file command*

Parameter **Description**

ERRORLEVEL *number*

this is a **condition** which is true if the previous program executed had an exit code equal to or greater than, *number*. (When a program finishes, it returns an exit code via MS-DOS.) You can use this condition to perform other tasks that are based on the previous program's exit code.

string1 == *string2*

this is a **condition** which is true if the two strings are identical. The strings cannot contain separators, such as commas, semicolons, equal signs, and spaces. Also, they must match exactly because the comparison is case-sensitive. Either or both strings may be replaceable parameters. For example, both of the following can be included in a batch file:

```
IF %1==EXAMPLE
IF %1==%2
```

When specifying no string, you must use double quotes, for example: IF "%1"==""

EXIST *drive:\path\file*

this is a **condition** which is true if the specified file exists. The file name can include wildcards.

command command to execute if the condition (parameters described above) is true. If the condition is false, MS-DOS ignores the command.

Example

To do this:	Type this:
Do ERRORLEVEL processing on a program to format a disk. Send the program processor to the label named END—only if no errors occur when you format a disk in drive A:	ECHO OFF :BEGIN FORMAT A: IF ERRORLEVEL 0 GOTO END ECHO An error occurred during formatting. :END ECHO End of batch file.
Create a batch file that uses the DIR command to find a file	@ECHO OFF REM Sample batch file IF "%1"=="" GOTO need_file IF NOT EXIST %1 GOTO no_file DIR %1 GOTO END :no_file ECHO File %1 does not exist GOTO END :need_file ECHO Please supply a file name :END

Notes

1. BACKUP, DISKCOMP, DISKCOPY, FORMAT, GRAFTABL, KEYB, REPLACE and RESTORE are the only MS-DOS commands that return an ERRORLEVEL. MS-DOS provides this facility for application programs that return exit codes. These exit codes can be tested by the ERRORLEVEL parameter of the IF command. Commands that produce exit codes are described in the "MS-DOS Command Descriptions" chapter.

PAUSE

The PAUSE command suspends the execution of a batch file and allows you to display a message (or an instruction to be performed) before resuming execution.

Syntax

PAUSE *comment*

Parameter	Description
comment	to display a special message. Unless ECHO is off, PAUSE will display this message before the "`Strike a key`" prompt.

Operation

The PAUSE command temporarily suspends the execution of a batch file. An optional message of up to 121 characters can be displayed. This is followed by the prompt:

```
Strike a key when ready ...
```

Pressing Ctrl Break or Ctrl C terminates the execution of the batch file. Pressing any other key resumes execution of the batch file.

Example

To do this:	Type this:
In a batch file, insert a line to pause the batch file execution and prompt the user with the words "Insert disk in drive A:"	`PAUSE Insert disk in drive A:`

REM

The REM command allows you to put comments in a batch file.

Syntax

REM *comment*

Parameter	**Description**
comment	a line of text to help you identify and remember what your batch file does. The message can contain up to 123 characters and will be displayed by MS-DOS if it is on the same line as the REM command. You can also use REM with out *comment* to add blank spaces for readability. If ECHO is OFF, the message isn't displayed.

Example

To do this:	Type this:
In a batch file, insert a line that tells other programmers that the batch file does an automatic backup of drive C: to drive A:	`REM This batch file backs up C: to A:`

SHIFT

The SHIFT command allows access to more than 10 replaceable parameters.

Syntax

```
SHIFT
```

Operation

Each time the SHIFT command is executed, the values assigned to the replaceable parameters are shifted; %1 becomes %0, %2 becomes %1, and so on. The tenth parameter entered with the command when the batch file was invoked is assigned to %9.

With careful planning and use of the SHIFT command, the limit of 10 replaceable parameters can effectively be removed. Note however, that even with the SHIFT command, only 10 parameters can be active at any one time. In addition, there is no way to retrieve parameters which have been **shifted out** (that is, once a SHIFT is executed, the parameter %0 that existed before the shift cannot be recovered). The following example demonstrates the SHIFT command. The values assigned to each replaceable parameter are shown below:

Before SHIFT	**After SHIFT**
%0 = Alpha	%0 = Beta
%1 = Beta	%1 = Gamma
%2 = Gamma	%2 = Delta
%3 = Delta	

The Alpha is lost, %3 is set to the previous value of %4, and if more than 9 parameters were entered with the command, %9 is set to the 10th one.

5

System Configuration

This chapter examines the various ways you can expand MS-DOS to suit the different system configurations and requirements of your HP computer.

Overview

MS-DOS is an extremely flexible operating system. It provides support for many different system components and configurations. This flexibility is the result of two different MS-DOS features: **configuration commands** and **device drivers.**

What Are Configuration Commands?

MS-DOS **configuration commands** uses configuration commands to set up your computer. The file that contains a list of these commands is called CONFIG.SYS. If you used SELECT to install MS-DOS on you hard disk drive C:, a CONFIG.SYS file was placed in the root directory. MS-DOS looks for this file and executes the commands in it when ever you start or restart your computer. Here is an example of what you may find in the CONFIG.SYS file:

```
BREAK=ON
BUFFERS=20
FILES=8
LASTDRIVE=E
SHELL=C:\DOS\COMMAND.COM /P /E:256
DEVICE=C:\DOS\ANSI.SYS
INSTALL=C:\DOS\FASTOPEN.EXE C:=(50,25)
```

These commands enhance the performance of your computer for such things as speed in reading files, creating virtual drives, modifying the colors of your display. You may change these and add others at any time by using Edlin or any other word processor that creates unformatted files. We'll examine the commands you may use in CONFIG.SYS in this chapter.

What Are Device Drivers?

Device drivers are special programs which add support for devices and peripherals, or enable you to use new features of the MS-DOS operating system.

There are two types of device drivers: character device drivers and block device drivers. **Character device drivers** (such as AUX, CLOCK$, CON, NUL, COM1, COM2, COM3, COM4, LPT1, LPT2, LPT3, and PRN) provide support for character devices, such as keyboards, display screens, modems, printers, and plotters.

Block device drivers (such as those for drive A:, B:, C:, and D:) provide support for block devices, such as disk drives and devices which emulate disk drives (virtual disks).

MS-DOS has several built-in character and block devices to provide basic functionality to your computer. A user can replace these or add other device drivers, called installable device drivers, by including a DEVICE configuration command in the CONFIG.SYS file.

Additional Device Drivers Provided With MS-DOS

MS-DOS provides several files which contain installable device drivers that you can use to enhance the functionality of your computer. For more information on using these files, refer to the "Installable Device Drivers Provided With MS-DOS" section in this chapter.

Configuration Command Descriptions

There are several MS-DOS configuration commands. Each operates in a manner similar to the MS-DOS internal commands. With the exception of the BREAK command, configuration commands cannot be used outside of the CONFIG.SYS file. If you enter a configuration command at the MS-DOS prompt, MS-DOS displays the following error message:

```
Bad command or file name
```

Configuration Command Syntax Conventions

Many configuration commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the command and its parameters. This manual uses the following conventions for command syntax:

CAPS — Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Blue — Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics — Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that you should type the name of your file.

... — An ellipsis indicates that you can repeat the parameters as many times as necessary.

Enter all punctuation (except the ellipsis), such as commas, colons and equal signs, exactly as shown in the syntax.

BREAK

The BREAK command enables or disables extended Ctrl Break and Ctrl C checking by MS-DOS.

Syntax

```
BREAK=ON
```

or

```
BREAK=OFF
```

Parameter **Description**

`ON` causes MS-DOS to check for Ctrl Break and Ctrl C during all MS-DOS operations.

`OFF` causes MS-DOS to check for Ctrl Break and Ctrl C only during standard input/output, standard printer, and standard auxiliary operations. OFF is the default setting.

Operation

MS-DOS automatically checks for a Ctrl Break or Ctrl C during all standard I/O, printer, and auxiliary operations. If extended Ctrl Break and Ctrl C checking are enabled, MS-DOS checks during all MS-DOS operations. This configuration command functions identically to the BREAK command.

Examples

To do this:	Type this in CONFIG.SYS:
Enable extended checking each time you start your system	`BREAK=ON`
Display status of BREAK	`BREAK`

BUFFERS

The BUFFERS command specifies the number of disk buffers that MS-DOS allocates in memory when you start your system.

Syntax

BUFFERS=*n*,*m* /X

Parameter **Description**

n number of buffers. Valid values are 1 to 99. (If /X is specified, the maximum number of disk buffers is 10,000 or the largest number of buffers that fit in memory, whichever is less.)

The default value is 2 except:

If you have a flexible disk drive greater than 360 KB, the default value is 3.

If your system memory size is greater than 128 KB, the default value is 5.

If your system memory size is greater than 256 KB, the default value is 10.

If your system memory size is greater than 512 KB, the default value is 15.

m maximum number of sectors that can be read or written in one I/O operation (1 to 8). The default is 1. When used **with** *n*, include comma separator.

/X places the buffers in expanded memory. This parameter fails if all expanded memory is used.

Operation

A **disk buffer** is a 528-byte storage location in memory that MS-DOS uses to hold data while reading and writing to disk.

The use of buffers can increase the speed of certain applications. As data is read from the disk, it is stored in a buffer. The buffers are used by MS-DOS in a manner that ensures that the most recently read data is in one of the buffers.

The performance of applications that perform a large number of random reads and writes on a data file can be increased by using additional buffers. As the number of buffers increases, the chances increase that the data requested by the application program is already in one of the buffers. Thus, MS-DOS can retrieve the data from memory instead of disk, which speeds up program performance.

For applications that perform mostly sequential read and write operations, there is little performance gain from an increased number of buffers.

You need to try different values to determine the number of buffers that yields the maximum performance. In general, applications that perform large amounts of random data access perform best with between 10 and 25 buffers. Systems with a large number of subdirectories perform best with between 20 and 30 buffers. Remember, though, that each buffer takes up approximately 528 bytes of system memory space, so the more buffers you have, the less memory you will have available for applications. If you have LIM 4.0 expanded memory available, use the /X parameter to store the buffers in expanded memory—this will help free up system memory for your applications.

Examples

To do this:	**Type this in CONFIG.SYS:**
Set the number of buffers to 10	`BUFFERS=10`
In your CONFIG.SYS file, specify a command to set the number of buffers to 20, and store buffers in expanded memory	`BUFFERS=20 /X`

Notes

1. Each buffer increases the resident size of MS-DOS by 528 bytes and reduces the amount of memory available to the application program by the same amount. If possible, use the /X parameter to store buffers in expanded memory.

2. If the number of buffers is too large, it takes more time to search through the buffers than to read the data from disk. As a result, in certain cases, additional buffers actually decrease system performance instead of increase it.

COUNTRY

The COUNTRY command selects the display format for the system time and date, currency symbol, and decimal separator based on the specified country. It also prepares two system code pages and selects one as the active system code page. For additional information on code pages, refer to the "How to Use Code Pages" appendix.

Syntax

```
COUNTRY=xxx,yyy,drive:\path\file
```

Parameter	Description
xxx	3-digit country code. This is the same code used by the telephone system. It is used to change the format for time, date, decimal separator, and currency symbol to match those of the country code you specify. The default value is 001 (United States). The country codes for each country are listed below:

Country	Country Code	System Code Pages
Arabic-speaking *	785	437
Australia	061	437, 850
Belgium	032	437, 850
Canadian-French	002	863, 850
Denmark	045	865, 850
Finland	358	437, 850
France	033	437, 850
Germany	049	437, 850
Hebrew-speaking *	972	437
Italy	039	437, 850
Japan *	081	932, 437
Korea *	082	934, 437
Netherlands	031	437, 850
Norway	047	865, 850

Country	Country Code	System Code Pages
China (PRC) *	086	936, 437
Portugal	351	860, 850
Spain	034	437, 850
Sweden	046	437, 850
Switzerland	041	437, 850
Taiwan *	088	938, 437
United Kingdom	044	437, 850
United States	001	437, 850

* These code pages can only be used with country supplement hardware and the appropriate localized version of DOS.

yyy system code page you want to use when you start your system. MS-DOS automatically prepares two system code pages for each country; *yyy* specifies which of the two you want to use. The default value is country dependent; it's the first value listed in the table above under "System Code Pages." When used with *xxx,* include comma separator.

For a list of the characters in each code page, refer to the appendix "How to Use Code Pages."

drive:\path\file name of the file that contains all the country information. The default file name is COUNTRY.SYS. If the location of your file is not in the root directory, you will also need to specify a drive and path to the file. When used **without** *yyy,* include extra comma, for example:

```
COUNTRY=032,,C:\DOS\COUNTRY.SYS
```

Examples

To do this:	Type this:
In your CONFIG.SYS file, specify a command to select the Norwegian country code, but leave the code page and country information file at the defaults (865 and COUNTRY.SYS)	`COUNTRY=047`
In your CONFIG.SYS file, specify a command to select the Belgian country code, and specify a the configuration information file which is located in the DOS directory of drive C:	`COUNTRY=032,,C:\DOS\COUNTRY.SYS`

Notes

1. The COUNTRY command doesn't translate MS-DOS messages or prompts, or change the video or keyboard character set.
2. If your country isn't listed above, pick the country code that most closely matches your needs.
3. Country codes 358 and 972 assume United States code page 437, but include country-specific date and time conventions.

DEVICE

The DEVICE command allows MS-DOS to load an installable device driver onto the system by specifying the file that contains it.

Syntax

```
DEVICE=drive:\path\file parm1 ...
```

Parameter	Description
drive:	drive that contains the device driver.
path	path to the file that contains the device driver.
file	name of the file that contains the device driver.
parm1	first parameter to be passed to the device driver.

Operation

The DEVICE command allows you to specify a file containing an installable device driver (for more about device drivers, refer to the "Overview" section at the beginning of this chapter). **Installable device drivers** allow you to support additional devices and reconfigure existing devices. MS-DOS loads the device driver from this file and incorporates it into the resident portion of MS-DOS.

MS-DOS provides several installable device drivers, for more information on these, refer to the section "Installable Device Drivers Provided With MS-DOS" in this chapter.

Examples

To do this:	Type this in CONFIG.SYS:
Load the ANSI.SYS installable driver to enable you to use ANSI escape sequences	`DEVICE=ANSI.SYS`
Load the RAMDRIVE.SYS installable driver (located in the DOS directory of drive C:)	`DEVICE=C:\DOS\RAMDRIVE.SYS`

Notes

1. The device drivers COUNTRY.SYS and KEYBOARD.SYS are loaded automatically by MS-DOS. Do not try to load either of these with the DEVICE command. If you do, it will "hang" or lock your system.
2. The size of the resident portion of MS-DOS is increased by the size of each device driver installed.
3. You can create your own files which contain installable device drivers. Refer to Microsoft's *MS-DOS Programmer's Reference* manual for additional information.

DRIVPARM

The DRIVPARM command allows you to define parameters for block devices (such as disk or tape drives) when you start your computer, overriding the original MS-DOS device driver settings.

Syntax

DRIVPARM=/D:*number* /C /F:*factor* /H:*heads* /I /N /S:*sectors* /T:*tracks*

Parameter **Description**

/D:*number* physical drive number, ranging from 0 to 255. This means that drive number 0 is A:, 1 is B:, 2 is C:, etc.

/C indicates that the physical disk drive referenced has change-line support in the hardware. This means that the physical disk drive can sense that the drive door has been opened. If the door is open, the device driver will assume that the drive does not have a disk in it yet. If you specify this switch, MS-DOS assumes that the physical disk drive has the ability to support change-line error detection. See the documentation for your disk drive to see if it supports change-line.

/F:*factor* specifies the device type (form factor). The default value is 2. *factor*=form factor index, where:

0 = 160/180K bytes, or 320/360K bytes
1 = 1.2 megabytes
2 = 720K bytes (3.5-inch disk)
5 = Hard disk
6 = Tape drive
7 = 1.44 megabytes (3.5-inch disk)

The default values for the following switches depend upon the form factor specified with /F. If you do not specify /F, DRIVPARM uses a default of 720K bytes (3.5-inch disk).

/H:*heads* maximum head number, ranging from 1 to 99. The default value is 2.

`/I` specifies an electrically-compatible 3.5-inch disk drive. (Electrically-compatible drives are installed in your computer and use your existing flexible disk controller.) The drive is connected to the flexible disk controller as either A: or B:. Use /I if your computer's ROM BIOS does not support 3.5-inch disk drives. (Some older HP Vectra systems do not have a ROM BIOS that supports 3.5-inch disk drives.)

`/N` specifies a nonremovable block device (hard disk).

`/S:`*sectors* number of sectors per track, ranging from 1 to 99. The default value is 9.

`/T:`*tracks* number of tracks per side on the block device, ranging from 1 to 999.

Operation

This command allows you to define parameters for block devices when you start MS-DOS, overriding the original MS-DOS device driver settings. If you want to create a new logical drive and associate it with a physical drive, see the DRIVER.SYS section in this chapter. By contrast, DRIVPARM will modify the parameters of an existing physical drive and not create a new logical drive.

Example

To do this:	**Type this in CONFIG.SYS:**
To make tapes to be used in another type of tape drive, override the default device driver settings on a your tape drive (which is set to write 20 tracks of 40 sectors) to now write 10 tracks of 99 sectors each (this assumes that your tape drive supports this new configuration)	`DRIVPARM=/D:3 /F:6 /H:1 /S:99 /T:10`

FCBS

The FCBS command specifies the number of files that can be concurrently open via FCBs (File Control Blocks).

Syntax

```
FCBS=x,y
```

Parameter	Description
x	maximum number of files that can be open at any one time via FCBs. Valid values are 1 to 255; the default value is 4 files. *x* must be greater than or equal to *y*.
y	number of open files that MS-DOS cannot close automatically if an application program tries to open more than *x* files via FCBs. This is also known as the number of **protected files** because they're protected from being closed. Valid values are 1 to 255; the default value is 0 files.

Operation

MS-DOS provides two ways to open and access files: FCBs and Handles. The FCBS command allows you to specify the number of files that can be open (accessed) by way of FCBs. The FILES command allows you to specify the number of files that can be open by way of Handles. For additional information on FCBs and Handles, refer to the *MS-DOS Programmer's Reference* manual.

MS-DOS keeps track of which file opened via an FCB is the least recently used. If an application program attempts to open more than *x* files, the action taken by MS-DOS depends on whether file-sharing is loaded.

If file-sharing is loaded (the SHARE command has been executed), MS-DOS closes the least recently used file prior to opening the new file. The *y* parameter protects the first *y* files from being closed in this manner. If *y* is set equal to *x*, MS-DOS doesn't close any files when an application program tries to open more than *x* files. Instead, the application program is unable to open the new file.

FCBS

If file-sharing isn't loaded, there isn't a limit to the number of files that can be open concurrently via FCBs. Thus, no action is taken by MS-DOS provided the SHARE command hasn't been executed.

Example

To do this:	Type this in CONFIG.SYS:
Allow 10 files to be open concurrently via FCBs, plus protect the first 5 files	`FCBS=10,5`

Notes

1. If an application program uses more than one FCB to refer to the same file, MS-DOS still considers this as only one FCB used.
2. The FCBS command increases the size of the resident portion of MS-DOS.

FILES

The FILES command specifies the number of files that can be concurrently open via Handles.

Syntax

```
FILES=n
```

Parameter **Description**

n maximum number of files that can be open at any one time via handles. Possible values are 8 to 255; the default value is 8 files.

Operation

MS-DOS provides two ways to open and access files: Handles and FCBs (File Control Blocks). The FILES command allows you to specify the number of files that can be open by way of Handles. The FCBS command allows you to specify the number of files that can be open by way of FCBs. For additional information on Handles and FCBs, refer to Microsoft's *MS-DOS Programmer's Reference* manual.

Example

To do this:	Type this in CONFIG.SYS:
Allow 10 files to be concurrently open via handles	`FILES=10`

Notes

1. Each Handle over 8 increases the resident size of MS-DOS by 48 bytes.
2. A process can open up to 65,534 files. However, the total number of unique files that can be open on the system at one time is *n*.

INSTALL

Executes a FASTOPEN, KEYB, NLSFUNC, or SHARE command during CONFIG.SYS processing.

Syntax

INSTALL=*drive:\path\file parm1 ...*

Parameter	Description
file	name of file must be FASTOPEN.EXE, KEYB.EXE, NLSFUNC.EXE, or SHARE.EXE.
parm1	the first optional command parameter to pass to the file.

Operation

With the INSTALL command, you can execute a FASTOPEN, KEYB, NLSFUNC, or SHARE command when CONFIG.SYS is being processed.

Example

To do this:	Type this in CONFIG.SYS:
Add INSTALL command to the FASTOPEN command to prevent FASTOPEN from being reissued each time you run AUTOEXEC.BAT	`INSTALL=FASTOPEN.EXE C:100`

LASTDRIVE

The LASTDRIVE command specifies the last valid drive. This sets the maximum number of drives that can be accessed on the system.

Syntax

```
LASTDRIVE=x
```

Parameter	Description
x	the last valid drive. Possible values are A to Z; the default value is drive E (5 drives).

Example

To do this:	Type this in CONFIG.SYS:
Specify drive G: as the last valid drive on your system	`LASTDRIVE=G`

Notes

1. Each block device driver, disk drive, and subdirectory assigned via the SUBST command requires a drive name. If the total of these exceeds 5, you need to put a LASTDRIVE command in the CONFIG.SYS file to extend the last valid drive.
2. On a computer that is not connected to a network, if the last valid drive isn't high enough to accommodate the block device drivers and disk drives, MS-DOS ignores the specified value and determines the last valid drive for you. However, when determining the value of the last valid drive, MS-DOS doesn't take into consideration any subdirectories assigned via the SUBST command.

REM

The REM command allows comments to be entered in the CONFIG.SYS file.

Syntax

REM *comment*

Parameter	**Description**
comment	is any string.

Operation

The REM command allows comments to be place in CONFIG.SYS which you might use for clarification of a particular command.

Example

To do this:	Type this in CONFIG.SYS:

Identifies the country code used in the COUNTRY command	`REM Set country code to France`
	`COUNTRY=033`

SHELL

The SHELL command specifies the shell (top-level command processor) your computer will start from.

Syntax

SHELL=*drive:\path\file parm1* ...

Parameter	Description
drive:	drive that contains the alternate command processor.
path	path to the file that contains the alternate command processor.
file	name of the file that contains the alternate command processor. The default for MS-DOS is COMMAND.COM.
parm1	first parameter to be passed to the alternate command processor.

Operation

The SHELL command installs a command processor. MS-DOS sets the COMSPEC environment variable equal to the *drive:\path\file* specified in the SHELL command in CONFIG.SYS. The default command processor is COMMAND.COM, but system programmers who write their own command processors should use the SHELL command to specify the name of their shell program.

The SHELL command does not have any parameters. However, if the new command processor does have parameters, you can include those parameters in the syntax.

Examples

To do this:	Type this in CONFIG.SYS:
Substitute, for COMMAND.COM, a new command processor located in the root directory of drive C: called PAMCODE.COM that uses a parameter called ROOT	`SHELL=C:\PAMCODE.COM ROOT`
Substitute, for COMMAND.COM, a new command processor located in the NEWDOS directory of drive C: called NEWCMD.COM that uses a parameter called /P	`SHELL=C:\NEWDOS\NEWCMD.COM /P`

Notes

1. If you're a system programmer and you're developing your own top-level command processor, remember to include provisions for handling interrupts 22H, 23H, and 24H, and for reading and executing commands. The batch processor and internal commands reside in COMMAND.COM. As a result, they must be duplicated in your command processor or they won't be available to the user.

2. The operating system uses the COMSPEC environment setting to determine which file to use when reloading any transient part of the command processor. To make sure that the same command processor is used for reloading (the transient portion only), use the SET command with COMSPEC to point to that command processor.

STACKS

The STACKS command allows you to override the default number of stack frames and the default size of each stack frame.

Syntax

```
STACKS=n,s
```

Parameter	Description
n	number of stack frames. Valid values are 0, and 8 to 64; the default value is 0 for 8088- and 8086-based systems (such as the Vectra CS) and 9 for all other systems.
s	size (in bytes) of each stack frame. Valid values are 0, and 32 to 512; the default value is 0 for 8088- and 8086-based systems (such as the Vectra CS) and 128 for all other systems.

Operation

Each time a hardware interrupt occurs, MS-DOS uses a stack frame from the stack pool. After MS-DOS finishes processing the interrupt, it returns the stack frame to the stack pool. If a number of hardware interrupts occur in rapid succession, MS-DOS may need to use more stack frames than it has available. The STACKS command allows you to increase the number and size of stack frames, and thus resolve this resource problem.

Caution

Do not decrease the number of stack frames or the size of each stack frame below the default value. Doing so could cause a system failure or unexpected results.

Example

To do this:	Type this in CONFIG.SYS:
Increase the number of stack frames to 20	`STACKS=20,128`

Notes

1. When you increase stack resources, you decrease the amount of available memory. For this reason, we recommend that you increase the number of stacks first, without increasing the size of each stack frame. Then, if that doesn't solve the problem, increase the size of each stack frame.

Installable Device Drivers Provided With MS-DOS

MS-DOS provides these installable device drivers that you can use with the DEVICE command and enable you to enhance the functionality of your computer:

- ANSI.SYS contains a character device driver that allows you to use the ANSI standard terminal escape sequences to extend the screen and keyboard display features of your system.
- DISPLAY.SYS contains a character device driver that supports code page switching on the console device.
- DRIVER.SYS contains a block device driver that allows you to access a physical drive by referencing a logical drive.
- HIMEM.SYS contains a block device driver that allows MS-DOS programs on some 80286 and 80386 computers to utilize a 64 KB region of memory located just above the 1 MB boundary.
- INDSKBIO.SYS contains a block device driver that provides your HP Vectra personal computer compatibility with 3.5-inch disks used with the **HP 150, HP 150II, HP 110,** and **HP Portable PLUS** personal computers. It also provides internal 3.5-inch disk drive support for early model HP Vectras.
- PRINTER.SYS contains a character device driver that supports code page switching on the following devices: PRN, LPT1, LPT2, and LPT3.
- RAMDRIVE.SYS contains a block device driver that allows you to convert system RAM into a virtual disk.

A detailed description of each driver is contained in this section.

ANSI.SYS and Escape Sequences

The ANSI.SYS installable device driver lets you use ANSI escape sequences. An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) developed by the American National Standards Institute (ANSI). You can use these sequences to define functions for MS-DOS. Specifically, you can change graphics functions and affect the movement of the cursor.

Syntax

DEVICE=*drive*:\path\\ANSI.SYS /X

Parameter	**Description**
drive:	drive where ANSI.SYS is located.
path	path to ANSI.SYS.
/X	allows expanded keys to be remapped independently. On the 101-key keyboard, for example, there are two (Home) keys. One (Home) key is on the numeric keypad and the other is in the block of cursor-control keys. The two (Home) keys are considered to be the same unless the /X switch is specified.

Operation

When this device driver is installed, it replaces the built-in CON device driver. As a result, ANSI standard terminal escape sequences are executed by the standard input and output devices. Applications, programs, and batch files which send output to the standard device (CON) can send these escape sequences. When installed, ANSI.SYS will interpret them and perform the specified function.

ANSI Escape Sequences

The following parameters appear in ANSI escape sequences:

Parameter	**Description**
esc	ASCII escape code—all escape sequences begin with this character. *esc* is 1BH or 27 decimal. To generate the escape code, (1) enter the MS-DOS PROMPT command followed by `$e` or (2) use an application or a programming language such as GW-BASIC. **Do not** type the letters `ESC` or press ESC.
`[`	this character is the second character in the sequence.
Pn	Numeric parameter—a decimal number that you specify with ASCII digits.
Ps	Selective parameter—a decimal number that you use to select a subfunction. You may specify more than one subfunction by separating the parameters with semicolons.
Pl	Line parameter—a decimal number that you specify with ASCII digits.
Pc	Column parameter—a decimal number that you specify with ASCII digits.

This section lists and explains valid ANSI escape sequences for MS-DOS:

Escape Sequence	**Function**
esc `[` *Pl;Pc* `H`	Cursor Position (CUP)
esc `[` *Pl;Pc* `f`	Horizontal & Vertical Position (HVP) CUP and HVP move the cursor to the position specified by the parameters. When no parameters are provided, the cursor moves to the home position (the upper-left corner of the screen).
esc `[` *Pn* `A`	Cursor Up (CUU) This sequence moves the cursor up *Pn* lines without changing columns. If the cursor is already on the top line, MS-DOS ignores the CUU sequence.

esc [*Pn* B

Cursor Down (CUD)
This sequence moves the cursor down *Pn* lines without changing columns. If the cursor is already on the bottom line, MS-DOS ignores the CUD sequence.

esc [*Pn* C

Cursor Forward (CUF)
The CUF sequence moves the cursor forward *Pn* columns without changing lines. If the cursor is already in the far right column, MS-DOS ignores the CUF sequence.

esc [*Pn* D

Cursor Backward (CUB)
This escape sequence moves the cursor back *Pn* columns without changing lines. If the cursor is already in the far left column, MS-DOS ignores the CUB sequence.

esc [6 n

Device Status Report (DSR)
The console driver outputs an RCP sequence when it receives the DSR escape sequence.

esc [s

Save Cursor Position (SCP)
The console driver saves the current cursor position. The RCP sequence can restore this position.

esc [u

Restore Cursor Position (RCP)
This sequence restores the cursor position to the value it had when the console driver received the SCP sequence.

esc [2 J

Erase Display (ED)
The ED sequence erases the screen. The cursor then goes to the home position.

esc [K

Erase Line (EL)
This sequence erases from the cursor to the end of the line (including the cursor position).

esc [*Ps; . . . ;Ps* m

Set Graphics Rendition (SGR)
The SGR escape sequence calls the graphic functions specified by the following numeric parameters. These

functions remain until the next occurrence of an SGR escape sequence.

Graphics Functions:

0 All attributes off
1 Bold on
4 Underscore (monochrome display only)
5 Blink on
7 Reverse video on
8 Concealed on

Foreground colors:
30 Black
31 Red
32 Green
33 Yellow
34 Blue
35 Magenta
36 Cyan
37 White

Background colors:
40 Black
41 Red
42 Green
43 Yellow
44 Blue
45 Magenta
46 Cyan
47 White

Parameters 30 through 47 meet the ISO 6429 standard.

esc [= *Ps* h
esc [= h
esc [= 0 h
esc [? 7 h

Set Mode (SM)
The SM escape sequence changes the screen width or type to one of the following numeric parameters:

Screen width parameters

0	40 25 B&W
1	40 25 color
2	80 25 B&W
3	80 25 color
4	320 200 color
5	320 200 B&W
6	640 200 B&W
7	Wraps at the end of each line
14	640 200 color
15	640 350 mono
16	640 350 color
17	640 480 color
18	640 480 color
19	320 200 color

esc `[ =` *Ps* `l`
esc `[ = l`
esc `[ = 0 l`
esc `[ ? 7 l`

Reset Mode (RM)
Parameters for RM are the same as for SM (Set Mode), except parameter 7 resets the mode that causes wrapping at the end of each line.

esc `[`*code;string;* ... `p`

Allows redefinition of keyboard keys to a specified string where:

string is either the ASCII code for a single character or a string contained in quotation marks. For example, both 65 and A can be used to represent an uppercase A.

code is one or more of the following values that represent keyboard keys. Semicolons shown in this table must be entered in addition to the required semicolons in the command.

Key	Code Alone	Code Shift	Code Ctrl	Code Alt
F1	0;59	0;84	0;94	0;104
F2	0;60	0;85	0;95	0;105
F3	0;61	0;86	0;96	0;106
F4	0;62	0;87	0;97	0;107
F5	0;63	0;88	0;98	0;108
F6	0;64	0;89	0;99	0;109
F7	0;65	0;90	0;100	0;110
F8	0;66	0;91	0;101	0;111
F9	0;67	0;92	0;102	0;112
F10	0;68	0;93	0;103	0;113
F11	0;133	0;135	0;137	0;139
F12	0;134	0;136	0;138	0;140
Home	0;71	55	0;119	—
Up Arrow	0;72	56	—	—
Pg Up	0;73	57	0;132	—
Left Arrow	0;75	52	0;115	—
Down Arrow	0;77	54	0;116	—
End	0;79	49	0;117	—
Pg Dn	0;81	51	0;118	—
Ins	0;82	48	—	—
Del	0;83	46	—	—
PrtSc	—	—	0;114	—
A	97	65	1	0;30
B	98	66	2	0;48
C	99	67	3	0;46
D	100	68	4	0;32
E	101	69	5	0;18
F	102	70	6	0;33
G	103	71	7	0;34
H	104	72	8	0:35
I	105	73	9	0;23
J	106	74	10	0;36
K	107	75	11	0;37

Key	Code Alone	Code Shift	Code Ctrl	Code Alt
L	108	76	12	0;38
M	109	77	13	0;50
N	110	78	14	0;49
O	111	79	15	0;24
P	112	80	16	0;25
Q	113	81	17	0;16
R	114	82	18	0;19
S	115	83	19	0;31
T	116	84	20	0;20
U	117	85	21	0;22
V	118	86	22	0;47
W	119	87	23	0;17
X	120	88	24	0;45
Y	121	89	25	0;21
Z	122	90	26	0;44
1	49	33	—	0;120
2	50	64	—	0;121
3	51	35	—	0;122
4	52	36	—	0;123
5	53	37	—	0;124
6	54	94	—	0;125
7	55	38	—	0;126
8	56	42	—	0;127
9	57	40	—	0;128
0	48	41	—	0;129
-	45	95	—	0;130
=	61	43	—	0;131
Tab	9	0;15	—	—
Null	0;3	—	—	—

Examples

To do this:	Type this:
Using the PROMPT command, exchange the backslash and question-mark keys using literal strings to denote the keys	`PROMPT $e["\";"?"p$e["?";"\"p`
Using the PROMPT command, exchange the backslash and question-mark keys using each key's ASCII value to denote the key	`PROMPT $e[92;63p$e[63;92p`

Using the PROMPT command, restore the backslash and question-mark keys to their original meanings	`PROMPT $e[92;63p$e["?";"\"[`
	or
	`PROMPT $e[92;92p$e[63;63p`

Using the PROMPT command, make it so that all commands typed at the prompt are displayed in inverse blue	`PROMPT $e[37;44m`
Using a programming language, create a batch command with an escape sequence to home the cursor, clear the display, change to reverse video and print the message "Hello"	@ECHO *esc*[2J*esc*[12;40m*esc*[7m Hello

DISPLAY.SYS

DISPLAY.SYS is an installable device driver that supports code page switching for the console device.

Syntax

```
DEVICE=drive:\path\DISPLAY.SYS CON:= (type,hwcp,n,m)
```

Parameter	Description
drive:	drive that contains DISPLAY.SYS.
path	path to DISPLAY.SYS.
type	type of video/graphics adapter card. Valid types are `MONO`, `CGA`, and `EGA`. If you have a VGA card, use EGA as the type.
hwcp	code page built into the video/graphics adapter card. Valid values are 437 (United States), 850 (Multilingual), 860 (Portugal), 863 (French-Canadian), and 865 (Norway). In most cases, code page 437 is built into the card.
n	number of additional code pages that can be supported. Valid values are 0 to 12. This value is hardware dependent. The value for MONO and CGA must be 0 (because they can't support any additional code pages). The default value for EGA is 1. When used **without** *hwcp,* include the extra comma separator, for example: `DEVICE=C:\DOS\DISPLAY.SYS CON:=(EGA,,2)`
m	number of sub-fonts that can be supported for each code page. Values are 0 and 2. MONO and CGA must be 0 (because they can't support any additional code pages). The default value for EGA is 2. You cannot use *m* without *n.*

Operation

DISPLAY.SYS provides support for code page switching on the console (CON) device. For additional information on code page switching, refer to the appendix "How to Use Code Pages."

Examples

To do this:	Type this in CONFIG.SYS:
Tell MS-DOS that you have an EGA card with code page 437 built-in and you want to support 2 additional code pages	`DEVICE=DISPLAY.SYS CON:=(EGA,437,2)`
Tell MS-DOS that you have an EGA card and want to support 2 additional code pages, but your DISPLAY.SYS file is in the DOS directory of drive C:	`DEVICE=C:\DOS\DISPLAY.SYS CON:=(EGA,,2)`

Notes

1. If you want to include both the ANSI.SYS and DISPLAY.SYS device drivers in your CONFIG.SYS file, the ANSI.SYS driver must come before the DISPLAY.SYS driver.

DRIVER.SYS

DRIVER.SYS provides the ability to access a physical drive by referencing a logical drive, and loads support for external disk drives.

Syntax

`DEVICE=`*drive:*\path\\`DRIVER.SYS /D:`*number* `/T:`*tracks* `/S:`*sectors* `/H:`*heads* `/F:`*factor* `/N /C`

Parameter **Description**

drive: drive that contains DRIVER.SYS.

path path to DRIVER.SYS.

`/D:`*number* specifies the physical drive number. Valid values for *number* are 0 to 255. Flexible and hard disk drives are numbered separately. Flexible disk drive numbers start with 0; hard disk drive numbers start with 128. The most commonly used values are listed below:

0 = first internal flexible disk drive
1 = second internal flexible disk drive
2 = first external flexible disk drive
3 = second external flexible disk drive
128 = first hard drive
129 = second hard drive

`/T:`*tracks* specifies the number of tracks per side. Valid values for *tracks* are 1 to 999; the default value is 80 tracks per side.

`/S:`*sectors* specifies the number of sectors per track. Valid values for *sectors* are 1 to 99; the default value is 9 sectors per track.

`/H:`*heads* specifies the maximum number of heads. Valid values for *heads* are 1 to 99; the default value is 2 heads.

`/F:`*factor* specifies the device type. Valid values for *factor* are 0, 1, 2, and 7; the default value is device type 2. Device types are:

0 for a 160/180 or 320/360 KB drive
1 for a 1.2 MB drive
2 for a 720 KB drive
7 for a 1.44 MB drive

`/N` indicates the physical device is a hard drive (or any other non-removable block device).

`/C` indicates changeline (doorlock) support is required. This means that the device driver will be able to tell whether the door of the flexible disk drive is open or closed.

Operation

DRIVER.SYS allows you to copy files to and from the same disk drive. This is useful if you have more than one flexible disk drive on your system and the drive types are different.

It also allows you to load support for external disk drives that don't come with their own software drivers.

Let's say you have two flexible disk drives (with drive A: being 1.2 MB drive, and drive B: a 360 KB drive) and a hard drive (drive C:). To be able to reference the first flexible disk drive as drive A: or drive D:, enter the following command in the CONFIG.SYS file:

```
DEVICE=DRIVER.SYS /D:0 /T:80 /S:15 /H:2 /C /F:1
```

Now, you can copy files from drive A: to drive D: by simply switching the flexible disks in the 1.2 MB drive. For example, you can enter the command:

```
COPY FILE1.TXT D:
```

Examples

To do this:	Type this in CONFIG.SYS:
Assign the 1.2 MB drive A: the drive letters A: and D: (this command will assign the next available drive letter which, in this case, happens to be D:)	`DEVICE=DRIVER.SYS /D:0 /T:80 /S:18 /H:2 /C /F:7`
Load support for your first external disk drive, a 720 KB drive, (MS-DOS will assign the next available drive letter to it). In addition, the DRIVER.SYS file is located in the DOS directory on drive C:	`DEVICE=C:\DOS\DRIVER.SYS /D:2 /T:80 /S:9 /H:2 /C /F:2`
Assign the external 720 KB drive the next two available drive letters	`DEVICE=DRIVER.SYS /D:2 /T:80 /S:9 /H:2 /C /F:2` `DEVICE=DRIVER.SYS /D:2 /T:80 /S:9 /H:2 /C /F:2`

HIMEM.SYS

HIMEM.SYS is an installable device driver which conforms to XMS (eXtended Memory Specification) Version 2.0. It allows MS-DOS programs on 80286 and 80386 systems with extended memory to access the extended memory in a standard, machine-independent manner.

Syntax

DEVICE=*drive:\path*\HIMEM.SYS /HMAMIN=*h* /NUMHANDLES=*n*

Parameter	Description
drive:	drive that contains HIMEM.SYS file.
path	path to HIMEM.SYS file.
/HMAMIN=*h*	sets the minimum amount of space (in kilobytes) in the High Memory Area (HMA) that a TSR (Terminate and Stay Resident) program can use. The minimum value for *h* is 0 and the maximum is 63, with a default of 0. The 0 default allows “first come, first served” access to the HMA. When this parameter is used, this message will be displayed: `Minimum HMA size set to` *h*
/NUMHANDLES=*n*	For advanced users only: this sets the maximum number of Extended Memory Block (EMB) handles that can be used at any given time. The minimum value for *n* is 1 and the maximum is 128, with a default is 32. Each additional handle requires an additional 6 bytes of resident memory. When this parameter is used, this message will be displayed: *n* `extended memory handles available.`

Operation

HIMEM.SYS allows systems to utilize a 64 KB region of **extended memory** located just above the 1 MB boundary for code and data, some MS-DOS programs (such as Microsoft Windows) utilize this region.

Use of the /HMAMIN parameter ensures that MS-DOS Terminate and Stay Resident (TSR) programs which use less space than the minimum number you set will not be placed in the HMA. This number becomes important when two TSR programs which use the HMA are installed. Setting this number to the request size of the largest HMA-using TSR ensures maximum benefit of this area of memory.

If you have **expanded memory** (EMS, not XMS), you can usually configure some part of that memory as extended memory (64 KB would be optimal), leaving the rest as expanded.

If other extended memory programs are also being loaded which do not use the XMS interface to manage the extended memory, they should be configured to leave at least 64KB of extended memory free.

To use HIMEM.SYS, with your computer, you need the following:

- A computer that uses the 80286 or 80386 processor.
- Extended memory (built-in or on an accessory card).

Examples

To do this:	Type this in CONFIG.SYS:
Load HIMEM.SYS (located in the DOS directory of drive C:) to allow MS-DOS programs to use the High Memory Area above 1 MB	`DEVICE=C:\DOS\HIMEM.SYS`
Allow MS-DOS programs to use at least 40 KB of the High Memory Area	`DEVICE=HIMEM.SYS /HMAMIN=40`

Notes

1. HIMEM.SYS must be installed before any other device drivers that make calls to it. Thus, any device driver which uses XMS calls should be loaded after HIMEM.SYS.

INDSKBIO.SYS

INDSKBIO.SYS provides an HP Vectra compatibility with double-sided 3.5-inch flexible disks used with the **HP 110, Portable PLUS, 150,** and **HP 150II** personal computers. In addition, you need to use this driver if you are using a 3.5-inch internal flexible disk drive with an early model HP Vectra (see the note at end of this section).

Syntax

```
DEVICE=drive:\path\INDSKBIO.SYS /D:number /HP110
```

Parameter	**Description**
drive:	drive that contains INDSKBIO.SYS file.
path	path to INDSKBIO.SYS file.
number	specifies the physical drive number. Valid values are: 0 for the first flexible disk drive 1 for the second flexible disk drive 2 for the third flexible disk drive 3 for the fourth flexible disk drive
/HP110	allows data exchange with the **HP 110.**

Operation

Use INDSKBIO.SYS if you have an HP Vectra personal computer with an internal 3.5-inch disk drive and you intend to share (read and write) data on double-sided 3.5-inch disks used with:

- The **HP 110** portable computer
- The **HP Portable PLUS** portable computer
- The **HP 150** personal computer
- The **HP 150II** personal computer

In addition, you **must** use INDSKBIO.SYS if you have an early model HP Vectra with an internal 3.5-inch internal flexible disk drive in order for the drive to operate properly (see the note at end of this section).

Examples

To do this:	Type this in CONFIG.SYS:
Make a 3.5-inch flexible drive A: in an HP Vectra ES, QS, or RS compatible with **HP 150, 150II,** and **Portable PLUS,** where the file INDSKBIO.SYS is located in the DOS directory of you HP Vectra's drive C:	`DEVICE=C:\DOS\INDSKBIO.SYS /D:0`
Make a 3.5-inch flexible drive A: in an HP Vectra ES, QS, or RS compatible with drives in the first example and also the **HP 110** drives	`DEVICE=C:\DOS\INDSKBIO.SYS /D:0 /HP110`

Make a 3.5-inch flexible drive B: in an early model HP Vectra (*see note*) compatible with all of the drives in the previous two examples	`DRIVPARM=/D:1 /F:7 /T:80 /S:18 /H:2 /C`
	`DEVICE=C:\DOS\INDSKBIO.SYS /D:1 /HP110`

Note

An early model HP Vectra has just the words "Hewlett-Packard *Vectra*" on the nameplate on the front of the computer (without any identifier, such as ES, QS, or RS). These computers will only support a 3.5-inch flexible disk drive if it is in the B: drive slot (second drive slot down from top). In addition, only these early model computers require the DRIVPARM statement as shown in the above example.

PRINTER.SYS

PRINTER.SYS provides support for code page switching on parallel ports LPT1, LPT2, and LPT3. (PRN can be substituted for LPT1 when referring to the first parallel port.) For additional information on code page switching, refer to the appendix "How to Use Code Pages."

Syntax

DEVICE=*drive:*\path\\PRINTER.SYS LPT*n:*= (*type*,*hwcp*, ... ,*n*)

Parameter **Description**

drive: drive that contains PRINTER.SYS.

path path to PRINTER.SYS.

LPT*n:* printer device. Valid printer devices are LPT1, LPT2, and LPT3. PRN can be used instead of LPT1. Up to 3 printer devices can be specified.

type type of printer. Valid printer types are 4201 (IBM 4201 Proprinter family, and IBM 4202 Proprinter XL), 4208 (IBM 4207 or 4208, and IBM Proprinter X24 or XL24), and 5202 (IBM 5202 Quietwriter III). If you don't have one of these types of printers, you can't use code page switching with your printer at this time.

hwcp code page(s) built into the printer. Valid values are 437 (United States), 850 (Multilingual), 860 (Portugal), 863 (French-Canadian), and 865 (Norway). Some printers can support a pair of code pages, such as 437, or 850, for example:

```
DEVICE=C:\DOS\PRINTER.SYS LPT1:=(5202,437,850)
```

n number of additional code pages that can be supported. Valid values are 0 to 12. This value is printer dependent. When used **without** *hwcp,* include the extra comma separator, for example:

```
DEVICE=C:\DOS\PRINTER.SYS LPT1:=(4201,,2)
```

Example

Let's say you wanted to install code page switching support for two print devices (LPT1 and LPT2) with the LPT1 printer an IBM 5202 Quietwriter III and the LPT2 printer an IBM 4201 Proprinter. In addition, both printers have code page 437 built-in and the LPT2 printer supports one additional code page. You would type this line in CONFIG.SYS:

```
DEVICE=PRINTER.SYS LPT1:=(5202,437,0) LPT2:=(4201,437,1)
```

RAMDRIVE.SYS

RAMDRIVE.SYS allows a portion of your system's RAM memory to be used as a disk drive. RAM memory used to emulate or act like a physical disk drive is referred to as a **RAM Disk** or **Virtual Disk.** A virtual disk can be used in the same manner as a physical disk drive, except you cannot use a few of the MS-DOS commands (such as CHKDSK, DISKCOPY, and FORMAT) on a virtual disk. RAMDRIVE.SYS allows either RAM memory, expanded, or extended memory (at or above 1 MB) to be used as a virtual disk. In addition, you can install more than one virtual disk at a time.

Syntax

DEVICE=*drive:\path*\RAMDRIVE.SYS *disksize sectorsize entries* /E

or

DEVICE=*drive:\path*\RAMDRIVE.SYS *disksize sectorsize entries* /A

Parameter	**Description**
drive:	drive that contains RAMDRIVE.SYS.
path	path to RAMDRIVE.SYS.
disksize	virtual disk size in KB. Valid values are 1 to the amount of memory you have in your system; the default value is 64 KB.
sectorsize	sector size in bytes. Valid values are 128, 256, and 512. The default value is 128 bytes. Smaller sector sizes conserve space; larger sector sizes increase performance. If you're going to use RAMDRIVE to store many small files, consider specifying one of the smaller sector sizes.
entries	number of directory entries that the root directory of the virtual disk can contain. Valid values are 2 to 512; the default value is 64 directory entries.
/E	instructs RAMDRIVE.SYS to create the virtual disk in extended memory (memory at or above 1 MB—you must have this extra memory to use this parameter), even though

the driver code is to be installed in low memory. /E and /A cannot be used together.

/A instructs RAMDRIVE.SYS to create the virtual disk in expanded memory. You need to have extra memory installed that supports the Lotus/Intel/Microsoft Expanded Memory Specification. /A and /E cannot be used together.

Operation

There is one big advantage and one big disadvantage to using a virtual disk instead of a flexible or hard disk. The advantage is speed. Disk operations on a virtual disk are significantly faster than on a hard disk, and much faster than on a flexible disk. The disadvantage is that the contents of a virtual disk are lost whenever you restart or turn off your system.

This means that you need to transfer files to the virtual disk when you start your system. Then, you need to transfer the files back to a flexible or hard disk before you restart or turn off your system. Use the COPY or XCOPY command to transfer files to and from a virtual disk.

RAMDRIVE.SYS automatically adjusts the virtual *disksize* in the following situations:

If the size you specify is less than 1 KB or greater than the amount of memory in your system, RAMDRIVE.SYS uses the default value of 64 KB.

If your system has less than 64 KB of available memory at the time you're attempting to install RAMDRIVE, MS-DOS displays an error message and doesn't install RAMDRIVE.

RAMDRIVE.SYS also adjusts the number of directory *entries* upward to the nearest sector size boundary. Thus, if you specify a *sectorsize* of 256 and the number of directory entries as 17, 24 directory entries are generated. Twenty-four directory entries at 32 bytes per entry occupy 768 bytes, which is a multiple of the sector size.

If you specify a virtual disk size that's too small to hold the FATs (File Allocation Tables), the root directory, and two additional sectors, RAMDRIVE.SYS decreases the directory size until these conditions are met. If the directory size reaches one sector and the conditions still aren't met, MS-DOS displays an error message and doesn't install RAMDRIVE.SYS.

To install more than one virtual disk, insert additional `DEVICE=RAMDRIVE.SYS` commands in the CONFIG.SYS file. Once a virtual disk has been installed, the following sign-on message appears each time you start your system:

```
RAMDRIVE Version 4.00 virtual disk x
```

where *x* is the drive letter assigned to the virtual disk. Below this message, the values of the three parameters are displayed as shown:

```
Buffer size:      disksize
Sector size:      sectorsize
Directory entries: entries
```

Example

To do this:	Type this in CONFIG.SYS:
Install a RAMDRIVE.SYS (located in the DOS directory of drive C:) virtual disk with default parameters	`DEVICE=RAMDRIVE.SYS`
Install a virtual disk of 1024 KB in expanded memory in the CONFIG.SYS file	`DEVICE=RAMDRIVE.SYS 1024 /A`

Notes

1. We recommend that you only put applications and MS-DOS commands on a virtual disk. Don't put data files on it. Also, you can use the AUTOEXEC.BAT file to load commonly used applications onto a virtual disk every time you start your system.
2. When a virtual disk is operating in extended memory, interrupt servicing is suspended during data transfers. If frequent interrupts occur during data transfers, some of the interrupts can be lost. If you experience this problem, try installing the virtual disk in non-extended memory.

6

Redirecting Input and Output

Standard Input and Output Devices

Input always comes from the **standard input device** and output always goes to the **standard output device.** When you start your computer, the keyboard is assigned as the default standard input device and the display screen is assigned as the default standard output device. The keyboard and display screen together are referred to as the **console** or CON device.

MS-DOS allows you to change the standard input and output devices from the default to other (physical) devices, or even files. There are a number of ways to change the standard input and output devices. In this chapter, we'll discuss changing them through redirection.

Redirecting Input and Output

It is often advantageous to receive input from a file or a device other than the keyboard, or to send output to a file or a device other than the display screen. This process is called **redirection.**

MS-DOS allows you to redirect input so that it doesn't come from the default standard input device; it comes from a file or another device instead. The same is true of output. MS-DOS allows you to redirect output so that it doesn't go to the default standard output device; it goes to a file or another device instead. In both cases, the file (or device) becomes the new standard input or output device, temporarily replacing the default standard input or output device.

Redirection has the following important characteristics:

- Input and output are redirected with an MS-DOS command.
- Input and output are redirected independently of each other.
- Redirection is temporary. It is only in effect while an MS-DOS command or application program is executing.

Redirecting input and output has many uses. You can put an application program's keyboard input into a file and then redirect the input so that it comes from that file instead of the keyboard. Thus, the file acts somewhat like a batch file. Also, you can redirect output so that it goes to a printer or a file instead of the display screen. If output goes to a printer, you get a printed copy; if output goes to a file, you get a permanent copy that you can use in the future.

Command Syntax

Many commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the command and its parameters.

This manual uses the following conventions for command syntax:

CAPS Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Blue Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that you should type the name of your file.

... An ellipsis indicates that you can repeat the parameters as many times as necessary.

Enter all punctuation (except the ellipsis), such as commas, colons and equal signs, exactly as shown in the syntax.

Redirecting Input

You can redirect input so that it comes from a file or a device instead of the keyboard.

Syntax

command < *drive:\path\file*

or

command < *device*

Parameter	**Description**
command	MS-DOS command, or application program.
<	input redirection symbol.
drive:	drive where input file is located.
path	path to input file.
file	name of input file.
device	character device capable of input (for example: AUX, COM1, COM2, COM3, COM4).

Operation

If input comes from a file, MS-DOS reads characters from the specified file as they are requested by an MS-DOS command (or an application program) until the command finishes executing. When using this form of redirection, it is important that all input required by the command is contained in the file. If the command attempts to read more input than is contained in the file, it stops executing, and you need to exit by holding down Ctrl and pressing Break.

If input comes from a device, MS-DOS reads characters from the specified device. Input can be redirected from any character device capable of input.

Examples

To do this:	Type this:
Send input to the SORT command filter from a file called CLIENT.LST which contains an unsorted listing, instead of from the standard input device (the keyboard)	`SORT <CLIENT.LST`
Send input to the SORT command filter from the AUX device, instead of from the standard input device (the keyboard)	`SORT <AUX`

In both of the above examples, the default standard **output** device (that is, the display screen) remains unchanged. As a result, the sorted output appears on the display screen.

Redirecting Output

Output can be redirected so that it goes to a file or a physical device instead of the display screen.

Syntax

command > *drive*:\path\file

and

command >> *drive*:\path\file

or

command > *device*

Parameter	**Description**
command	MS-DOS command, or application program.
>	output redirection symbol.

>>	output redirection symbol used to append (add) data to the end of an existing file. This parameter ensures that the data in an already existing file is not overwritten.
drive:	drive where output file is located.
path	path to output file.
file	name of output file.
device	character device capable of output (for example: PRN, AUX, LPT1, LPT2, LPT3, COM1, COM2, COM3, COM4).

Operation

If output is redirected to a file, MS-DOS creates the specified file and stores all of the characters sent to the standard output device in that file. If the file already exists on the disk, MS-DOS overwrites its original contents.

If output is redirected to a device, MS-DOS sends characters to the specified device. Output can be redirected to any character device capable of output.

Examples

To do this:	Type this:
Send output from the DIR command to a file called DIRLIST.LST, instead of to the standard output device (the display)	`DIR >DIRLIST.LST`
Send the output from the DIR command to the printer, instead of to the standard output device (the display)	`DIR >PRN`
Send output from the DIR command and append it to the end of a file called DISKDIR.TXT which already contains data	`DIR >>DISKDIR.TXT`

In the above examples, the default standard input device (that is, the keyboard) remains unchanged. As a result, the input comes from the keyboard.

Piping Input and Output

Piping is an extension of redirection. **Piping** allows two or more MS-DOS commands (or application programs) to be "chained" together. That is, the output from the first command serves as the input to the next.

Syntax

command1 | *command2* ...

Parameter	**Description**
command1	first MS-DOS command, or application program in chain.
|	piping symbol. Chains commands or programs together.
command2	second MS-DOS command, or application program in chain.

Operation

Piping occurs when MS-DOS encounters two or more commands (or programs) separated by the vertical bar (|) character in a command.

When piping is used, MS-DOS executes the commands in the order they're encountered on the command. In addition, MS-DOS creates a temporary file to store the output from each command except the last command. It assigns the filename extension **.$$$** to each temporary file. When piping is complete, MS-DOS deletes the temporary files.

Example

To do this:	**Type this:**
Pipe the output from the DIR command to the SORT command filter, then pipe the output from SORT to the MORE command filter to display the sorted directory listing one screen at a time	`DIR | SORT | MORE`

In the above example, MS-DOS creates a temporary file to store the output from the DIR command. This temporary file is used as input to the SORT command. Then, MS-DOS creates another temporary file to store the output from the SORT command. This second temporary file is used as input to the MORE command. Finally, MS-DOS deletes both temporary files.

Using Filters

The final topic relating to input and output is filters. A **filter** is any program that reads data from the standard input device, modifies or examines it, and then sends it to the standard output device.

MS-DOS provides three filters: FIND, MORE, and SORT.

- FIND searches input for a specified string of characters.
- MORE displays output one screen at a time.
- SORT sorts input in ascending or descending order and displays the sorted output.

Filters are used extensively with piping, as we've already seen in the previous section. For additional information on MORE, FIND, and SORT, refer to the chapter "MS-DOS Command Descriptions."

Tips on Using Redirection, Piping, and Filters

There are several points to remember when redirecting input and output, piping input and output, and using filters.

- These three capabilities won't work with every application program. To work, a program must be written to read from the standard input device and write to the standard output device. Therefore, any program that bypasses MS-DOS for character input and output (that is, using BIOS interrupts or going to the hardware, itself) won't be able to perform these functions.
- In most cases, piping should only be used with filters. A filter is a specially written application program that uses input and output, interprets an end-of-file marker from the standard input device as a "terminate program" command, and is generally programmed to perform correctly as a filter. The

use of piping with an application program that doesn't meet this criteria can lead to unpredictable or unsatisfactory results.

- The temporary files created by piping are stored in the current directory on the active drive. Therefore, to be able to use piping, the disk in the active drive can't be write-protected.
- MS-DOS allows the use of redirection, piping, and filters in the same command. The following command pipes the output from the DIR command through the SORT filter. Then, the sorted output is redirected to the file DISKDIR.LST.

  ```
  DIR | SORT >DISKDIR.LST
  ```

  It is also possible to redirect output to a device. The following command redirects the sorted directory listing to the system printer instead of the file DISKDIR.LST.

  ```
  DIR | SORT >PRN
  ```

- MS-DOS sends some error messages to a logical device referred to as the **Standard Error Device.** Output to this device (for example, error messages) can't be redirected. Using the previous example, if a disk read error occurs during the execution of this command, the error message appears on the display screen instead of the system printer.

7

MS-DOS Editing and Function Keys

MS-DOS Editing Keys

With MS-DOS, you don't have to type the same sequences of keys repeatedly, because the most recently typed command is automatically placed in a special storage area called a template. When you type a command and press Enter, MS-DOS executes the command and saves a copy of the command in the template. You can then recall or modify the command using the MS-DOS editing keys. Using the template and the MS-DOS editing keys allows you to take advantage of the following MS-DOS features:

- You can repeat a command instantly by pressing two keys.
- With a minimum of typing, you can edit and execute a command that is similar to a previous one.
- If you make a mistake while typing a command, you can edit and retry it without having to retype the entire command.

The following table describes the MS-DOS editing keys.

MS-DOS Editing Keys

Key	Editing Function
F1	Copies one character from the template to the screen.
F2	Copies all characters from the template to the screen, up to the character you specify.
F3	Copies all remaining characters in the template.
Del	Skips over (does not copy) a character in the template.
F4	Skips over (does not copy) characters in the template, up to the character you specify.
Esc	Voids the current input and leaves the template unchanged. Displays a backslash (\) after the voided input.
Ins	Enters/exits insert mode.
F5	Makes the new line the new template.
F6	Puts a Ctrl Z (1AH) end-of-file character in the new template.
←	Deletes one character on the screen and moves the cursor back one character in the template.

Examples

Creating a Template

When MS-DOS executes a command it also saves the command in the template. To repeat the command, just press two keys: F3 and Enter. MS-DOS displays the command on the screen when you press F3, and executes the command a second time when you press Enter.

For example, to see the directory information for a file named INVEST.MNT, you might type the following command:

```
DIR INVEST.MNT
```

To see the directory information a second time, press F3 and Enter.

Changing the Template

If you want to use a command that is similar to the last command you used, modify the contents of the template instead of typing the new command.

For example, to display information about a file named INVEST.RPT, you can modify the contents of the template. Press F2 and type the letter **M**, and MS-DOS copies all characters from the template to the command at the MS-DOS prompt, up to but not including the *M*. MS-DOS displays:

```
DIR INVEST._
```

(The underline represents your cursor.) Now type the letters RPT to get the following result:

```
DIR INVEST.RPT_
```

To execute the command, press Enter.

Then, to execute the command "TYPE INVEST.RPT," type the word TYPE and then press the following sequence of keys: Ins, spacebar, F3, Enter.

This is how the sequence of keys works:

Keys	Effect of Keys	Result on Screen
T Y P E	*TYPE* replaces *DIR* (and the space after it) in the template	`C>TYPE_`
Ins and the spacebar	Inserts a space	`C>TYPE _`
F3	Copies the rest of the template to the command at the MS-DOS prompt	`C>TYPE INVEST.RPT_`
Enter	Executes the command	`C>_`

Correcting Errors in the Template

If you make a mistake as you type a command, you can save the command in the template **without executing the command.** This allows you to use the special editing keys to correct your mistake.

For example, if you type "PYTE INVEST.RPT" instead of "TYPE INVEST.RPT," you can use the special editing keys to correct the misspelling. Do not press Enter. Instead, press the following sequence of keys: F5, Del, Del, F1, Ins. Type YP. Then press F3 and Enter.

This is how it works:

Keys	Effect of Keys	Result on Screen
F5	Stores command in the template	C>PYTE INVEST.RPT@ _
Del	Skips over 1st template character	C>PYTE INVEST.RPT@ _
Del	Skips over 2nd template character	C>PYTE INVEST.RPT@ _
F1	Copies 3rd template character	C>T_
Ins Y P	Inserts Y and P	C>TYP_
F3	Copies rest of template	C>TYPE INVEST.RPT_
Enter	Executes the command	C>

The MS-DOS Control Characters

A control character affects the command at the MS-DOS prompt. For example, you use Ctrl C to stop executing the current command, and you use Ctrl S to suspend the screen output from a command.

Note

When you type a control sequence, such as Ctrl C, you must hold down the Ctrl key and then press the C key.

The following table shows the MS-DOS control characters and describes what they do.

Control Character	What It Does
Ctrl C	Aborts the current command.
Ctrl H	Removes the last character from a command, and erases that character from the terminal screen.
Ctrl J	Inserts a physical end-of-line, but does not clear the command. Use the LINEFEED key to extend the current logical line beyond the physical limits of the terminal screen.
Ctrl N	Causes echoing of output to a lineprinter.
Ctrl P	Causes terminal output to a lineprinter.
Ctrl S	Pauses output displayed on the screen. Press Ctrl S again to resume.
Ctrl X	Cancels the current command, clears the command, and then outputs a backslash (\), ENTER, and LINEFEED. Ctrl X does not affect the template used by the special editing commands.

The Line Editor (Edlin)

What Can Edlin Do?

Edlin is the MS-DOS line editor. You can use Edlin to create text files and save them on disks. You can also use Edlin to delete, change, and insert lines in files. Even though it isn't a word processor, it's easy to use Edlin to create and revise files such as memos, letters, reports, or GW-BASIC programs.

How Edlin Works

Edlin divides the text from a file into lines, each line containing up to 253 characters. It gives each line a number and always numbers the lines consecutively. However, even though you see these line numbers on the screen when you use Edlin, they are not part of the file.

When you insert lines of text in a file, Edlin automatically adjusts the line numbers after the inserted text. Similarly, when you delete lines in a file, Edlin automatically renumbers the lines following the deleted text.

Starting Edlin

To start Edlin, type

```
EDLIN file
```

and press (Enter). *File* is the file you want to edit. If you are creating a new file, *file* should be the name of the file you want to create. If Edlin does not find the *file*, it creates a new file with the name that you specify. For example, if you want to create a file called BUDGET.JUN, you would type

```
EDLIN BUDGET.JUN
```

and press Enter.

Creating a New File With Edlin

Once you type the command to start Edlin, Edlin displays the following:

```
New file
*_
```

The Edlin prompt is an asterisk (*). To begin entering text, type

```
I
```

(for the Insert command). Press Enter after each line of text you type. Press Ctrl C when you are done inserting text.

Editing an Existing File With Edlin

To edit an existing file, type

`EDLIN` *file*

and press Enter, where *file* is the name of the existing file. When Edlin finds the *file,* it loads it into memory. If your computer has enough memory to load the entire file, Edlin displays the following message:

```
End of input file
*
```

You can then edit the file, using the Edlin commands.

If the file is too large to be loaded into memory, Edlin loads lines from the file until memory is 3/4 full, and displays the asterisk (*) prompt. You can then edit the portion of the file that is in memory.

To edit the rest of the file, you must save some of the edited lines on a disk to free memory. Edlin will then be able to load the remaining unedited lines from a disk into memory. (For more information on editing large files, see the descriptions of the W (Write) and A (Append) commands in this chapter.)

How to End Edlin and Save Your Changes

When you finish your editing session and the cursor is at the asterisk (*) prompt, save your file by typing

```
E
```

(for the End/Save command) and press Enter. Edlin saves your updated file. In addition, Edlin saves a copy of the file as it was before you edited it. This file has the same file name and the extension .BAK.

Caution

You cannot update a file that has the extension ".BAK." To update a ".BAK" file, use the MS-DOS RENAME command and change the file's extension.

If a .BAK file already exists and it is a read-only file, Edlin **will not** save your changes. Edlin does not detect this condition when you first start to edit a file. However, when you end your editing with the E command, you will see the following message:

```
Access denied - file.BAK
```

Using the MS-DOS Editing Keys With Edlin

You can use the MS-DOS editing keys to edit your text files. For more information, refer to the chapter "MS-DOS Editing and Function Keys."

Some Tips for Using Edlin Commands

- You can use paths in commands. For example, by typing the following command, you can edit a file named REPORT.MAY in a subdirectory named \SHARPE\BUDGET:

  ```
  EDLIN \SHARPE\BUDGET\REPORT.MAY
  ```

- You can refer to lines with numbers relative to the current line. The current line is identified with an asterisk (*). To indicate lines before the current line, use a minus sign with a number; to indicate lines after the current line, use a plus sign with a number. For example, to list 10 lines before the current line, the current line, and 10 lines after the current line, you could type this command:

  ```
  -10,+10L
  ```

- Edlin ignores spaces between the line number and command. For the examples in this chapter, spaces are omitted.

- Generally, Edlin allows you to type one command after another on the same line. However, if you want to use the Edlin Single Line Edit command to edit a specific line, you must separate the line number from the other commands with a semicolon. For example, the following command edits line 15, then displays lines 10 through 20 on the screen:

  ```
  15;-5,+5L
  ```

Using Control Characters

- When using control key sequences, press and hold the Ctrl key, then press the control character (such as Z or C or V).

- You can insert a control character, such as Ctrl C, into text by using the quotation mark character, Ctrl V, before it while in Insert. Ctrl V tells MS-DOS to recognize the next **capital** letter typed as a control character. For example, to search for the first occurrence of Ctrl Z (the end-of-file mark) in a file, type S and then press Ctrl V Z.

  To insert Ctrl V into the text, press Ctrl V V.

- The Ctrl Z character is usually an end-of-file identifier for Edlin. If you have Ctrl Z characters elsewhere in your file, you must tell Edlin that these other control characters do not mean end-of-file. To tell Edlin to ignore the Ctrl Z characters in the file and to show you the entire file, use the /B parameter when you start Edlin. For example, the following command lets you start editing the file MACRO.ASM and ignores any Ctrl Z characters:

  ```
  EDLIN MACRO.ASM /B
  ```

The Edlin Commands

This rest of this chapter contains descriptions of all the Edlin commands, listed in alphabetical order. Use this chapter as a reference.

Edlin Syntax Conventions

Many Edlin commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the command and its parameters. This manual uses the following conventions for command syntax:

CAPS
Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Blue
Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics
Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that you should type the name of your file.

Enter all punctuation, such as commas, colons and equal signs, exactly as shown in the syntax.

A (Append)

Syntax

*n*A

Parameter	Description
n	the number of lines that you want to read into memory.

Operation

This command appends lines from disk to memory. If you are editing a large file that is too large to read into memory all at once, you can use the A (Append) command. This command lets you read in portions of your file to memory as you need to work on them.

When you start Edlin, it reads as many lines as possible into memory. If the size of your file exceeds available memory, you must edit your file in stages. That is, after you have edited the first part of a large file, you must write lines that you have already edited onto your disk. For information about how to write edited lines to your disk, see the W (Write) command in this chapter. Then you can load unedited lines from your disk into memory by using the A (Append) command.

For example, if you wanted to read the remaining 100 lines of a file into memory, you would type: `100A`

Notes

1. If you do not specify *n*, Edlin adds lines to the available memory until it is 3/4 full, but does nothing if available memory is already 3/4 full. If available memory is already full, you may be able to free memory by quitting other applications that may be running or by restarting MS-DOS.
2. After the A (Append) command reads the last line of the file into memory, Edlin displays the message "`End of input file.`"

C (Copy)

Syntax

line,*line*,*line* C

Parameter **Description**

line,line,line the first and second *line* parameters specify the range of *lines* that you want to copy. If you omit the first or second *line* parameter, Edlin defaults to the current line (the line Edlin displays with an * next to the line number). The third *line* specifies the line before which Edlin will place the copied lines.

count the number of times you want the range of lines to be copied.

Operation

The C (Copy) command copies a range of lines to a specified line number, and when used with the *count* parameter, copies this range as many times as you want.

You must not overlap the line numbers or you will get an "`Entry error`" message. For example, this command would result in an error message:

```
3,20,15C
```

If you do not specify a number for the *count* parameter, Edlin copies the lines one time and automatically renumbers the file after the copy.

For example, if you want to copy lines 1 through 5 of a file and duplicate them one time, beginning on line 6 (lines 1 through 5 and 6 through 10 will be identical), you would type: `1,5,6C`

D (Delete)

Syntax

```
        D
```

Parameter	Description
line,line	range of lines you want to delete. If you type a period (.), or omit the first *line* parameter, Edlin defaults to the current line (the line with the * next to the line number). If you omit the second *line* parameter, Edlin deletes just the first line.

Operation

The D (Delete) command deletes a specified range of lines in a file. When you delete lines, Edlin automatically renumbers the file.

Examples

To do this:	Type this:
Delete line 7 and renumber line 8 and all following lines	`7D`
Delete the block of text on lines 22 through 32	`22,32D`
Delete a range of lines beginning with the current line (which happens to be line 7) through line 11	`,11D`

E (End/Save)

Syntax

```
E
```

Operation

The E (End/Save) command saves your edited file to disk, renames the original input file *file*.BAK, and then exits Edlin. If you created the file during this editing session, Edlin does not create a backup (.BAK) file.

The E (End/Save) command takes no parameters. This means that you must select the directory and drive that you want to save the file on when you start Edlin. If you don't select a drive when you start Edlin, it saves the file on the disk in the default drive. However, you can still copy the file to a different drive by using the MS-DOS COPY command.

Before using the E command to save your file, make sure that the disk contains enough free space for the entire file. If it doesn't, Edlin may not be able to write the entire file to the disk. The edited file will be lost, although Edlin may have saved part of the file on the disk.

Notes

1. If a .BAK file already exists and it is a read-only file, Edlin **will not** save your changes. Edlin does not detect this condition when you first start to edit a file. However, when you end your editing with the E command, you will see the following message:

   ```
   Access denied - file.BAK
   ```

I (Insert)

Syntax

line I

Parameter	Description
line	number of the line that you want the new line inserted before. If you omit this parameter, or specify a period (.), Edlin uses the current line number (the line with the * next to the line number). To specify the line immediately following the last line, enter a number larger than the last line, or enter the pound sign (#).

Operation

The I (Insert) command allows you to insert text immediately before the specified line. If you are creating a new file, you must type the I (Insert) command before you can insert a new line of text. Text begins on line 1, and the next line number appears automatically each time you press Enter.

Edlin remains in insert mode until you press Ctrl C. When you finish the insertion and exit insert mode, the line immediately following the inserted lines becomes the current line. Edlin automatically increments the line numbers that follow the inserted section by the number of lines that you inserted.

If you do not specify *line*, the default is the current line number and Edlin inserts the lines before the current line. If *line* is a number larger than the last line number, or if you specify a pound sign (#) as *line*, Edlin appends the inserted lines to the end of the file. In this case, the last line that you inserted becomes the current line.

Examples

To do this:	Type this:
Insert text before line 8 (making the line you insert the new line 8, and making the old line 8 now line 9)	`8I`
Insert a blank line immediately before the current line	`I`

L (List)

Syntax

line,line L

Parameter **Description**

line,line range of lines to display. If you specify a period (.) in place of the first line parameter, Edlin displays lines starting from the current line (the line with the * next to the line number). If you specify only one of the *line* parameters, Edlin uses default values.

Operation

The L (List) command displays a range of lines, including the two lines specified.

Examples

To do this:	Type this:
Display lines 5 through 10	`5,10L`
Display 23 lines, beginning 11 lines before the current line and ending with line 40	`,40L`
Display 23 lines, starting with line 5	`5L`
Display 23 lines—beginning with the 11 lines before the current line	`L`

Notes

1. If the specified line is more than 11 lines before the current line, the display will be the same as if you omitted both parameters.

Line Edit

Syntax

line

Parameter **Description**

line the line number of text you want to edit. If you specify a period (.) instead of a line number, Edlin will edit the current line (the line with the * after the line number).

Operation

The *line* parameter allows you to specify the line number of text you want to edit. When you type a line number as a command, Edlin displays the line number and the text on that line; then, on the line below, Edlin reprints the line number. Now you can retype the line, or use the Edlin editing keys to edit it. The existing text of the line serves as the template until you press Enter.

For example, to edit line 5 of a file, type: `5`

If you press Enter without typing a line number, Edlin edits the line after the current line. When you have edited the line, simply press Enter.

Caution If you press Enter while the cursor is in the middle of a line, Edlin deletes the remainder of the line.

M (Move)

Syntax

line,+*line*,*line*`M`

Parameter	Description
line,+*line*,*line*	The first and second *line* parameters specify the range of lines that you want to move. If you specify a period (.) in place of the first *line* parameter, it refers to the current line number (the line Edlin displays with an * after the line number). The third *line* parameter specifies the line to which you want to move the first *line* in the range. To specify the line immediately following the last line, enter a number larger than the last line, or enter the pound sign (#).

Operation

The M (Move) command lets you transfer a block of text to another location in a file. Edlin automatically renumbers the lines after it moves them. For example, the following command moves the text from the current line—plus 25 lines—to line 100:

```
,+25,100M
```

If the line numbers that you specify overlap, Edlin displays an "`Entry error`" message.

Examples

To do this:	Type this:
Move lines 16 through 18 to line 33	`16,18,33M`
Move the current line (which happens to be line 16), plus two more lines (which would be lines 17 and 18), to line 1	`,+2,1M`

P (Page)

Syntax

[illegible]P

Parameter	Description
line,line	the first *line* parameter specifies the line at which Edlin starts displaying. If you type a period (.), Edlin starts the page at the current line (the line with the * before the line number). If you omit the first *line,* Edlin starts the page at the line after the current line. The second *line* parameter specifies how many lines appear on each page. If you do not type the second *line* parameter, Edlin lists 23 lines on each page.

Operation

The P (Page) command displays a file one screen at a time. For example, to view lines 100 through 200 of a file and see the text one screen at a time, for example, type: `100,200P`

Q (Quit/Do Not Save)

Syntax

```
Q
```

Operation

The Q (Quit/Do not Save) command is useful if you don't want to make any changes to a file. This command exits to the MS-DOS operating system and does not save any editing changes. If you use the Q (Quit/Do not Save) command, Edlin prompts you to make sure you don't want to save the changes. If you want to save changes as you exit Edlin, use the E (End/Save) command.

Notes

1. When you exit Edlin, it erases any previous copy of the file that has a .BAK extension. But if you quit Edlin (Q) and reply "`Y`" (for "Yes") to the "`Abort edit (Y/N)?`" message, Edlin will not delete your previous backup copy.

R (Replace)

Syntax

line,line ?R*text1* Ctrl Z *text2*

Parameter	**Description**
line,line	the range of lines for text strings to be replaced. If you specify a period in place of the first *line* parameter, the range of lines starts at the current line (the line with the * after the line number). If you omit the first *line* parameter, the line range starts with the line after the current line.
?	causes Edlin to ask "`O.K.?`" before lines are replaced.
text1	new text string Edlin will use in replace.
Ctrl Z	control key sequence to use if you want to replace all *text1* text strings in the file to blank spaces.
text2	old text string to be replaced.

Operation

The R (Replace) command replaces all occurrences of a string of text in a range of lines with a different string of text. Each time Edlin finds *text1,* it replaces it with *text2.* Then Edlin displays each line that changes.

If a line contains two or more replacements, it is displayed once for each change. If you include a question mark (?) in your command, Edlin asks "`O.K.?`". If you type `Y` (for Yes) or press Enter, *text2* replaces *text1,* and Edlin looks for the next occurrence of *text1.* If you press any other key in response, Edlin does not make the change for that occurrence of *text1.* When Edlin has made all the changes, the asterisk prompt reappears. When you do not specify *text1,* the R command assumes the old (any previous) value. If this is the first replacement that you have done during this editing session, and if you do not specify *text1,* the command ends. If you do not specify *text2,* you must end *text1* by pressing Enter.

If you omit the first *line* parameter, Edlin uses the line after the current line, by default. The default for the second *line* parameter is the line following the last line of the file (represented by the symbol #).

If you end *text1* with a (Ctrl) (Z) and do not specify *text2*, Edlin assumes you want blank spaces for *text2*.

Examples

To do this:	Type this:
In lines 5 through 10, replace all occurrences of the word *our* with the word *the*	`5,10Rour` (Ctrl) (Z) `the`
Change the word *mine* in a 20-line file to the word *ours*, but prompt for whether it is O.K. for each change	`1,20?Rmine`(Ctrl) (Z)`ours`
Delete all occurrences of the word *clients* from the file	`Rclients`(Ctrl) (Z)
Replace *clients* with the string previously (last specified in S or R command) used for *text2*	`Rclients`
Make the string previously (last specified in S or R command) used for *text1* become the previously specified *text2*	`R`

S (Search)

Syntax

line,line ?S*text*

Parameter	Description
line,line	specify the range of lines for Edlin to search. If you specify a period in place of the first *line* parameter, the range of lines starts at the current line (the line with the * after the line number). If you omit the first *line* parameter, the line range starts with the line after the current line.
?	causes Edlin to display the first line with matching text and prompts you with "`O.K.?`". If "Y" (Yes), Edlin discontinues the search for matching text strings, if "N" (No), Edlin continues to search for the next matching text strings.
text	text string you want Edlin search for.

Operation

The S (Search) command searches a range of lines for a string of text. Edlin displays the first line that matches the text string; that line then becomes the current line. To continue to the next occurrence, press Enter.

If you include the question mark parameter (?), Edlin displays the first line with matching text and prompts you with the message "`O.K.?`." If you press either "Y" (for Yes) or Enter, this line becomes the current line and the search ends. If you press any other key, the search continues until another match is found, or until all lines have been searched. (The search ends when Edlin displays the "`Not found`" message.)

If you do not type the first *line* number, Edlin defaults to the line after the current line; and if you do not type the second *line* number, it defaults to # (the line after the last line of the file).

If you omit the *text* parameter, Edlin uses the text from any previous S or R (Replace) command. If this is the first S or R command you have used this session, and you have not specified a search string, the S command ends immediately.

Examples

To do this:	Type this:
Search for the first occurrence of the word *to*	`2,12Sto`
Search through several occurrences of the word *to* until the correct string is found	`1,?Sto`

T (Transfer)

Syntax

*line*T*file*

Parameter	Description
line	line number where the file is to be inserted. If you specifying a period (.), Edlin inserts the file at the current line number (the line Edlin displays with an * after the line number). To specify the line immediately following the last line, enter a number larger than the last line, or enter the pound sign (#).
file	name of file to be transferred into present file.

Operation

The T (Transfer) command puts the contents of one file into another file, or into the text you are typing. Edlin inserts the file name at the line number you give in the *line* parameter, and then automatically renumbers the lines. If you omit the line number, Edlin inserts the text on the current line.

For example, to copy a file named IRSHARPE.MEM to line 12 of the file you are editing, type: `12TIRSHARPE.MEM`

W (Write)

Syntax

```
nW
```

Parameter	Description
n	number of lines you want to write to disk.

Operation

The W (Write) command writes a specified number of lines to disk. The *n* parameter specifies the number of lines that you want to write to the disk. You need this command only if the file you are editing is too large to fit into memory. When you start Edlin, it reads lines from your file until memory is 3/4 full.

To edit the remainder of your file, you must write the edited lines in memory to your disk. Then you can load additional unedited lines from your disk into memory by using the A (Append) command.

For example, to write 125 lines of a file to disk, type: `125W`

Notes

1. If you do not specify the number of lines for Edlin to write, it writes lines until memory is 3/4 full. But it does not write any lines to your disk until memory is more than 3/4 full. Also, Edlin renumbers all of the lines so that the first remaining line becomes line number 1.

Link: An Object File Linker

You need to read this chapter only if you are writing programs in a computer language that creates object (.OBJ) files.

Introduction

The Microsoft^{R} 8086 Object Linker (Link) creates executable programs by combining object files (also called object code modules) generated by the Microsoft Macro Assembler (MASM) or by compilers for high-level languages, such as C or Pascal. Link copies the resulting program to an executable (.EXE) output file. You can then run the program by typing the file's name at the MS-DOS prompt.

To use Link, you must create one or more object files, then submit these files, along with any required library files, to Link for processing. Link can process programs that contain up to one megabyte of code and data.

Starting and Using Link

This section explains three methods for starting and using Link to create executable programs. These methods, which you can also mix, let you specify Link files by

Method 1: Answering prompts

Method 2: Using a command

Method 3: Using a response file

Once you start Link, it will either process the files you supply or prompt you for additional files. You can stop Link at any time by pressing Ctrl C.

Method 1: Using Prompts to Specify Link Files

When you enter `LINK`, at the MS-DOS prompt, Link displays the following four prompts which you must answer with the correct path before it can create your executable file:

Prompt **Response**

`Object Modules [.OBJ]:` Type the name or names of the object files you wish to link. If you do not supply extensions for these files, Link uses .OBJ by default. If you have more than one file, make sure you separate each with spaces or plus signs (+). If you have more files than can fit on one line, enter a plus sign (+) as the last character on the line. Link then prompts you for additional object files.

`Run File [`*file*`.EXE]:` Type the name of the executable file you wish to create, and press Enter. If you do not supply a file name or extension, Link uses the first *file* you entered at the Object Modules prompt with the extension .EXE by default.

`List File [NUL.MAP]:` Type the name of the map file you wish to create. If you do not supply an extension, Link uses .MAP by default. If you don't want a map file, press Enter without typing one.

`Libraries [.LIB]:` Type the names of any library files containing routines or variables referenced but not defined in your program. If you give more than one name, make sure you separate the names with spaces or plus signs (+). If you don't supply extensions, Link uses .LIB by default. If you have more names than can fit on one line, enter a plus sign (+) as the last character on the line. Link then prompts you for additional files.

Method 2: Using a Command to Specify Link Files

You can also create an executable program by typing `LINK`, followed by the names of the files you wish to process. In the command syntax below, words and parameters in CAPS should be typed as shown, parameters in blue are optional, and *italics* are used to show variables:

```
LINK objectfiles,executablefile,mapfile,libraryfile parameters ;
```

File or Parameter — **Description**

objectfiles includes the name or names of object files that you want to link together. Link requires at least one object file. The default extension is .OBJ.

executablefile is an optional placeholder for the name you wish to give the executable file that Link will create. The default name is that of the first object file in the command with the default extension of .EXE. The comma separator must be included with or without this parameter.

mapfile is the name of the file that receives the map listing. The default extension is .MAP. If you specify the /MAP or /LINENUMBERS parameter, Link creates a map file even if you don't specify one in your command. The comma separator must be included with or without this parameter.

libraryfiles includes the name or names of the libraries containing routines that you wish to link to create a program. If you do not supply an extension, Link supplies the extension .LIB. The comma separator must be included with or without this parameter.

parameters control the operation of Link. You can use any of the parameters listed in the section "The Link Parameters." You can specify parameters anywhere on the command line.

You can also use a semicolon anywhere after the object file to terminate the command line.

If you give more than one object file or library file, you must separate the names by spaces or plus signs.

Method 3: Using a Response File to Specify Link Files

You can create a program by listing, in a response file, the names of all the files to be processed, and by giving the name of the response file with the Link command. The simplest way to use a response file is with a command with the following syntax:

LINK @\path\file

You can also specify a response file at any prompt, or at any position in a command. The input from the response file is treated exactly as though you had typed it at the Link prompts or in a command. However, any (Enter)/LINEFEED combinations in the file are treated the same as if you had pressed (Enter) in response to a prompt, or typed a comma in a command.

The Format of a Response File

You can name the response file anything you like. The file content has the following general form:

objectfiles
executable
mapfile
libraryfiles

You can omit any elements that have already been provided at prompts or with a command.

You must place each group of files on a separate line. If you have more names than can fit on one line, you can simply continue the names on the next line by entering a plus sign as the last character in the line. If you do not supply a file for a group of files, you must leave an empty line. You can have parameters on any line.

You can place a semicolon on any line in the response file. When Link encounters the semicolon, it automatically supplies default names for all files you have not yet named in the response file. It ignores the remainder of the response file.

The Library (.LIB) Files

Using Search Paths With Libraries

You can direct Link to search directories and disk drives for the libraries you have named in a command by either specifying one or more search paths with the library names, or by assigning the search paths to the environment variable before you invoke Link. Environment variables are explained under the SET command in the "MS-DOS Command Descriptions" chapter.

The Map (.MAP) File

The map file lists the names, load addresses, and lengths of all segments in a program. It also lists the names and load addresses of any groups in the program, the program start address, and messages about any errors it may have encountered. If the /MAP parameter is used in the Link command, the map file lists the names and load addresses of all public symbols.

The Temporary Disk File: VM.TMP

Link normally uses available memory for the Link session. If it runs out of available memory, it creates a temporary disk file named VM.TMP in the current working directory.

Note

Do not use the name VM.TMP for your own files, since when Link creates the temporary file, it destroys any previous file that has the same name.

The Link Parameters

Link parameters specify and control the tasks that Link performs. All parameters begin with a slash (/). You can abbreviate a parameter name as long as your abbreviation contains enough letters to distinguish the specified parameter from other parameters.

Many of the Link parameters set values in the MS-DOS program header. You will understand these parameters better if you understand how the header is organized. The program header is described in the *MS-DOS Programmer's Reference* and in some reference books on MS-DOS.

Parameter — **Description**

`/CPARMAXALLOC:`*number* or `/C:`*number*

sets the maximum number of 16-byte paragraphs needed by a program when it is loaded into memory. The *number* can be any integer in the range from 1 to 65,535. It must be a decimal, octal, or hexadecimal number. Octal numbers must begin with a zero, and hexadecimal values must begin with a leading zero followed by a lowercase x, for example, 0x2B. If *number* is less than the minimum number of paragraphs needed by the program, Link ignores your request and sets the maximum value equal to the minimum needed.

If you do not specify a *number,* MS-DOS will allocate the largest contiguous block of available memory.

`/EXEPACK` or `/E`

directs Link to remove sequences of repeated bytes (typically nulls) and optimize the load-time relocation table before creating the executable file. Executable files linked with the /EXEPACK parameter may be smaller, and, thus, load faster than files linked without the parameter. However, the Microsoft Symbolic Debug Utility (SYMDEB) cannot be used with packed files.

`/DOSSEG` or `/DO`

causes Link to arrange all segments in the executable file according to the MS-DOS segment-ordering convention. This convention has the following rules:

- All segments having the class name CODE are placed at the beginning of the executable file.
- Any other segments that do not belong to the group, DGROUP, are placed immediately after the CODE segments.
- All segments belonging to DGROUP are placed at the end of the file.

`/DSALLOCATE` or `/D`

directs Link to reverse its normal processing when assigning addresses to items belonging to the group named DGROUP. Normally, Link assigns the offset 0000H to the lowest byte in a group. If you use /DSALLOCATE, Link assigns the offset FFFFH to the highest byte in the group. The result is data that appear to be loaded as high as possible in the memory segment containing DGROUP.

Typically, you use the /DSALLOCATE parameter with the /HIGH parameter to take advantage of unused memory before the start of the program. Link assumes that all free bytes in DGROUP occupy the memory preceding the program. To use the group, you must set a segment register to the start address of DGROUP.

`/HELP` or `/HE`

causes Link to write a list of the available parameters on the screen. If you ever need a reminder of the available parameters, you may find this list convenient. Do not give a file name when using the /HELP parameter.

`/HIGH` or `/H`

sets a program's starting address to the highest possible address in free memory. If you don't use the /HIGH parameter, Link sets the program's starting address as low as possible in memory.

`/LINENUMBERS` or `/LI`

directs Link to copy the starting address of each program source line to a map file. The starting address is actually the address of the first instruction that corresponds to the source line. You can use the MAPSYM program to copy line-number data to a symbol file, which can then by used by SYMDEB.

Link copies the line number data only if you give a map file name in the Link command, and only if the given object file has line-number information. Line numbering is available in some high-level-language compilers, including Microsoft FORTRAN and Pascal, versions 3.0 and later, and Microsoft C, versions 2.0 and later. MASM does not copy line number information to the object file. If an object file has no line number information, Link ignores the /LINENUMBERS parameter.

`/MAP` or `/M`

causes Link to produce a listing of all public symbols declared in your program. This list is copied to the map file that Link creates. The /MAP parameter is required if you want to use SYMDEB for symbolic debugging.

If you do not specify a map file in a Link command, you can use the /MAP parameter to force Link to create one. Link gives the forced map file the same name as the first object file specified in the command. It also adds the default extension .MAP.

`/NODEFAULTLIBRARYSEARCH` or `/NOD`

directs Link to ignore any library names it may find in an object file. A high-level language compiler may add a library name to an object file to ensure that a default set of libraries is linked with the program. Using this parameter overrides these default libraries and lets you explicitly name the libraries you want by including them in the Link command.

/NOGROUPASSOCIATION or **/NOG**

directs Link to ignore group associations when assigning addresses to data and code items.

This parameter exists strictly for compatibility with older versions of FORTRAN and Pascal (Microsoft versions 3.13 or earlier, or any IBM version prior to 2.0). You should never use the /NOGROUPASSOCIATION parameter except to link with object files produced by those compilers, or with the run-time libraries that accompany the old compilers.

/NOIGNORECASE or **NOI**

differentiates between uppercase and lowercase letters in symbol names.

If you are linking modules created with MASM to modules created with a case-sensitive language such as C, make sure public symbols have the same sensitivity in both modules. Another alternative would be to use the /ML or /MX parameter to make public variables in MASM case-sensitive. Then link with the /NOIGNORECASE parameter.

/OVERLAYINTERRUPT:*number* or **/O:***number*

sets the interrupt number of the overlay loading routine to number. This parameter overrides the normal overlay interrupt number (03FH).

number can be any integer value in the range from 0 to 255. It must be a decimal, octal, or hexadecimal number. Octal numbers must have a leading zero, and hexadecimal numbers must start with a leading zero followed by a lowercase x, for example, 0x3B.

MASM does not have an overlay manager. Therefore, you can use this parameter only if you are linking with a run-time module from a language compiler that supports overlays. Check your compiler documentation, since you may not be able to use this parameter with some compilers.

You should not use interrupt numbers that conflict with the standard MS-DOS interrupts.

/PAUSE or /P

causes Link to pause before writing the executable file to disk so that you can swap disks before Link writes the executable (.EXE) file to disk.

If you specify /PAUSE, Link displays the following message before creating the run file:

```
About to generate .EXE file
Change diskette in drive x:
and press <ENTER>
```

This message appears after Link has read data from the object files and library files, and after it has written data to the map file, if you specified one. After it writes the executable file to disk, it displays the following message:

```
Please replace original diskette
in drive letter and press <ENTER>
```

Do not remove the disk used for the VM.TMP file, if such a file has been created. If the temporary disk message appears when you have specified /PAUSE, you should press Ctrl C to terminate the Link session. Rearrange your files so that the temporary file and the executable file can be written to the same disk, then try again.

`/SEGMENTS:`*number* or
`/SE:`*number*

directs Link to process no more than *number* segments per program. If it encounters more than the given limit, Link displays an error message, and stops linking. You use *number* to override the default limit of 128 segments.

If you do not use /SEGMENTS, Link allocates enough memory space to process up to 128 segments. If your program has more than 128 segments, you will need to set the segment limit higher to increase the number of segments that Link can process. If you get the following Link error message, you should set the segment limit lower:

```
Segment limit set too high
```

The *number* can be any integer value in the range from 1 to 1024. It must be a decimal, octal, or hexadecimal number. Octal numbers must have a leading zero, and hexadecimal numbers must start with a leading zero followed by a lowercase x, for example, 0x4B.

`/STACK:`*size* or `/ST:`*size*

sets the program stack to the number of bytes given by size. Link usually calculates a program's stack size automatically, basing it on the size of any stack segments given in the object files. If you do use /STACK, Link uses the value you type in place of any value it may have calculated. The size can be any positive integer in the range from 1 to 65,535. This value can be a decimal, octal, or hexadecimal number. Octal numbers must begin with a zero, and hexadecimal numbers must begin with a leading zero followed by a lowercase x, for example, 0x1B.

By using the EXEMOD utility, you can also change the stack size after linking.

10

Debug

The Debug utility is a debugging program that provides a controlled testing environment for binary and executable object files. Edlin, the MS-DOS line editor, is used to alter source files; Debug is Edlin's counterpart for binary files.

Debug eliminates the need to reassemble a program to see if a problem has been fixed by a minor change. It allows you to alter the contents of a file or the contents of a CPU register, and then immediately re-execute a program to check the validity of the changes made.

All Debug commands may be aborted at any time by pressing Ctrl C. Ctrl S suspends the display, so that you can read it before the output scrolls away. Pressing any key other than Ctrl C or Ctrl S restarts the display.

How to Start Debug

Debug may be started two ways. By the first method, you type all commands in response to the Debug prompt (a hyphen). By the second method, you type all commands on the line used to start Debug.

Method 1: `DEBUG`

Method 2: `DEBUG` *file parameters*

Method 1: DEBUG Without Parameters

To start Debug using Method 1, simply type the following:

```
DEBUG
```

Debug responds with the hyphen (-) prompt, signaling that it is ready to accept your commands. Since you didn't specify a filename, you can use other commands to work on current memory, disk sectors, or disk files.

Notes

- When Debug (version 3.0) is started, it sets up a program header at offset 0 in the program work area. In previous versions of Debug, you could overwrite this header. You can still overwrite the default header if you don't give a *file* to Debug. If you are debugging a .COM or .EXE file, however, do not tamper with the program header below address 5CH, or Debug will terminate.
- Do not restart a program after the following message is displayed:

  ```
  Program terminated normally
  ```

  You must reload the program with the N (Name) and L (Load) commands for it to run properly.

Method 2: DEBUG With the File Name and Parameters

To start Debug using a command, you must use the following syntax:

```
DEBUG file parameters
```

where *file* is the file to be debugged and *parameters* is the list of file parameters that are to be passed to the program *file.*

For example, to start Debug and debug the file FILE.EXE, enter:

```
DEBUG FILE.EXE
```

Debug would then load FILE.EXE into memory starting at 100 (hexadecimal) in the lowest available segment. The BX:CX registers are loaded with the number of bytes placed into memory.

Debug Command Information

Each Debug command consists of a single letter followed by one or more parameters. If a syntax error occurs in a Debug command, Debug reprints the command and indicates the error with a caret (^) and the word `Error` as in the following example,

```
DCS:100 CS:110
   ^ Error
```

Notice that when typing commands and parameters you may use any combination of uppercase and lowercase letters. As you work with Debug, you can use the control characters and special editing functions described in the "MS-DOS Editing and Function Keys" chapter.

Debug Syntax Conventions

Debug commands have required or optional parameters which modify or enhance the function of the command. The command syntax shows you how to use the Debug command parameters. This manual uses the following conventions for command syntax:

CAPS — Commands and parameters in CAPS should be typed as shown in the syntax. (When actually entering the command or parameter, you may use uppercase or lowercase characters, or a combination.)

Blue — Parameters printed in blue are optional. In most cases you can use one or more of the optional parameters in a command.

Italics — Parameter characters or words shown in *italics* indicate variables. You must supply the specific number or text value for a variable. For example, if you see the word *file* in the command syntax, it means that

you should type the name of your file.

... — An ellipsis indicates that you can repeat the parameters as many times as necessary.

Enter all punctuation (except the ellipsis), such as commas, colons and equal signs, exactly as shown in the syntax.

Debug Command Descriptions

All Debug commands accept parameters, except the Q (Quit) command. Parameters may be separated by delimiters (spaces or commas), but a delimiter is required only between two consecutive hexadecimal values. Thus, the following commands are equivalent:

```
DCS::100 110
D CS:100 110
D,CS:100,110
```

Parameter Definitions

The following parameters are used in Debug commands. Refer to these when necessary.

Parameter **Definition**

drive: A one-digit hexadecimal value that indicates which drive a file will be loaded from or written to. The valid values are 0–3, where 0=A:, 1=B:, 2=C:, 3=D:.

byte A two-digit hexadecimal value placed in or read from an address or register.

record A one-digit to three-digit hexadecimal value that indicates the logical record number on the disk and the number of disk sectors to be written or loaded. Logical records correspond to sectors; however, since they represent the entire disk space, their numbering differs.

value A hexadecimal value of up to four digits that specifies a port number or the number of times a command should repeat its functions.

address A two-part designation containing either an alphabetic segment register or a four-digit segment address plus an offset value. You may omit the segment name or segment address, in which case the default segment DS is used for all commands except A, G, L, T, U, and W, for which the default segment is CS. All numeric values are hexadecimal.

Following is a sample *address:*

```
CS:0100
04BA:0100
```

Notice that the colon is required between the segment name (whether numeric or alphabetic) and the offset value.

range

Contains two addresses: for example, *address address;* or one address, an L, and a value: for example, *address* L *value* where *value* is the number of lines on which the command should operate (L80 is assumed). The second type of *range* cannot be used if another hexadecimal value follows, since the hexadecimal value would be interpreted as the second *address* of the *range.*

Here are some sample ranges:

```
CS:100 110
CS:100 L 10
CS:100
```

The following example, however, is illegal:

```
CS:100 CS:110
       ^ Error
```

The limit for *range* is 10000 (hexadecimal). To specify a*value* of 10000 with only four digits, type 0000 (or 0).

list

A series of *byte* values or *strings. List* must be the last parameter on the command line.

Following is a sample *list:*

```
FCS:100 42 45 52 54 41
```

string

Any number of characters enclosed in quotation marks. The quotation marks may be either single (‘ ’) or double (" "). If the delimiter marks must appear within a *string,* you must use the double quotation marks.

For example, the following strings are legal:

```
"This ‘string’ is okay."
```

```
"This ""string"" is okay."
```

However, this string is illegal:

```
""This "string" is not okay.""
```

Notice that the double quotation marks are not necessary in the following strings:

```
"This string is not necessary."
‘This ""string"" is not necessary.’
```

the ASCII values of the characters in the string are used as a *list* of byte values.

count Number of 16KB pages to allocate.

Handle EMS Handle number.

lpage Logical page of Handle to map.

ppage Physical page where logical page will map into.

A (Assemble)

Assembles 8086/8087/8088 mnemonics directly into memory.

Syntax

A*address*

Operation

If it finds a syntax error, Debug responds with the following message, then redisplays the current assembly address:

```
^Error
```

All numeric values are hexadecimal and you must type them as 1–4 characters. Also, you must specify prefix mnemonics *in front of* the opcode to which they refer. You may type them on a separate line, however.

The segment override mnemonics are CS:, DS:, ES:, and SS:. The mnemonic for the far return is RETF. String manipulation mnemonics must explicitly state the string size. For example, use MOVSW to move word strings, and use MOVSB to move byte strings.

The assembler will automatically assemble short, near, or far jumps and calls, depending on byte displacement, to the destination address. You may override these jumps and calls by using a NEAR or far prefix, as in the following example:

```
0100:0500 JMP  502        ; a 2s-byte short jump
0100:0502 JMP  NEAR 505   ; a 3-byte near jump
0100:505  JMP  FAR  50A   ; a 5-byte far jump
```

You may abbreviate the NEAR prefix to NE, but the FAR prefix cannot be abbreviated.

Debug cannot tell whether some operands refer to a word memory location or to a byte memory location. In this case, the data type must be explicitly stated with the prefix, WORD PTR or BYTE PTR. Acceptable abbreviations are WO and BY. For example,

```
NEG     BYTE PTR [128]
```

```
DEC      WO [SI]
```

Debug also cannot tell whether an operand refers to a memory location or to an immediate operand. So it uses the common convention that operands enclosed in square brackets refer to memory. For example,

```
MOV      AX,21  ; Load AX with 21H
MOV      AX,[21]; Load AX with the
                ; contents
                ; of memory location 21H
```

Two popular pseudo-instructions are available with the A (Assemble) command: the DB opcode, which assembles byte values directly into memory; and the DW opcode, which assembles word values directly into memory. Following are examples of both:

```
DB       1,2,3,4,"THIS IS AN EXAMPLE"
DB       ‘THIS IS A QUOTATION MARK: "’
DB       "THIS IS A QUOTATION MARK: ’"

DW       1000,2000,3000,"BACH"
```

The A command supports all forms of register indirect commands. For example,

```
ADD      BX,34[BP+2].[SI-1]
POP      [BP+DI]
PUSH     [SI]
```

All opcode synonyms are also supported, as in the next example:

```
LOOPZ    100
LOOPE    100

JA       200
JNBE     200
```

For 8087 opcodes, the WAIT or FWAIT prefixes must be explicitly specified, as in the last example:

```
FWAIT FADD ST,ST(3)     ; This line assembles
                        ; an FWAIT prefix
```

C (Compare)

Compares the portion of memory specified by *range* to a portion of the same size beginning at the specified *address.*

Syntax

C*range address*

Operation

If the two areas of memory are identical, there is no display, and debug returns with the MS-DOS prompt. If there are differences, they are displayed in this format:

address1 byte1 byte2 address2

Example

The following commands have the same effect:

```
C100,1FF 300
```

or:

```
C100L100 300
```

Each command compares the block of memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

D (Dump)

Displays the contents of the specified *range* of memory.

Syntax

D*range*

Operation

If you specify a *range* of addresses with the D (Dump) command, the contents of the *range* are displayed. If you don't use parameters with the D command, 128 bytes are displayed at the first address (DS:100) after the address displayed by the previous D command.

The dump is displayed in two portions: a hexadecimal dump (each byte is shown in hexadecimal value) and an ASCII dump (the bytes are shown in ASCII characters). Nonprinting characters are denoted by a period (.) in the ASCII portion of the display. Each display line shows 16 bytes, with a hyphen between the eighth and ninth bytes. Each displayed line begins on a 16-byte boundary.

Examples

If you type the command

```
DCS:100 110
```

Debug displays the dump in the following format:

```
04BA:0100 42 45 52 54 41 ... 4E 44 TOM SAWYER
```

If you simply type `D`, the display is formatted as just described. Each line of the display begins with an address incremented by 16 from the address on the previous line. Each subsequent `D` (typed without parameters) displays the bytes immediately following those last displayed.

If you type the following command, the display is formatted as described above, but 20H bytes are displayed:

```
DCS:100 L 20
```

If you then type the following command, the display is formatted as described above, but all the bytes in the range of lines from 100H to 115H in the CS segment are displayed:

```
DCS:100 115
```

E (Enter)

Enters byte values into memory at the specified address.

Syntax

E*address list*

Operation

If, when using the E (Enter) command, you type the optional *list* of values, the byte values are replaced automatically. (If an error occurs, no byte values are changed.)

If you type the *address* without the optional *list*, Debug displays the *address* and its contents, repeats the *address* on the next line, and then waits for your input. At this point, the E (Enter) command waits for you to perform one of the following actions:

- Replace a byte value with a value you type. Simply type the value after the current value. If the one you type is not a legal hexadecimal value or if it contains more than two digits, the illegal or extra character is not echoed.
- Press the spacebar to advance to the next byte. To change the value, simply type the new value as described in the previous action. If, when you press the spacebar, you move beyond an 8-byte boundary, Debug starts a new display line with the address displayed at the beginning.
- Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, typing the hyphen returns the current position to the previous byte. When you type the hyphen, a new line is started with its address and byte value displayed.
- Press (Enter) to terminate the E command. (Enter) may be pressed at any byte position.

Examples

Suppose you type the following command:

```
ECS:100
```

Now suppose that Debug displays the following:

```
04BA:0100  EB._
```

To change this value to, say, 41, type the number 41 at the cursor, as shown:

```
04BA:0100  EB.41_
```

To step through the subsequent bytes, you would press the spacebar until you saw the following:

```
04BA:0100  EB.41    10. 00. BC._
```

To change BC to the number 42, for instance, you would type the number at the cursor, as follows:

```
04BA:0100  EB.41    10. 00. BC.42_
```

notice that the value 10 should be 6F. To correct this value, you would type the hyphen as many times as needed to return to byte 0101 (value 10), then replace 10 with 6F:

```
04BA:0100  EB.41    10. 00. BC.42-
04BA:0102  00.-_
04BA:0101  10.6F_
```

Pressing Enter ends the E command and returns you to the Debug command level.

F (Fill)

Fills the addresses in the specified *range* with the values in the specified *list.*

Syntax

F*range list*

Operation

If the *range* contains more bytes than the number of values in the *list,* the *list* will be used repeatedly until all bytes in the *range* are filled.

If the *list* contains more values than the number of bytes in the *range,* the extra values in the *list* are ignored. If any of the memory in the *range* is not valid (bad or nonexistent), the error will occur in all succeeding locations.

Example

Suppose you type the following command:

```
F04BA:100 L 100 42 45 52 54 41
```

In response, Debug would fill memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values would then be repeated until all the 100H bytes were filled.

G (Go)

Executes the program currently in memory.

Syntax

G=*address* *addresses*

Operation

If you type the G (Go) command by itself, the program currently in memory executes as if it had run outside Debug.

If you set *=address,* execution of the G command begins at the address specified. The equal sign (=) is required so that debug can distinguish the start *=address* from the breakpoint *addresses.*

With the other optional *addresses* set, execution stops at the first *address* encountered, regardless of that address' position in the list of addresses that halt execution or program branching. When program execution reaches a breakpoint, the registers, flags, and decoded instruction are displayed for the last instruction executed. The result is the same as if you had typed the R (Register) command for the breakpoint address.

You may set up to ten breakpoints, but only at addresses containing the first byte of an 8086 opcode. If you set more than ten breakpoints, Debug returns the BP error message.

The user stack pointer must be valid and must have 6 bytes available for this command. The G command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, Code Segment register, and Instruction pointer are pushed on the user stack. (If the user stack is not valid or is too small, the operating system may crash.) An interrupt code (0CCH) is placed at the specified breakpoint address(es).

When Debug encounters an instruction with the breakpoint code, it restores all breakpoint addresses to their original instructions. If you don't halt execution at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

Example

Suppose you type the following command:

```
GCS:7550
```

The program currently in memory would execute up to the address 7550 in the CS segment. Debug would then display registers and flags, after which the G command would terminate.

After Debug has encountered a breakpoint, if you type the G command again the program runs as if you had typed the filename at the MS-DOS command level. The only difference is that program execution begins at the instruction after the breakpoint, rather than at the usual start address.

H (Hex)

Performs hexadecimal arithmetic on the two specified parameters.

Syntax

`H`*value value*

Operation

First, Debug adds the two parameters, then subtracts the second parameter from the first. The results of these actions are displayed on one line–first the sum, then the difference.

Example

Suppose you type the following command:

```
H19F 10A
```

In response, Debug would perform the calculations and then display the following result:

```
02A9    0095
```

I (Input)

Inputs and displays one byte from the port specified by value. A 16-bit port address is allowed.

Syntax

I*value*

Example

Suppose you type the following command:

```
I2F8
```

Suppose also that the byte at the port is 42H. Debug would input the byte and then display the following:

```
42
```

L (Load)

Loads a file into memory.

Syntax

L*address drive: record record*

Operation

Set BX:CX to the number of bytes read. The file must have been named either when you started Debug or with the N (Name) command. Both the Debug invocation and the N command format a file name properly in the normal format of a file control block at CS:5C.

If you use the L (Load) command without any parameters, Debug loads the file into memory beginning at address CS:100 and sets BX:CX to the number of bytes loaded. If you type the L command with an address parameter, loading begins at the memory location specified by the *address.* If you use the L command with all parameters included, absolute disk sectors are loaded, instead of a file.

Each *record* is taken from the specified *drive:* (the drive name is numeric: 0=A:, 1=B:, 2=C:, etc.). Debug begins loading with the first specified *record,* and continues until the number of sectors in the second *record* have been loaded.

Example

Suppose, once you have started Debug, that you type the following commands:

```
-NFILE.COM
```

Now, to load FILE.COM, you would simply type the L command.

Debug would then load the file and display the Debug prompt. Suppose now that you want to load only portions of a file or certain records from a disk. To do this, you would type the following:

```
L04BA:100 2 OF 6D
```

Debug would then load 109 (6DH) records, beginning with logical record number 15, into memory beginning at address 04BA:0100. Then, once it had loaded the records, Debug would simply return the hyphen (-) prompt.

If the file has an .EXE extension, it would be relocated to the load address specified in the header of the .EXE file. The *address* parameter is always ignored for .EXE files. The header itself is stripped off the .EXE file before it is loaded into memory. So, the size of an .EXE file on disk will differ from its size in memory.

If the file was named by the N (Name) command, or specified when you started Debug, as a .HEX file, then typing the L command with no parameters would cause Debug to load the file beginning at the address specified in the .HEX file. If the L command included the option *address,* Debug would add the address specified in the L command to the address found in the .HEX file to determine the start address at which to load the file.

M (Move)

Moves the block of memory specified by *range* to the location beginning at the specified *address.*

Syntax

`M`*range address*

Operation

Overlapping moves (i.e., moves where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that could be overwritten are moved first. For moves from higher to lower addresses, the sequence of events is to first move the data beginning at the block's lowest address and then work toward the highest. For moves from lower to higher addresses, the sequence is to first move the data beginning at the block's highest address and then work toward the lowest.

Notice that if the addresses in the block being moved will not have new data written to them, the data in the block *before the move* will remain. The M (Move) command copies the data from one area into another, in the sequence described, and writes over the new addresses. This action is why the sequence of the move is important.

Example

Suppose you type the following command:

```
MCS:100 110 CS:500
```

In response, Debug would first move address CS:110 to CS:510, then CS:10F to CS:50F, and so on until it has moved CS:100 to CS:500. To review the results of the move, you could type the D (Dump) command, using the same *address* you used with the M command.

N (Name)

Sets file names.

Syntax

N*file file . . .*

Operation

The N (Name) command performs two functions. First, it assigns a file name for a later L (Load) or W (Write) command. So if you start Debug without naming a file to be debugged, you must type the command `N` *file* before a file can be loaded. Second, the N command assigns file name parameters to the file being debugged. In this case, N accepts a list of parameters used by the file being debugged.

Notice that these two functions overlap. Consider, for example, the following set of Debug commands:

```
-NFILE1.EXE
-L
-G
```

The N command would use these commands to perform the following steps:

1. It would first assign the file name FILE1.EXE to the file to be used in any later L or W commands.
2. It would also assign the FILE1.EXE file name to the first filename parameter used by any program that is later debugged.
3. The L command would then load FILE1.EXE into memory.
4. The G (Go) command would cause FILE1.EXE to be run with FILE1.EXE as the single file name parameter (that is, FILE1.EXE would be run as if FILE1.EXE had been typed at the command level).

A more useful chain of commands might look like this:

```
-NFILE1.EXE
-L
-NFILE2.DAT FILE3.DAT
```

```
-G
```

In this example, the N command sets FILE1.EXE as the file name for the subsequent command, which loads FILE1.EXE into memory. The N command is then used again, this time to specify the parameters to be used by FILE1.EXE. Finally, when the G command is run, FILE1.EXE is executed as if FILE1 FILE2.DAT FILE3.DAT had been typed at the MS-DOS command level. Notice that if you were to execute a W command now, then FILE1.EXE—the file being debugged—would be saved with the name FILE2.DAT. To avoid this kind of result, you should always execute an N command before either an L or W command.

There are four regions of memory that can be affected by the N command:

Memory Location	Contents
CS:5C	FCB for file 1
CS:6C	FCB for file 2
CS:80	Count of characters
CS:81	All characters typed

The first file name parameter that you specify for the N command has a file control block (FCB) set up at CS:5C. If you name a second file name parameter, an FCB is set up for this parameter beginning at CS:6C. The number of characters typed in the N command (exclusive of the first character, N) is given at location CS:80.

The actual stream of characters given by the N command (again,exclusive of the letter N) begins at CS:81. Notice that this stream of characters may contain switches and delimiters that would be legal in any command typed at the MS-DOS command level.

Example

A typical use of the N command is as follows:

```
 DEBUG PROG.COM
-NPARAM1  PARAM2/C
-G
-
```

In this case, the G command executes the file in memory as if you had typed the following command line:

```
PROG PARAM1 PARAM2/C
```

Testing and debugging therefore reflect a normal run-time environment for PROG.COM.

O (Output)

Sends the specified *byte* to the output port specified by *value.* A 16-bit port address is allowed.

Syntax

O*value byte*

Example

Suppose you want Debug to output the byte value 4F to output port 2F8. To do this, you could type the following command:

```
O2F8 4F
```

P (Proceed)

Executes a loop, a repeated string instruction, a software interrupt, or a subroutine call to completion.

Syntax

P=*address number*

where *address* is the location of the first instruction to be executed and *number* is the number of instructions to execute.

Operation

The P (Proceed) command transfers control from Debug to the target program. The program executes without interruption until the loop, repeated string instruction, software interrupt, or subroutine call at address is completed, or until the specified number of machine instructions has been executed. Control then returns to Debug, which displays the contents of the target program's registers and the status of the flags.

If the *address* parameter does not include an explicit segment, Debug uses the target program's CS register; if *address* is omitted entirely, execution begins at the address specified by the target's CS:IP registers. The *address* parameter must be preceded by an equal sign (=) to distinguish it from *number.*

If the instruction at *address* is not a loop, a repeated string instruction, a software interrupt, or a subroutine call, the P command functions just like the T (Trace) command. The optional *number* parameter specifies the number of instructions to be executed before control returns to Debug. If *number* is omitted, Debug executes only one instruction. After each instruction is executed, Debug displays the contents of the target program's registers, the status of the flags, and the next instruction to be executed.

Caution The P command cannot be used to trace through ROM.

Example

Assume that the target program's location CS:143FH contains a call instruction. To execute the subroutine that is the destination of call and then return control to Debug, type:

```
-P =143F
```

Q (Quit)

Terminates the Debug utility.

Syntax

```
Q
```

Operation

The Q (Quit) command takes no parameters and exits Debug without saving the file you're currently working with. You are returned to the MS-DOS command level.

Example

To end the debugging session, type

```
    Q
```

and press (Enter). Debug terminates, and control ret.urns to the MS-DOS command level.

R (Register)

Displays the contents of one or more CPU registers.

Syntax

R *register-name*

Operation

If you do not type a *register-name*, the R (Register) command dumps the register storage area and displays the contents of all registers and flags.

If you do type a *register-name*, the 16-bit value of that register is displayed in hexadecimal, and a colon then appears as a prompt. You can now either type a *value* to change the register, or press (Enter) if you don't want a change.

AX	BP	SS
BX	SI	CS
CX	DI	IP
DX	DS	PC
SP	ES	F

(IP and PC both refer to the Instruction Pointer.)

Any other entry for *register-name* results in a BR error message.

If you type F as the *register-name*, Debug displays each flag with a two-character alphabetic code. To change any flag, type the opposite two-letter code. The flags are then either set or cleared.

The flags are listed in the following table with their codes for SET and CLEAR:

Flag Name	Set	Clear
Overflow	OV	NV
Direction	DN (Decrement)	UP (Increment)
Interrupt	EI (Enabled)	DI (Disabled)
Sign	NG (Negative)	PL (Plus)
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE (Even)	PO (Odd)
Carry	CY	NC

Whenever you type the RF command, the flags are displayed (in the order shown in the previous table) in a row at the beginning of a line. At the end of the list of flags, Debug displays a hyphen (-).

You may enter new flag values in any order as alphabetic pairs. You do not have to leave spaces between these values. To exit the R command, press (Enter). Any flags for which you did not specify new values remain unchanged.

If you type more than one value for a flag, Debug returns a DF error message. If you enter a flag code other than one of those shown in the table above, Debug returns a BF error message. In both cases, the flags up to the error in the list are changed; those flags at and after the error are not. When you start Debug, the segment registers are set to the bottom of free memory, the Instruction Pointer is set to 0100H, all flags are cleared, and the remaining registers are set to zero.

Example

If you type the following command, Debug displays all registers, flags, and the decoded instruction for the current location:

```
R
```

If the location is CS:11A, for example, the display will look similar to this:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000
SI=005C DI=0000 DS=04BA ES=04BA SS=04BA CS=04BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21          INT     21
```

If you then type the following command,

```
RF
```

Debug will display these flags:

```
NV UP DI NG NZ AC PE NC - _
```

Now, you could type any valid flag designation, in any order, with or without spaces. For example,

```
NV UP DI NG NZ AC PE NC - PLEICY
```

In response, Debug would display the Debug prompt. To see the changes, type either the R or RF command:

```
RF
NV UP EI PL NZ AC PE CY - _
```

Press Enter to leave the flags this way or to specify different flag values.

S (Search)

Searches the specified *range* for the specified *list* of bytes.

Syntax

S*range list*

Operation

The *list* may contain one or more bytes, each separated by a space or comma. If the *list* contains more than one byte, only the first address of the byte string is returned. If the *list* contains only one byte, all addresses of the byte in the *range* are displayed.

Example

Suppose you type the following command:

```
SCS:100 110 41
```

Debug would display a response similar to

```
04BA:0104
04BA:010D
-type:
```

T (Trace)

Executes one instruction and displays the contents of all registers, flags, and the decoded instruction.

Syntax

T=*address value*

Operation

If you include the *=address* option in the T (Trace) command, tracing occurs at the specified *=address.* The *value* option causes Debug to execute and trace the number of steps specified by *value.*

The T command uses the hardware trace mode of the 8086 or 8088 microprocessor. Consequently, you may also trace instructions stored in ROM (Read Only Memory).

Example

Suppose you type the following command:

```
T
```

In response, Debug would return a display of the registers, flags, and decoded instruction for that one instruction. Assuming, for this example, that the current position is 04BA:011A, Debug might return the following display:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000
SI=005C DI=0000 DS=04BA ES=04BA SS=04BA CS=04BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21          INT     21
```

If you type the following command,

```
T=011A 10
```

Debug executes sixteen (10 hex) instructions beginning at 011A in the current segment, and then displays all registers and flags for each instruction as it is executed. The display scrolls away until the last instruction is executed, and then stops. Now you can see the register and flag values for the last few instructions performed.

Remember that if you want to study the registers and flags for any instruction (at any time), you can press (Ctrl) (S) to stop the display from scrolling.

U (Unassemble)

Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

Syntax

U*range*

Operation

The display of disassembled code looks like a listing for an assembled file. If you type the U (Unassemble) command without parameters, 20 hexadecimal bytes are disassembled at the first address after that displayed by the previous U command. If you type the U command and include the *range* parameter, Debug disassembles all bytes in *range.* But if you specify *range* only as an *address,* 20H bytes are disassembled.

Examples

Using the U command, suppose you type the following:

```
U04BA:100 L10
```

In response, Debug would disassemble 16 bytes, beginning at address 04BA:0100:

```
04BA:0100  206472    AND  [SI+72],AH
04BA:0103  69        DB   69
04BA:0104  7665      JBE  016B
04BA:0106  207370    AND  [BP+DI+70],DH
04BA:0109  65        DB   65
04BA:010A  63        DB   63
04BA:010B  69        DB   69
04BA:010C  66        DB   66
04BA:010D  69        DB   69
04BA:010E  63        DB   63
04BA:010F  61        DB   61
```

Now, suppose you type the following:

```
U04BA:0100 0108
```

The display would now show

```
04BA:0100  206472    AND  [SI+72],AH
04BA:0103  69        DB   69
04BA:0104  7665      JBE  016B
04BA:0106  207370    AND  [BP+DI+70],DH
```

If the bytes in some addresses are altered, the disassembler alters the instruction statements. You can then type the U command for the changed locations, for the new instructions viewed, and for the disassembled code used to edit the source file.

W (Write)

Writes the file being debugged to a disk file.

Syntax

W*address drive: record record*

Operation

If you do not use parameters with the W (Write) command, BX:CX must already be set to the number of bytes tc be written; the file is written beginning from CS:100. If you type the W command with just an *address,* the file is written beginning at that *address.* If you have used a G (Go) or T (Trace) command, you must reset BX:CX before using the W command without parameters.

Notice that if a file is loaded and modified, the name, length, and starting address are all set correctly to save the modified file (as long as the length has not changed).

You must have named the file either with the initial Debug startup command or with the N (Name) command (refer to the N command described earlier in this chapter). Both the Debug startup command and the N command properly format a file name in the normal format of a file control block at CS:5C.

If you include parameters when you use the W command, the write begins from the memory address specified; the file is written to the specified *drive:* (the drive name is numeric: 0=A:,1=B:, 2=C:, etc.). Debug writes the file beginning at the logical record number specified by the first *record.* Debug then continues to write the file until the number of sectors specified in the second *record* have been written.

Caution Writing to absolute sectors is *extremely risky* because the process bypasses the file handler.

Example

If you type the following command, Debug writes the contents of memory to the disk in drive B:, beginning with the address CS:100. The data written starts in the disk logical record number 37H and consists of 2BH records.

```
   WCS:100 1 37 2B
```

When the write is complete, Debug displays its prompt again.

XA (Allocate Expanded Memory)

Allocates a specified number of pages as expanded memory (EMS) pages. Displays "Handle created" if successful.

Syntax

`XA` *count*

where *count* is the hexadecimal number of 16K pages to allocate.

Operation

If the specified number of pages is available, Debug displays a message indicating the number of the Handle created. If it is not, Debug displays an error message.

Example

To allocate eight EMS pages, type the following:

```
XA 8
```

If eight pages of EMS memory are available, Debug displays a message similar to the following:

```
Handle created=0003
```

XD (Deallocate Expanded Memory)

Deallocates a Handle.

Syntax

`XD` *handle*

where *handle* is the handle to be deallocated.

Operation

If the Handle number is valid, Debug deallocates the Handle and displays a message.

Example

To deallocate Handle 0003, type the following:

```
XD 0003
```

If the Handle is deallocated successfully, Debug displays the following message:

```
Handle 0003 deallocated
```

XM (Map Expanded Memory Pages)

Maps an EMS logical page, belonging to the specified EMS Handle, into an EMS physical page.

Syntax

`XM` *lpage ppage handle*

where *lpage* is the number of the page among the pages allocated to the *handle* specified and *ppage* is the number of the physical page into which *lpage* is to be mapped.

Example

To map logical page 5 of EMS Handle 0003 to physical page 2, type the following:

```
XM 5 2 0003
```

XS (Get Expanded Memory Status)

Displays EMS information.

Syntax

```
XS
```

Operation

When the Debug XS command is executed, information about expanded memory is displayed in the following format:

```
Handle %1 has %2 pages allocated
Physical page %1 = Frame segment %2
   %1 of a total %2 EMS pages have been allocated
   %1 of a total %2 EMS handles have been allocated
```

Example

To display EMS information, type the following:

```
XS
```

Debug will then display information similar to the following:

```
Handle 0000 has 0000 pages allocated
Handle 0001 has 0002 pages allocated

Physical page 00 = Frame segment C000
Physical page 01 = Frame segment C400
Physical page 02 = Frame segment C800
Physical page 03 = Frame segment CC00
   2 of a total 80 EMS pages have been allocated
   2 of a total FF EMS handles have been allocated
```

Debug Error Messages

During a debug session, you may receive any of the following error messages. Each error ends the debug command under which it occurred, but does not end debug itself.

Error Code	Definition
BF	Bad flag You attempted to change a flag, but the characters you typed were not one of the acceptable pairs of flag values. See the R (register) command for the list of acceptable flag entries.
BP	Too many breakpoints You specified more than ten breakpoints as parameters to the G (go) command. Retype the G command using ten or fewer breakpoints.
BR	Bad register While using the R command, you typed an invalid register name. See the R command for the list of valid register names.
DF	Double flag You typed two values for one flag. You may specify a flag value only once per RF command.

A

How to Use Code Pages

Introduction

MS-DOS 4.0 provides national language support through the use of language-specific code pages. If you live in, or work with, a country other than the United States, you may choose to use the MS-DOS commands that support code-page switching.

What Is a Code Page?

A **code page** is a table that defines the character set you are using. Each **character set** contains 256 entries specific to a country or language. The characters are translated from the code page table and used by your keyboard, screen, and printer. An example of a character set is the set of letters, numbers, and symbols (such as accent marks) used by French-Canadians. When the character set is put in a table for use by MS-DOS, it becomes the Canadian-French code page.

There are two types of code pages: hardware and prepared. A **hardware** code page is built into a device. For example, a printer manufactured for use in Portugal has a Portuguese hardware code page in it. Many devices can use only their own hardware code page.

Prepared code pages are provided in code-page information (.CPI) files in your software. MS-DOS includes the following prepared code pages:

- 437—United States code page.
- 850—Multilingual code page. This code page includes all characters for most languages of European, North American, and South American countries.
- 860—Portuguese code page.
- 863—Canadian-French code page.

- 865—Nordic code page. This code page includes all characters for the Norwegian and Danish languages.

Devices That Can Use Multiple Languages

The following devices can use prepared code pages to switch from one language to another.

Enhanced Graphics Adapter (EGA) or compatible

IBM Personal System/2 display adapter (VGA) or compatible

IBM Proprinter Model 4201 or compatible

IBM Proprinter Model 4208 or compatible

IBM Quietwriter III Printer Model 5202 or compatible

For example, if you have an EGA, you can switch between the United States code page and the Multilingual code page by entering the appropriate MS-DOS commands. With prepared code pages, a single system can support multiple national languages.

National Language Support Codes

When you use commands to set up your system for a national language, MS-DOS checks to make sure the screen, printer, and keyboard codes you request work together. The valid combinations of prepared code pages, country codes, and keyboard codes are defined in the COUNTRY.SYS and KEYBOARD.SYS files. The following combinations are supported:

Country or Language	Country Code	Prepared Code	Keyboard Code
Belgium	032	437,850	BE
Canadian-French	002	863,850	CF
Denmark	045	865,850	DK
Finland	358	437,850	SU
France	033	437,850	FR
Germany	049	437,850	GR
Italy	039	437,850	IT
Latin America	003	437,850	LA
Netherlands	031	437,850	NL
Norway	047	865,850	NO
Portugal	351	860,850	PO
Spain	034	437,850	SP
Sweden	046	437,850	SV
Switzerland *	041	437,850	SF, SG
United Kingdom	044	437,850	UK
United States	001	437,850	US

* Both Swiss-French and Swiss-German use country code 041.

For example, if you use country code 002 (Canadian-French), you can use prepared code pages 863 and 850 and the CF (Canadian-French) keyboard code. You could not use the US (United States) keyboard code. If you enter a country code 002 with a US keyboard code, MS-DOS displays an error message when you restart the system.

How to Use Code Pages

Setting the System Code Page

MS-DOS uses several commands to support a national language with a code page. To set up your system to support a character set, you need to do the following:

In your CONFIG.SYS file:

- Use the country configuration command to control country-specific characteristics such as the time format, date format, currency symbol, and character-sorting sequence.
- Use a device configuration command to tell MS-DOS the hardware code page in a device and to allocate buffers for prepared code pages.

Note The "System Configuration" chapter explains each configuration command. Examples of CONFIG.SYS commands are shown later in this chapter. Also, remember that when you change your CONFIG.SYS file, you must restart MS-DOS to enable the new settings.

In your AUTOEXEC.BAT file:

- Use an NLSFUNC command to load the memory-resident national support functions. If you forget to issue the NLSFUNC command, MS-DOS will not allow you to specify code pages or keyboard codes.
- Use a MODE CP PREPARE command to prepare code pages for each device that supports code-page switching.
- Use a KEYB command to select the initial keyboard layout.
- Use a CHCP command to select the code pages for all prepared devices. MS-DOS automatically prepares two system code pages and selects the primary code page for your country automatically. If you want to use the other code page prepared for your country, you can use the CHCP command again.

Note For an explanation of each command, see the "MS-DOS Command Descriptions" chapter. Examples of how to use the commands are shown in this chapter.

Several examples, using different hardware and prepared code pages, are provided on the following pages. The examples show you how to set up your system to support national languages. The examples assume all MS-DOS files are in the directory \DOS on drive C:.

Setting Country and Keyboard Codes

The first example uses only a hardware code page; code-page switching is not used. The commands tell MS-DOS to accept the country information for France (code page 437) and loads the French keyboard program. This example covers most European countries.

In your CONFIG.SYS file, type the following:

```
COUNTRY=033 ,,C:\DOS\COUNTRY.SYS
```

In your AUTOEXEC.BAT file, type the following:

```
KEYB FR ,,C:\DOS\KEYBOARD.SYS
```

Using Hardware and Prepared Code Pages

The next example uses two code pages—one hardware and one prepared. In this instance, the system you're setting up uses the German code page provided in the EGA and the United States code page provided by MS-DOS. In your CONFIG.SYS file, type the following:

```
COUNTRY=049 ,,C:\DOS\COUNTRY.SYS
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,437,1)
```

These CONFIG.SYS statements tell MS-DOS that the hardware code page (437) in the enhanced graphics adapter should be used and that you are allocating space for one prepared code page. Note that EGA in the DISPLAY.SYS statement is also used if you have VGA.

In your AUTOEXEC.BAT file, type the following:

```
NLSFUNC
MODE CON CODEPAGE PREPARE=((850)C:\DOS\EGA.CPI)
KEYB GR ,,C:\DOS\KEYBOARD.SYS
CHCP 437
```

These AUTOEXEC.BAT commands do the following:

- Load memory-resident, national support functions.
- Load prepared code page 850 into memory, getting the code page from the EGA.CPI file.
- Load the German (GR) keyboard program.
- Start the system with the United States (437) code page.

How to Set Device Code Pages

The following example uses two prepared code pages—no hardware code pages are used—to set up the system to support Denmark/Norway and the Multilingual national languages.

In your CONFIG.SYS file, type the following:

```
COUNTRY=045 ,,C:\DOS\COUNTRY.SYS
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,2)
```

These CONFIG.SYS commands tell MS-DOS that the EGA's hardware code page is not used and that you are allocating space for two prepared code pages.

In your AUTOEXEC.BAT file, type the following:

```
NLSFUNC
MODE CON CODEPAGE PREPARE=((865,850) C:\DOS\EGA.CPI)
KEYB DK ,,C:\DOS\KEYBOARD.SYS
CHCP 865
```

These AUTOEXEC.BAT commands do the following:

- Load memory-resident, national support functions.
- Load prepared code pages 865 and 850 into memory, getting the code pages from the EGA.CPI file.

- Load the Denmark (DK) keyboard program.
- Start the system with the Denmark/Norway (865) code page.

Setting Parallel Printer Code Pages

The last example also uses two prepared code pages. And it assumes you have an IBM Proprinter model 4201 connected to LPT1. The system is set up to support Denmark/Norway and the Multilingual national languages.

In your CONFIG.SYS file, type the following:

```
COUNTRY=045 ,,C:\DOS\COUNTRY.SYS
DEVICE=C:\DOS\DISPLAY.SYS CON=(EGA,,2)
DEVICE=C:\DOS\PRINTER.SYS LPT1=(4201,,2)
```

These CONFIG.SYS commands tell MS-DOS that the hardware code page is not used and that you are allocating space for two prepared code pages in each device.

In your AUTOEXEC.BAT file, type the following:

```
NLSFUNC
MODE CON CODEPAGE PREPARE=((865,850)C:\DOS\EGA.CPI)
MODE LPT1 CODEPAGE PREPARE=((865,850)C:\DOS\4201.CPI)
KEYB DK,,C:\DOS\KEYBOARD.SYS
CHCP 865
```

These AUTOEXEC.Bat commands do the following:

- Load memory-resident, national support functions.
- Load prepared code pages 865 and 850 into memory. For the EGA, they get the code pages from the EGA.CPI file. For the printer, they get the code pages from the 4201.CPI file.
- Load the Denmark (DK) keyboard program.
- Start both devices with the Denmark/Norway (865) code page.

How to Switch Between Code Pages

After using the CONFIG.SYS and AUTOEXEC.BAT file commands to set up your system for multiple national languages, use the following command to change to a different code page on all devices or for a single prepared device:

```
CHCP code page
```

or

```
MODE device CP SELECT=code page
```

where *code page* is one of the pages set up in AUTOEXEC.BAT and *device* is CON or LPT1.

How to Display Current Code Pages

You can display the current prepared and selected code pages for your console screen or a parallel printer by using the MODE command in the following form:

```
MODE device CODEPAGE
```

For example, to display the current code pages for your console screen device, type the following:

```
MODE CON CODEPAGE
```

MS-DOS displays a message similar to this one:

```
Active codepage for device CON is 437
hardware codepages:
   Codepage 850
prepared codepages:
   Codepage 437
   Codepage 850
   Codepage not prepared
   Codepage not prepared
MODE Status Codepage function completed
```

How to Refresh Lost Code Pages

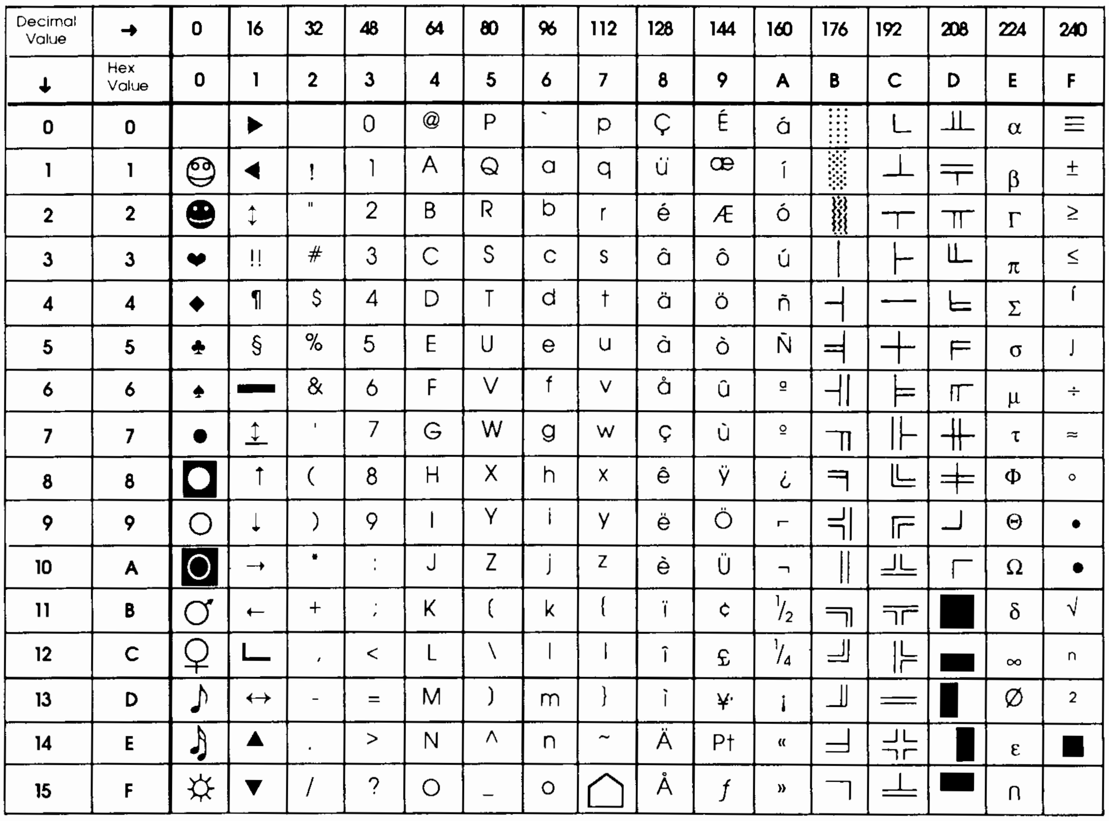
It is possible for prepared code pages to be lost due to hardware errors or other reasons. For example, if you prepared code pages for your printer, and then turned off the printer, the current code page may be lost. You can use the REFRESH parameter with the MODE command to restore the lost code page.

To illustrate, suppose you had selected the Canadian-French code page (863) as the active code page for your console screen (CON). But, because of a hardware error, the active code page was lost. You could type the following commands to reinstate the active code pages for your screen:

```
MODE CON CODEPAGE PREPARE=((863)EGA.CPI)

MODE CON CP REFRESH
```

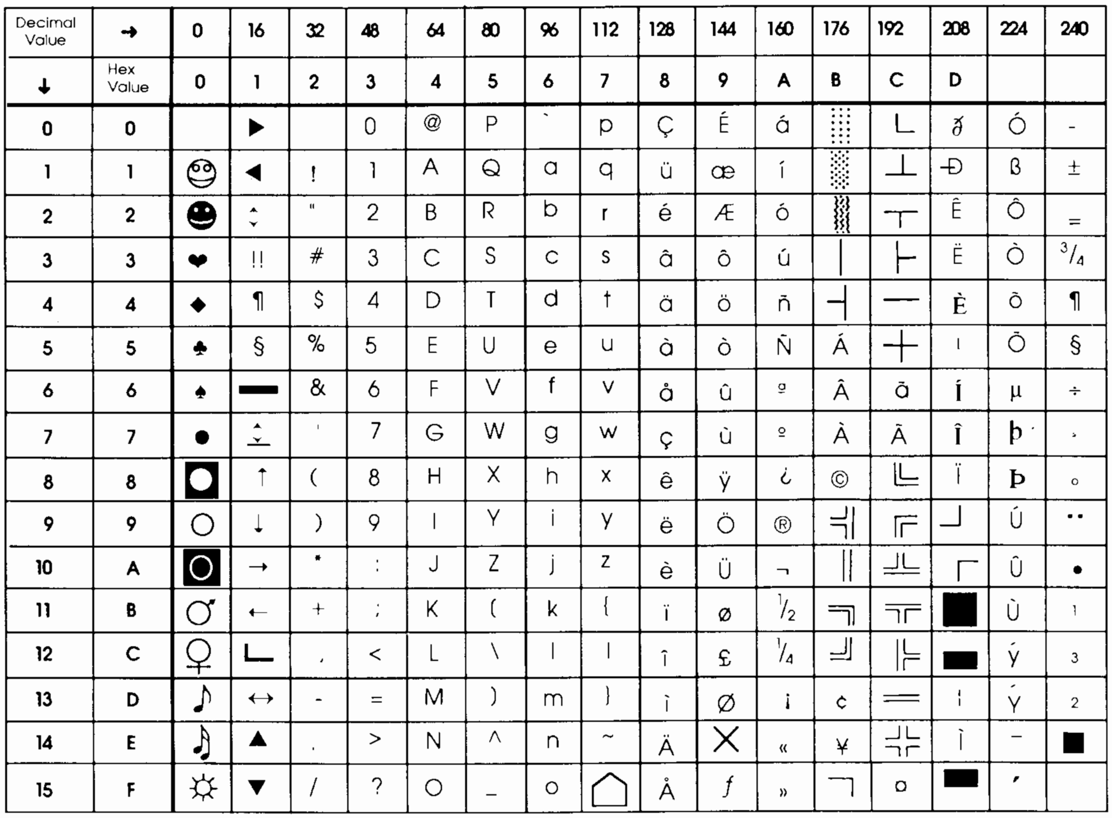
Code Page Tables



Decimal Value	→	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
↓	Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0		►		0	@	P	`	p	Ç	É	á	░	└	╨	α	≡
1	1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
2	2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7	7	•	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	◙	→	•	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	●
11	B	♂	←	+	;	K	(	k	{	ï	¢	½	╗	╦	█	δ	√
12	C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
13	D	♪	↔	-	=	M	)	m	}	ì	¥	¡	╜	═	▌	Ø	²
14	E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╛	╬	▐	ε	■
15	F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	╧	▀	∩	

FIGURE E-1. Code page 437 (United States)

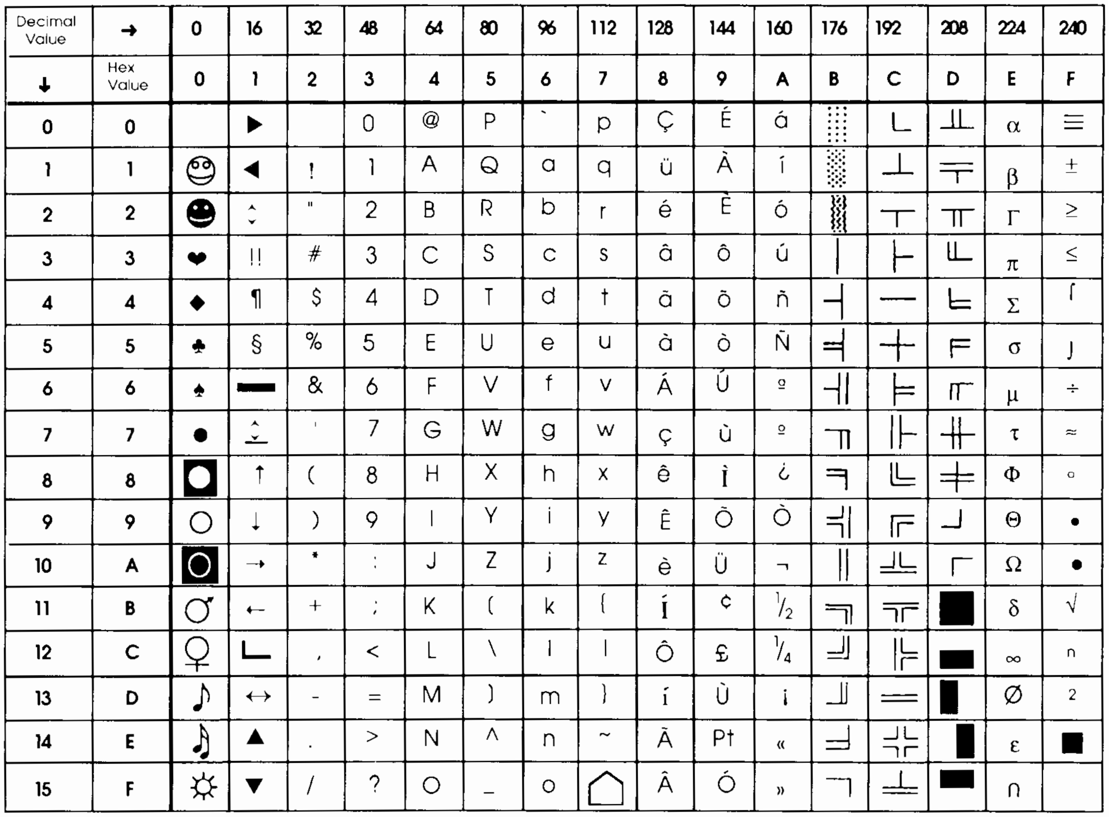
Figure A-1. Code Page 437 (United States)



Decimal Value	→	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
↓	Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D		
0	0		►		0	@	P	`	p	Ç	É	á	░	└	ð	Ó	-
1	1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	Đ	ß	±
2	2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	Ê	Ô	‗
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	Ë	Ò	¾
4	4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	È	õ	¶
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	Á	┼	ı	Õ	§
6	6	♠	▬	&	6	F	V	f	v	å	û	ª	Â	ã	Í	µ	÷
7	7	•	↨	'	7	G	W	g	w	ç	ù	º	À	Ã	Î	þ	¸
8	8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	©	╚	Ï	Þ	°
9	9	○	↓	)	9	I	Y	i	y	ë	Ö	®	╣	╔	┘	Ú	¨
10	A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Û	•
11	B	♂	←	+	;	K	(	k	{	ï	ø	½	╗	╦	█	Ù	¹
12	C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	ý	³
13	D	♪	↔	-	=	M	)	m	}	ì	Ø	¡	¢	═	¦	Ý	²
14	E	♫	▲	.	>	N	^	n	~	Ä	×	«	¥	╬	Ì	¯	■
15	F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	»	┐	¤	▀	´	

FIGURE E-2. Code page 850 (Multilingual)

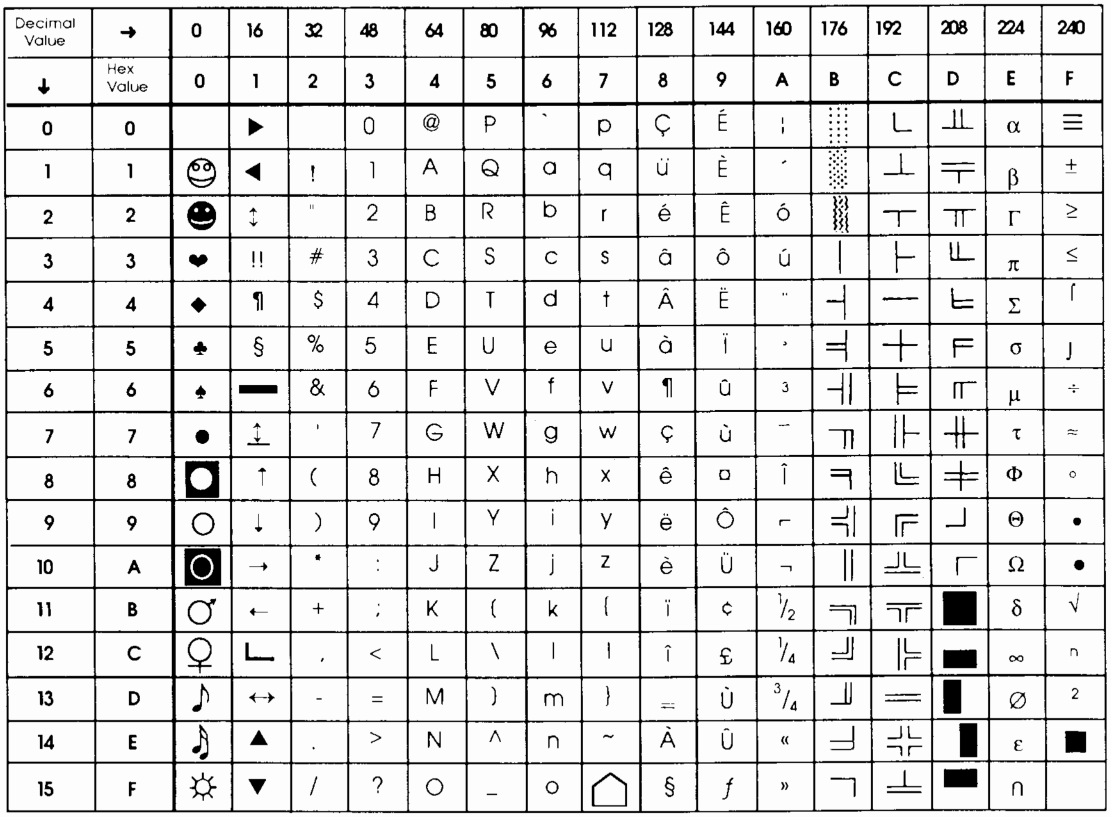
Figure A-2. Code Page 850 (Multilingual)



Decimal Value	→	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
↓	Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0		►		0	@	P	`	p	Ç	É	á	░	└	╨	α	≡
1	1	☺	◄	!	1	A	Q	a	q	ü	À	í	▒	┴	╤	β	±
2	2	☻	↕	"	2	B	R	b	r	é	È	ó	▓	┬	╥	Γ	≥
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	4	♦	¶	$	4	D	T	d	t	ã	õ	ñ	┤	─	╘	Σ	⌠
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	♠	▬	&	6	F	V	f	v	Á	Ú	ª	╢	╞	╓	µ	÷
7	7	●	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	8	◘	↑	(	8	H	X	h	x	ê	Ì	¿	╕	╚	╪	Φ	°
9	9	○	↓	)	9	I	Y	i	y	Ê	Õ	Ò	╣	╔	┘	Θ	•
10	A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	•
11	B	♂	←	+	;	K	[	k	{	Í	¢	½	╗	╦	█	δ	√
12	C	♀	∟	,	<	L	\	l	|	Ô	£	¼	╝	╠	▄	∞	n
13	D	♪	↔	-	=	M	]	m	}	í	Ù	¡	╜	═	▌	Ø	2
14	E	♫	▲	.	>	N	^	n	~	Ã	Pt	«	╛	╬	▐	ε	■
15	F	☼	▼	/	?	O	_	o	⌂	Â	Ó	»	┐	╧	▀	∩	

FIGURE E-3. Code page 860 (Portugal)

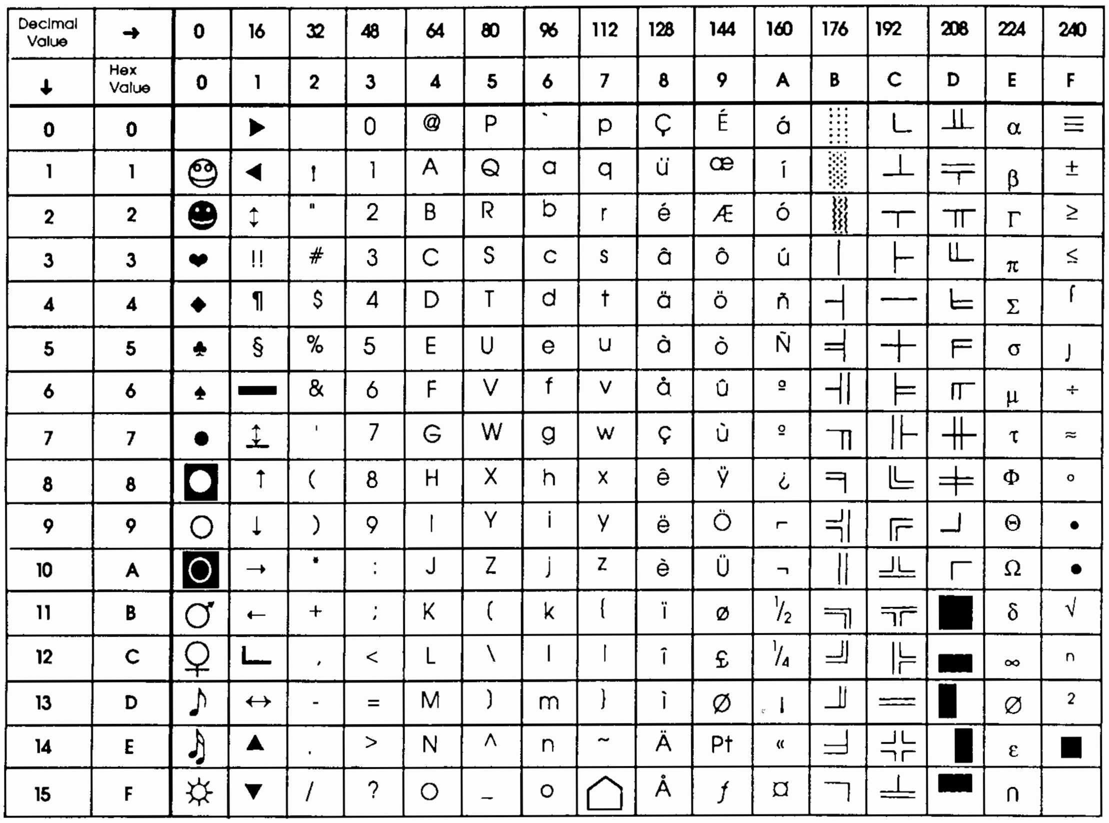
Figure A-3. Code Page 860 (Portugal)



Decimal Value	→	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
↓	Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0		►		0	@	P	`	p	Ç	É	¦	░	└	╨	α	≡
1	1	☺	◄	!	1	A	Q	a	q	ü	È	´	▒	┴	╤	β	±
2	2	☻	↕	"	2	B	R	b	r	é	Ê	ó	▓	┬	╥	Γ	≥
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	4	♦	¶	$	4	D	T	d	t	Â	Ë	¨	┤	─	╘	Σ	⌠
5	5	♣	§	%	5	E	U	e	u	à	Ï	¸	╡	┼	╒	σ	⌡
6	6	♠	▬	&	6	F	V	f	v	¶	û	³	╢	╞	╓	μ	÷
7	7	•	↨	'	7	G	W	g	w	ç	ù	¯	╖	╟	╫	τ	≈
8	8	◘	↑	(	8	H	X	h	x	ê	¤	Î	╕	╚	╪	Φ	°
9	9	○	↓	)	9	I	Y	i	y	ë	Ô	⌐	╣	╔	┘	Θ	•
10	A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	•
11	B	♂	←	+	;	K	[	k	{	ï	¢	½	╗	╦	█	δ	√
12	C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
13	D	♪	↔	-	=	M	]	m	}	‗	Ù	¾	╜	═	▌	∅	²
14	E	♫	▲	.	>	N	^	n	~	À	Û	«	╛	╬	▐	ε	■
15	F	☼	▼	/	?	O	_	o	⌂	§	ƒ	»	┐	╧	▀	∩	

FIGURE E-4. Code page 863 (Canada-French)

Figure A-4. Code Page 863 (Canada-French)



Decimal Value	→	0	16	32	48	64	80	96	112	128	144	160	176	192	208	224	240
↓	Hex Value	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0		►		0	@	P	`	p	Ç	É	á	░	└	╨	α	≡
1	1	☺	◄	!	1	A	Q	a	q	ü	æ	í	▒	┴	╤	β	±
2	2	☻	↕	"	2	B	R	b	r	é	Æ	ó	▓	┬	╥	Γ	≥
3	3	♥	‼	#	3	C	S	c	s	â	ô	ú	│	├	╙	π	≤
4	4	♦	¶	$	4	D	T	d	t	ä	ö	ñ	┤	─	╘	Σ	⌠
5	5	♣	§	%	5	E	U	e	u	à	ò	Ñ	╡	┼	╒	σ	⌡
6	6	♠	▬	&	6	F	V	f	v	å	û	ª	╢	╞	╓	µ	÷
7	7	•	↨	'	7	G	W	g	w	ç	ù	º	╖	╟	╫	τ	≈
8	8	◘	↑	(	8	H	X	h	x	ê	ÿ	¿	╕	╚	╪	Φ	°
9	9	○	↓	)	9	I	Y	i	y	ë	Ö	⌐	╣	╔	┘	Θ	•
10	A	◙	→	*	:	J	Z	j	z	è	Ü	¬	║	╩	┌	Ω	•
11	B	♂	←	+	;	K	[	k	{	ï	ø	½	╗	╦	█	δ	√
12	C	♀	∟	,	<	L	\	l	|	î	£	¼	╝	╠	▄	∞	ⁿ
13	D	♪	↔	-	=	M	]	m	}	ì	Ø	¡	╜	═	▌	∅	²
14	E	♫	▲	.	>	N	^	n	~	Ä	Pt	«	╛	╬	▐	ε	■
15	F	☼	▼	/	?	O	_	o	⌂	Å	ƒ	¤	┐	╧	▀	∩	

FIGURE E-5. Code page 865 (Norway)

Figure A-5. Code Page 865 (Norway)

Dead Keys

Belgium (Vectra PC Keyboard same as France)

Belgium (Vectra Enhanced Keyboard)
437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù ñÑ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúÁÉÍÓÚ àèìòùÀÈÌÒÙ ãõñÃÕÑ

Canadian French (Vectra PC Keyboard and Enhanced Keyboard)
863: ëïüËÏÜ âêîôûÂÊÎÔÛ éóúÉ àèùÀÈÙ çÇ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ çÇ

Denmark and Norway (Vectra PC Keyboard)
865: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ

Denmark and Norway (Vectra Enhanced Keyboard)
865: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù ñÑ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ ãõñÃÕÑ

Finland and Sweden (Vectra PC Keyboard)
437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ

Finland and Sweden (Vectra Enhanced Keyboard)
437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù ñÑ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ ñÑ

France (Vectra PC Keyboard and Enhanced Keyboard)
437: äëïöüÿÄÖÜ âêîôû
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ

Germany (Vectra PC Keyboard and Enhanced Keyboard)
437: áéíóúÉ àèìòù
850: áéíóúÁÉÍÓÚ àèìòùÀÈÌÒÙ

Italy
There are no dead keys.

Latin America (Vectra PC Keyboard same as Spain)

Latin America (Vectra Enhanced Keyboard)

437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúÁÉÍÓÚ àèìòùÀÈÌÒÙ

Netherlands (Vectra PC Keyboard same as U.S.)

Netherlands (Vectra Enhanced Keyboard)

437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù ñÑ çÇ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ ãõñÃÕÑ çÇ

Portugal (Vectra PC Keyboard same as U.S.)

Portugal (Vectra Enhanced Keyboard)

860: üÜâêôÂÊÔ áéíóú àèìòù ãõñÃÕÑ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ ãõñÃÕÑ

Spain (Vectra PC Keyboard and Enhanced Keyboard)

437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúÁÉÍÓÚ àèìòùÀÈÌÒÙ

Swiss French and Swiss German (Vectra PC Keyboard and Enhanced Keyboard)

437: äëïöüÿÄÖÜ âêîôû áéíóúÉ àèìòù ñÑ
850: äëïöüÿÄËÏÖÜ âêîôûÂÊÎÔÛ áéíóúýÁÉÍÓÚÝ àèìòùÀÈÌÒÙ ãõñÃÕÑ

United Kingdom There are no dead keys.

United States There are no dead keys.

B

The HP Vectra Disk Cache Program

Description and Use of HP Vectra Disk Cache

HP Vectra Disk Cache is a disk caching program designed for computers with flexible disks or a hard disk that may also have expanded or extended memory. To use HP Vectra Disk Cache, enter one of the forms of the program's syntax at the MS-DOS prompt, or in a batch file.

Syntax

In the command syntax below, commands and parameters in CAPS should be typed as shown, parameters in blue are optional, and *italics* are used to show variables.

To install a cache with the optimized defaults (recommended):

```
HPDCACHE
```

To install a customized cache (advanced users):

```
HPDCACHE /A+ /B± /C+ /D± /E+ /E:xxxx /H± /O± /P± /Q± /R:xxx /S:xxx
/T± /T:xx /V± /W± /-drive /?
```

After your cache has been installed:

```
HPDCACHE /D /E /F /M /P /U /?
```

Installation Parameters	**Description**
/A+	puts the cache in expanded memory (/A+ or /A). The cache will use as much memory as is available. If you have expanded memory, /A+ is the default.

`/B+` or `/B-`
copies (/B+ or /B), or does not copy (/B-), data to and from the cache in batches of sectors. The default for base and expanded memory is /B+. For extended memory, which turns interrupts off, the default is /B+, which transfers data one sector at a time.

`/C+`
puts the cache in base memory (/C+ or /C). This is memory directly addressable by MS-DOS from 64 KB up to a maximum of 640 KB (the default cache size is 96 KB). The default is /C+ when there is no expanded or extended memory available.

`/D+` or `/D-`
increases (/D+ or /D), or does not increase (/D-), performance when writing to flexible disk drives. The default is /D+.

`/E+`
puts the cache in extended memory (/E+ or /E). The cache will use as much memory as is available. If you have extended memory (and no expanded memory), /E+ is the default.

`/E:`*xxxx*
allows HP Disk Cache to coexist with programs that use the same extended memory space in a way HP Disk Cache cannot detect. HP Disk Cache allocates from the top-most memory limit down. *xxxx* represents the bottom-most limit of the cache in kilobytes.

`/H+` or `/H-`
increases performance (/H+ or /H), or specifies regular performance (/H-), when writing to the hard disk. This parameter is compatible with all HP hard disk drives, but may be incompatible with some third-party hard disk subsystems. The default is /H+.

`/O+` or `/O-`
gives (/O+ or /O), or does not give (/O-), reads priority over writes during disk transfers. /O+ may increase performance when used with /D+ (for diskettes), /H+ (for hard disks), and /Q+ (when copying files from one directory or disk to another). The default is /O-.

`/P+` or `/P-`
displays (/P+ or /P), or does not display (/P-), all parameters in effect when HP Disk Cache is started. The default is /P-.

/Q+ or /Q-

specifies (/Q+ or /Q), or does not specify (/Q-) a faster return to the MS-DOS prompt during disk transfers. This parameter allows a fast return to MS-DOS prompt when using /D+ or /H+ parameters. **Caution!** Do not remove a diskette from a drive until the disk activity light is off.

/R:*xxx*

allocates *xxx* KB of memory (of the type already specified) for programs loaded after HP Disk Cache. Allocates the rest of available memory to the cache. Do not use this parameter with /S.

/S:*xxx*

allocates *xxx* KB of memory for the cache. The default cache size is 96 KB for base memory; for expanded or extended memory, the cache size is adjusted to take up all available expanded or extended memory. Do not use this parameter with /R:*xxx*.

/T+ or /T-

reads (/T+ or /T), or does not read (/T-), part or all of the track into cache memory when a disk read is requested. The amount read is the track buffer size. When /T+ is specified, the track buffer amount is one full track. This number could be 17 or 32 sectors, depending on your hard disk. The default is /T+. If your cache is in base memory, you may want to specify /T- to save memory.

/T:*xx*

allows you to specify the number of sectors in the track buffer size.

/V+ or /V-

uses (/V+ or /V), or does not use (/V-), a drive's hardware diskette change detector, instead of software, to detect when a flexible disk has been removed from a drive. This parameter can be used for high capacity 5.25-inch (1.2 MB) and 3.5-inch drives that have this capability. This increases performance. The default is /V+.

/W+ or /W-

checks (/W+), or does not check (/W-), data being written to the cache to see if it already has identical data. If there is no change to the data, nothing is written to the disk. The default is /W+.

B

/-*drive* does not provide caching function for the specified *drive* name. For example, if you have two hard disk drives and do not wish to cache drive D:, specify: `HPDCACHE /-D`.

/? displays all available installation parameters.

Post-Installation Parameters **Description**

/D disables cache and sets measurements to zero. This leaves HP Disk Cache in memory, but turns off all caching functions.

/E enables cache after previously disabled with /D.

/F flushes the cache and resets measurements to values in effect when program is started. This is useful when making tests of cache performance.

/M displays the measurements of the cache. This parameter displays the number of disk transfer requests made by the system and applications, the number of actual physical transfers from the disk, the number of disk transfers saved by the cache, and the percentage of overall transfer requests saved by the cache.

/P displays all parameters now in effect.

/U un-installs HP Disk Cache. MS-DOS allows only the last-installed memory resident program to be un-installed. To use this parameter, any programs loaded after HP Disk Cache must have been un-installed.

/? displays all available post-installation parameters.

Operation

HP Disk Cache automatically keeps copies of recently used disk data in memory. Your computer retrieves this information from the cache more quickly without having to read the disk every time. This improves performance. Many applications such as accounting packages, spelling checkers, and data base programs, run faster and more efficiently using disk caching. The improvement you get depends on the way your application accesses the disk, how much data is accessed, and how often.

HP Disk Cache can be installed in the following types of memory:

base memory directly addressable by MS-DOS—up to 640 KB. For best performance, specify the cache size as large as possible up to 512 KB.

expanded memory above the 640 KB limit using the Lotus^R/Intel^R/Microsoft^R (LIM) specification, or the Enhanced Expanded Memory Specification. For best performance, specify the cache size as large as possible up to 16,384 KB.

extended memory above 1 MB. It is not directly addressable by applications using MS-DOS. It can be used by HP Disk Cache, and by programs such as RAMDrive. For best performance, specify a cache size as large as possible up to 16,384 KB.

When starting HP Disk Cache without parameters (using the optimized defaults), it first looks to install the cache in expanded memory. If there is no expanded memory, it then looks for extended memory. If there is no extended memory, it creates the cache in base memory.

The parameters are pre-set to optimize disk cache performance on most computers for most applications and work loads. However, in some situations additional performance may be achieved by modifying the default parameters.

When using HP Disk Cache on DOS-based network workstations or servers, only local drives will be cached; network (or remote) drives will not be cached. For best results, use HP Disk Cache's optimized defaults, or specify a large cache size.

To have HP Disk Cache start automatically, put your command in your AUTOEXEC.BAT file. (If your AUTOEXEC.BAT file contains a `DOSSHELL` command line, any command you add must come before `DOSSHELL`—not after it.) If you wish to have a different set of parameters for each of your often used applications, you may wish to create different batch files to customize the cache to run with each application. The batch file would set up the appropriate size cache before loading the application itself.

Examples

To do this:	Type this:
Allow HP Disk Cache to install itself in the most optimal way for your computer	`HPDCACHE`
Put a line in AUTOEXEC.BAT to allow HP Disk Cache to automatically install. Where the HPDCACHE.COM file is located in the DOS directory of drive C:	`C:\DOS\HPDCACHE`

Set up HP Disk Cache in expanded memory and reserve 256 KB of expanded memory for applications that will be loaded later	`HPDCACHE /A+ /R:256`
	or
	`HPDCACHE /A /R:256`

Set up HP Disk Cache in extended memory, and set size of the cache to 384 KB	`HPDCACHE /E+ /S:384`
	or
	`HPDCACHE /E /S:384`

Set up a cache of 256 KB in base memory, and read sectors off the disk 8 at a time (instead of the default of 4)	`HPDCACHE /S:256 /T:8`
Set up a cache of 96 KB in base memory, but do not cache drive D:. Drive C: will still be cached	`HPDCACHE /-D`
Set the cache lower limit so as not to interfere with a 512 KB RAM disk in extended memory. (Hint: although your maximum addressable amount of base memory may be 640 KB, there is an additional 384 KB reserved for programs, so the formula you use to find the lowest limit for your cache would be to add your **total** base memory to your RAM disk size: 1024 + 512 = 1536)	`HPDCACHE /E:1536`
Set up optimized cache, but do not store disk writes frequently done by applications in the cache	`HPDCACHE /H-`

Notes

1. If one of your applications does not have enough memory to run after a cache is installed, decrease the amount of memory used by HP Disk Cache.
2. If you want to use HP Disk Cache with an external drive, you must place the line `DEVICE=HPDCACHE.SYS` directly after your external hard disk driver line in CONFIG.SYS. If HPDCACHE.SYS is not located in the root directory, be sure to specify the proper path.
3. If your computer does not have expanded or extended memory, HP Disk Cache may not improve performance with programs such as Microsoft^{R} Windows^{TM} 2.0 (or later), or Aldus PageMaker^{TM}.
4. 80386-based computers (such as HP Vectra RS and QS) include memory manager software which can allocate all or part of the extended memory as expanded memory. We recommend placing HP Disk Cache in expanded memory in this case. (Your HPEMM/386 version number must be A.01.02 or later to work with HP Disk Cache.)
5. If your computer has problems with HP Disk Cache installed, it is possible that the problem you have is not related to HP Disk Cache. Start your system without HP Disk Cache, or use the /U parameter to un-install the cache, and see if the problem persists. If the problem persists, check your CONFIG.SYS and AUTOEXEC.BAT files for drivers and memory resident programs which may be interfering with HP Disk Cache. Remove them, then add them back one at a time until you find the problem.
6. You should set BUFFERS in your CONFIG.SYS file to 3 or 4 for best performance with HP Disk Cache. However, you may need to experiment to determine the best minimum BUFFERS size to work with all of your applications.
7. You may use the FASTOPEN command in conjunction with HP Disk Cache; however, FASTOPEN will probably not provide any performance improvements over using HP Disk Cache alone.
8. If you experience trouble using high-speed communications (such as with a 2400 baud modem) or networking software, and you are using the extended memory support parameter /E+, try also specifying /B- in your HP Disk Cache command to remedy the problem.

C

MS-DOS Message Directory

Introduction

There are three types of messages that you could see on your screen:

- MS-DOS command and utility messages
- MS-DOS device error messages
- Application program messages

MS-DOS utility and device error messages are listed in this appendix. For instructions about error messages related to non-MS-DOS software, see your application's documentation.

Disk and Device Errors

If a disk or device error occurs at any time during a command or program, MS-DOS displays an error message and includes this prompt:

```
Abort, Ignore, Retry, Fail?_
```

MS-DOS waits for you to type one of the following responses:

A Abort. End the program requesting the disk read or write.

I Ignore. Ignore the bad sector and pretend the error did not occur. This may result in lost data.

R Retry. Repeat the operation. You should use this response when you have corrected the error (for example, with Not ready or Write protect errors).

F Fail. This causes the current MS-DOS system operation to end (fail) and the application to continue.

Note

For some flexible disk device errors, the Ignore option will not be displayed.

Usually, you will want to recover by first typing `R` (to try again). If the second attempt fails, type `A` to terminate the process.

MS-DOS Messages

This section lists MS-DOS messages in alphabetical order, describes their causes, and provides information on how to correct them. It identifies the list of sources in brackets ([ ]) for each message.

`10 Mismatches - ending compare`

Explanation: COMP assumes the files are too different after 10 mismatches and stops. [COMP]

`64K High Memory Area is available.`

Explanation: HIMEM has properly installed itself in the system (no errors). [HIMEM.SYS]

`/A+ parameter ignored -- conflicts with /E+.`

Explanation: You may not select both expanded (/A) and extended (/E) memory, re-enter the HPDCACHE command with the correct parameter(s). [HP DISK CACHE]

`Abort edit (Y/N)?`

Explanation: MS-DOS displays this message when you choose the Edlin Q (Quit/Don't Save) command. The Q command exits the editing session without saving any editing changes. Type `Y` (for Yes) or `N` (for No). [EDLIN]

`Access denied`

Explanation: You tried to replace a write-protected, read-only, or locked file. [ATTRIB, FIND, PRINT, REPLACE, XCOPY]

`Active Code Page:` *xxx*

Explanation: *xxx* is the code page currently being used by the system. [CHCP]

`Active Code Page for device` *ddd* `is` *xxx*

Explanation: *xxx* is the code page currently being used by the device *ddd.* [MODE]

`Active Code Page not available from con device`

Explanation: The code page that the system is currently using is not supported on the console (screen) you are using. [KEYB]

`Advanced support request ignored for Drive` *x.*

Explanation: You have requested advanced support (/H), but your hard disk controller is not industry standard compatible. Basic support will be used. [HP DISK CACHE]

`All available space in the Extended DOS Partition is assigned to logical drives`

Explanation: There is no more room for logical drives in the extended partition. [FDISK]

`All files canceled by operator`

Explanation: MS-DOS displays this message when you specify the /T parameter with the PRINT command. [PRINT]

```
All files in.directory will be deleted!
Are you sure (Y/N)?
```

Explanation: MS-DOS displays this message if you try to delete all files in the working directory using the *.* wildcard. Type `Y` (for Yes) to delete all files, or **`N`** (for No). [MS-DOS]

`All logical drives deleted in the Extended DOS Partition`

Explanation: Any logical drives previously associated with the extended DOS partition on your disk are now removed. [FDISK]

`Allocation error, size adjusted`

Explanation: The size of the file indicated in the directory was not consistent with the amount of data actually allocated to the file. The file was truncated to match the amount of data allocated. [CHKDSK]

`All specified file(s) are contiguous`

Explanation: All are written sequentially on the disk and there isn't any fragmentation. [CHKDSK]

```
An error occurred while installing DOS
Press Enter to continue, or Esc to exit SELECT
```

Explanation: This is an unexpected error and you should respond appropriately to the prompt. [SELECT]

`ANSI.SYS must be installed to perform requested function`

Explanation: The screen function requested requires ANSI.SYS be loaded. [MODE]

`APPEND already installed`

Explanation: You have already used the APPEND command once since you turned on your computer. Now you are trying to use either the /X or /E parameter with this command. These parameters are only valid the first time you type the APPEND command. If you want to change the APPEND parameter, reboot your computer. Then type the APPEND command with the parameter you want to use. [APPEND]

`APPEND/ASSIGN Conflict`

Explanation: You cannot use the APPEND command on an assigned drive. Cancel the drive assignment before using the APPEND command with this drive again. [APPEND]

`Attempting to recover allocation unit` *x*

Explanation: Format found a bad allocation unit while trying to format the disk. [FORMAT]

`/B invalid with a black and white printer`

Explanation: Since you cannot print a background color on a black and white printer, you cannot use the /B parameter here. [GRAPHICS]

`*** Backing up files to drive x: *** Diskette Number:` *n*

Explanation: MS-DOS displays this message while backing up files to the specified drive. Be sure to label backup disks with the appropriate backup disk number for use in restoring them later. [BACKUP]

`Bad command or file name`

Explanation: The command cannot find the program you asked it to run. Check to see that you typed the command properly, and that the file or command is on the disk or in the command path. [MS-DOS]

`Bad or missing Command Interpreter`

Explanation: MS-DOS cannot find the COMMAND.COM file on the disk; either the file is missing from the root directory, or the file is invalid. You also receive this message if command.com has been moved from the directory it was originally in when you started MS-DOS. Either restart the system with a disk that contains the COMMAND.COM file, or copy the COMMAND.COM file from your backup MS-DOS master disk onto the disk used to start MS-DOS. [MS-DOS]

`Bad or missing filename`

Explanation: You specified a device incorrectly in the CONFIG.SYS file. Check the accuracy of the device command in the CONFIG.SYS file. [MS-DOS]

`Bad or Missing Keyboard definition file`

Explanation: MS-DOS cannot find the KEYBxx file that you specified with the KEYB command. Check to see that the file you specified exists on the disk. Also check to see that your path includes the directory in which this file resides. Then, retype the command. If you get this message again, the keyboard.sys or KEYB.COM file may be corrupted. [KEYB]

`Bad Partition Table`

Explanation: This message means that there is no DOS partition on the hard disk. You must run FDISK to create a DOS partition on your hard disk. [FORMAT]

`Bad unit error reading drive` *x:*

Explanation: Invalid subunit numbers were passed to the device driver. [MS-DOS DEVICE ERROR]

`Batch file missing`

Explanation: A batch file name was referenced that could not be found. [COMMAND]

`Baud rate required`

Explanation: MODE COM*x* commands must have at least a baud rate specified for MODE to initialize the COM port. [MODE]

`Cache size requested too small.`

Explanation: You have requested a cache size less than the minimum supported by HP Disk Cache. [HP DISK CACHE]

```
Cannot CHDIR to path -
tree past this point not processed
```

Explanation: CHKDSK is checking the structure of the directory and is unable to go to the specified directory. All subdirectories underneath this directory will not be verified. To correct this error automatically, specify the CHKDSK /f parameter. [CHKDSK]

`Cannot CHDIR to root`

Explanation: CHKDSK is checking the tree structure of the directory and is unable to return to the root directory. CHKDSK is not able to continue checking the remaining subdirectories. Try to restart MS-DOS. If this error persists, the disk is unusable. [CHKDSK]

`Cannot Chkdsk a Network drive`

Explanation: You cannot check drives that are redirected over the network. [CHKDSK]

`Cannot Chkdsk a SUBSTed or ASSIGNed drive`

Explanation: You cannot check drives that have been substituted or assigned. [CHKDSK]

`Cannot create a zero size partition`

Explanation: You are trying to create a partition with a size of 0 cylinders. You must allocate a minimum of 1 cylinder to any partition you create. [FDISK]

`Cannot create extended DOS partition without primary DOS partition on disk 1`

Explanation: You are trying to create an extended DOS partition, but your first hard disk does not contain a primary DOS partition. First, create the primary DOS partition on your first hard disk. Then, if you have more room on that disk, or if you have a second hard disk, you can create an extended DOS partition. [FDISK]

`Cannot create Logical DOS drive without an Extended DOS Partition on the current drive`

Explanation: You must create an extended DOS partition before any logical drives may be created. [FDISK]

`Cannot delete Extended DOS Partition while logical drives exist`

Explanation: Your disk has one or more logical drives assigned to it. These must be removed before you can delete an extended DOS partition. Delete all logical drives by using FDISK. Then delete the extended DOS partition. [FDISK]

`Cannot DISKCOMP to or from an ASSIGNed or SUBSTed drive`

Explanation: One of the drives that you specified is a drive that you created using the ASSIGN or SUBST command. [DISKCOMP]

`Cannot DISKCOMP to or from a network drive`

Explanation: You cannot compare disks on drives that have been redirected over the network. [DISKCOMP]

`Cannot DISKCOPY to or from an ASSIGNed or SUBSTed drive`

Explanation: One of the specified drives was created with the ASSIGN or SUBST command. [DISKCOPY]

`Cannot DISKCOPY to or from a network drive`

Explanation: You cannot copy disks to or from drives that have been redirected over the network. [DISKCOPY]

`Cannot do binary reads from a device`

Explanation: The copy cannot be done in binary mode when you are copying from a device. You should either not use the /B parameter, or you should use the /A parameter to specify an ASCII copy. [COPY]

`Cannot edit .BAK file--rename file`

Explanation: You attempted to edit a file that had a file name extension of .BAK (a backup copy created by Edlin). If you must edit a file that has an extension of .BAK, you must either rename or copy the file and give it a different extension. [EDLIN]

`Cannot find FORMAT.COM`

Explanation: You tried to back up to an unformatted disk and MS-DOS couldn't find the FORMAT command. [BACKUP]

`Cannot find GRAPHICS profile`

Explanation: You must give the path to the GRAPHICS.PRO file or MS-DOS will look in the current directory. [GRAPHICS]

`Cannot find System Files`

Explanation: You tried to use a drive that did not have the system files in the root directory. [FORMAT]

`Cannot format an ASSIGNed or SUBSTed drive`

Explanation: You attempted to format a drive currently mapped to another drive by the ASSIGN or SUBST command. Run ASSIGN or SUBST again and clear all drive assignments. [FORMAT]

`Cannot FORMAT a Network drive`

Explanation: You cannot format drives that are redirected over the network. [FORMAT]

`Cannot FORMAT nonremovable drive` *x:*

Explanation: You are trying to back up files with the /f parameter. MS-DOS will not allow you to format the target disk specified. Be sure you want to back up files to a hard disk. If you do, you must use a hard disk that is formatted already. [BACKUP]

`Cannot JOIN a Network drive`

Explanation: You cannot join drives that are redirected over the network. [JOIN]

`Cannot LABEL a Network drive`

Explanation: You cannot label a drive that is shared on a network server station. [LABEL]

`Cannot LABEL a SUBSTed or ASSIGNed drive`

Explanation: You cannot label a drive if it has been substituted with the SUBST command or assigned with the ASSIGN command. Check the command to be sure you specified a valid file name. [LABEL]

`Cannot load COMMAND, system halted`

Explanation: MS-DOS cannot reload the command processor. Reboot MS-DOS on your system. [MS-DOS]

`Cannot perform a cyclic copy`

Explanation: When you are using the /s parameter, you may not specify a target that is a subdirectory of the source. [XCOPY]

`Cannot read file allocation table`

Explanation: The file allocation on your disk has been damaged. [RECOVER]

`Cannot recover . entry, processing continued`

Explanation: The . entry (working directory) is defective and cannot be recovered. [CHKDSK]

`Cannot recover .. entry,`
`Entry has a bad attribute (or link or size)`

Explanation: The .. entry (parent directory) is defective and cannot be recovered. If you have specified the /F parameter, CHKDSK tries to correct the error automatically. [CHKDSK]

`Cannot RECOVER a Network drive`

Explanation: You cannot recover files on drives that are redirected over the network. [RECOVER]

`Cannot setup expanded memory`

Explanation: This message indicates that the expanded memory (EMS) card in your system is not functioning properly. [FASTOPEN]

`Cannot specify default drive`

Explanation: If you don't leave enough space on the disk for SYS to install the system files and two unused entries in the root directory of the drive to which you want to transfer the operating system files, then SYS won't be able to specify the disk drive you want. [SYS]

`Cannot start COMMAND, exiting`

Explanation: Either the value of Files in config.sys needs to be increased before restarting MS-DOS, the path to find the command processor was set incorrectly, or there was insufficient memory available to load the command processor. [MS-DOS]

`Cannot SUBST a Network drive`

Explanation: You cannot substitute drives that are redirected over the network. [SUBST]

`Cannot SYS to a Network drive`

Explanation: You cannot transfer the system files to drives that are redirected over the network. For more information about the net print command, see the *Microsoft Networks User's Guide.* [SYS]

`Cannot use FASTOPEN for drive` *x:*

Explanation: FASTOPEN works only with local, hard disks and can work with a maximum of four disks at a time. You may be trying to use

FASTOPEN over a network, with a flexible disk, or with more than four disks at one time, none of which is possible with FASTOPEN. [FASTOPEN]

`Cannot use PRINT - Use NET PRINT`

Explanation: You must use the Net Print command to print files. [PRINT]

`Cannot XCOPY to a reserved device`

Explanation: You must specify as your target a file or block device (not a character device) that is not NUL and, then, enter the XCOPY command again. [XCOPY]

`CHDIR .. failed, trying alternate method`

Explanation: When checking the tree structure, CHKDSK was not able to return to a parent directory. It will try to return to that directory by starting over at the root and searching again. [CHKDSK]

`CHKDSK not available on drive` *x*

Explanation: This is an informational message indicating that you are trying to CHKDSK an alternate file system which cannot be found. [CHKDSK]

xxxxxxxx `code page drive cannot be initialized`

Explanation: MS-DOS cannot start either the PRINTER.SYS or DISPLAY.SYS program. Check the device command in your CONFIG.SYS file. You probably included an illegal parameter. For more information, see the "System Configuration" chapter. [MS-DOS]

`Code page not prepared`

Explanation: You have selected a code page that has not yet been prepared for the system, or one that does not have the correct font to support the current video mode. To prepare a code page for the system, use the MODE PREPARE command. If you have installed the DISPLAY.SYS installable device driver, be sure the device command in your CONFIG.SYS file allows for additional subfonts. For more information, see the "System Configuration" chapter. [MODE]

`Code page xxx not prepared for all devices`

Explanation: You have selected a code page that is not currently supported by a device. First, be sure your device supports code-page switching, and that it is currently on-line. If the device supports code-page switching, use the MODE PREPARE command to prepare the device for the code page. Then retry the CHCP command. [CHCP]

`Code page` *`xxx`* `not prepared for system`

Explanation: CHCP is unable to select a code page for the system. First, make sure that NLSFUNC is installed. If you have not used the device command in your CONFIG.SYS file to install device drivers, you may now retry the CHCP command. If you are using installable device drivers with your system, you must use the MODE PREPARE command to prepare the specific code page for each device on your system. Then retype the CHCP command. [CHCP]

`Code page operation not supported on this device`

Explanation: You have specified a device and code page combination which MS-DOS does not recognize as valid. Check to see that the device you specified exists and that you have listed a valid code page. Also check to see that code page is supported on that device. [MODE]

`Code page requested` *`xxx`* `is not valid for given keyboard code`

Explanation: The keyboard code and code page specified are not compatible. Retype the KEYB command with a compatible keyboard code and code page. [KEYB]

`Code page specified has not been designated`

Explanation: You have typed the KEYB command with an option the system doesn't recognize. You must first prepare the associated code page for your console screen device. Use the MODE PREPARE command to prepare the associated code page for CON. Then retype the KEYB command. [KEYB]

`Code page specified has not been prepared`

Explanation: You have typed the KEYB command with an option the system doesn't recognize. You must first prepare the associated code page for your console screen device. Use the MODE PREPARE command to prepare the associated code page for CON. Then retype the KEYB command. [KEYB]

```
Code page specified is inconsistent with invoked code page
```

Explanation: This warning message lets you know that the KEYB option you've selected does not coincide with the code page for your console screen device (CON). Use the MODE SELECT command if you also want to change the code page for CON. [KEYB]

```
Code page specified is inconsistent with selected code page
```

Explanation: This warning message lets you know that the KEYB option you've selected does not coincide with the code page for your console screen device (CON). Use the MODE SELECT command if you also want to change the code page for CON. [KEYB]

```
Code pages cannot be prepared
```

Explanation: You have either specified a duplicate code page for this device or tried to prepare more than the total number of code pages supported for this device. Check the device command in your CONFIG.SYS file to see how many prepared code pages are allowed for this device. Use the /STATUS parameter of the MODE command to find out which code pages are already prepared for this device. For more information, see the "System Configuration" and "MS-DOS Command Descriptions" chapters. [MODE]

```
Compare error at OFFSET x
```

Explanation: This is an informational message indicating that the files COMP is comparing contain different hexadecimal values at the displayed offset (also in hexadecimal) into the file. [COMP]

```
Compare error on disk
side x, track x
```

Explanation: DISKCOMP found a difference on the disk in the specified drive, side *x*, track *x*. [DISKCOMP]

```
Compare OK
```

Explanation: DISKCOMP displays this message if the disks are identical. [DISKCOMP]

```
Compare process ended
```

Explanation: DISKCOMP displays this message if a fatal error occurred during the comparison. [DISKCOMP]

`COM port does not exist`

Explanation: You have specified an invalid COM port. [MODE]

x `Contains` *x* `non-contiguous blocks`

Explanation: The disk contains fragmented files. If you want to copy this disk, you should use the COPY or XCOPY command instead of the DISKCOPY command. The new COPY will then store the new files sequentially. [CHKDSK]

`Content of destination lost before copy`

Explanation: The source file that you specified in the COPY command was overwritten before the copy process completed. Refer to the COPY command for the proper syntax. [COPY]

`Convert directory to file (Y/N)?`

Explanation: The directory is no longer usable because it contains too much invalid information. Type `Y` (for Yes) if you want to convert it to a file and then use DEBUG, or type `N` (for No) if you don't. [CHKDSK]

`Convert lost chains to files (Y/N)?`

Explanation: CHKDSK displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type `Y` (for Yes) in response to this prompt, CHKDSK recovers the lost blocks it found when checking the disk. CHKDSK then creates a proper directory entry and a file for each lost chain with a file name of the form: *filennnn*.CHK. If you type `N` (for No), CHKDSK frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks. [CHKDSK]

`Copy process ended`

Explanation: Diskcopy could not copy the entire disk. Use the COPY or XCOPY command to copy specific files onto the disk. [DISKCOPY]

`Corrections will not be written to disk`

Explanation: There are errors on the disk, but CHKDSK will not correct them because you did not specify the /F parameter. You must specify the CHKDSK /F parameter to correct disk errors. [CHKDSK]

Current drive is no longer valid

Explanation: Your current drive is either a network drive or a disk drive with no disk in it. You must either change to a valid drive or, if using a disk drive, make sure you have a disk in it and that the drive door is closed. [MS-DOS]

Current keyboard code: *xx* code page: *yyy* Current CON code page: *zzz*

Explanation: This message displays the current keyboard code and its associated code page, and the current code page used by your console screen device (CON). [KEYB]

Current keyboard does not support this code page

Explanation: The code page selected is not compatible with the current keyboard code. Check the code page you have selected. If it is correct, change the keyboard code with the KEYB command. [KEYB]

Data error reading drive *x:*

Explanation: MS-DOS could not read the data from the disk properly. This is often due to a defective disk. Try typing R (for Retry) several times, or type A (for Abort) to end the program. (It's a good idea to make a new copy of the disk, because if it's defective, you may lose information.) [MS-DOS DEVICE ERROR]

Device Error during Status

Explanation: MS-DOS found an error with the specified device when it was checking the status of that device. The problem may be due to a device that does not support code pages, a device not properly prepared for code-page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code-page switching. Consult the hardware vendor if you are unsure. [MODE]

Device Error during Prepare

Explanation: MS-DOS found an error with the specified device when preparing that device for code-page switching. The problem may be due to a device that does not support code pages, a device not properly prepared for code-page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code-page switching. Consult the hardware vendor if you are unsure. [MODE]

`Device Error during Select`

Explanation: MS-DOS found an error with the specified device. The problem may be due to a device that does not support code pages, a device not properly prepared for code-page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code-page switching. Consult the hardware vendor if you are unsure. [MODE]

`Device Error during write of font file to device`

Explanation: MS-DOS found an error when it tried to write the font file to the specified device. The problem may be due to a device that does not support code pages, a device not properly prepared for code-page switching, a device which cannot support more code pages than those already prepared, or a device with a bad or irregular font file. Check the device command in your CONFIG.SYS file. Make sure that the command syntax and limits for subfonts and additional code pages are all correct. Also check to see if your device supports code-page switching. Consult the hardware vendor if you are unsure. [MODE]

`Device or code page missing from font file`

Explanation: MS-DOS did not find a definition of the indicated code page for this device in the font file. Use the mode command to specify another code page for this device. Also check to see that the font file supports the code page you want to use. This error also may cause specified code pages to be undefined. Use the mode command to prepare and refresh lost code pages. [MODE]

`Device nnn not prepared`

Explanation: No code page has been prepared for this device. [MODE]

`Directory is joined`

Explanation: CHKDSK does not process directories that are joined. Use the JOIN /D command to unjoin the directories, and then run CHKDSK again. [CHKDSK]

`Directory is totally empty, no . or ..`

Explanation: The specified directory does not contain references to working and parent directories. Delete the specified directory and recreate it. [CHKDSK]

`Directory not empty`

Explanation: You can only join onto an empty directory. [JOIN]

`Disk error reading (or writing) drive x:`

Explanation: MS-DOS could not read the data from the disk properly. This is often due to a defective disk. Try typing `R` (for Retry) several times, or type `A` (for Abort) to end the program. (It's a good idea to make a new copy of the disk, because if it's defective, you may lose information.) [MS-DOS DEVICE ERROR]

`Disk error reading (or writing) FAT x`

Explanation: One of your File Allocation Tables has a defective sector in it. MS-DOS automatically uses the other FAT. You should copy all your files onto another disk. To correct this error automatically, you simply specify the CHKDSK /F parameter. [CHKDSK]

`Disk full. Edits lost`

Explanation: Edlin was not able to save your file due to lack of disk space. You should always make sure that there is enough room on the default disk to save your file before you use the Edlin E (End/Save) command. You should also make sure that the default disk is not write-protected. [EDLIN]

`Disk full error writing to BACKUP Log File`

Explanation: The disk to which you were writing the log file is full. To restart, press any key or Ctrl Break. [BACKUP]

```
Disk unsuitable for system disk
```

Explanation: The Format program detected a bad track on the disk where system files should reside. You should use this disk to store data only. [FORMAT]

```
Do not specify filename(s) Command format: DISKCOMP d: d:[/1] [/8]
```

Explanation: You specified an incorrect parameter or gave a file name in addition to a drive name. [DISKCOMP]

```
Do not specify filename(s) Command format: DISKCOPY d: d:[/1]
```

Explanation: You specified an incorrect parameter or gave a file name in addition to a drive name. [DISKCOPY]

```
Do you see the leftmost 0? (Y/N)
```

Explanation: Mode displays this message to help you align the test pattern on your screen. Type `Y` (for Yes) if you can see the leftmost 0 in the test pattern, or type `N` (for No) if you want to shift the display to the right. [MODE]

```
Do you see the rightmost 9? (Y/N)
```

Explanation: Mode displays this message to help you align the test pattern on your screen. Type `Y` (for Yes) if you can see the rightmost 9 in the test pattern, or type `N` (for No) if you want to shift the display to the left. [MODE]

```
Do you wish to use the maximum
size for a DOS partition and make
the DOS partition active (Y/N)........ []
```

Explanation: You are formatting your hard disk. Type `Y` (for Yes) and press the Enter key if you want to format your entire hard disk as the primary DOS partition. Otherwise, type `N` (for No) and press the Enter key. [FDISK]

```
Does name specify a file name
or directory name on the target (F = file D = directory)?
```

Explanation: XCOPY displays this prompt if the target directory does not exist. TypeF if the name specifies a file, or `D` if the target specifies a directory that does not currently exist. [XCOPY]

`DOS command line parameters supported`

Explanation: This is an informational message. GRAFTABL will display command parameters. [GRAFTABL]

`Drive deleted`

Explanation: You deleted a hard drive from the system. [FDISK]

`Drive has been changed or deleted`

Explanation: You changed or deleted a hard drive on the system. [FDISK]

`Drive letter must be specified`

Explanation: You did not specify the drive letter for the drive that you want to format. You must specify the name of the drive to format. [FORMAT]

```
Drive types or diskette types
not compatible
```

Explanation: You must have the same size and type of disks to run these commands. For example, you cannot copy from a single-sided disk to a double-sided disk, or compare a high-density disk with a low-density disk. You should use FC if you want to compare the files on the disks. If you want to copy the disk, you can use copy or XCOPY. You could also reformat the target disk so that it's the same type as the source disk, or use a disk of the same type. [DISKCOMP, DISKCOPY]

`Duplicate file name or file not found`

Explanation: You tried to rename a file to a file name that already exists, or the name you specified could not be found. [RENAME]

`Duplicate parameters not allowed`

Explanation: You cannot specify a parameter twice. [GRAPHICS]

`/E+ parameter ignored -- not 80286 system.`

Explanation: Use of extended memory is not supported on the HP Vectra CS. [HP DISK CACHE]

`End of input file`

Explanation: The entire file was read into memory. If the file was read in sections, this message indicates that the last section of the file is in memory. [EDLIN]

`Enter current Volume Label for drive` *x:*

Explanation: MS-DOS asks you to enter the current volume label for verification before it formats the hard disk in the specified drive. If you do not know what the volume label is, press Ctrl C to abort this command, and give the VOL command for the specified drive. Then give the FORMAT command again. [FORMAT]

`Entry error`

Explanation: The last command you typed contained a syntax error. Retype the command with the correct syntax and press the Enter key. [EDLIN]

`Entry has a bad attribute (or link or size)`

Explanation: This message may be preceded by one or two periods that show which subdirectory is invalid. If you have specified the /F parameter, CHKDSK tries to correct the error automatically. [CHKDSK]

`EOF mark not found`

Explanation: While comparing non-text files, COMP could not find the end of the valid data in the last block of files being compared. [COMP]

```
ERROR: An Extended Memory Manager is already installed.
XMS Driver not installed.
```

Explanation: HIMEM can only be installed once. Installing it more than one time will result in an error for the second and subsequent installations. [HIMEM.SYS]

`Error during read of Font file`

Explanation: MS-DOS found an error when it tried to read the font file for the code page specified. [MODE]

```
ERROR: HIMEM.SYS requires an 80x86-based machine.
XMS Driver not installed.
```

Explanation: HIMEM can only be installed on a computer system which has an 80286 or 80386 microprocessor. [HIMEM.SYS]

```
ERROR: HIMEM.SYS requires DOS 3.00 or higher.
XMS Driver not installed.
```

Explanation: HIMEM can only be used on systems with MS-DOS Version 3.00 or higher. [HIMEM.SYS]

```
ERROR: No available extended memory was found.
```

Explanation: HIMEM can only be installed on a computer with extended memory. [HIMEM.SYS]

```
ERROR: Unrecognized A20 hardware.
```

Explanation: HIMEM cannot recognize the A20 hardware of your system. If this occurs, it is probably because the system is not one supported by this release of HIMEM. Contact your sales representative to see if an XMS driver exists for your machine. [HIMEM.SYS]

```
Error in country command
```

Explanation: You used the incorrect syntax for the COUNTRY command in your CONFIG.SYS file. For the correct syntax of this configuration command, see the “System Configuration” chapter. [MS-DOS]

```
Error in .EXE file
```

Explanation: The .exe file you have asked MS-DOS to load has an invalid internal format. You cannot run this program. Check to make sure that you are using the correct version of MS-DOS. [MS-DOS]

```
Error loading operating system
```

Explanation: You were unable to load your operating system from the hard disk due to a disk error. If you restart the system and the error continues, restart it again using your MS-DOS disk and then the SYS command to transfer a new copy of MS-DOS to your hard disk. [FDISK]

```
Error opening log file
```

Explanation: MS-DOS cannot open the backup log file. Check the drive and path specified with this command. Remember that the log file cannot be located on the target drive. If you did not specify a file name for the log, then the error occurred when MS-DOS tried to open and create BACKUP.LOG on the source disk. [BACKUP]

```
Error reading directory
```

Explanation: You could not read the directory due to bad sectors within the directory or FAT structure. If the error occurred on the hard disk, try to format it again to make it usable; otherwise, throw it out. [FORMAT]

```
Error reading fixed disk
```

Explanation: After five tries, FDISK still cannot read the startup record of the current hard disk. Consult your system's manuals or call your vendor. [FDISK]

```
Error reading GRAPHICS profile
```

Explanation: MS-DOS cannot read the Graphics profile. Make sure the disk drive door is closed. [GRAPHICS]

```
Error reading/writing partition table
```

Explanation: Format could not read or write the partition table. You should run fdisk on the disk and then try formatting it again. [FORMAT]

```
Errors found, F parameter not specified
Corrections will not be written to disk
```

Explanation: CHKDSK found errors on the disk. If you have not specified the /F parameter, CHKDSK continues printing messages but will not correct the errors. You should run CHKDSK with the /F parameter if you want to correct the problems encountered by the CHKDSK command. [CHKDSK]

```
Errors on list device indicate that it
may be off-line. Please check it.
```

Explanation: Your printer is not turned on. [PRINT]

`Error writing directory (FAT or fixed disk)`

Explanation: Either the File Allocation Table (FAT) or the directory has bad sectors. If the error occurred on your hard disk, try to format it again to make it usable; otherwise, throw it out. [FORMAT, FDISK]

`Error writing to device`

Explanation: You tried to send too much data to a device, so MS-DOS was unable to write the data to that device. [MS-DOS]

`EXEC failure`

Explanation: MS-DOS either found an error when reading a command, or the FILES command in the CONFIG.SYS file is set too low. Increase the value of the FILES command in the CONFIG.SYS file, and restart MS-DOS. [MS-DOS]

`Expanded Memory Failure; Function =` *x*`; Error Code =` *y*.

Explanation: The Expanded Memory Manager returned error indicated. Restart your computer. A message from your Expanded Memory Manager may provided you with more explanation. [HP DISK CACHE]

`Expanded Memory not available`

Explanation: This is an informational message indicating that you do not have an expanded memory card installed in your system. [MS-DOS]

`Extended DOS partition already exists`

Explanation: You cannot create another extended DOS partition. [FDISK]

`/F Flush request ignored.`

Explanation: You have requested that the cache be flushed at the time the disk cache program is started. This parameter is used to flush the program **after** you have started the program. [HP DISK CACHE]

`Failure to access code page font file`

Explanation: MS-DOS cannot open the font file for the specified code page. Check to see that you typed font file name and its path name correctly. Also check the CONFIG.SYS file to see that the device driver for this device has been properly installed. If the CONFIG.SYS file is incorrect, correct it and restart MS-DOS before retyping the MODE command. [MODE]

`Failure to access device:` *xxx*

Explanation: You are trying to specify a code page for a particular device, but MS-DOS cannot access the device listed. Retype the command using an existing device. Make sure you are typing the device name correctly. [MODE]

`Fastopen already installed`

Explanation: FASTOPEN is already installed on the system. [FASTOPEN]

`FCB unavailable reading (or writing) drive` *x:*

Explanation: An unusual error has occurred. This error usually requires an experienced programmer to fix it. Type `R` (for Retry) or `A` (for Abort). [MS-DOS DEVICE ERROR]

`fc: cannot open filename - No such file or directory`

Explanation: One of the files that you specified doesn't exist. Check the directory for the correct file name. [FC]

`fc:` *file* `longer than` *file*

Explanation: After reaching the end of one of the files in a file comparison, the other file still has data left that was not yet compared. [FC]

`fc: incompatible switches`

Explanation: You have specified parameters that are not compatible. (For example, /B and /L.) You should not combine binary and ASCII comparison parameters. [FC]

`fc: no differences encountered`

Explanation: The files are the same. [FC]

`fc: out of memory`

Explanation: You do not have enough memory to perform the comparison. [FC]

`File allocation table bad`

Explanation: The disk may be defective. Run CHKDSK /F to check the disk. [MS-DOS]

`File allocation table bad drive x:`

Explanation: This message means that the disk was not formatted or was formatted improperly. It could also mean that an operating system other than MS-DOS is on the disk. Run CHKDSK /F to check the disk. If this message is displayed again, you must reformat the disk. [CHKDSK]

`File filename canceled by operator`

Explanation: MS-DOS displays this message when you specify the /T parameter with the PRINT command. [PRINT]

`File cannot be converted`

Explanation: The input file is not in the correct format. [EXE2BIN]

`File cannot be copied onto itself`

Explanation: The source file name you specified is the same as the target file name. [COPY, REPLACE, XCOPY]

`File creation error`

Explanation: You tried to add a new file name or replace a file that already exists in the directory, or there was not enough space for the file. If the file already exists, it is a read-only file and cannot be replaced. This error message may also occur if the root directory is full, out of files, or if the file name is the same as a volume or directory, or a hidden (or system) file. [MS-DOS, EDLIN, RESTORE, XCOPY]

`File is READ-ONLY`

Explanation: The file is designated read-only, so you may not change it. [EDLIN]

`File name must be specified`

Explanation: You did not specify a filename when you started Edlin. You should type the Edlin command followed by a file name. [EDLIN]

`File not found`

Explanation: MS-DOS could not find the file that you specified, or you tried to rename a file with a name already in the directory. Check to see that you entered the file name correctly. [CHKDSK, EDLIN, FC, FIND, PRINT, RECOVER, RENAME, XCOPY]

`File not in PRINT queue`

Explanation: The file that you specified was not in the print queue, so you cannot remove it from the queue. Check to see that you entered the file name correctly. [PRINT]

`Files are different sizes`

Explanation: This is an informational message indicating why COMP did not compare the files. [COMP]

`FIND: Access denied`

Explanation: You cannot access the file. Make sure that the

disk is not write-protected, read-only, or locked. [FIND]

`FIND: File not found`

Explanation: MS-DOS could not find the file that you specified. Make sure you have typed the file name correctly. [FIND]

`FIND: Invalid number of parameters`

Explanation: You specified either too many or too few parameters in the command. [FIND]

`FIND: Invalid Parameter`

Explanation: One of the parameters you specified is wrong. [FIND]

`FIND: Read error in filename`

Explanation: The FIND command could not read the specified file. [FIND]

`FIND: Syntax error`

Explanation: Check to make sure that you have typed the command correctly. [FIND]

`First cluster number is invalid, entry truncated`

Explanation: The file directory entry contains an invalid pointer to the data area. If you specified the /F parameter, the file is truncated to a zero-length file. [CHKDSK]

`FIRST diskette bad or incompatible`

Explanation: DISKCOMP cannot recognize the format on the source disk. You should run CHKDSK to help you identify the problem. [DISKCOMP]

`Fixed backup device` *x* `is full`

Explanation: This is an informational message indicating that you can no longer use that device for backing up files. [BACKUP]

`Fixups needed - base segment hex:`

Explanation: The source (.EXE) file contained information indicating that a load segment is required for the file. You must specify the absolute segment address where the finished module is to be located. [EXE2BIN]

`Font File contents invalid`

Explanation: MS-DOS cannot use the contents of the font file specified. Make sure you are typing the name of the font file correctly. Retype the command. If this message is displayed again, your font file may have been altered or corrupted. Recopy this file from the master MS-DOS disk. Type the command again. This error may also cause existing selected code pages to be undefined. Use the mode command to prepare these code pages again, and to refresh them. [MODE]

`For cannot be nested`

Explanation: You cannot nest For commands in a batch file. [MS-DOS]

`Format failure`

Explanation: MS-DOS could not format the disk. This message is usually displayed with an explanation as to why the command failed. [FORMAT]

`Format not supported on drive` *x:*

Explanation: You cannot use Format to format this drive. You may have specified device parameters that your computer cannot support. [FORMAT]

`Function not supported on this computer`

Explanation: Your machine cannot do the requested function due to a lack of the appropriate adapter or device. [MODE]

`General failure reading` (or `writing`) `drive` *x:*

Explanation: An unusual error has occurred. This error usually requires an experienced programmer to fix it. Type `R` (for Retry) or `A` (for Abort). [MS-DOS DEVICE ERROR]

`Graphics characters already loaded`

Explanation: The GRAFTABL command displays this message if you have already loaded the table of graphics characters into memory. [GRAFTABL]

`Has invalid cluster, file truncated`

Explanation: The file directory entry contains an invalid pointer to the data area. If you specified the /F parameter, the file is truncated to a zero-length file. [CHKDSK]

```
HP Disk Cache Program already installed.
You must un-install before reinstalling with different parameters.
```

Explanation: You have tried to start the HP Disk Cache program after it has already installed a disk cache. [HP DISK CACHE]

`Illegal device name`

Explanation: Your computer does not recognize this device name. [MODE]

`Incompatible system size`

Explanation: The system files occupy more space on the source disk than is available on the target disk. You cannot use the SYS command to transfer the system files to this disk. [SYS]

`Incorrect APPEND Version`

Explanation: You are not using the MS-DOS 4.0 APPEND command. You are using another incompatible version. [APPEND]

`Incorrect DOS Version`

Explanation: Some MS-DOS utilities will not run on older versions of the operating system, and many are written to run only on the exact version of MS-DOS that they were created for. You must use the correct version of MS-DOS to run this command. [MS-DOS]

`Incorrect DOS Version, use DOS 2.00 or later`

Explanation: Some MS-DOS utilities will run only on MS-DOS version 2.00 or later versions. [LINK]

`Incorrect number of parameters`

Explanation: You specified either too many or too few parameters in the command. [JOIN, SUBST]

`Incorrect parameter`

Explanation: One of the parameters you specified is wrong. [ASSIGN, SHARE]

`Infinite retry on parallel printer timeout`

Explanation: Your printer is probably off-line or not ready. If the printer appears to be ready, you may have to press the (Ctrl) (Alt) (Del) keys to reset the computer. [MODE]

`Infinite retry not supported on network printer`

Explanation: Your system cannot sense printer errors through the network interface. You should either use a printer with a direct connection or not specify retry. [MODE]

`Insert backup diskette n into drive` *x:*

Explanation: This message prompts you for the *n*th backup disk. Put the next disk into the specified drive. Be sure to label each backup disk in the appropriate order for use when restoring the files. [BACKUP, RESTORE]

`Insert backup source diskette in drive` *x*

Explanation: This message prompts you for the backup source disk. [BACKUP]

`Insert disk with command.com in drive` *x*

Explanation: This message prompts you for the disk with the COMMAND.COM file on it. [COMMAND]

C

```
Insert diskette with batch file
and press any key when ready
```

Explanation: The disk containing your batch file is not in the drive you originally specified. Reinsert the disk that contains the batch file in the appropriate drive. [MS-DOS]

```
Insert DOS diskette in drive x:
and strike ENTER when ready
```

Explanation: You typed the FORMAT /S command, but the disk in the default drive does not contain MS-DOS system files. Insert a disk with the files IBMBIO.COM and IBMDOS.COM in the drive specified and press any key. [FORMAT]

```
Insert FIRST diskette into drive x:
```

Explanation: This message prompts you for the first disk that you want to compare. [DISKCOMP]

```
Insert new diskette for drive x:
```

Explanation: This message prompts you for the disk you want to format. [FORMAT]

```
Insert restore target diskette into drive x:
```

Explanation: MS-DOS displays this prompt if you are restoring files to a flexible. Put the target disk into the specified drive. [RESTORE]

```
Insert SECOND diskette in drive x:
```

Explanation: This message prompts you for the disk that you want to compare with the first disk. [DISKCOMP]

```
Insert source disk
```

Explanation: This message prompts you to put the source disk (disk containing your source files, of files to be backed up) into the drive. [BACKUP]

```
Insert system diskette in drive x:
and strike any key when ready
```

Explanation: SYS needs a disk from which to read the IBMBIO.COM and IBMDOS.COM files. Insert a system disk into the specified drive and press any character or number key to start the system copy process. [SYS]

`Insert TARGET diskette into drive x:`

Explanation: DISKCOPY displays this message to prompt you to place the target disk (disk you are copying files to) into the specified drive. If your computer has one flexible drive, your source and target disks are the same drive. [DISKCOPY]

`Insufficient disk space`

Explanation: The disk is full and does not contain enough room to perform the specified operation. [MS-DOS, REPLACE, SORT, XCOPY]

`Insufficient memory`

Explanation: There is not enough memory in your computer to perform the specified operation. Before retrying this operation, you must free memory by deleting files. In Edlin, you may be able to free memory by typing a W (Write) command followed by an A (Append) command. [BACKUP, CHKDSK, DISKCOMP, DISKCOPY, EDLIN, REPLACE, RESTORE, SORT, XCOPY]

```
Insufficient room in root directory.
Erase files in root and repeat CHKDSK
```

Explanation: CHKDSK always recovers lost files into the root directory. In this case, your root directory is full. Delete some files in your root directory, or move them to another directory to make room for the lost files. [CHKDSK]

`Intermediate file error during pipe`

Explanation: The pipe operation uses temporary files on the disk that are deleted automatically once the piping process is complete. An error has occurred in one of these files. Make sure that there is enough room on the disk for the temporary file and that the disk is not write-protected, and then try the command again. [MS-DOS]

`Internal error`

Explanation: This message indicates an error in the utility. [FC, MODE, SHARE]

```
Internal stack overflow
System Halted
```

Explanation: The system tried to use more stacks than were available. This caused a series of hardware interrupts and halted the system. Restart MS-DOS. Then edit your CONFIG.SYS file and allocate more stack resources. For more information, see the "System Configuration" chapter. [MS-DOS]

```
Invalid argument
```

Explanation: You have specified an invalid argument. Refer to the chapter "MS-DOS Command Descriptions" for the correct syntax of the command, and try again. [BACKUP, FC, RESTORE]

```
Invalid baud rate specified
```

Explanation: You have specified an incorrect baud rate. Valid choices are 110, 150, 300, 600, 1200, 2400, 4800, 9600 and 19200. You must specify at least the first two digits of the baud rate. [MODE]

```
Invalid characters in volume label
```

Explanation: The volume label should only contain up to 11 alphanumeric characters. [FORMAT, LABEL]

```
Invalid code page specified
```

Explanation: You selected an invalid code page number. Retype the command with the correct code page. [CHCP]

```
Invalid combination of parameters
```

Explanation: You must enter the command parameters in the correct order. [MS-DOS]

```
Invalid COMMAND.COM
Insert COMMAND.COM disk in default drive
and strike any key when ready
```

Explanation: The program you have just run used up almost all of memory. MS-DOS must now reload the COMMAND.COM file from disk. However, either MS-DOS cannot find COMMAND.COM on the disk, or the copy it has found is the incorrect version. Insert a disk that contains a copy of command.com into the default drive (it must be the same version with which you started MS-DOS). [MS-DOS]

`Invalid country code or code page`

Explanation: MS-DOS found an invalid country code or code page number in your CONFIG.SYS file. Correct the country command in your CONFIG.SYS file. [MS-DOS]

`Invalid current directory`

Explanation: Your disk has an invalid directory on it. You may be able to recover some of the files on this disk by copying them with the COPY command. Otherwise, you must replace the disk. [CHKDSK]

`Invalid date`

Explanation: You specified an invalid date in response to the date prompt. Enter a valid date in the correct format. Refer to the "MS-DOS Command Descriptions" chapter for the proper syntax and format of the DATE command. [DATE, XCOPY]

`Invalid Date/Time`

Explanation: You specified an invalid date with one of the BACKUP command parameters. Refer to the "MS-DOS Command Descriptions" chapter for the proper syntax of the BACKUP command, then try again. [BACKUP]

`Invalid device`

Explanation: The device specified was not AUX, CON, NUL, or PRN. [MS-DOS]

`Invalid device parameters from device driver`

Explanation: MS-DOS displays this message when the number of hidden sectors is not evenly divisible by the number of sectors per track (that is, the partition does not start on a track boundary). This might happen if you tried to format a hard disk that previously had been formatted with MS-DOS 2.x without first running FDISK, or if you have set the device driver parameters incorrectly. Check the CONFIG.SYS file for incorrect device or DRIVPARM commands. [FORMAT]

`Invalid directory`

Explanation: The directory you specified either does not exist or is invalid. Check to see that you entered the directory name correctly. [MS-DOS]

`Invalid disk change reading (or writing) drive` *x:*

Explanation: You changed the disk in a drive when you weren't supposed to. Put the disk back in the drive and type `R` (for Retry). [MS-DOS DEVICE ERROR]

`Invalid drive in search path`

Explanation: The drive does not exist. [MS-DOS]

`Invalid drive or filename`

Explanation: You did not type a valid drive name or file name. Enter a valid drive name or file name. [EDLIN, RECOVER]

`Invalid drive specification`

Explanation: The drive is incorrect or does not exist. Enter a valid drive name. [BACKUP, CHKDSK, DISKCOMP, DISKCOPY, FORMAT, LABEL, PRINT, REPLACE, RESTORE, SYS, TREE, XCOPY]

`Invalid entry, please enter` *x*

Explanation: You must enter a value that is within the acceptable range. [FDISK]

`Invalid entry, please press Enter`

Explanation: The last character you can enter is the percent sign. If you want to enter additional characters, press (Enter) or (Backspace) to change your entry and then press (Enter) again. [FDISK]

`Invalid environment size specified`

Explanation: You gave an invalid number of bytes with the /E parameter. You must specify a number between 160 and 32,768 (bytes). [COMMAND]

`Invalid extent entry`

Explanation: You must enter a value for *m* that is within the appropriate range of from 1 through 999. [FASTOPEN]

`Invalid file/directory entry`

Explanation: You must enter a value of *n* that is within the appropriate range of from 10 through 999. If you do not provide a value, the default is 34. [FASTOPEN]

`Invalid keyboard code specified`

Explanation: You selected an invalid keyboard code with the KEYB command. Retype the command with the correct keyboard code. [KEYB]

`Invalid number of parameters`

Explanation: Either you did not specify an option or string, or you specified the wrong number of parameters in the command. [ATTRIB, BACKUP, FC, FIND, RECOVER, RESTORE, XCOPY]

`Invalid parameter (or parameter combination)`

Explanation: One of the parameters you specified is wrong or does not exist. Refer to the "MS-DOS Command Descriptions" chapter to make sure you are using the correct parameters. [CHKDSK, FASTOPEN, APPEND, REPLACE]

`Invalid parameter(s)`

Explanation: One of the parameters you specified is wrong or does not exist. Refer to the "MS-DOS Command Descriptions" chapter to make sure you are using the correct parameters. [BACKUP, CHKDSK, DISKCOMP, DISKCOPY, EDLIN, FIND, FORMAT, JOIN, MODE, PRINT, REPLACE, RESTORE, SORT, SUBST, SYS, TREE, XCOPY]

`Invalid partition table`

Explanation: MS-DOS found invalid information in the hard disk's partition information. Use FDISK to examine and correct it. [FDISK]

`Invalid path`

Explanation: The path is not correct. Make sure the path is less than 63 characters and that the filename characters and format are valid. [APPEND, MS-DOS]

`Invalid path, not directory, or directory not empty`

Explanation: You are unable to remove the directory requested for one of the specified reasons. [MS-DOS]

`Invalid path (or file not found)`

Explanation: You have entered a path or file name that does not exist. Enter a valid path or file name with the command. [ATTRIB, BACKUP, COPY, RESTORE, TREE, XCOPY]

`Invalid path or parameter`

Explanation: You specified a file or directory that does not exist. Enter a valid path or file name with the APPEND command. [APPEND]

`Invalid profile statement on line` *x*

Explanation: You used an invalid word in the profile statement on line *x*. Enter a valid word on the specified line. [GRAPHICS]

`Invalid STACK parameter`

Explanation: The syntax of the STACK command in your CONFIG.SYS file includes an invalid parameter. See the "System Configuration" chapter for the correct syntax of that configuration command. [MS-DOS]

`Invalid sub-directory entry`

Explanation: The subdirectory that you specified either does not exist or is invalid. Check whether you typed the subdirectory name correctly. [CHKDSK]

`Invalid switch type`

Explanation: You specified a parameter that is either not valid, in the wrong order, or duplicated. Check the parameters specified and correct them. [FASTOPEN]

`Invalid syntax`

Explanation: You used the wrong syntax when typing a command. See the "MS-DOS Command Descriptions" chapter for the correct syntax of an MS-DOS command. [MS-DOS]

`Invalid syntax on DISPLAY.SYS code page driver`

Explanation: You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load DISPLAY.SYS. See the "System Configuration" chapter for the correct syntax of DEVICE, and for information about DISPLAY.SYS. [MS-DOS]

Invalid syntax on PRINTER.SYS code page driver

Explanation: You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load PRINTER.SYS. See the "System Configuration" chapter for the correct syntax of DEVICE, and for information about PRINTER.SYS. [MS-DOS]

Invalid syntax on PRINTER.SYS code page switching device drivers

Explanation: You used the wrong syntax when you typed the DEVICE command in your CONFIG.SYS file to load PRINTER.SYS. See the "System Configuration" chapter for the correct syntax of DEVICE, and for information about PRINTER.SYS. [MODE]

Invalid time

Explanation: You specified an invalid time. Refer to the "MS-DOS Command Descriptions" chapter for the correct syntax, and try the command again. [TIME]

Invalid Volume ID

Explanation: MS-DOS displays this message if you enter a volume label that doesn't match the label on the hard disk you want to format. It then quits the format process. Use the VOL command to find out what the volume label for the hard disk is, then try the command again. [FORMAT]

Is cross linked on allocation unit *x*

Explanation: MS-DOS found two files using the same allocation unit. You can use the /F parameter, which may truncate the file. [CHKDSK]

x is not a choice, Please enter *y*-*z*

Explanation: You tried to select an invalid option *x*. Select a valid parameter from the range shown (*y*-*z*). [FDISK]

KEYB has not been installed

Explanation: No alternate keyboard code has been installed for your system. If you want to use a keyboard code other than the default U.S. (QWERTY) keyboard, use the KEYB command to install it. [MS-DOS]

```
Label not found
```

Explanation: Your batch file contains a GOTO command to a nonexistent label. [MS-DOS]

```
Last backup diskette not inserted
Insert last backup diskette in drive x:
Strike any key when ready
```

Explanation: This message prompts you for the final backup disk. [BACKUP]

```
*** Last file not backed up ***
```

Explanation: MS-DOS could not back up the last file on the disk. This message may occur if there is no more room on the target disk. It may also occur if there was an error in the source file or on the target disk. You may have to back up this file separately to another disk. [BACKUP]

```
Line too long
```

Explanation: During an Edlin R (Replace) command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. You should divide the long line into two lines and retry the R command. [EDLIN]

```
List output is not assigned to a device
```

Explanation: When you first type the PRINT command, MS-DOS asks you what device you want to specify as a printer. This message appears if Print is set up for a device that does not exist. [PRINT]

```
Lock violation reading (or writing) drive x:
```

Explanation: A program tried to access part of a file that another program was using. Type `A` (for Abort), or wait awhile and type `R` (for Retry). [MS-DOS DEVICE ERROR]

```
x lost allocation units found in !!y!! chains Convert lost chains to
files (Y/N)?
```

Explanation: MS-DOS displays this message if it finds information on the disk that isn't allocated properly in the disk's File Allocation Table. If you type `Y` (for Yes) in response to this prompt, CHKDSK recovers the lost blocks it found when checking the disk. CHKDSK then creates a proper directory entry and a file for each lost chain with the file name of the form:

filennnn.CHK. If you did not specify the /F parameter, MS-DOS displays: *x* bytes would be freed. If you type **N** (for No), CHKDSK frees the lost blocks so that they can be reallocated and does not recover any data that was in those lost blocks. If you did not specify the /F parameter, CHKDSK does nothing. [CHKDSK]

`Logging to file` *x*

Explanation: The BACKUP command is writing a backup log to the file specified. [BACKUP]

`Logical DOS drive created, drive letters changed or added`

Explanation: You have created or revised one or more logical drives. [FDISK]

`LPT`*x*`: not rerouted`

Explanation: MODE could not reroute the parallel printer port. Check to see whether you have specified the proper parameters. [MODE]

`LPT`*x*`: rerouted to COM`*x*`:`

Explanation: Output on the parallel printer port will now be sent to this serial communications port. [MODE]

`LPT`*x*`: set for 80`

Explanation: The parallel printer port has been set for 80 columns. [MODE]

`LPT`*x*`: set for 132`

Explanation: The parallel printer port has been set for 132 columns. [MODE]

`/M parameter ignored -- no measurements available.`

Explanation: You have requested measurements when the disk cache program is started. Measurements will be available once you have executed other commands after the disk cache is installed. [HP DISK CACHE]

`Maximum number of logical DOS drives installed`

Explanation: You have installed the maximum number of logical DOS drives allowed by MS-DOS. You may not create any more logical DOS drives. [FDISK]

```
Memory allocation error.
Cannot load MS-DOS, system halted
```

Explanation: Restart MS-DOS. If this error persists, make a new copy of the MS-DOS disk from your backup copy of the system disk. [MS-DOS]

```
--More--
```

Explanation: This means that there is more information to be displayed. To see the next screen of data, press the spacebar. [MORE]

```
MORE: Incorrect DOS version
```

Explanation: The MORE command does not run on MS-DOS versions before 2.0. [MORE]

```
Must enter both /T and /N parameters
```

Explanation: You must enter both parameters together. [FORMAT]

```
Must specify COM1, COM2, COM3 or COM4
```

Explanation: You must specify a serial port. [MODE]

```
Must specify destination line number
```

Explanation: You did not specify the destination line number for an Edlin C (Copy) or M (Move) command. Retype the command with a destination line number. [EDLIN]

```
Must specify ON or OFF
```

Explanation: The command requires either an ON or an OFF argument. [MS-DOS]

```
Name of list device [PRN]:
```

Explanation: This prompt appears the first time that PRINT is run and the /D parameter is not specified. You can specify the name of any valid device, which then becomes the print output device. If you press the (Enter) key, MS-DOS uses the default list device PRN. [PRINT]

```
New file
```

Explanation: Edlin prints this message if it does not find a file with the name you specified. If you are creating a new file, ignore this message. If you do not intend to create a new file, check to see whether you have correctly typed the name of the file that you wish to edit. [EDLIN]

`NLSFUNC already installed`

Explanation: NLSFUNC stays resident in memory once it is initialized. You have already loaded it into memory. [NSLFUNC]

`No Append`

Explanation: No path has been appended. If you would like to append a path for data files, use the APPEND command. [APPEND]

`No appended directories`

Explanation: You did not specify a path with the APPEND command. [APPEND]

`No code page has been selected`

Explanation: No code pages have been selected for the system. If you would like to select a code page, use the CHCP command. [CHCP]

`No drive specified`

Explanation: You must specify the physical drive number in the DEVICE command in your CONFIG.SYS file. For further information, see the "System Configuration" chapter. [DRIVER.SYS]

`No drives found to cache.`

Explanation: No hard disks were found to cache, therefore no disk cache was started. If you have an external hard disk, refer to the "HP Vectra Disk Cache Program" appendix before starting the disk cache program. [HP DISK CACHE]

`No Expanded Memory Manager found; Check CONFIG.SYS file.`

Explanation: The Expanded Memory Manager was not installed successfully. Refer to the documentation that came with your expanded memory board for information on installing the Expanded Memory hardware and software. [HP DISK CACHE]

`No Extended DOS Partition to delete`

Explanation: You chose the wrong option for the current hard disk. To review information on your hard disk, choose Display Partition Data from the FDISK Options menu. [FDISK]

No files added (or replaced)

Explanation: The REPLACE command did not add or replace any files. [REPLACE]

No files found filename

Explanation: Replace could not find matching source or target files. [REPLACE]

No fixed disks present

Explanation: You must determine the cause of the problem and act appropriately. [FDISK]

No free file handles.
Cannot start COMMAND.COM, exiting

Explanation: Restart MS-DOS. If this message recurs, increase the files command value in the CONFIG.SYS file. [MS-DOS]

No Logical DOS Drive(s) to delete

Explanation: This is an informational message indicating that there are no logical DOS drivers in this partition. Press the ESC key to return to the previous menu. [FDISK]

No logical drives defined

Explanation: There are no logical drives defined for your system. [FDISK]

No paper error writing device dev

Explanation: The printer is either out of paper or not turned on. [MS-DOS DEVICE ERROR]

No partitions to delete

Explanation: You chose the wrong option for this hard disk. Choose another option from the FDISK Options menu. [FDISK]

`No partitions to make active`

Explanation: You chose the wrong option for this hard disk. From the FDISK Options menu, choose Create DOS partition or logical DOS drive to create a partition, and then Set active partition to make it the active one. [FDISK]

`No path`

Explanation: You typed `Path` and pressed the Enter key to find out what your search path is, but you didn't set a command search path. [PATH]

`No primary DOS partition to delete`

Explanation: You have selected the FDISK option to delete your primary DOS partition, but that partition does not exist. [FDISK]

`No room for system on destination disk`

Explanation: There is not enough room for the system files on the target disk. Delete some files to make room for the system files or use another disk. You may need to reformat the disk to put the system on it. [SYS]

`No room in directory for file`

Explanation: You tried to create or save a file to the root directory, but it is either full, or you specified an invalid disk drive or file name. Check the command that you used to start Edlin for an invalid file name or disk drive entry. If your command contains no invalid entries, you should run the CHKDSK program for the specified disk drive. If the status report shows that the disk directory is full, and if there is still enough memory left on the disk, you may be able to create the file in a subdirectory. (This is because subdirectories are not limited in size as is the root directory.) Otherwise, remove the disk and replace it with another formatted disk. [EDLIN]

`No source drive specified`

Explanation: You must specify a source drive (drive that contains the files you wish to backup). [BACKUP]

`No space left on device`

Explanation: You cannot back up or restore any more files, and you cannot send any more output from a file comparison to your disk because the target disk is now full. You should probably delete some of the files on the disk to make more room. [BACKUP, FC, RESTORE]

`No space to create a DOS partition`

Explanation: You do not have enough space on your current hard disk to create a DOS partition. Either remove or reduce the size of the existing partition, and then run FDISK again to create the DOS partition(s). [FDISK]

`No space to create logical drive`

Explanation: You are trying to create a logical drive, but there is no space available to do so. [FDISK]

`No sub-directories exist`

Explanation: You have specified the /S parameter, but the directory does not contain subdirectories. [TREE]

`No such file or directory`

Explanation: One or more of the files or directories that you specified does not exist. [BACKUP, FC, RESTORE]

`No system on default drive`

Explanation: You must use a disk with the system files on it (the DOS disk) and try again. [SYS]

`No target drive specified`

Explanation: You must specify a target drive for this command. [BACKUP]

`Non-DOS disk error reading (or writing) drive` *x:*

Explanation: MS-DOS does not recognize the disk format because the disk is missing information or contains another operating system. Try running the CHKDSK command to correct the problem. (See the "MS-DOS Command Descriptions" chapter for information about CHKDSK.) If running CHKDSK does not solve the problem, you should reformat the disk by using the format command-even though this will destroy all the files on the disk. [MS-DOS DEVICE ERROR]

```
Non-system disk or disk error
Replace and strike any key when ready
```

Explanation: Replace the disk with the proper disk and press any alphanumeric key to continue. [FORMAT, SYS]

```
*** Not able to back up (or restore) file ***
```

Explanation: This message may occur if there was an error in the source file or on the target disk. Use the CHKDSK command on the source disk to see if you can determine the problem. [BACKUP]

```
Not enough memory
```

Explanation: There is not enough memory for MS-DOS to run the command. [JOIN, SHARE, SUBST]

```
Not enough memory for HP Disk Cache program.
```

Explanation: There is not enough free memory in your system to start the HP Disk Cache program. [HP DISK CACHE]

```
Not enough room to merge the entire file
```

Explanation: There was not enough room in memory to hold the file during an Edlin T (Transfer) command. You must free some memory by writing some files to a disk or by deleting some files before transferring this file. [EDLIN]

```
Not found
```

Explanation: You specified an Edlin S (Search) or R (Replace) command that was unable to find a further occurrence of the specified search or replace string. [EDLIN]

```
Not ready error reading (or writing) drive x:
```

Explanation: The device (usually a drive or printer) specified in the error message is not ready to accept or transmit data. This often happens when the disk drive door is open. If this is the problem, close the door and type R (for Retry), or check to see if the printer is on and ready to print. [MS-DOS DEVICE ERROR]

`O.K.?`

Explanation: This prompt occurs during Edlin S (Search) or R (Replace) command processing. If you press any key except `Y` (for Yes) or the Enter key, the search or replace process continues. [EDLIN]

`One or more CON code pages invalid for given keyboard code`

Explanation: KEYB examined all prepared code pages, and has found that at least one code page is incompatible for your screen console device (CON). This is only a warning to let you know that your keyboard and screen console device are working from different code pages. [KEYB]

`Only non-bootable partitions exist`

Explanation: None of the partitions left can boot (start) MS-DOS. [FDISK]

`Only partitions on drive 1 can be made active`

Explanation: You are trying to create an active partition on a hard disk other than that found on the first hard disk drive. This is not allowed. [FDISK]

`Out of environment space`

Explanation: There is not enough room in the program environment to accept more data. To increase the size of the existing environment, use the /E parameter with the COMMAND command or remove some of the existing environment variables by using the SET command. [COMMAND, MS-DOS]

`Parameters not compatible`

Explanation: You have specified parameters that cannot be used together. [FORMAT, REPLACE]

`Parameters not compatible with fixed disk`

Explanation: You have used a parameter that is not compatible with the specified drive. [FORMAT]

`Parameters not supported`

Explanation: You have specified parameters that MS-DOS does not support. [MS-DOS, FORMAT]

`Parameters not supported by Drive`

Explanation: MS-DOS displays this message when the device driver for this drive does not support Generic IOCtl function requests. [FORMAT]

`Partition selected (x) is not bootable,`
`active partition not changed`

Explanation: You are trying to change active partitions, but MS-DOS cannot be booted from the partition selected. [FDISK]

`Path(name) too long`

Explanation: The path you specified was too long. You may have to change directories to use this command with files in deep subdirectories. [PRINT, REPLACE, XCOPY]

`Path not found`

Explanation: You specified an invalid path. [CHKDSK, REPLACE, SUBST, XCOPY]

x `percent of disk formatted`

Explanation: This is an informational message indicating the percentage of the disk already formatted. [FORMAT]

`Press any key to begin adding (replacing) file(s)`

Explanation: When you specify the /W parameter, MS-DOS displays this message to prompt you to start replacing files. [REPLACE]

`Press any key to begin copying file(s)`

Explanation: This message provides a pause to allow you to change disks and then press a key before copying begins. [XCOPY]

`Press any key to begin formatting` *x:*

Explanation: This prompt is issued before you format a disk. Press any key to begin the format process. Or, if you wish to end this command, press Ctrl C at the same time. [FORMAT]

`Press any key to begin recovery of the` *n* `file(s) on drive` *x:*

Explanation: This prompt is issued before you recover a disk or file. Press any key to begin the recovery. Recovered files are named filennnn.rec. If you wish to end this command, press Ctrl C at the same time. [RECOVER]

`Press any key when ready ...`

Explanation: This prompt gives you time to insert the appropriate disks before copying them. When you have inserted the disks into the appropriate drives, press any key to begin the diskcopy process. Or, if you wish to end this command, press Ctrl C at the same time. [DISKCOMP, DISKCOPY]

`Previous Code Page:` *x*

Explanation: This status message can indicate either that GRAFTABL was or was not used already to load a code character table and whether or not it has been altered. [GRAFTABL]

`Previously prepared code page replaced`

Explanation: This command changed the selected code page for a specific device by using another prepared code page. [MODE]

`Primary DOS partition already exists`

Explanation: You are trying to create a primary DOS partition, but one already exists. If there is space available on your hard disk, try to create an extended DOS partition instead. [FDISK]

`Primary DOS partition created`

Explanation: You have successfully created a primary DOS partition on your disk. [FDISK]

`Primary DOS partition deleted`

Explanation: You have deleted the primary DOS partition from your disk. [FDISK]

`Printbox ID not in GRAPHICS profile`

Explanation: You must specify a printbox size (ID) that matches the first operand of a PRINTBOX command in the GRAPHICS profile. [GRAPHICS]

`Printer error`

Explanation: The printer is off or is not ready to print. [MODE]

`PRINT queue is empty`

Explanation: There are no files waiting to be printed. [PRINT]

`PRINT queue is full`

Explanation: There is only room for 10 files in the list of files waiting to be printed. You can make room for more by using the PRINT /Q parameter. The limit is 32 files. [PRINT]

`Printer type not in GRAPHICS profile`

Explanation: You can only specify printer types that are defined in the GRAPHICS profile. See the GRAPHICS command in the "MS-DOS Command Descriptions" chapter for a list of the supported printer types. [GRAPHICS]

`Probable non-DOS disk Continue (Y/N)?`

Explanation: The disk you are using is not recognized by this version of MS-DOS. The disk was either created by another system with a format that is not supported on this version of MS-DOS, or it is not an MS-DOS disk. Do not continue processing if CHKDSK returns this message for a flexible disk. If this message returns for a hard disk, the information describing the characteristics of the disk to MS-DOS has been destroyed. In this case, you may continue CHKDSK processing by typing `Y` (for Yes). This error may mean that the File Allocation Table (FAT) is bad and that the disk is unusable. [CHKDSK]

`Problem with drive` *x*.
`Correct and then press any key.`

Explanation: There may be bad sectors on your disk. If you suspect this to be true, you have the following options:

1. If you have a test utility that detects (and removes from use) bad sectors on the disk, run it.

   OR,

2. Backup all files on the hard disk, reformat the hard disk, then restore all files back on to the hard disk. Reformatting the hard disk should detect (and remove from use) bad sectors.

If you are sure that your hard disk has no bad sectors, specify the /H-parameter to instruct HP Disk Cache to use basic support for your hard disk. [HP DISK CACHE]

`Processing cannot continue`

Explanation: There is not enough memory in your machine to run CHKDSK for this disk. You must obtain more memory to run CHKDSK. [CHKDSK]

`Profile statement out of sequence on line` *x*

Explanation: This message indicates where you need to correct the sequence of the profile statement. [GRAPHICS]

`Program too big to fit in memory`

Explanation: You need more memory to run your application. It is possible that some programs you have run are still using some memory. You may try to restart MS-DOS; however, if you still receive this message, you still need more memory. [MS-DOS]

`RAMDrive: Computer must be PC-AT or PC-AT compatible`

Explanation: The /E parameter was specified on a computer other than an HP Vectra PC (or 100% IBM PC-AT compatible), or a 386 with extended memory. Ramdrive.sys was not installed. Remove the /E parameter. Try installing RAMDrive in the system memory. [RAMDRIVE.SYS]

`RAMDrive: Expanded Memory Manager not present`

Explanation: You included the /A parameter in the RAMDRIVE command. However, RAMDRIVE could not find the expanded memory manager because your CONFIG.SYS file does not contain the appropriate information. RAMDRIVE.SYS was not installed. Consult your expanded memory card documentation for the correct installation instructions. [RAMDRIVE.SYS]

`RAMDrive: Expanded Memory Status shows error`

Explanation: MS-DOS detected an error in your expanded memory adapter while trying to install RAMDRIVE. RAMDRIVE.SYS was not installed. Consult your expanded memory adapter documentation on memory errors. [RAMDRIVE.SYS]

`RAMDrive: Incorrect DOS version`

Explanation: RAMDRIVE requires DOS 2.x, DOS 3.x, or DOS 4.0. Windows 2.0 requires DOS 3.0 or higher. RAMDRIVE.SYS was not installed. Upgrade to DOS 3.0 or higher. [RAMDRIVE.SYS]

`RAMDrive: Insufficient memory`

Explanation: Your computer has some memory available, but not enough to set up a RAMDRIVE. RAMDRIVE.SYS was not installed. Free some extended memory or obtain more memory. [RAMDRIVE.SYS]

`RAMDrive: Invalid parameter`

Explanation: The parameters you specified in your CONFIG.SYS entry for RAMDRIVE.SYS are not correct. RAMDRIVE.SYS was not installed. Check to see if you specified too many parameters, if one of your numeric parameters is not valid, if you specified conflicting parameters (i.e., only one of either /E or /U may be specified), or if you specified too many parameters. Change the RAMDRIVE.SYS command in your CONFIG.SYS file to conform to the usage described. [RAMDRIVE.SYS]

`RAMDrive: I/O error accessing drive memory`

Explanation: During the set up of the RAMDRIVE, an error was detected in the memory being accessed for RAMDRIVE. RAMDRIVE.SYS was not installed. Run the memory test for the memory on which you were attempting to install a RAMDRIVE. [RAMDRIVE.SYS]

`RAMDrive: No extended memory available`

Explanation: Your computer has no memory available for RAM drives. RAMDRIVE.SYS was not installed. Free some extended memory or obtain more memory. [RAMDRIVE.SYS]

RATE and DELAY must be specified together
RATE=X
DELAY=X

Explanation: You must specify both the typematic rate and the delay value when you change the setting of either. [MODE]

Read error, COUNTRY.SYS

Explanation: MS-DOS cannot read the COUNTRY.SYS file. Retry the command. If you get the same message, the COUNTRY.SYS file is probably corrupted. Restore the file from backup. [MS-DOS]

Read error in filename

Explanation: MS-DOS could not read the entire file. [EDLIN, FIND]

Read error, KEYBOARD.SYS

Explanation: MS-DOS cannot read the keyboard.sys file. Retry the command. If you get the same message, the KEYBOARD.SYS file is probably corrupted. Restore the file from backup. [MS-DOS]

Read fault error reading drive *x:*

Explanation: MS-DOS is unable to read data from the device (usually a disk drive). Check to see that the disk is properly inserted in the drive, then type R (for Retry). [MS-DOS]

Reading source file(s) ...

Explanation: XCOPY is now reading the source files that you specified. [XCOPY]

Reinsert diskette for drive *x:*

Explanation: Reinsert the disk being formatted in the indicated drive. [FORMAT]

Replace *file*? (Y/N)

Explanation: MS-DOS displays this prompt if you specify REPLACE with the /W parameter. Type Y (for Yes) if you want to replace the existing file, or type N (for No) if you do not want to replace the file. [REPLACE]

`Replace the file (Y/N)?`

Explanation: The file that you want to restore from backup already exists on your target disk. Type `Y` (for Yes) and press the (Enter) key to overwrite the file. Type `N` (for No) and press the (Enter) key if you don't want to replace the file on your target disk with the file from the backup disk. [RESTORE]

`Requested logical drive size exceeds the maximum available space`

Explanation: You are trying to create a logical drive that is larger than the space available. [FDISK]

`Requested partition size exceeds the maximum available space`

Explanation: You are trying to create a partition on your hard drive that is larger than the space available. [FDISK]

`Requested Screen Shift out of range`

Explanation: You cannot shift the display any farther. [MODE]

`Required font not loaded`

Explanation: DISPLAY.SYS doesn't have the font size to do what you asked. Change the number of subfonts (*m*) to the maximum in the DEVICE=DISPLAY.SYS command in your CONFIG.SYS file, and then restart MS-DOS. [MODE]

`Required profile statement missing before line` *x*

Explanation: You left out a profile statement before line *x* and need to add it. [GRAPHICS]

`Resident part of PRINT installed`

Explanation: This is the first message that MS-DOS displays when you issue the PRINT command. It means that to process the PRINT command with other processes, available memory has been reduced by several thousand bytes. [PRINT]

`Resident portion of MODE loaded`

Explanation: Part of the MODE command is now resident in memory, and available memory has been reduced by several thousand bytes. [MODE]

C

`Resident portion of NLSFUNC loaded`

Explanation: NLSFUNC stays resident in memory once it is initialized. This informational message lets you know that you have already loaded NLSFUNC into memory. [NSLFUNC]

`Restore file sequence error`

Explanation: You have restored files in the wrong order. You must insert the backup disks in the same order that they were backed up. [RESTORE]

`*** Restoring files from drive` *x:* `*** Diskette:` *n*

Explanation: This message is displayed during the restore process. [RESTORE]

`Resync failed. Files are too different`

Explanation: FC compares what can be loaded into memory. If no lines match in the portion of the files in the buffer space, FC displays this message. [FC]

`Same drive specified more than once`

Explanation: You tried to activate FASTOPEN for the same drive more than once. There is no need to reactivate it for the same drive. [FASTOPEN]

`Same parameter entered twice`

Explanation: This is an informational message indicating that your have entered either /T, /N, /F, or /V twice and need to re-enter the correct parameter. [FORMAT]

`SECOND diskette bad or incompatible`

Explanation: The second disk does not contain the same format as the first disk, or DISKCOMP does not recognize the format of the second disk. You should run CHKDSK to help you identify the problem. [DISKCOMP]

`Sector not found error reading (or writing) drive` *x:*

Explanation: This error usually means the disk has a defective spot so that MS-DOS cannot find the requested information on it. You should copy all files from the disk onto a good disk and then try to reformat the defective disk. [MS-DOS DEVICE ERROR]

`Sector size too large in file` *file*

Explanation: The specified device driver loaded by CONFIG.SYS uses a sector size larger than that of any other device driver on the system. You cannot run this device driver. [MS-DOS]

`Seek error reading (or writing) drive` *x:*

Explanation: MS-DOS is unable to locate the information on the disk. Make sure that the disk is properly inserted in the drive, or try a different drive. [MS-DOS DEVICE ERROR]

`SHARE already installed`

Explanation: Share can only be installed once. [SHARE]

`Sharing violation reading drive` *x:*

Explanation: A program tried to access a file that another program was currently using. Type **A** (for Abort), or wait awhile and type **R** (for Retry). [MS-DOS DEVICE ERROR]

`SORT: Incorrect DOS version`

Explanation: Sort does not run on MS-DOS versions before 2.0. [SORT]

`SORT: Insufficient disk space`

Explanation: The disk is full. [SORT]

`SORT: Insufficient memory`

Explanation: There is not enough memory to run the SORT command. [SORT]

`Source and target drives are the same`

Explanation: You specified the same drive for the source and target disks. [BACKUP, RESTORE]

`Source disk is Non-removable`

Explanation: This is an informational message indicating that the source disk is a hard disk. [BACKUP]

`SOURCE diskette bad or incompatible`

Explanation: Your disk either has bad sectors or is incompatible with the source drive type (e.g.; a high-capacity disk in a 320/360K disk drive). [DISKCOPY]

`Source does not contain backup files`

Explanation: You are attempting to restore files from a disk that does not contain backup files: [RESTORE]

`Source path required`

Explanation: You did not specify a source path for the REPLACE command. [REPLACE]

`Specified COMMAND search directory bad`

Explanation: The SHELL command in the CONFIG.SYS file is incorrect. Make sure that the COMMAND.COM file exists and that MS-DOS can find it. [MS-DOS]

```
Specified drive does not exist,
or is non-removable
```

Explanation: You cannot compare or copy hard disks with this command. You must specify the name of a valid flexible drive. [DISKCOMP, DISKCOPY]

`Strike a key when ready ...`

Explanation: This prompt occurs during command processing and is always accompanied by another message. This message is also displayed if you have inserted a PAUSE command in a batch file. Usually, MS-DOS asks you to insert disks into appropriate drives before this prompt. To begin command processing, press any character, any number key, the spacebar, or the Enter key. [MS-DOS]

`Syntax error`

Explanation: You have entered a command incorrectly. Check to make sure you have typed the command correctly and have used appropriate quotation marks. [ATTRIB, FIND, MS-DOS]

`Syntax errors in GRAPHICS profile`

Explanation: You have entered a command incorrectly. Check to make sure you have typed the command correctly. [GRAPHICS]

`System transferred`

Explanation: The system files were transferred during FORMAT or SYS command processing. [FORMAT, SYS]

`Target cannot be used for backup`

Explanation: Either the target disk has an unrecognizable format, or it is bad. You should either try to format the disk with the format command, run CHKDSK on it to determine the problem, or not use the disk. [BACKUP]

`Target diskette bad or incompatible`

Explanation: Either the target disk has an unrecognizable format, or it is bad. Try to format the disk with the FORMAT command, or run CHKDSK on it to determine the problem. [DISKCOPY]

`Target diskette may be unusable`

Explanation: Either the target disk has an unrecognizable format, or it is bad. Try to format the disk with the FORMAT command, or run CHKDSK on it to determine the problem. [DISKCOPY]

`Terminate batch job (Y/N)?`

Explanation: If you press Ctrl C at the same time while in batch mode, MS-DOS asks you whether or not you wish to end batch processing. Type `Y` (for Yes) to end processing, or type `N` (for No) to continue. [MS-DOS]

`The current active keyboard table is` *xx* `with code page:` *yyy*
`The current active CON code page is` *zzz*

Explanation: This is an informational message that shows the current keyboard code, code page for the system, and code page for the console (screen). [KEYB]

`The last file was not restored`

Explanation: There was not enough room on the target disk for the file, or the last file was bad. Use the CHKDSK command to determine the problem. [RESTORE]

`The only bootable partition on drive 1 is already marked active`

Explanation: You are trying to change the active partition. The active partition must reside on the first hard disk drive on your system and must be bootable. The only bootable (from which your computer can start) partition on the first hard disk drive is already the active partition. [FDISK]

`There is not enough free memory in your system`
`Your system has xxxK bytes of free memory.`

Explanation: You have selected a cache or reserve size that exceeds the memory capacity of your system. Try a different value for the /S or /R parameter. If you are also using a RAM disk, you may wish to reduce its size or eliminate it altogether.

If you have requested the cache to be installed in conventional memory, try instead to install the cache in expanded or extended memory if you have it. [HP DISK CACHE]

`Too many drive entries`

Explanation: You can use FASTOPEN with up to four hard drives. You have tried to specify a fifth drive. [FASTOPEN]

`Too many extent entries`

Explanation: The total number of extent entries (*m*) cannot exceed 999. [FASTOPEN]

`Too many file/directory entries`

Explanation: The total number of file or directory entries (*n*) cannot exceed 999. [FASTOPEN]

`Too many files open`

Explanation: MS-DOS could not open the .BAK file or write the volume label due to the lack of available system file Handles. Increase the value of the FILES command in the CONFIG.SYS file. [EDLIN, LABEL]

`Too many name entries`

Explanation: The total number of entries specified for a drive exceeded the maximum of 999. [FASTOPEN]

`Too many open files`

Explanation: MS-DOS could not open the files that you want to compare due to the lack of available system file Handles. Increase the value of the files command in the CONFIG.SYS file. [BACKUP, FC, RESTORE, XCOPY]

`Top level process aborted, cannot continue`

Explanation: You chose to abort and the error is unrecoverable. You must use another disk. [COMMAND]

`Track 0 bad - disk unusable`

Explanation: The FORMAT command can accommodate defective sectors on the disk, except for those near the beginning. You must use another disk. [FORMAT]

`Tree past this point not processed`

Explanation: This is an informational message indicating that CHKDSK cannot continue processing past this directory path because of the above-mentioned error. [CHKDSK]

`/U Cannot un-install -- Other programs above.`

Explanation: You have requested that your disk cache be un-installed after loading one or more programs after it. You will need to exit or un-install these programs before un-installing the disk cache.

`/U parameter ignored -- HP Disk Cache is not installed.`

Explanation: You have requested that a disk cache to be un-installed when it currently is not installed.

`Unable to access Drive` *x*

Explanation: Fdisk cannot access your drive *x*. Turn off your system and then restart. [FDISK]

`Unable to create directory`

Explanation: MS-DOS could not create the directory you specified. Check to see that there is not a name conflict. You may have a file with the same name, or the disk may be full. [MKDIR, XCOPY]

`Unable to create KEYB table in resident memory`

Explanation: MS-DOS tried to create a country-specific table for the keyboard code specified, but failed. Check the amount of available memory. There may not be enough memory available to create this table. [KEYB]

`Unable to perform refresh operation`

Explanation: PRINTER.SYS doesn't have a copy of the code page in its Random Access Memory (RAM) to download to the printer. You must prepare and select the code page and make sure that the number of buffers (n) in DEVICE=PRINTER.SYS is equal to or greater than 1. [MODE]

`Unable to reload with profile supplied`

Explanation: You already loaded GRAPHICS once and didn't allocate enough memory for a second, different profile. Start your system again. [GRAPHICS]

`Unable to shift Screen`

Explanation: MODE is unable to shift the test pattern on the screen any farther. [MODE]

`Unable to write BOOT`

Explanation: If the first track of the disk or DOS partition is bad, the Boot record cannot be written on it. You must use another disk and retry the FORMAT command. [FORMAT]

`Unexpected DOS Error` *n*

Explanation: An unexpected error *n* occurred, where *n* is the MS-DOS error number. [REPLACE]

`Unrecognized command in CONFIG.SYS`

Explanation: There is an invalid command in your CONFIG.SYS file. Refer the "System Configuration" chapter for a list of valid statements. [MS-DOS]

`Unrecognized parameter ignored: /`*x*

Explanation: You have selected on option that is not recognized by the HP Disk Cache Program. [HP DISK CACHE]

`Unrecoverable error in directory`
`Convert directory to file (Y/N)?`

Explanation: This message is displayed if CHKDSK is unable to correct an error in a directory. If you respond `Y` (for Yes) to this prompt, CHKDSK converts the bad directory into a file. You can then fix the directory or delete it. If you respond `N` (for No) to this prompt, you may not be able to write to or read from the bad directory. [CHKDSK]

`Unrecoverable read error on drive` *x*
`side` *n*, `track` *n*

Explanation: After four attempts, the data still could not be read from the indicated side and track. Try the command again with a new disk, or copy all files from the damaged disk to a new disk. Either reformat the bad disk or discard it. [DISKCOMP]

`Unrecoverable read (or write) error on drive` *x:*

Explanation: MS-DOS is unable to read or write data to the specified device. Make sure that the disk is properly inserted in the disk drive. Then type `R` (for Retry). If the error occurs again, type `A` (for Abort). [MS-DOS DEVICE ERROR]

`Usage: FC [/A] [/B] [/C] [/L] [/LB n] [/W]`
`[/T] [/N] [/nnnn] file1 file2`

Explanation: One of the parameters that you have specified is invalid. [FC]

`VERIFY is off` (or on)

Explanation: This message tells you the current setting of the VERIFY command. [MS-DOS]

`Volume in drive` *x:* `has no label`

Explanation: This is an informational message displayed in response to the DIR, LABEL, or VOL command. [DIR, LABEL, VOL]

`Volume in drive` *x:* `is filename`

Explanation: This is an informational message displayed in response to the DIR, LABEL, or VOL command. [DIR, LABEL, VOL]

```
Volume label (11 characters, ENTER for none)?
```

Explanation: This message is displayed when you specify the LABEL command or the /V parameter in the FORMAT command. Type a volume label, or press the Enter key to indicate that you do not want a volume label for this disk. It's a good idea, though, to specify a volume label to help you identify your disks. [FORMAT, LABEL]

```
Volume label does not match
```

Explanation: You must enter the correct volume label for this drive. [FDISK]

```
Volume serial number is xxxx-xxxx
```

Explanation: This is an informational message. [MS-DOS]

```
WARNING, ALL DATA ON NON-REMOVABLE
DISK DRIVE x: WILL BE LOST!
Proceed with Format (Y/N)?
```

Explanation: This message appears when you try to format a hard disk that already contains data. If you type `Y` (for Yes), the data on the disk will be erased. If you do not want the files on your hard disk erased, type `N` (for No). Copy the files to a flexible disk and repeat the format command. [FORMAT]

```
Warning - directory full
```

Explanation: The root directory is too full for RECOVER processing. Delete some files in the root directory to free space for the recovered files, and try the command again. [RECOVER]

```
Warning! Diskette is out of sequence
Replace diskette or continue if okay
Strike any key when ready
```

Explanation: You should restore the diskettes in the order that you backed them up. [RESTORE]

```
Warning! File file
is a hidden (or read-only) file
Replace the file (Y/N)?
```

Explanation: This message prompts you as to whether you want to replace a hidden or read-only file. Type Y (for Yes) if you want to restore the hidden or read-only file from the backup disk. Type N (for No) if you do not want to restore this file. [RESTORE]

```
Warning! File file
was changed after it was backed up
Replace the file (Y/N)?
```

Explanation: This message prompts you as to whether you want to replace a backup file that has been changed. Type Y (for Yes) if you want to restore this file, or type N (for No) if you do not. [RESTORE]

```
WARNING! File x
is a read-only file
Replace the file (Y/N)?
```

Explanation: This message prompts you as to whether you want to replace a read-only file. Type Y (for Yes) if you want to restore the read-only file from the backup disk. Type N (for No) if you want to restore this file. [RESTORE]

```
Warning! Files in the target drive
\Backup (or root) directory will be erased
```

Explanation: MS-DOS found files in the target drive, and you did not specify the /A parameter of the BACKUP command to append files. [BACKUP]

```
Warning: HPDCACHE.SYS placed after non-disk driver.
```

Explanation: Either your external hard disk is not turned on, or the HPDCACHE.SYS driver has not been placed after the external hard disk driver in your CONFIG.SYS file. [HP DISK CACHE]

```
Warning! No files were found to back up
```

Explanation: MS-DOS did not find any files to back up on the disk you specified. [BACKUP]

`Warning! No files were found to restore`

Explanation: MS-DOS did not find the file that you wanted to restore from the backup disk. [RESTORE]

`Warning: Read error in EXE file`

Explanation: The amount read was less than the size of the header. This is a warning message only. [EXE2BIN]

`WARNING: The High Memory Area is unavailable.`

Explanation: HIMEM cannot find enough memory to use the High Memory Area. HIMEM will not be able to process any requests for the High Memory Area. However, HIMEM will remain installed to process any requests for the Extended Memory Data Blocks. [HIMEM.SYS]

`WARNING: The A20 Line was already enabled.`

Explanation: HIMEM detected the A20 hardware already enabled when it was installing, which is an abnormal condition. It is most commonly caused by another program which is controlling the A20 line. HIMEM will remain installed and attempt to work properly, however it will never disable the A20 line. [HIMEM.SYS]

`Warning! The partition marked active is not bootable`

Explanation: The active DOS partition must be bootable. [FDISK]

`Write failure, diskette unusable`

Explanation: This message indicates that a DOS critical error occurred when SYS was writing data to the disk. You must rerun the SYS command with a different disk. [SYS]

`Write fault error writing drive` *x:*

Explanation: MS-DOS is unable to write data to the specified device. Make sure that the disk is properly inserted in the disk drive. Then type **R** (for Retry). If the error occurs again, type **A** (for Abort). [MS-DOS DEVICE ERROR]

```
Write protect error
Format terminated
```

Explanation: The disk you are trying to format is write protected. [FORMAT, BACKUP, DISKCOPY]

```
Write protect error writing drive x:
```

Explanation: You tried to write data on a write-protected disk. If the disk has a write-protect tab on it, you must remove the tab before you can write on the disk. (You should consider first why the disk was write-protected.) If the disk doesn't have a write-protect notch, you cannot write on that disk. [MS-DOS DEVICE ERROR]

```
/-x Drive x cannot be cached -- sector size incompatible.
```

Explanation: Drive *x* has a sector size which is not a multiple of 512 bytes. [HP DISK CACHE]

```
/-x Drive x cannot be cached -- physical unit unknown
```

Explanation: If drive *x* is a RAM disk, HP Disk Cache is simply informing you that it cannot cache a RAM disk. This is not a problem since there is no point in caching a RAM disk.

Another possibility is that drive *x* is a remote drive in a network. In this case, it is inappropriate to cache a remote drive. (You may want to install a disk cache on the system where the remote drive resides.)

You may also see this message if drive *x* is installed in your system in such a way that MS-DOS accesses it without using the BIOS (Basic Input Output System). In the case of external hard disk drives, refer to the "HP Vectra Disk Cache" appendix. [HP DISK CACHE]

Index

HEWLETT
PACKARD

Printed in Singapore 07/89
Part Number D1151-90001