MS-DOS 2.1 User's Guide

COPYRIGHT

Published by arrangement with Microsoft Corporation, whose software has been customized for use on various desktop microcomputers produced by VICTOR. Portions of the text hereof have been modified accordingly.

VICTOR Publications
380 El Pueblo Road
Scotts Valley, California 95066
(408) 438-6680

TRADEMARKS

NOTICE

VICTOR makes no representations or warranties of any kind whatsoever with respect to the contents hereof and specifically disclaims any implied warranties of merchantability or fitness for any particular purpose. VICTOR 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.

VICTOR reserves the right to revise this publication from time to time and to make changes in the content hereof without obligation to notify any person of such revision or changes.

Note: The following MS-DOS 2.1 utilities have been copyrighted by MetaWare Inc.: search, fgrep, concat, tail, wc, ls, mv.

First VICTOR printing February, 1984.
Second VICTOR printing January, 1985.
Rev. A

ISBN 0-88182-128-4 Printed in U.S.A.

IMPORTANT SOFTWARE DISKETTE INFORMATION

For your own protection, do not use this product until you have made a backup copy of your software diskette(s). The backup procedure is described in the user's guide for your computer.

Please read the DISKID file on your new software diskette. DISKID contains important information including:

- ▶ The part number of the diskette assembly.
- ▶ The software library disk number (for internal use only).
- ▶ The date of the DISKID file.
- ▶ A list of files on the diskette, with version number, date, and description for each one.
- ▶ Configuration information (when applicable).
- ▶ Notes giving special instructions for using the product.
- ▶ Information not contained in the current manual, including updates, any known bugs, additions, and deletions.

To read the DISKID file onscreen, follow these steps:

1. Load the operating system.
2. Remove your system diskette and insert your new software diskette.
3. Enter —

   ```
   type diskid(cr)
   ```

4. The contents of the DISKID file is displayed on the screen. If the file is large (more than 24 lines), the screen display will scroll. Type ALT-S to freeze the screen display; type ALT-S again to continue scrolling.

CONTENTS

4. Using the Line Editor (EDLIN)

5. Using the CP/M-86 Emulator

6. MS-DOS Command Editing, Filtering, and Piping

7. MS-DOS Commands

APPENDIXES

FIGURES

TABLES

CHAPTERS

MANUAL CONVENTIONS

This manual uses the following conventions:

- When commands are explained for the first time, the command format is set off in boldface, such as:

  DO filename /switch(cr)

  —The parts of the command that you type exactly as shown (such as the command name) are all-uppercase.

  —Parts of the command that you vary or omit to fit the situation (such as filenames and drive names) are all-lowercase.

  —When you type a command, you can use upper- or lowercase for any command elements.

- In examples and command formats, pressing the Return key is shown as (cr), and typing a space is shown as (sp).
- Some examples show the screen, including what you type and the system's response. In these screen examples, what you type is underlined and all-lowercase. For example:

```
   A>print myfile(cr)
```

- In the text, names of commands, files, and programs appear in all-uppercase. Examples are COPY and TEXT.DOC.
- The ALT (Alternate function) key acts as the "Control" key on your computer. In text, "ALT-" represents the ALT key. For example, ALT-C represents typing the Alternate function of key C. In screen examples, however, the ALT key is shown as a caret (^), such as ^C.

PREFACE

This user's guide contains these eight chapters:

- Chapter 1 discusses introductory subjects such as the structure of MS-DOS; how the system uses memory and disk space; and how to load MS-DOS.
- Chapter 2 describes the types of disk files: system files and utilities, data/text files, and batch files. This chapter also describes how to use device files, and gives the rules for naming files and for using wildcard characters.
- Chapter 3 discusses the hierarchical (tree-structured) MS-DOS directory system, and tells how to use subdirectory names, create and delete directories, move between directories, and find files in the directory system.
- Chapter 4 explains how to use EDLIN, a text editor used mainly to edit source files.
- Chapter 5 describes the CP/M-86 Emulator, which runs CP/M-86 programs under MS-DOS.
- Chapter 6 tells how to edit MS-DOS commands, how to redirect input/output, and how to "filter" and "pipe" data.
- Chapter 7 describes all the MS-DOS commands—internal, external, and batch commands, plus system configuration commands. This chapter gives the syntax for each command, explains the parameters used with each command, and gives examples of usage.
- Chapter 8 lists and describes error messages the system may display during bootup and during normal operations.

INTRODUCTION

MS-DOS (Disk Operating System) is a group of programs that:

▶ Manage files and directories

▶ Process commands

▶ Control application programs designed for MS-DOS

▶ Control the operation of your keyboard, disk drive(s), and printer(s)

The first part of this chapter gives short descriptions of the system files and utilities that make up MS-DOS. Also included is a discussion of how MS-DOS uses memory (RAM) and disk space. The last part of the chapter describes the system configuration, and the procedures for loading MS-DOS and for entering the date and time.

MS-DOS SYSTEM STRUCTURE 1.1

MS-DOS consists of the following files/programs:

▶ System Files

COMMAND.COM—a command processor that accepts commands from the keyboard and runs the programs that process these commands.

CONFIG.SYS—a collection of system initialization routines and the commands for changing the system configuration (see Chapter 7.3).

MSDOS.SYS—the actual operating system program that manages files and application programs. MSDOS.SYS is a hidden file.

▶ Command Files (Utility Programs)

CHKDSK displays the amount of disk space occupied by files and tells you how much space is available for storage. CHKDSK also scans the directory for errors and can correct allocation errors.

COMMAND loads the command processor (COMMAND.COM) into memory.

CONCAT concatenates (combines) files, displays the files onscreen, or sends them to a file.

CPM is the CP/M-86 Emulator. It runs CP/M-86 programs under MS-DOS control.

DISKCOPY copies the contents of one diskette onto another diskette.

EDLIN is the MS-DOS line editor program.

FILCOM (File Comparison utility) compares two files and displays differences between the two.

FGREP searches through files or directories for a specified expression, and then prints the names of the files and the lines containing the expression.

FIND searches for a specified string of text.

FORMAT prepares a diskette to receive MS-DOS files.

LS lists a directory in alphabetical order, or lists selected parts of a directory according to specified constraints.

MODCON changes the "soft console" (keyboard and character set files) used by your computer.

MORE displays the contents of files one screen at a time.

MV moves files (makes a copy of a file, and deletes the original), and renames files.

PORTSET changes the transmission settings (baud rate, parity, stop bits, and bits per character) for the serial output port(s).

PRINT queues a list of files for printing.

RDCPM displays a CP/M-86 diskette directory or transfers CP/M-86 files onto an MS-DOS diskette.

RECOVER recovers lost data from a file or from a whole disk.

SDCOPY copies the contents of one diskette onto another diskette. SDCOPY requires only one floppy drive.

SEARCH selects files from a given set of files according to matching constraints, and performs a specified action on the matching files.

SORT sorts specific lines of data from a file or arranges the file in ascending or descending order.

SYS copies the hidden system files from one diskette or hard disk volume to another.

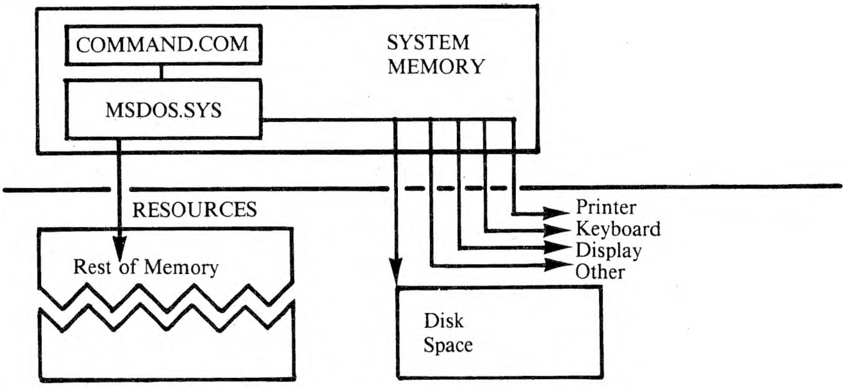
TAIL displays the final lines of a file.

WC counts words and/or lines in a file or a group of files.

Note: Chapter 7.4 contains detailed descriptions of all the MS-DOS commands in alphabetical order. Refer to Chapter 7.4 for explanations of commands as you read the earlier chapters.

The hidden system file MSDOS.SYS contains programs that execute automatically when MS-DOS loads into memory. COMMAND.COM and MSDOS.SYS control the system resources. The relationship between these files and system resources is shown in Figure 1-1.

Figure 1-1: MS-DOS and Its Resources



The system's resources are its internal memory (RAM) and external storage (disk space), plus connections to the CRT, keyboard, printer(s), other peripheral devices, and a network configuration, if any.

Memory and the MS-DOS file system can be thought of as the external organization of system resources. MS-DOS supports "device-independent I/O." This means that you can treat devices (such as your printer, keyboard, and CRT) as if they were files when you do input/output processing (see Chapter 2.2).

The system's memory and its disk space are described in Chapters 1.1.1 and 1.1.2.

MS-DOS controls main memory (RAM) as well as disk space and other devices. MS-DOS can load files into memory as data files or as executable files. The actual loading of files is performed by MSDOS.SYS, the lowest level of the MS-DOS operating system. COMMAND.COM supervises the loading of executable files.

Most programs return control to MS-DOS after normal or unexpected termination of the program.

MS-DOS may overlay (write over) the transient portion of COMMAND.COM to make room for a particularly large executable file. After such a file is executed, MS-DOS automatically loads the overlaid part of COMMAND.COM back into memory from disk, and normal execution of COMMAND.COM resumes. MS-DOS attempts to reload the transient portion from the root directory of the default drive at startup (not from the current default drive, if the default drive has changed) or from the root directory of the drive specified in the CONFIG.SYS file.

If MS-DOS finds an incorrect version of COMMAND.COM when it attempts to load the overlaid section, it displays a message such as:

```
Invalid COMMAND.COM
  COMMAND.COM expected at X:\COMMAND.COM
  <CR> for retry, or new COMMAND.COM path =
```

where drive X is the original default drive. Insert the system diskette you used to load MS-DOS (or any diskette with the version of COMMAND.COM that was originally loaded). To use another drive or path for reloading COMMAND.COM, type the name of that drive or path name. Now MS-DOS will use that drive or path each time it must reload COMMAND.COM during this operating session.

1.1.2 DISK SPACE

In MS-DOS, disk space is divided into four parts:

- The reserved sectors contain information used each time MS-DOS is loaded.
- The directory contains information about each file on a given diskette. This information includes the file's complete filename, its size, and the time and date of its last modification.
- The file allocation table (FAT) contains location information for the data making up each file on a diskette.
- Files occupy the great majority of disk space. An individual file does not necessarily reside in contiguous sectors on the diskette; its contents may be "scattered" on the diskette to decrease wasted disk space.

1.2 SYSTEM CONFIGURATION

With MS-DOS your system configuration can include these peripheral devices:

- One parallel printer
- One serial printer
- A MODEM option (telephone hookup)
- A CODEC option (voice recorder)

To set up your operating system for your particular configuration, refer to the *Operator's Reference Guide*, and the discussions of the system configuration commands and the CAUX, CLST, CTTY, and PORTSET commands in Chapter 7.

LOADING MS-DOS 1.3

Before you start working with your system, make a backup of your MS-DOS system diskette using the DISKCOPY program. To load MS-DOS, insert the system diskette and press the reset button, as described in the *Operator's Reference Guide.*

Once MS-DOS is in main memory, it prompts you to enter the date:

```
Enter new date:_
```

MS-DOS keeps track of time, and it records the current date and time on each file that you create or change. If you set the date to today's date, MS-DOS can date your work accurately. Follow these guidelines:

- Type the date as numbers separated by slashes or by hyphens, in one of these formats:

  —month/day/year or month-day-year (for American OS)

  —day-month-year or day/month/year (for European OS)

- For the month, enter a number from 1 to 12. For the day, enter a number from 1 to 31. For the year, enter a number from 80–99 or from 1980 to 2099. Then press Return to submit your entry to MS-DOS.
- Use the Backspace key to correct mistakes.
- Do not type the name of the day.
- If you do not want to type in a date, press the Return key. Then MS-DOS uses the system creation date or the last valid date setting as the current date.

To enter the date of February 25, 1984, you would type:

2/25/84 or 2-25-84 (American OS)

or

25-2-84 or 25/2/84 (European OS)

If the date you enter is not in the correct format or contains a number outside the valid ranges (such as 13 for the month), MS-DOS displays:

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

After MS-DOS accepts your entry for the date, it displays its current time setting and prompts you to reset the time:

```
Enter new time:_
```

Note: The AUTOEXEC.BAT file and the SHELL command in the CONFIG.SYS file can instruct MS-DOS to load the command processor without prompting you for the date and time of bootup. The date and time prompts, however, will appear by default each time COMMAND.COM loads.

To enter the time, you must use a 24-hour clock. For example, 1 pm is 13:00, and 11 pm is 23:00. When MS-DOS displays the time, it gives the hour, minute, second, and half second (0:00:00.0).

To change the time:

▶ Use the numbers 0–23 for the hour and 0–59 for the minutes. You do not have to make an entry for seconds. Do not type the half second.

- Separate the time with colons (hour:minute:second).
- Use the Backspace key to correct mistakes.
- Press the Return key when you want MS-DOS to accept your entry.

If you have made an error, after you press the Return key MS-DOS displays:

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

Retype the time.

After MS-DOS accepts your entry for the time, the MS-DOS command prompt and the cursor appear on your screen. By default, the prompt displays the name of the drive from which MS-DOS was loaded. For example, if MS-DOS is loaded from drive A, the prompt is:

```
A>_
```

The command prompt indicates that MS-DOS is in main memory and is ready for you to enter an MS-DOS command from the keyboard. You can change the MS-DOS prompt to be any letter(s), word, or phrase with the PROMPT command.

You can check and change the date or time while you are using MS-DOS. Type the word “date” or “time” and press Return. MS-DOS shows the date or time and asks you to reset it. Press Return to leave it as is, or follow the rules for entering the date/time presented in this section.

2 THE MS-DOS FILE SYSTEM

This chapter describes basic facts about MS-DOS files, including:

- Types of files you can use with MS-DOS
- Conventions for naming files
- Rules for using wild-card characters
- How to create and run batch files
- The function of the AUTOEXEC.BAT file
- How to use peripheral devices (device files) for input/output

MS-DOS stores and processes data as files. A file is made up of one or more related characters (up to the data storage capacity of a diskette). MS-DOS uses files as locations for writing or reading data. A file's location can be a diskette, or a peripheral device, such as the keyboard, screen, or printer. The data in most files has been input from the keyboard.

TYPES OF FILES 2.1

MS-DOS recognizes these types of files:

- System files
- Command files (utility programs)
- Data/text files
- Batch files
- Source files

System files are predefined programs that:

- Run the computer and its devices
- Control other programs loaded into memory from disk, such as word processing programs
- Process data entered from the keyboard

A **command file** is a predefined program on disk. You run the program by typing the filename, followed by optional parameters. The terms "external command" and "utility program" both refer to command files. Utilities or external commands perform basic file and diskette management tasks (such as formatting and copying diskettes) that are necessary for every application. Each utility program is stored on disk with a filename extension of .COM or .EXE.

A **data/text file** is a collection of alphanumeric characters that can be in document form, such as a memo. It can also be a list of data, such as a mail list, payroll statement, or the output of an application program, like MS-BASIC.

A **batch file** is a sequence of MS-DOS commands. Each batch file performs a specific task or tasks when you load the batch file (see Chapter 2.5).

A **source file** contains English-like program language statements that you create. Your computer uses a memory-resident translation program called a compiler that puts each source statement line into a format that it can process.

FILENAME CONVENTIONS 2.2

The filename conventions described here are rules for specifying the name of a file, the filename extension, and the drive name.

FILENAMES 2.2.1

Names for MS-DOS files can be one to eight alphanumeric characters. The characters you can use are:

- The letters A through Z (either upper- or lowercase)
- The numbers 0 through 9
- The special characters
  ! @ # $ % & ( ) - ’ ^ { } ~ ‘

If you use any other characters, MS-DOS displays:

```
Invalid filename
```

Also, do not give any of your files the device names listed in Table 2-2.

FILENAME EXTENSIONS 2.2.2

An optional filename extension can be added to a filename. The extension is one to three alphanumeric characters preceded by a period, like this:

MAIL.LST

The valid characters for extensions are the same as for filenames. You can use extensions to identify types of files. For example, you might use your initials as the extension for the files you create. MS-DOS automatically defines the meanings of several file extensions, listed in Table 2-1.

If you specify more than three characters for the extension, MS-DOS accepts only the first three characters. If a data/text file has a filename extension, you must include the extension with the filename when you make a request for the file. If you want to load a command or batch file, you can omit the extension (.COM, .EXE, or .BAT) when you type the filename.

Table 2-1: MS-DOS File Extensions

FILE EXTENSION	MS-DOS INTERPRETATION
.ASM	8086 assembler language source code
.BAK	Backup file created by application
.BAS	BASIC source code (MS-BASIC)
.BAT	Batch command file
.COB	COBOL source code
.COM	Executable command file
.CRF	Cross-reference
.DAT	Data file (assumed to be ASCII)
.EXE	Relocatable executable file
.FOR	FORTRAN source code
.INT	Intermediate compiled code
.LIB	Library file
.LST	Listing of compilation or assembly
.MAP	List file from linker program
.OBJ	Relocatable object code module
.OVR	Overlay module
.PAS	Pascal source file
.PLM	PL/M source code
.PRN	Listing of compilation or assembly
.REF	Cross-reference listing
.$$$	Temporary system-generated file

Extensions other than those automatically defined by MS-DOS (listed in Table 2-1) or defined by your application program can be assigned at your discretion. For example, you might use the extension .TXT for all text files or .MEM for memos.

DRIVE NAMES 2.2.3

The drive name is the letter name (A–O) of the disk drive where the file is currently located. Place a colon between the drive name and the filename itself. For example, to indicate that the file MAIL.LST is on drive B, type:

```
b:mail.lst
```

If you give a filename without a drive name, MS-DOS searches for the file on the default drive.

To fully identify a file for MS-DOS, you may need to include the name of the subdirectory that contains the file. See Chapter 3 for information on using directory names.

The following description summarizes the format for a filename:

```
drivename:filename.ext
```

drivename:
- Is one letter (A–O)
- Is followed by a colon
- Does not need to be specified when the file is on the default drive

filename
- Contains up to 8 characters
- Does not use " / [ ] + = , ; : | < > \

.ext
- Is preceded by a period (.)
- Contains up to 3 characters
- Does not use " / [ ] + = , ; : | < > \
- Is optional except for the file types shown in Table 2-1

2.3 FILENAME WILD-CARD CHARACTERS

The ? and * wild-card characters replace existing characters in a filename. You can use wild-card characters to match a specified portion of a filename with existing filenames.

Wild-card characters can be used with these commands:

COPY	PRINT
DEL	RDCPM
DIR	REN

The ? wild-card character used in a filename with an MS-DOS command matches any existing character (or none) in that position.

For example, if your disk directory in the default drive has several similar filenames, such as:

```
CHAPT1
CHAPT2
CHAPT3
```

You can specify:

```
dir chapt?(cr)
```

to display only the filenames that match the first five characters of the filename in the command. The sixth character can be any character, as indicated by the ?.

The * wild-card character replaces any existing character in that position and all subsequent characters.

For example, if you specify:

```
dir ch*(cr)
```

MS-DOS displays all the filenames that start with CH. A matching filename can be 2 to 8 characters long, and the other characters in the filename can be any character. Using the previous directory example, MS-DOS displays CHAPT1, CHAPT2, and CHAPT3. Filenames such as CHARLES.MEM, CHART.DAT, or CHECKS are displayed also.

If you specify:

```
dir *.bat(cr)
```

MS-DOS displays the names of all the batch files (indicated by .BAT) in the current directory. *.* in a command indicates all the filenames in a directory. For example, using *.* with the DEL command deletes all the file entries from the directory. (MS-DOS asks if you are sure before it deletes the files. Make sure that you have backed up any files you need.)

You can use ? and * together in a command. If your directory contains:

83JOBS.PUB
82JOBS.PUB
83JOBS.ACT

Entering:

```
dir 8?jobs.p*(cr)
```

displays these filenames:

```
83JOBS.PUB
82JOBS.PUB
```

83JOBS.ACT is not displayed, because .A does not match .P.

2.4 DEVICE NAMES

MS-DOS can send data to or receive data from a device. MS-DOS interprets device names as the names of peripheral devices (see Table 2-2). You can use these device names as parameters in MS-DOS commands to refer to devices as if they were files. A device name, however, never has a file extension or drive name because device names are not disk files, but input/output devices. “Logical device” is another term for a device name.

Table 2-2: MS-DOS Device Names

NAME	DEVICE TYPE
AUX	An input or output auxiliary device.
CON	Keyboard input to the CPU, or CPU output to the screen.
LST	List device (printer).
NUL	A nonexistent device for a command that requires a filename. Use this device when you do not want to create a file.
PRN	Same as LST.
PARALLEL	The name for the device driver in the PPORT.EXE file.
SERIALA	The name for the device driver in the PORTA.EXE file.
SERIALB	The name for the device driver in the PORTB.EXE file.

This COPY command tells MS-DOS to copy data from the CON device file (the keyboard) to a file on disk called PHN.LST:

```
copy con phn.lst(cr)
```

After a COPY CON command, the next lines you type are temporarily stored in memory until you type ALT-Z and press the Return key. Then the text you have entered is stored in the file PHN.LST. Data to or from CON (or any device) must always be ended with ALT-Z, followed by a Return.

You can also redirect output to a device by using the > symbol (described in Chapter 6). For example, this command sends the directory to the printer (LST):

```
dir > lst(cr)
```

When you specify a device name, such as LST, be sure that the correct device type is assigned to LST (see the CLST command in Chapter 7.4).

BATCH FILES 2.5

A batch file is a sequence of MS-DOS commands that perform a specific task. You can create batch files with the line editor EDLIN, with the COPY CON command (described in Chapter 7), or with any word processing program. You must give batch files the extension .BAT. To run the sequence of commands stored in a batch file, type the name of the file without the extension.

Many of the examples in this manual use batch files to illustrate how you can perform several MS-DOS commands by typing a single batch command. Because batch files can perform many different functions, writing batch files is similar to programming, and the files themselves are like short programs "written in MS-DOS." This chapter gives the basic information you need to start using and writing batch files, but you will find examples of batch file usage throughout this manual.

The following example is a batch file named NEWDISK, containing a sequence of commands to display the main directory of a formatted disk and check the directory for consistencies:

```
REM This is a file to check formatted disks
REM It is named NEWDISK.BAT
PAUSE Insert disk in drive B
DIR B:
CHKDSK B:
```

Once the NEWDISK batch file is saved on disk, you can run the two commands (DIR B: and CHKDSK B:) by typing the batch filename: NEWDISK. PAUSE and REM (Remark) are commands used only for batch files. These commands and the other commands used only for batch files (FOR, IF, GOTO, ECHO, and SHIFT) are described in Chapter 7.4.

2.5.1 BATCH FILE CONVENTIONS

1. To run a batch file, specify the batch filename (without the .BAT extension). If the batch file contains replaceable parameters, follow the filename with the values you want to substitute for the parameters (see Chapter 2.5.2).
2. If you press ALT-C while a batch file is being processed, MS-DOS displays:

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

   Type Y to ignore the rest of the commands in the batch file and to return to the system prompt. Type N to stop the current command being processed. MS-DOS then processes the next command in the batch file.

3. You can specify the name of another batch file as the last command in the batch file. MS-DOS then invokes that batch file from the one being processed.

REPLACEABLE PARAMETERS 2.5.2

Replaceable parameters are "dummy values" in a batch file. They are numbers preceded by a percent sign (such as %1 and %2). When you run a batch file, you include the value(s) you want to use in place of the replaceable parameters in the file.

You can specify ten replaceable parameters in a batch file (%0 to %9). %0 is always replaced by the drive name (if there is one designated) and the name of the batch file. %1 is replaced by the first value following the filename in the command you enter to run the batch file. %2 is replaced by the second value, and so on. You can use more than ten replaceable parameters by using the SHIFT command.

For example, you can use the COPY CON command to create a phone list file:

```
copy con phn.lst(cr)
Mary Smith (415) 497-0110(cr)
Engineering (408) 438-6680(cr)
George Jones (916) 555-0900(cr)
Marketing (408) 438-7000(cr)
^Z(cr)
```

Then you can create a batch file named SEARCH that uses the FIND command to look through PHN.LST for a dummy value (the replaceable parameter %1):

```
copy con search.bat(cr)
find "%1" phn.lst(cr)
^Z(cr)
```

When you load the batch file, include the value or name that you want to substitute for the replaceable parameter in the batch file. For example, specifying:

search 408(cr)

replaces the parameter %1 with 408. MS-DOS then processes the SEARCH batch file with this value and displays the command from the file:

```
A>find "408" phn.lst
```

The FIND command then displays all the lines from the PHN.LST file that have 408 in them:

```
---------- phn.lst
Engineering (408) 438-6680
Marketing (408) 438-7000
```

2.5.3 THE AUTOEXEC.BAT FILE

AUTOEXEC.BAT is the name of a batch file that MS-DOS always looks for and processes immediately after it loads into memory. If there is no AUTOEXEC.BAT file, MS-DOS prompts for the date and time and then displays its command prompt. When there is an AUTOEXEC.BAT file, MS-DOS does not display the time and date prompts unless the DATE and TIME commands (described in Chapter 7.4) are included in the AUTOEXEC.BAT file.

Create an AUTOEXEC.BAT file when you want a command or sequence of commands to operate each time MS-DOS loads. For example, suppose your principal application is MS-BASIC. You can use COPY CON to create an AUTOEXEC.BAT file that takes you straight into MS-BASIC when you boot up:

```
copy con autoexec.bat(cr)
msbasic(cr)
^Z(cr)
```

The command "MSBASIC" tells MS-DOS to load the file MSBASIC.COM (the MS-BASIC language) into memory. If this AUTOEXEC file and the MSBASIC.COM file are on your system diskette, each time MS-DOS loads into memory, it displays the first command from the file:

```
A>msbasic
```

Next MS-BASIC loads into memory and displays its command prompt (Ok). You can now use MS-BASIC. To return to MS-DOS, use the MS-BASIC command to exit to the operating system.

If you create an AUTOEXEC.BAT file and want to process it immediately, specify:

```
autoexec(cr)
```

or reload MS-DOS by pressing the reset button.

3

THE MS-DOS SUBDIRECTORY SYSTEM

MS-DOS allows you to create and use a hierarchical subdirectory system. This chapter describes the subdirectory system and how to work with it. The first topic is a general introduction, followed by the rules for using subdirectory names to locate files.

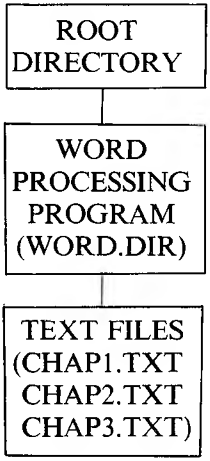
These commands are described in this chapter:

- DIR displays the current directory.
- MKDIR creates a subdirectory.
- CHDIR changes from one directory to another.
- RMDIR deletes an empty subdirectory.

A **directory** is a list of the files on a diskette, including the filenames, the file size, and the date and time each file was created. MS-DOS enters the filename and date and time in the directory each time you create or update a file.

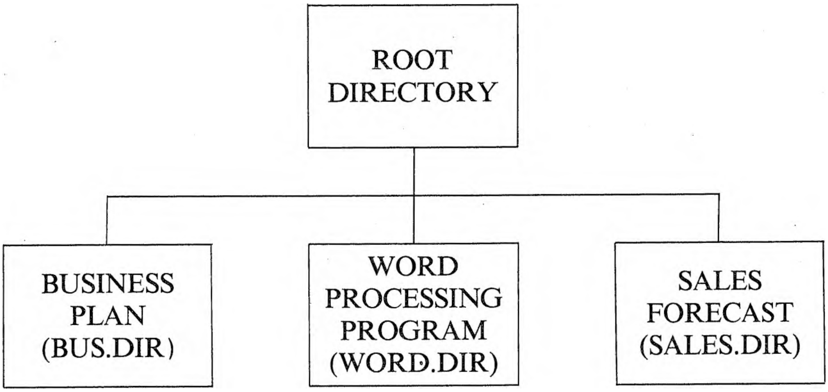
MS-DOS has a main directory to which you can add subdirectories. Subdirectories are hierarchical like a family tree or organization chart. The main or "root" directory is created by the FORMAT program during a disk format or by the AUTOSET program during hard disk initialization. The root directory is the start of your subdirectory system. The root directory contains files and can also contain the names of subdirectories. Each subdirectory in turn can contain files and other subdirectories.

For example, you could create a subdirectory called WORD.DIR and place your word processing application program in that subdirectory:


ROOT DIRECTORY
WORD PROCESSING PROGRAM (WORD.DIR)
TEXT FILES (CHAP1.TXT CHAP2.TXT CHAP3.TXT)

Then you can store all the text files you create with your word processing program in the word processing subdirectory, WORD.DIR.

You can group your files into subdirectories for each application:


ROOT DIRECTORY
BUSINESS PLAN (BUS.DIR)
WORD PROCESSING PROGRAM (WORD.DIR)
SALES FORECAST (SALES.DIR)

When you save a file, MS-DOS puts the filename in the directory you're currently using or in another directory that you specify. A file in one subdirectory can have the same name as a file in another directory because MS-DOS keeps each directory separate.

You can search for a file in one subdirectory rather than searching through many kinds of files in one large main directory. Chapter 3.1 describes how to specify the subdirectory name (or path) for a file.

USING DIRECTORY NAMES (PATHS) TO LOCATE FILES 3.1

When you use hierarchical directories, you must tell MS-DOS where to find your files in the directory structure. In addition to the filename, you give the name of the subdirectory where the file is located.

Giving the subdirectory name means specifying a path for MS-DOS to follow through the directory hierarchy. A path or subdirectory name:

- Is a sequence of directory names, each separated from the previous one by a backslash (\).
- Leads MS-DOS through the directory hierarchy. Each \ leads to a lower subdirectory level.
- Can end with a filename or a directory name.
- Can go through only existing subdirectories.
- Must not exceed 63 characters.

The syntax for a path to a file or subdirectory is:

```
[drivename:][[\]directoryname[\directoryname...]][filename.ext]
```

drivename
is the drive containing the file or directory you want. You can omit the drive name if the file or directory is on the default drive.

\
An initial backslash always indicates the root directory. If a path begins with a backslash, MS-DOS begins searching for the file or directory at the root directory. Otherwise, MS-DOS searches downward from the current directory.

directoryname
is the name of a subdirectory (created with the MKDIR command, described in Chapter 3.3). If the first subdirectory listed has no backslash, it is directly below the current directory. Each additional \directoryname indicates another subdirectory level below the root directory or below the current directory.

filename.ext
is the full filename and extension of the file you want. The filename is always last in the path. If the file is in the current directory, you do not need to specify a path with the filename.

MS-DOS recognizes two abbreviated path names:

.
is shorthand for the current directory. You can use the . notation with the DIR command to display the current subdirectory.

..
is shorthand for the current directory's parent directory. MS-DOS uses the . and .. directories to create the hierarchical tree directory structure. You can use the .. notation with DIR, CHDIR, and DEL.

3.1.1 EXAMPLES OF PATHS

\WORD.DIR and \MGR1.NAM
are subdirectory names. The initial \ indicates the root directory. Both WORD.DIR and MGR1.NAM are first-level subdirectories (directly below the root directory).

\WORD.DIR\CHAP1.TXT and \MGR1.NAM\JOBS\CLERK
are full path names. The file or subdirectory CHAP1.TXT is in the first-level subdirectory WORD.DIR. Likewise, the file or subdirectory CLERK is in the subdirectory JOBS under the first-level subdirectory MGR1.NAM.

WORD.DIR\DAVID and JOBS\BENEFITS
are relative path names that refer to the current directory level. DAVID is a file or directory in the subdirectory WORD.DIR below the current directory. The file or directory BENEFITS is in the JOBS subdirectory, which is listed in the current directory. If the current directory is the root directory, these path names are the same as \WORD.DIR\DAVID and \JOBS\BENEFITS.

WORKING WITH FILES IN SUBDIRECTORIES 3.1.2

Any of the commands that use filenames can also use paths or subdirectory names.

You locate a file by specifying a path for MS-DOS to follow through the directory hierarchy. The filename is always the last name in the path. When the file is in the current directory, you can omit the path and give just the filename. To get to a file in the previous directory, you can use .. as the path.

Displaying Files Onscreen

To display the CHAP1.TXT file in the WORD.DIR subdirectory on the default drive, enter:

```
type \word.dir\chap1.txt(cr)
```

The first \ tells MS-DOS to start the search for the file at the root directory. MS-DOS finds the subdirectory WORD.DIR listed in the root directory. In the WORD.DIR subdirectory, MS-DOS finds the CHAP1.TXT file and displays the file on the screen.

If WORD.DIR is the current directory, you can specify the filename without a path, like this:

```
type chap1.txt(cr)
```

If you include the drive name in a command, the drive name must precede the path. For example, this command displays CHAP1.TXT from the WORD.DIR subdirectory on drive B:

```
type b:\word.dir\chap1.txt(cr)
```

Copying Files

To copy a file from one directory to another, you must specify the path to the file to be copied and the path to the directory where the file is to be copied. You use the COPY internal command to copy files. For example, to copy the CHAP1.TXT file from the WORD.DIR subdirectory on drive A into the root directory on drive B, and to rename the file INTRO, enter:

```
copy a:\word.dir\chap1.txt b:intro(cr)
```

Deleting Files

To delete files from a directory, you use the DEL (Delete) internal command. If the file you want to delete is in the current directory, you do not need to specify a path to the file. For example, if you are using the WORD.DIR directory, you can delete a file from that directory with the command:

```
del chap1.txt(cr)
```

If you are using any other directory, you must include the path to the file you want to delete. For example:

```
del b:\word.dir\chap1.txt(cr)
```

deletes CHAP1.TXT from the WORD.DIR subdirectory on drive B.

You can use the .. notation with DEL to refer to the previous directory. For example, this command deletes MYFILE from the parent directory of the current directory:

```
del ..\myfile(cr)
```

PATHS AND EXTERNAL COMMANDS 3.1.3

External commands are programs stored in disk files. When you are working with more than one directory, you may want to put all the external commands in one directory to keep the other directories uncluttered. If you do this, you must be in the subdirectory containing the external command in order to load that command.

For example, suppose you store all the external commands in the BIN first-level subdirectory. To load and run the CHKDSK program, you must first move to the BIN subdirectory.

Instead of moving to the subdirectory, you can set up the path to the subdirectory with the PATH command. For this example, you would enter this PATH command:

```
path x:\bin(cr)
```

where x is the name of the default drive.

Now MS-DOS searches the current directory and the BIN subdirectory whenever you specify an external command. You only have to specify PATH once in an operating session (see PATH in Chapter 7.4).

3.2 LISTING A DIRECTORY

The DIR command, shown in Chapter 7.4, lists the contents of a directory. You can list a directory onscreen or send it to a device file. To list a subdirectory with DIR, you can use the path notation shown in Chapter 3.1.1.

For example, you can display the current directory on the default drive by entering:

dir(cr)

If you are in the WORD.DIR subdirectory, DIR displays that subdirectory. To display the WORD.DIR subdirectory on the default drive (regardless of the directory you are in), enter:

dir \word.dir(cr)

The directory display for the WORD.DIR subdirectory looks like this:

```
            Volume in drive A has no ID
            Directory of A:\word.dir

    .                <DIR>         5-05-83   10:09a
    ..               <DIR>         5-05-83    2:15p
    DAVID.L          <DIR>         5-20-83    3:40p
    CHAP1.TXT                345   6-15-83   11:13a
                   4 File(s)    370688 bytes free
```

This DIR display gives the following information:

- ▶ No Volume ID was assigned to the disk during formatting.
- ▶ This directory is WORD.DIR, a first-level subdirectory.
- ▶ The single period (.) indicates the current directory. The . is displayed only for subdirectories, not for the root directory. MS-DOS automatically creates the . entry when you create a directory.

- Two periods (..) indicates the "parent" directory (the subdirectory level above this one).
- The WORD.DIR subdirectory has another subdirectory (DAVID.L) below it.
- WORD.DIR contains the file CHAP1.TXT, which is 345 bytes, and was created or last modified at 11:13 am on June 15, 1983.

To display the current directory, you can use the shorthand notation:

```
dir .(cr)
```

To display the parent directory (one level above the current directory), enter:

```
dir ..(cr)
```

MAKING A SUBDIRECTORY 3.3

You can create a subdirectory by entering a MKDIR (Make Directory) command from the keyboard or from a batch file. (See the descriptions of the MKDIR and FOR commands in Chapter 7.4.) You can shorten MKDIR to MD. In the MKDIR command give the name of the directory you want to create. MS-DOS creates a subdirectory as a special kind of file with a <DIR> attribute.

The conventions for naming a directory are the same as for naming a file:

- 1–8 alphanumeric characters
- An optional filename extension of 1–3 alphanumeric characters preceded by a period

For example, suppose you want to create a subdirectory named MGR1.NAM under the root directory on the disk in the default drive. If you are in the root directory, you can enter:

```
mkdir mgr1.nam(cr)
```

If you are in another subdirectory, you can enter:

```
md \mgr1.nam(cr)
```

You can verify that MS-DOS created the subdirectory by entering the DIR command. Once a directory exists, you can add other subdirectories to it. For example, you can make a subdirectory called JOBS beneath MGR1.NAM. Enter:

```
md \mgr1.nam\jobs(cr)
```

Another procedure for creating a subdirectory is first to move into the parent directory, and then to create the subdirectory. You move from one directory to another with the CHDIR command, described in Chapter 3.4. For example, entering:

```
chdir mgr1.nam(cr)
mkdir jobs(cr)
dir(cr)
```

changes to the MGR1.NAM subdirectory, creates the JOBS directory, and displays the current directory, MGR1.NAM:

```
Volume in drive A has no ID
Directory of A:\mgrl.nam

.          <DIR>      6-30-83      9:00a
..         <DIR>      8-09-83     10:09a
JOBS       <DIR>      8-09-83     10:34a
       3 File(s)    370688 bytes free
```

If you create and save a file now, MS-DOS saves it in the MGR1.NAM subdirectory. For example, entering this COPY CON command creates and saves a file named TITLES in the MGR1.NAM subdirectory:

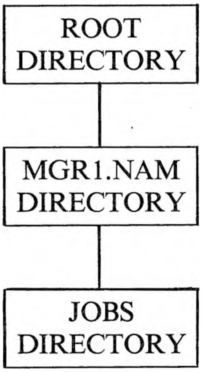
```
copy con titles(cr)
Publication Manager(cr)
Managing Editor(cr)
Copy Editor(cr)
^Z(cr)
```

Now the name of the TITLES file, its size, and the date and time it was created are displayed in the MGR1.NAM directory:

```
Volume in drive A has no ID
Directory of A:\mgr1.nam

.        <DIR>      6-30-83     9:00a
..       <DIR>      8-09-83    10:09a
JOBS     <DIR>      8-09-83    10:34a
TITLES           51 8-09-83    11:20a
       4 File(s)    370637 bytes free
```

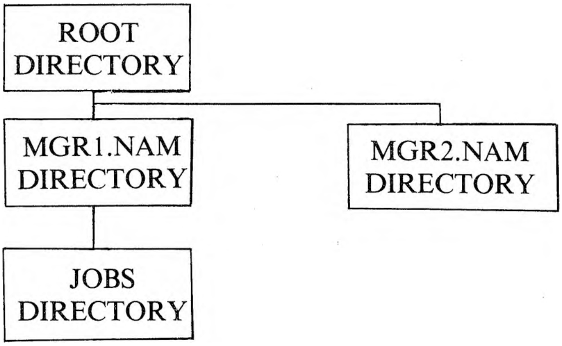
The full path to JOBS is \MGR1.NAM\JOBS, and the path to the TITLES file is \MGR1.NAM\TITLES. Each \ indicates a subdirectory level below the root directory. MGR1.NAM is the first subdirectory (indicated in the directory by the .), and JOBS is the second subdirectory:



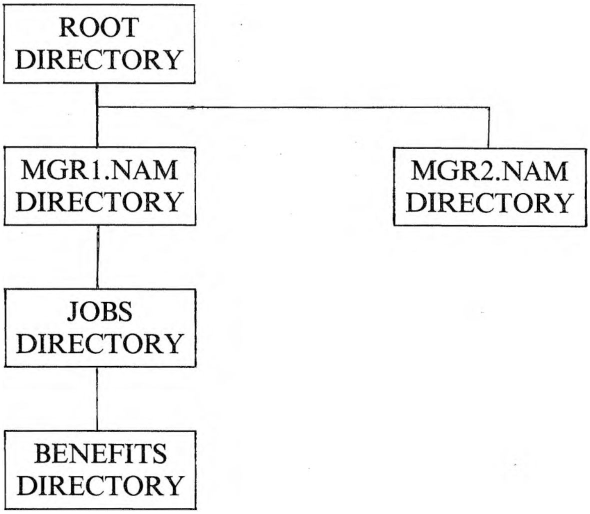
To add another subdirectory, such as MGR2.NAM, at the same level as MGR1.NAM, you must tell MS-DOS to start at the root directory. Enter:

```
mkdir \mgr2.nam(cr)
```

The first \ in a MKDIR command indicates the root directory to MS-DOS. You can omit the initial \ if you are at the root directory. MS-DOS creates MGR2.NAM and puts the new directory name in the root directory:



You can add a third directory level beneath JOBS, such as a subdirectory named BENEFITS:



If you are currently at the JOBS subdirectory, you can create BENEFITS by entering:

mkdir benefits(cr)

If you are not at JOBS, enter:

mkdir \mgr1.nam\jobs\benefits(cr)

Because each subdirectory name is listed in the preceding directory, you cannot create a subdirectory unless the parent directory exists. MS-DOS cannot create a new subdirectory unless a previous directory already exists. If you load MS-DOS and try to create several new subdirectories in the same command, such as:

mkdir mgr2.nam\mgr3.nam\mgr4.nam(cr)

MS-DOS responds with:

```
Unable to create directory
A>_
```

Likewise, MS-DOS does not allow you to give a file the same name as its subdirectory. This is because subdirectories are listed as files in a directory, and duplicate filenames are not allowed.

MOVING INTO A DIRECTORY 3.4

You move into a directory to:

- ▶ Display the directory.
- ▶ Create files or subdirectories.
- ▶ Delete files.
- ▶ Use files or programs in the directory.
- ▶ Remove subdirectories.

The CHDIR (Change Directory) command tells MS-DOS to move from one directory to another. You can shorten CHDIR to CD.

CHDIR without any parameters displays the name of the directory currently being used by MS-DOS. CHDIR with only a subdirectory name moves you to that directory if it is listed in the current directory. If you enter CHDIR and a subdirectory name that is not listed in the current directory, MS-DOS displays:

```
Invalid directory
```

Using the examples in Chapter 3.3, if you are in the root directory, you can move into the next subdirectory (MGR1.NAM) by entering:

```
chdir mgr1.nam(cr)
```

or

```
cd \mgr1.nam(cr)
```

The first \ in a CHDIR command indicates the root directory. If a directory name follows the backslash, it names the first subdirectory below the root directory. Each additional \ in the command is another subdirectory level in the path to the specified directory. The command:

```
chdir \mgr1.nam\jobs\benefits(cr)
```

places you in the BENEFITS subdirectory. To move up to the MGR1.NAM subdirectory, enter:

```
chdir \mgr1.nam(cr)
```

To get to JOBS from MGR1.NAM (the current directory), enter:

```
chdir jobs(cr)
```

If you are not at MGR1.NAM, you must indicate the path to JOBS:

```
chdir \mgr1.nam\jobs(cr)
```

You can change to the level above the current directory level by specifying .. with CHDIR:

```
chdir ..(cr)
```

This command moves you to the directory above the one you are now using. For example, if you are at JOBS and you specify CHDIR .., MS-DOS moves up to MGR1.NAM.

A CHDIR command with only a \, such as:

```
chdir \(cr)
```

always returns you to the root directory. You should enter a CHDIR \ when you finish working in a subdirectory. If you do not, you might save files in the wrong subdirectory. CHDIR \ ensures that MS-DOS is at the root directory.

REMOVING A SUBDIRECTORY 3.5

The RMDIR (Remove Directory) command removes an empty subdirectory from the directory hierarchy. You can abbreviate RMDIR to RD.

Before you can remove a subdirectory, you must delete all filenames (with DEL) and all lower subdirectories (with RMDIR). An empty directory lists only the . and .. notations, which cannot be deleted because they are part of the MS-DOS hierarchy system.

You can remove only one subdirectory at a time. For example, if you want to remove the BENEFITS subdirectory, first delete any files in BENEFITS. If you are in the BENEFITS subdirectory, move to another directory (use the CHDIR command as described in Chapter 3.4). To remove the BENEFITS subdirectory if you are in the JOBS subdirectory, enter:

```
rmdir benefits(cr)
```

You can also specify a full path from the root directory. For example:

```
rmdir \mgr1.nam\jobs\benefits(cr)
```

removes the BENEFITS subdirectory from the JOBS subdirectory if BENEFITS is already empty.

If you are currently using the MGR1.NAM directory, you can specify the path to the empty BENEFITS subdirectory and remove it:

```
rmdir jobs\benefits(cr)
```

If the directory you name does not exist, or if you try to remove the current directory or one that is not empty, MS-DOS displays:

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

4

USING THE LINE EDITOR (EDLIN)

EDLIN is a line editor program used mainly to create and edit source files (such as MS-DOS batch files). You can also use EDLIN to create or edit text. To input text such as memos or documents, however, it is simpler and more efficient to use a word processing program.

You can use two types of commands for editing with EDLIN:

▶ File commands, which create or edit files.

▶ Line commands, which edit characters within a single line of a file.

The file commands are unique to EDLIN. The line commands are the MS-DOS command-line editing functions (described in Chapter 6).

To use EDLIN, load the program as shown in Chapter 4.1. Chapter 4.1.1 shows you how to create a batch file called NEWDISK.BAT. This batch file is used in examples throughout this chapter. Chapter 4.1.2 shows you how to edit an existing file.

Chapters 4.2 and 4.3 discuss how to use the file and line commands, respectively. These chapters introduce you to the format of the commands. Chapter 4.2.3 is a reference section summarizing the EDLIN file commands, and Chapter 4.3.2 summarizes the line commands.

4.1 INVOKING EDLIN

The EDLIN.COM file must be available on disk in order to load the program. Also, if you are using subdirectories, MS-DOS must be able to find EDLIN.COM in the directory system as described in Chapter 3. (If it is not in the current subdirectory, give its subdirectory location in the EDLIN command, or enter a PATH command to indicate its location to MS-DOS.)

You load EDLIN and the file you want to edit at the same time. To invoke EDLIN, type:

```
edlin filename(cr)
```

The filename you enter with the EDLIN command can be either the name of an existing file or the name of the file you are creating. Include the drive name if the file is not on the default drive. To edit a data/text file in a subdirectory, either change to that subdirectory before loading EDLIN or give the subdirectory name with the filename when you invoke EDLIN. Chapter 3 describes how to use the MS-DOS subdirectory system.

4.1.1 CREATING A FILE

To create a new file, enter EDLIN followed by the name of a file that does not exist. Include the drive on which the file is to be saved unless the file is to be saved on the default drive.

For example, if you want to create the file NEWDISK.BAT in subdirectory MGR1.NAM on the diskette in drive B, enter:

```
edlin b:\mgr1.nam\newdisk.bat(cr)
```

EDLIN creates a new, empty file, and displays the message "New file" and its command prompt (*):

```
New file
*_
```

To enter text into the file, type the Insert command (I) at the * prompt and press Return. EDLIN then numbers and displays the first line for your entry:

```
New file
*I(cr)
          1:*_
```

Now you can type in the text of the file. End each line with a Return. EDLIN then displays the next line number.

Here is an example of an MS-DOS batch file created with EDLIN. This file displays the directory of a formatted disk and checks the disk directories for consistency.

```
A>edlin b:\mgrl.nam\newdisk.bat(cr)
New file
*I(cr)
    1:REM This is a file to check formatted
      disks(cr)
    2:REM It is named NEWDISK.BAT(cr)
    3:PAUSE Insert formatted disk in drive B(cr)
    4:DIR B:(cr)
    5:CHKDSK B:(cr)
```

You can stop text entry without saving any of the file by entering the Quit (Q) command. When you finish entering data, type an ALT-C (Interrupt) or ALT-Z (Terminate Text Entry command) on the next line and press Return. Either ALT-C or ALT-Z stops Insert mode and returns you to the * prompt.

To display the file you created, enter the List command (L). EDLIN displays the file and numbers the lines. (Line numbers are not saved in the file on disk.) When you insert new lines or delete existing lines, EDLIN adjusts the line numbers accordingly. The line numbers always run from 1 through the number of the last line.

To save the file on disk, type the End command (E). You are returned to the MS-DOS command prompt.

4.1.2 EDITING AN EXISTING FILE

To edit an existing file, specify EDLIN, the drive and subdirectory containing the file, and the filename with its extension. For example, if you want to make changes to the file NOTICE.MEM in the WORD.DIR subdirectory on the default drive, enter:

```
edlin \word.dir\notice.mem(cr)
```

EDLIN loads the file into memory. If EDLIN can load the entire file, it displays the following message and the asterisk command prompt (*):

```
End of input file
*_
```

If EDLIN cannot load the entire file, it fills up 3/4 of available memory with the first part of the file and does not display the "End of input file" message.

To edit a file larger than memory:

1. Display the portion of the file in memory with the List Line command (L, described in Chapter 4.2.3).
2. Edit the portion of the file in memory.
3. Save the edited portion of the file on disk with the Write command (W). Add the rest of the lines from the file by using the Append command (A, described in Chapter 4.2.3).

USING FILE COMMANDS 4.2

Use the Insert command (I) to create an EDLIN file, as described in Chapter 4.1.1. After the file is created, you can use the EDLIN file commands to:

- ▶ Add lines from disk to memory if the file is too large for memory (Append).
- ▶ Delete one or more lines from the file (Delete).
- ▶ Modify existing lines (Edit).
- ▶ Add one or more lines of text to a file (Insert).
- ▶ Display the series of lines that you want to change (List).
- ▶ Replace previous text with new text within a certain number of lines in the file (Replace).
- ▶ Search through a specified number of lines in a file for a string of characters that you want to change (Search).
- ▶ Write edited portions of a large file from memory to disk (Write).

File commands are single letters (such as A for Append) with optional parameters. The parameters select, display, or change one or more lines in a file. You type the letter of the command and the parameters you want from the keyboard. The parameters and the commands are summarized in Table 4-1 and in Chapter 4.2.3.

4.2.1 FILE COMMAND PARAMETERS

The parameters common to all the file commands are shown in Table 4-1.

Table 4-1: EDLIN File Commands and Parameters

COMMAND PARAMETER	PARAMETER DESCRIPTION	COMMANDS THAT USE PARAMETER
n	Number of lines.	
line	Represents the line number you select. For the line number, you can specify a decimal integer from 1–65534. If the number is larger than the lines in memory, the number indicates the line after the last existing line. Line numbers in a command must be separated from each other by a comma or a space. Lines can also be specified by:	A—Append lines W—Write lines
	. (a period) represents the current line (marked on the screen by an * between the line number and the first character of the line). # (the pound sign) specifies the line after the last line (same as specifying the number larger than the last line number in the file). (cr) (Return) directs EDLIN to use a default line number appropriate to the command.	D—Delete lines N—Edit lines I—Insert lines L—List text R—Replace a string of characters
?	Directs EDLIN to query for a yes or no response to an Ok? prompt.	R—Replace a string of characters S—Search for a string
string	Represents one or more characters to be found or replaced. Each string must end with an ALT-Z or a Return. There should be no spaces between the string parameters unless the space is part of the string. Do not enter spaces between string parameters and the command.	R—Replace a string of characters S—Search for a string in a line

4.2.2 FILE COMMAND CONVENTIONS

Some conventions are common to all the file commands. With any file command you can:

- ▶ Type commands with or without a space between the line number and the command. For example:

  ```
  5I
  ```

  can also be typed:

  ```
  5 I
  ```

- ▶ Reference lines relative to the current line.

  On the screen, the current line has an * between the line number and the first character of the line. You include a plus sign with a line number in a command to indicate the number of lines following the current line. A minus sign with a line number in a command indicates the number of lines before the current line.

  For example, this command directs EDLIN to begin inserting text one line before the current line on the screen:

  ```
  −1I
  ```

- ▶ Issue multiple commands on one command line. Separate multiple commands in a line with a semicolon. For example, if you want to edit a line and then display other lines, enter:

  ```
  10;−2,+1L(cr)
  ```

  This command lets you edit line 10, then display (List) the two lines before line 10 and the line following line 10. The comma after −2 is a syntax element for "line,line"—see the List command in Chapter 4.2.3.

  If you are issuing a multiple command with a string parameter, as in a Search or Replace command, separate the commands with an ALT-Z.

For example, this command:

```
2,10 SThis string^Z − 2; + 2L(cr)
```

searches for the first occurrence of the characters “This string” in lines 2 through 12. It then displays the 2 lines before the line that contains “This string” and the 2 lines after it.

If “This string” cannot be found, the 2 lines before the current line (noted on the screen with the *) and the 2 lines after the current line are displayed.

DESCRIPTIONS OF FILE COMMANDS 4.2.3

This chapter presents detailed descriptions of each EDLIN command for editing files. The descriptions are presented in alphabetical order.

APPEND LINES (A)

```
[n]A(cr)
```

A adds the number of lines, specified by n, from disk to the end of the lines currently in memory.

n represents the number of lines.

Use this command for large files that cannot fit into memory at one time. Edit the lines in memory, write them to disk with the Write command (W), then use the Append command (A) to put the rest of the file in memory.

If you enter a number with the A command, that number of lines is appended to the part of the file currently in memory. If no number is entered, the lines are read into memory until memory is 3/4 full. If memory is already full, no more lines can be appended.

When the last line of the file is in memory, EDLIN displays:

```
End of Input File
*_
```

DELETE LINES (D)

[line][,line]D(cr)

D deletes the lines specified. “Line” can be any of the parameters for a line described in Table 4-1. Without parameters, D deletes the current line.

line
deletes the specified line.

,line or ,#
deletes the current line (marked by the *), the specified line, and all lines between the current line and the specified line.

line, or line,line or line,#
deletes the specified lines and all lines between them.

The line immediately after the deleted lines becomes the current line (marked by the *).

For example, to delete lines 2 and 3 in the NEWDISK.BAT file:

```
1:REM This is a file to check formatted disks
2:REM This is a batch file created by EDLIN
3:REM This file is named NEWDISK.BAT
4:PAUSE Insert formatted disk in drive B
5:DIR B:
6:CHKDSK B:
```

Enter:

2,3 D(cr)

To verify the result, specify the List command (L):

```
*L(cr)
1:REM This is a file to check formatted disks
2:*PAUSE Insert formatted disk in drive B
3:DIR B:
4:CHKDSK B:
```

After lines 2 and 3 are deleted, line 4 becomes line 2 and is marked by the asterisk as the current line.

To delete a single line, such as line 3, enter:

3D(cr)

To see the result, specify the List command (L):

```
*L(cr)
1:REM This is a file to check formatted disks
2:PAUSE Insert formatted disk in drive B
3:*CHKDSK B:
```

When line 3 is deleted, line 4 becomes line 3 and is marked as the current line.

To delete the current line, now marked as line 3, enter:

D(cr)

Specify the L command at the * prompt, and the result is:

```
1:REM This is a file to check formatted disks
2:PAUSE Insert formatted disk in drive B
```

EDIT LINE

[n](cr)
[.](cr)
[#](cr)

n represents the line number.

. indicates the current line.

# indicates the line after the last line in the file.

See Table 4-1 for an explanation of the line number parameters.

EDLIN displays the line number and the text on that line. On the next line, it repeats the line number and displays the asterisk prompt and the cursor. You can start editing the line at the cursor position. You can use any of the line commands described in Chapter 4.3.2 to edit the line.

If you do not want to change the current line (marked by the *), and the cursor is at the beginning of the line, press the Return key to accept the line as it is. If you press Return while the cursor is in the middle of the line, the rest of the line is deleted.

For example, you can select line 1 with the Edit Line command:

```
*1(cr)
```

EDLIN displays the specified line:

```
1:*REM This is a file to check formatted disks
1:*_
```

You can now type in a new line 1, or you can use the line commands to edit the current line (the template, described in Chapter 4.3).

END EDIT (E)

E(cr)

E ends the editing session, saves the edited file on disk, changes the original file's extension to .BAK, and exits to MS-DOS. If the file was newly created during the editing session, no .BAK file is created.

Because the E command takes no parameters, you cannot select the drive on which to save the file. If you did not designate the drive when you invoked EDLIN, such as EDLIN B:NEWDISK, the file is saved on the default drive.

If the disk does not have enough free space for the entire file to be written, the write is aborted and the edited file is lost. Part of the file, however, may be written to the disk.

INSERT LINES OF TEXT (I)

```
[line]I(cr)
```

I inserts line(s) of text immediately before the specified line.

"Line" can be any of the parameters for a line described in Table 4-1.

You must enter the I command to create a new file. The first insert begins with line number 1. EDLIN numbers each new line sequentially each time you press Return. EDLIN remains in insert mode until you enter an ALT-Z or an ALT-C as your last line (ended with a carriage return).

If you insert lines between existing lines in a file, all line numbers following the inserted section are incremented by the number of lines inserted.

If you do not include a line number, the new lines are inserted immediately before the current line. If the specified line number is larger than the last line number or if # is specified as the line number, the inserted lines are appended to the end of the file. In this case, the last line inserted becomes the current line.

For example, to add lines before line 2 in the following file:

```
1:REM This is a file to check formatted disks
2:PAUSE Insert formatted disk in drive B
3:DIR B:
4:CHKDSK B:
```

Enter:

```
2 I(cr)
```

EDLIN displays:

```
2:*_
```

Enter:

```
2:REM This file is a batch file(cr)
3:REM This batch file is called NEWDISK(cr)
4:^Z(cr)
```

You can display the new file with the L command:

```
1:REM This is a file to check formatted disks
2:REM This file is a batch file
3:REM This batch file is called NEWDISK
4:*PAUSE Insert formatted disk in drive B
5:DIR B:
6:CHKDSK B:
```

To insert lines immediately before the current line (line 4 in the previous example), enter:

I(cr)

The screen displays:

```
4:*_
```

If you now enter:

```
4:*REM NEWDISK was created with EDLIN(cr)
5:*^Z(cr)
```

and type the L command to list the file, the result is:

```
1:REM This is a file to check formatted disks
2:REM This file is a batch file
3:REM This batch file is called NEWDISK
4:REM NEWDISK was created with EDLIN
5:*PAUSE Insert formatted disk in drive B
6:DIR B:
7:CHKDSK B:
```

LIST LINES (L)

[line][,line]L(cr)

"Line" can be any of the parameters for a line described in Table 4-1.

L lists the specified range of lines, as follows:

- With no parameters, L lists the 23 lines centered around the current line (marked by an *). The 23 lines are the 11 lines preceding the current line, the current line, and the 11 lines following the current line.
- line,line or line,# lists the specified range of lines.
- ,line lists the range of lines preceding the specified line.
- line, lists a range of lines following the specified line.

Assume the following file exists and is ready to edit:

```
1:REM This is a file to check formatted disks
2:REM It is named NEWDISK.BAT
3:PAUSE Insert formatted disk in drive B
4:DIR B:
5:CHKDSK B:
```

To list the entire file, enter:

```
L(cr)
```

All five lines are displayed, because L without parameters displays up to 23 lines of text.

To list a range of lines preceding line 3, enter:

```
,3L(cr)
```

The result is:

```
1:REM This is a file to check formatted disks
2:REM It is named NEWDISK.BAT
3:PAUSE Insert formatted disk in drive B
```

To list a range of lines following line 3, enter:

```
3,L(cr)
```

The result is:

```
3:PAUSE Insert formatted disk in drive B
4:DIR B:
5:CHKDSK B:
```

QUIT EDIT (Q)

Q(cr)

Q quits the editing session without saving the editing changes, and exits to MS-DOS. The Q command is a quick way to exit an editing session. When the Q command is given, EDLIN displays the message:

```
Abort edit (Y/N)?
```

Press Y to quit the editing session; press N (or any other key except ALT-C) to continue the editing session.

REPLACE STRING (R)

```
[line][,line][?]Rstring1[^Z]string2(cr)
```

R replaces string1 with string2, in the range of lines specified by "line".

"Line" can be any of the parameters for a line described in Table 4-1. If no line is indicated, R operates on the line after the current line (marked by an *) through the end of the file.

- line is the specified line to the end of the file.
- line, line is the first specified line through the second specified line.
- ,line is the line after the current line through the specified line.
- .,line is the current line through the specified line.
- line,. is the range of lines from the specified line through the current line.

? prompts you to OK each string replacement or deletion. When EDLIN encounters a matching string of text, it displays the line and the prompt "O.K.?". If you press Y or Return, EDLIN makes the replacement or deletion and searches for the next matching string. If you press any key other than Y or Return, EDLIN leaves the string as is and searches for the next matching string.

string1 is the string of characters to be replaced. string2 is the string of characters to replace string1. ALT-Z separates string1 from string2.

If you specify only one string of characters, R deletes all occurrences of the string within the specified range of lines.

In this example, the following file exists and is ready for editing:

```
1:REM This is a file to check a formatted disk
2:REM It is named NEWDISK.BAT
3:PAUSE Insert a formatted disk in drive B
4:DIR B:
5:CHKDSK B:
```

To replace all occurrences of “a” with “one” in lines 1 through 3, enter:

1,3Ra^Zone(cr)

The result is:

```
1:*REM This is one file to check one formonetted
  disk
2:It is nonemed NEWDISK.BAT
3:PAUSE Insert one formonetted disk in drive B
```

Some of these replacements produce nonsense words. To avoid unwanted substitutions, add the ? parameter:

1,3?Ra^Zone(cr)

Now each specified line is displayed, and you can confirm the changes. For example:

```
     1:*REM This is one file to check a formatted
       disk
O.K.?y
     1:*REM This is one file to check one
       formatted disk
O.K.?y
     1:*REM This is one file to check one
       formonetted disk
O.K.?n
     2:REM It is nonemed NEWDISK.BAT
O.K.?n
     3:PAUSE Insert one formatted disk in drive B
O.K.?y
```

After making the replacements you want, enter an L command to list the edited file:

```
*L(cr)
1:REM This is one file to check one formatted disk
2:REM It is named NEWDISK.BAT
3:*PAUSE Insert one formatted disk in drive B
4:DIR B:
5:CHKDSK B:
```

SEARCH FOR A STRING (S)

[line][,line][?]Sstring(cr)

S searches for the string in the range of lines specified by line. It displays the first line containing the string. This line becomes the current line if the ? parameter is not specified.

“Line” can be any of the parameters for a line described in Table 4-1. The syntax for “line” is given in the Replace command description. If you do not specify a line, S operates on the line after the current line through the end of the file.

The ? prompts you to OK the displayed line as the current line. When EDLIN encounters a matching string of text, it displays the line and the prompt “O.K.?”. If you press Y or Return, the displayed line becomes the current line. If you press any key other than Y or Return, EDLIN searches for the next matching string.

Assume the following file exists and is ready for editing:

```
1:REM This is a file to check a formatted disk
2:REM It is named NEWDISK.BAT
3:PAUSE Insert a formatted disk in drive B
4:DIR B:
5:CHKDSK B:
```

To search for the first "in" within lines 1 through 3, enter:

```
1,3Sin(cr)
```

The result is:

```
3:PAUSE Insert a formatted disk in drive B
```

To find "DIR" past line 3, specify:

```
3SDIR(cr)
```

The search starts at line 3 and continues to the end of the file. EDLIN displays the line containing the first "DIR":

```
4:DIR B:
```

To search the file for the first occurrence of "B", at the * prompt enter:

```
?SB(cr)
```

The result is:

```
        2:REM It is named NEWDISK.BAT
O.K.?_
```

If you respond Y, the search ends and you return to the * prompt. To continue the search, respond N.

WRITE LINES (W)

[n]W(cr)

W writes lines from memory to disk and makes room in memory for more lines to be appended (with the Append command) from disk to memory.

n represents the number of lines to be written.

If no parameter is given, lines are written to disk until memory is only 1/4 full. If a number is given, the specified number of lines is written to disk. As lines are written, subsequent lines are renumbered beginning with 1.

An A command following the W command appends lines from disk to any line remaining in memory.

4.3 USING LINE COMMANDS

When you create an EDLIN file, you type each line and press Return. Pressing the Return key sends a copy of the line to an allocated portion of memory for input and output known as a buffer. The buffer is a template or pattern of the line you entered as input. You can use the line commands to edit the template (a single line) and then insert the edited line in the file.

Line commands allow you to:

- Recall a line in one keystroke
- Change an erroneous line without retyping it
- Repeat a line and make minor changes without retyping

You complete the template line edit by pressing the Return key. The edited line becomes the new line in the file and is the current template. Each line can be up to 255 characters long (Chapter 6 gives instructions on editing command lines). When you save the file, the template becomes the disk file.

To enter a line command, press one of the numbered function keys at the top of your keyboard. The procedures are described in Chapter 4.3.1.

Editing New Lines

As you insert lines (ended with a carriage return) into a file, EDLIN numbers each line dynamically from 1 through the last line of the file. You can edit the characters in a single line while you are creating an EDLIN file by pressing the ALT key and the letter Z as the last line during input. Then at the * prompt, enter the number of the line you want to edit. Use any of the commands described in Table 4-2 to edit the line.

Editing Existing Lines

You can also edit a single line in an existing file on disk. To use the line commands for an existing file:

1. Get the file from disk as described in Chapter 4.1.2.
2. After the EDLIN prompt (*), enter the List command (L) to display the file.
3. At the * prompt, specify the number of the line you want to edit.
4. Use any of the line commands described in Table 4-2.

SUMMARY OF LINE COMMANDS 4.3.1

Table 4-2 defines the function-key line commands. You enter a line command by pressing one of the numerical function keys in the top row of your keyboard. Detailed descriptions of all the line commands are in Chapter 4.3.2.

Table 4-2: Function-Key Line Commands

COMMAND NAME	FUNCTION KEY	EFFECT
Insert	F1	Enters or exits insert mode (toggle).
New Line	F2	Makes the new line the new template.
Copy Character	F3	Copies one character from the template to the command line.
Skip Character	F4	Skips over (does not copy) one character in the template.
Skip Multiple Characters	F5	Skips over (does not copy) characters in the template up to a given character.
Copy Multiple Characters	F6	Copies all characters up to a specified character from the template to the command line.
Copy Template	F7	Copies all remaining characters in the template to the command line.
Void Input	ALT-X	Cancels the line just entered; keeps the template unchanged.

4.3.2 DESCRIPTIONS OF LINE COMMANDS

This section contains detailed descriptions of the MS-DOS commands for editing text within a single line. You enter line commands by using the top row of numerical function keys on your keyboard. For example, you enter F1 (Insert mode toggle) by pressing Function key 1. For a summary of the line commands, see Table 4-2. These commands are also used for editing command lines at the operating system level.

COPY CHARACTER (F3)

F3 copies one character from the template to the new command line. A space is considered one character.

Use F3 to advance the cursor one column across the line. Insert mode is automatically turned off if it was on.

Assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

Entering F3 copies the first character (R) from the template to the new line:

```
     1:*REM This is a file to check formatted
       disks
F3   1:*R_
```

Each time you enter F3, one more character appears:

```
F3   1:*RE_
F3   1:*REM_
```

COPY MULTIPLE CHARACTERS TO A GIVEN CHARACTER (F6)

F6 copies multiple characters up to a given character from the template.

Press F6 and type the letter where you want the copy to stop. EDLIN copies all the characters up to that character. If the template does not contain the character indicated by the letter key you pressed, nothing is copied. For example, if you type an uppercase letter when the character you want is lowercase, no characters are copied to the line.

F6 automatically turns off insert mode if it is on.

Assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

Entering F6 and a lowercase f copies all the characters up to the character f:

```
        1:*REM This is a file to check formatted
          disks
F6 f    1:*REM This is a_
```

COPY TEMPLATE (F7)

F7 copies all the remaining characters from the template.

Regardless of the cursor position at the time F7 is pressed, EDLIN displays the whole line. The cursor is positioned after the last character on the line.

F7 automatically turns off insert mode if it is on.

Assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

Entering F7 copies all the characters from the template to the new line:

```
         1:*REM This is a file to check formatted
           disks
F7       1:*REM This is a file to check formatted
           disks_
```

INSERT (F1)

F1 turns insert mode on or off. With insert mode on, you can insert characters into the line. With insert mode off, what you type replaces the existing template. The cursor moves to the right as each character is inserted.

In replace mode, all the characters entered replace the characters in the template. Replace mode is the default mode. When you start to edit a line, replace mode is in effect. If you press the Return key while in replace mode, the rest of the template is discarded.

Assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

If you enter F6 (Copy Multiple Characters to a given character) and press the letter key f, the result is:

```
        1:*REM This is a file
F6  f   1:*REM This is a_
```

To insert the word "batch", press F1, type the word batch, and press the space bar:

```
F1 batch(sp)   1:*REM This is a batch_
```

Now press F7 (Copy Template), and the rest of the template is copied to the line:

```
F7        1:*REM This is a batch file to check
            formatted disks_
```

If you press Return instead of entering F7, the rest of the template is deleted. Display the line with the List (L) command:

```
     1:*REM This is a batch_
  *_
```

NEW LINE (F2)

F2 creates a new template. The old line in the template is destroyed.

When you press F2, an at-sign (@) is displayed, and a carriage return/ line feed is inserted in the file. Insert mode is turned off.

Assume the screen shows:

```
    1:*REM This is a file to check formatted
      disks
    1:*REM_
```

To insert the word “directory”, press F6 (Copy Multiple Characters), the letter “k”, a space, the word “directory”, and F7 (Copy Template):

```
                  1:*REM This is a file to check
                    formatted disks
F6 k              1:*REM This is a file to chec
(sp)directory     1:*REM This is a file to check
                    directory
F7                1:*REM This is a file to check
                    directory disks_
```

If you want this line as the new template, enter F2:

```
F2     1:*REM This is a file to check directory
         disks
```

SKIP CHARACTER (F4)

F4 skips over one character in the template. The character is not put in the new line. The cursor does not move; only the template is affected.

For example, assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

Entering F4 skips over the first character, R. If you press F4 two more times, the characters E and M are skipped. To see how much of the line has been skipped, enter F7 (Copy Template). The characters skipped are not put in the new line and the cursor moves beyond the last character of the line:

```
                1:*REM This is a file to check
                  formatted disks
F4 F4 F4
F7
                1:*This is a file to check
                  formatted disks_
```

SKIP MULTIPLE CHARACTERS TO A GIVEN CHARACTER (F5)

F5 skips over multiple characters in the template up to the character you specify. If the template does not contain the specified character, nothing is skipped.

For example, assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

Enter F5 and type a character to skip over all the characters in the template up to the character you typed. The cursor does not move. To see how much of the line has been skipped, enter F7 to copy the template:

```
          1:*REM This is a file to check formatted
            disks
F5 c      1:*_
F7        1:*check formatted disks_
```

VOID INPUT (ALT-X)

ALT-X voids the current input line. The template, however, does not change.

ALT-X also prints a backslash (\), carriage return/line feed, and turns insert mode off if it is on. The cursor moves to the beginning of the line. ALT-X copies the line from the template as it was before ALT-X was pressed.

Assume the screen shows:

```
1:*REM This is a file to check formatted disks
1:*_
```

You can now change line 1. For example, type “REM OLD DISK” and press Return:

```
1:*REM This is a file to check formatted disks
1:*REM OLD DISK(cr)
```

To erase the input you just entered, press ALT-X:

```
          1:*REM This is a file to check formatted
             disks
          1:*REM OLD DISK(cr)
ALT-X     1:*_
```

Now you can create a new line to replace the template. If, however, you enter F7 (Copy Template), the original template is copied to the line:

```
F7    1:*This is a file to check formatted disks_
```

5

USING THE CP/M-86 EMULATOR

The CP/M-86 Emulator (CPM.EXE) runs CP/M-86 programs under MS-DOS. The emulation process is one-way, from CP/M-86 to MS-DOS. The Emulator supports commonly used CP/M-86 functions and provides a CP/M-86 environment for the program. The program is executed as under CP/M-86; recompilation, reassembly, or other translation is unnecessary.

EMULATOR OPERATION 5.1

To run a CP/M-86 program:

1. Transfer the CP/M-86 files onto an MS-DOS diskette, using the MS-DOS command RDCPM.
2. Type:

   cpm

   followed by a space, the CP/M-86 command line exactly as you would type it under CP/M-86, and a Return.

5.2 EMULATOR DESCRIPTION

The Emulator is a modular program. Each module handles a set of the CP/M-86 BDOS functions. These modules provide:

- Emulator initialization
- Program loader emulation
- Memory management emulation
- Run-time interface
- File system interface
- Byte-oriented device interface
- System functions

Emulator Initialization

The Emulator is loaded and resides at high memory. Thus the TPA (Transient Program Area) as seen by the emulated program corresponds to the CP/M-86 environment.

The input command line is formatted according to CP/M-86 CCP (Console Command Process) conventions. The specified program is loaded, the base page is initialized, and control is turned over to the program.

Program Loader

The CP/M-86 program loader is fully emulated. All three memory models are supported.

Memory Management

Memory management functions are fully emulated. Functions implied by CCP artifacts, such as "system reset" and "stay resident," however, are not supported.

The memory functions emulated are:

- Get maximum available memory
- Get maximum memory, absolute
- Get memory region
- Get absolute memory region
- Free memory
- Free all memory

Run-Time Interface

The run-time interface module handles the CP/M-86 BDOS interrupt and dispatches the appropriate routine. The Emulator provides a stack, internal to the Emulator, distinct from the program's init stack.

File System Interface

The standard CP/M-86 FCB (File Control Block) is used for all emulations, and the CP/M-86 requests are mapped into corresponding MS-DOS requests. All user-accessible FCB data is maintained by the Emulator identically to CP/M-86.

The system FCB data space is reserved, but the information in it is not valid. Because of the lack of sparse files under MS-DOS, programs that use certain random-write techniques may require more disk space.

Directory maintenance is functionally compatible. Maintenance functions provide a directory block with the index and all fields except the name and extent are cleared to zero. The remainder of the directory block contains inactive entries.

Two CP/M-86 functions cannot be supported because they use CP/M-86 directory information. These functions are "set file attribute" and search with "?" drive.

The Emulator supports the following file system interface functions:

- Open/Make (Function 15)
- Close (Function 16)
- Read sequential (Function 20)
- Write sequential (Function 21)
- Read random (Function 33)
- Write random (Function 34)
- Set random record (Function 36)
- Write random with zero fill (Function 40)
- Search first/next—with the exception noted (Functions 17 and 18)
- Chain to program (Function 47)
- Delete (Function 19)
- Rename (Function 23)
- File size (Function 35)

Byte-Oriented Device Interface

Byte-oriented logical devices are fully supported, including redirection via the I/O byte.

The following functions are supported:

- Console input (Function 1)
- Console output (Function 2)
- Reader input (Function 3)
- Punch output (Function 4)

- List output (Function 5)
- Direct console I/O (Function 6)
- Print string (Function 9)
- Read console buffer (Function 10)
- Get console status (Function 11)

System Functions

These system functions are supported:

- System reset—terminates emulation; does not allow program to remain resident (Function 0)
- Return version number (Function 12)
- Reset disk system—MS-DOS disk reset (Function 13)
- Select disk—uses MS-DOS function (Function 14)
- Return login vector—reflects all drives known to MS-DOS (Function 24)
- Return current disk—CP/M-86 equivalent (Function 25)
- Get DMA address—CP/M-86 equivalent (Function 47)
- Write-protect disk—marks disk as write-protected within the Emulator (Function 28)
- Get address—R/O vector; returns address of the Emulator R/O vector (Function 29)
- Reset drive—resets Emulator R/O vector (Function 37)
- Get/Set DMA base—CP/M-86 equivalent (Functions 51 and 52)
- Get SYSDAT address where the equivalent of SYSDAT information is found within the Emulator, with the same offset values as in CP/M-86 (Function 49)

These direct BIOS calls are supported:

- INIT terminates emulation (BIOS subroutines)
- WBOOT terminates emulation (BIOS subroutines)
- CONST, CONIN, CONOUT (BIOS subroutines)
- LIST, LISTST, PUNCH, READER (BIOS subroutines)
- HOME, SELDSK (BIOS subroutines)
- GET/SET IO BYTE (Functions 7 and 8)

These system functions are NOT supported:

- Get address—allocate (Function 27)
- Get/Set user code (Function 32)

These direct BIOS calls are NOT supported:

- SETTRK, SETSEC, SETDMA (BIOS subroutines)
- READ, WRITE, SECTRAN (BIOS subroutines)
- SETDMA, SETDMAB, GETSEGB (BIOS subroutines)

If you attempt to use any system functions that are not supported by the CP/M-86 Emulator, an error message is displayed and the Emulator terminates.

6

MS-DOS COMMAND EDITING, FILTERING, AND PIPING

This chapter describes how to enhance your control of MS-DOS commands. Chapter 6.1 describes two methods of editing command lines—using Alternate-character editing functions during input, and using function-key commands to redisplay and edit parts of the previous command line (the template).

Chapters 6.2 and 6.3 demonstrate two major processes—filtering and piping of data. These processes enable you to reorganize and output data between files or devices, and to use the output from one command as input to another command in the same command line.

COMMAND LINE EDITING 6.1

There are two ways to edit MS-DOS command lines:

- While you enter a command line from the keyboard, you can use Alternate-character functions listed in Table 6-1 to edit the line. For example, you can use ALT-P to send the command and its output to the printer.
- After you finish entering a command, you can use function-key commands to edit that command line. The function-key commands are described in Chapter 4.3. This chapter gives examples of how to use the Function keys to edit the previous command line.

Table 6-1: Command-Line Alternate-Character Editing

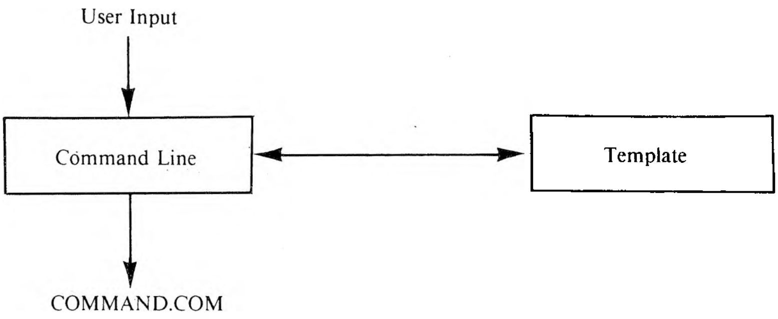
ALT-KEY SEQUENCE	FUNCTION
^C	Interrupts current command.
^H or Backspace	Removes last character from command line and erases character from screen.
^J	Inserts physical end-of-line, but does not empty command line. Uses linefeed to extend the current logical line beyond the physical limits of one screen line.
^N	Cancels echoing of output to the printer.
^P	Echoes console output to the printer.
^S	Suspends display of output to screen. Console (and printer, if output is being echoed to the printer) remains frozen until any key is pressed.
^X	Cancels the current line, empties the command line, and outputs a backslash (\), carriage return, and linefeed. The template is not affected.
^Z	Terminates text entry (end-of-file marker).

During command input, the following processes occur:

- The text you enter is held in a portion of memory called the command line buffer until you press the Return key.
- Pressing Return sends the contents of the command line buffer to COMMAND.COM for processing.
- At the same time, the command line is copied to the **template**, another buffer closely related to the command line buffer.
- When the MS-DOS command prompt reappears, the command line buffer is empty, ready for your input. The template, however, retains the preceding command line.

The relationship between the command line and the template is shown in Figure 6-1.

Figure 6-1: The Command Line and the Template



The template is like a short-term memory that "remembers" the last command line you entered. If you want to repeat your last command, you can redisplay the template instead of retyping the line. If your last command contained an error such as a misspelled word, you can edit the template with a minimum of retyping.

You use the numerical function keys on the top row of your keyboard to edit the command-line template. The function-key commands are the same as those used for editing the text-line template in EDLIN (see Chapter 4). The function-key commands are listed in Table 4-2. With them you can:

- Re-enter the previous command (the template) with one keystroke (F7).
- Edit and retry a command line that failed, by re-entering portions of the previous command line and typing in other, new portions.
- Enter a new command line similar to the previous command without retyping the entire line.

For example, the following command fails because COPY is misspelled:

copu memo b:letter(cr)

In response to this command, MS-DOS displays:

```
Bad command or filename
```

You can change COPU to COPY and re-enter the rest of the command without retyping the entire line:

1. Press F6 (Copy Multiple Characters) and the letter u. This copies the characters in the template up to the letter u into the new command line.

```
cop_
```

6

2. Press F4 (Skip Character). The next character in the template (u) is skipped.
3. Press F1 (Insert mode toggle) and type the letter y to complete "copy".
4. Press F7 (Copy Template). The rest of the characters from the template are copied to the new command line:

```
copy memo b:letter_
```

5. Press the Return key.

If you run a utility that uses a large portion of memory (such as SEARCH), the command-line template may be temporarily overwritten. For example, after running a SEARCH command you might press F7 to redisplay the previous command, but the template does not appear. In this situation, use the HISTORY internal command. You can restore the template by entering HI 1(cr), and then redisplay the template with F7, or edit the template with the other function keys. You can also type HISTORY(cr) to redisplay the entire command history, and select the command you want to edit and re-enter.

FILTERING DATA 6.2

MS-DOS can sort through the data in a file and arrange it in a different order. This process is called filtering. These commands filter data:

- FIND searches for a constant string of alphanumeric characters.
- SORT rearranges the data in the order that you specify in the command.
- MORE displays one screen of information at a time; press the Return key to display another screen of information.

You can use these commands with any MS-DOS files, such as MS-BASIC files. The examples given here use the following data file (PHN.LST) containing four records of names and phone numbers:

```
Washington, George 301 328-4370
Kennedy, John 617 736-2867
Lee, Robert 703 532-3370
Lincoln, Abraham 309 473-2800
```

You can filter the data by area code with the FIND and SORT commands. FIND searches for a numeric string of characters (the 309 area code):

```
find "309" b:phnlst.dat(cr)
```

MS-DOS displays:

```
----------b:phnlst.dat
Lincoln, Abraham 309 473-2800
```

The SORT command can arrange data in alphabetical order or the reverse, Z through A. The following command tells MS-DOS to arrange the PHN.LST data records in alphabetical order and to output the sorted data to the screen:

```
sort < b:phnlst.dat > con(cr)
```

The < and > symbols redirect input and output. The < in the previous command tells MS-DOS that the file PHNLST.DAT is the input for the SORT command. The > symbol tells MS-DOS to send the output (the sorted data) to the display unit (CON).

The preceding command displays this sorted data:

```
Kennedy, John  617 736-2867
Lee, Robert 703 532-3370
Lincoln, Abraham 309 473-2800
Washington, George 301 328-4370
```

You can also tell MS-DOS to create a new file and send sorted data to the new file. For example, if you name the file ALPHA.NUM in place of CON in the previous example, MS-DOS sends the alphabetized file records to ALPHA.NUM:

```
sort < b:phnlst.dat > b:alpha.num(cr)
```

PIPING DATA 6.3

To pipe data means to enter more than one command on a command line, separating commands with a vertical bar (|), the pipe separator. By piping commands you can use the output from one command as the input for another command. You can combine piping with filtering to rearrange existing files into new sorted files.

You can:

- ▶ Filter data from an existing data file and pipe it to a new file or other output source, such as the screen (CON) or printer (LST).
- ▶ Filter columnar data, such as a directory, by column number (the screen column where file size, date, or time is listed).
- ▶ Use batch processing to make filtering and piping data even easier.

FILTERING AND PIPING DATA FILES 6.3.1

Using the sample file shown in Chapter 6.2, you can filter the data by area code. You can then pipe the filtered area codes to the AREA.COD file by specifying:

```
find "301" b:phnlst.dat | sort > b:area.cod(cr)
```

The | pipe symbol tells MS-DOS to take the output from the command on the left side of the bar and use it as input for the command on the right side of the bar. The > symbol in the command sends all the records containing the 301 area code to a new file named AREA.COD.

The following command uses the >> symbol to append all the records containing the 309 area code to the end of the AREA.COD file:

```
find "309" b:phnlst.dat | sort >> b:area.cod(cr)
```

To verify that the sorted records are in AREA.COD, specify:

```
type b:area.cod(cr)
```

If you enter:

```
type b:phnlst.dat | sort > lst(cr)
```

MS-DOS gets the output of the TYPE B:PHNLST.DAT command and uses it as input for the SORT command. This SORT command sorts the file alphabetically and sends the sorted data to the printer.

6.3.2 FILTERING AND PIPING COLUMNAR DATA

If you type:

```
dir | sort /+16 | more(cr)
```

MS-DOS pipes the output of the DIR command as input to the SORT filter command. The sorting starts in column 16 of the directory display (file size). MS-DOS sends the directory, sorted by file size, to the display unit. MORE displays the directory one screen at a time with the message --MORE-- at the bottom of the screen. You can display another screenful of the directory by pressing Return.

FILTERING AND PIPING BATCH FILES 6.3.3

You can use filtering and piping in batch files. The next example creates a batch file PHONES.BAT that searches for a value (the %1 parameter) in the PHNLST.DAT file, sorts the records containing that value, and displays those records onscreen:

```
copy con phones.bat(cr)
find "%1" phnlst.dat | sort  > (cr)
^Z(cr)
```

You can use PHONES.BAT to search for any area code you want. Type PHONES and the area code you want sorted from the PHNLST file, such as:

```
phones 301
```

All the records from PHNLST.DAT with area code 301 are displayed.

7

MS-DOS COMMANDS

This chapter presents an overview of MS-DOS commands. Chapter 7.1 lists and describes the two types of MS-DOS commands (internal and external) as well as the batch commands (internal commands used in batch files). Chapter 7.2 describes the notation used in the syntax of commands in this manual.

Chapter 7.3 describes the special commands you use when you alter the system configuration contained in the CONFIG.SYS file. Finally, Chapter 7.4 presents the syntax of all the MS-DOS commands (excluding EDLIN, the CP/M-86 Emulator, and the system configuration commands). Commands are listed alphabetically, and examples of usage are included for most commands.

OVERVIEW OF COMMANDS 7.1

You use the MS-DOS commands to set up and manage your files and directories, the diskettes that contain your files, and the devices attached to your computer. This overview describes some of the MS-DOS commands and what you can do with them. For a complete listing of the commands, see Chapter 7.4.

You use the MS-DOS file commands to:

- ▶ Copy, delete, rename, display, or verify files (COPY, DEL, REN, TYPE, VERIFY)
- ▶ Copy the operating system files from one diskette or hard disk volume to another (SYS)
- ▶ Change the file format or rearrange the file contents (SORT, FIND)

- Queue a list of files for printing (PRINT)
- Transfer files from a CP/M-86 diskette to an MS-DOS diskette (RDCPM)
- Create batch files and control their execution (COPY CON, ECHO, SHIFT, and so on)

You use the MS-DOS directory commands to:

- Display the directory of a diskette (DIR)
- Create or remove directories (MKDIR, RMDIR)
- Change from one subdirectory to another, or check your position in the directory hierarchy (CHDIR)

You use the MS-DOS disk commands to:

- Prepare blank diskettes for MS-DOS use (FORMAT)
- Copy the contents of one diskette to another (DISKCOPY)
- Check the amount of space on a diskette, and check a directory for errors in file storage (CHKDSK)
- Recover data from a bad sector on a disk (RECOVER)

The MS-DOS device commands (CLS, CTTY, CLST, CAUX, and PORTSET) control the CRT, your printer, and your disk drives. These commands display information about your diskettes and disk drives. They can also change devices, such as changing output from the screen to the printer.

The system configuration commands (such as BREAK, BUFFERS, and DEVICE) enable you to control parts of the operating environment. All these commands can be used in the CONFIG.SYS file, which sets up the environment when MS-DOS loads into memory. One of the commands (BREAK) can also be used interactively. See Chapter 7.3 for a description of the configuration commands.

Other MS-DOS commands affect the way the operating system works. For example, there are commands to set the MS-DOS program interrupt (ALT-C) on or off, set up constants in the operating environment, and change the system command prompt.

INTERNAL AND EXTERNAL COMMANDS 7.1.1

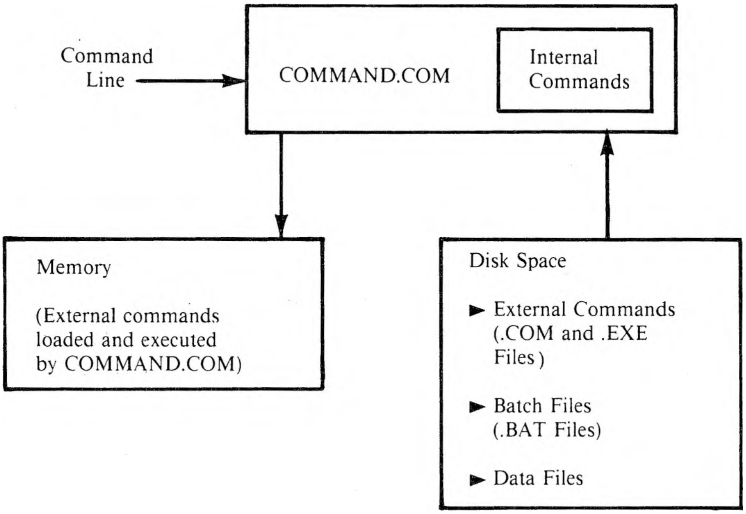
There are two basic types of MS-DOS commands: internal and external. Internal commands are a set of commands that exist within MS-DOS. External commands are stored on disk as separate files (with extension .COM or .EXE). External commands are also known as utility programs.

A program called the command processor interprets and processes all the MS-DOS commands, both internal and external. The command processor is in the system file COMMAND.COM on the MS-DOS system diskette. When you load MS-DOS, COMMAND.COM (or any command processor file named in the CONFIG.SYS file) is automatically loaded into memory.

The internal commands are part of the COMMAND.COM file. External commands, on the other hand, must be individually loaded into memory by COMMAND.COM each time you use that command. The internal commands remain in memory as long as COMMAND.COM is in memory.

The MS-DOS command interface is illustrated in Figure 7-1.

Figure 7-1: MS-DOS Command Interface


Command Line
COMMAND.COM
Internal Commands
Memory
(External commands loaded and executed by COMMAND.COM)
Disk Space
► External Commands (.COM and .EXE Files)
► Batch Files (.BAT Files)
► Data Files

Internal Commands

The internal commands are:

- BREAK
- CAUX
- CHDIR
- CLS
- CLST
- COPY
- CTTY
- DATE
- DEL
- DIR
- ECHO
- EXIT
- FOR–IN–DO
- HISTORY
- MKDIR
- PATH
- PROMPT
- REN
- RMDIR
- SET
- TIME
- TOD
- TYPE
- VOL
- VERIFY
- VER

These internal commands are not displayed in the directory. When you enter an internal command, it executes immediately whenever COMMAND.COM is in memory.

Some internal commands can use the pathing system described in Chapter 3 to obtain files from the MS-DOS subdirectory system. These commands are COPY, DEL, DIR, CHDIR, MKDIR, TYPE, and PATH.

External Commands

The external commands are:

CHKDSK
COMMAND
CONCAT
CPM
DISKCOPY
EDLIN
FILCOM
FGREP
FIND
FORMAT
LS
MODCON
MORE
MV
PORTSET
PRINT
RDCPM
RECOVER
SDCOPY
SEARCH
SORT
SYS
TAIL
WC

The command descriptions in Chapter 7.4 include all the external commands except CPM (see Chapter 5), EDLIN (see Chapter 4), and hard disk utilities such as AUTOSET (see the *Hard Disk User's Guide*).

Like all programs, the external commands are stored in command files on disk. You can identify them by the extension .COM or .EXE. You should keep all the MS-DOS commands in one directory separate from all other directories. Before you issue an external command, you must tell MS-DOS where the command file is located. You can use either the CHDIR command or the PATH command. You can then list the directory with the DIR command to verify that the command file is there. When you specify the command, MS-DOS reads the command from that directory. If the diskette in the drive does not contain the command file, MS-DOS displays:

```
Bad command or filename
```

7.1.2 BATCH COMMANDS

Batch commands are used only in batch files (see Chapter 2). Because batch commands are internal commands, COMMAND.COM must be loaded in main memory whenever you run a batch file that uses any of these commands:

ECHO
FOR-IN-DO
GOTO
IF-EXIST
PAUSE
REM
SHIFT

7.2 COMMAND SYNTAX AND NOTATION

Descriptions of all the MS-DOS commands are given in alphabetical order in Chapter 7.4. The syntax or format for each command is given first in the descriptions. Then the elements of the command are defined, and examples are presented to show how to use the commands. In examples, the entire command line is lowercase except for elements that require uppercase, such as proper nouns or command switches that are not valid in lowercase.

The notation used for command syntax is:

COMMAND NAMES
are shown in uppercase. You can, however, enter command names in upper- or lowercase.

LITERAL PARAMETERS
are shown in uppercase. These are parts of a command that must be entered literally if you want to include them. You can enter the parameters in upper- or lowercase, except as noted in the command descriptions.

variable parameters

are shown in lowercase. You can enter command parameters in upper- or lowercase.

[ ]

enclose optional parameters.

...

is an ellipsis indicating that you can repeat the preceding parameter.

filename.ext, drivename

represent the type of entry you must make. For example, when you see the word filename, you must enter the name of a specific file (with or without wild-card characters).

path

is the full or relative path through the subdirectory hierarchy to the subdirectory or file named in the command. In this chapter, the syntax for paths is:

[drivename:][[\]directoryname[\directoryname...]]

If the file or subdirectory you want is in the current directory on the default drive, you can omit the directory and drive names. The first \ in a path always refers to the root directory. Each additional \directoryname refers to a lower level subdirectory. Path syntax is fully described in Chapter 3.

/

indicates that the parameter (such as /C) is a switch. A switch causes MS-DOS to process the command as altered by the specified switch. When you enter a switch in the command line, separate it from commands and other switches with a space.

|

indicates that data is to be piped (see Chapter 6). The output generated by the command on the left side of the bar is sent as input to the command on the right side of the bar. Separate the | from other command elements with a space.

<

indicates that the input to the preceding command is to come from a source other than the keyboard. Enter a space before and after the <.

>

indicates that the output from the preceding command is to be sent to a file or device (such as the printer). Enter a space before and after the >.

>>

indicates that the output from the command is to be sent to the end of the file named in the command. Separate the >> from other command elements with a space, but do not type a space between the >s.

(cr)

indicates the Return key.

(sp)

indicates the Space bar.

^

7

indicates the ALT (Alternate) key. An ALT-key sequence (pressing ALT and another key) is shown as ALT- and the letter or number, such as ALT-C.

You must include all punctuation (commas, colons, or question marks) and the symbols \, |, <, and >> where shown, except for [ ], which indicate optional parameters, and the ellipsis. The ellipsis indicates that a parameter can be repeated.

NOTE: Because the symbols \, |, <, and > are literal parts of MS-DOS commands, they cannot be used in a filename. If you have any existing files that have \, |, <, or > in their filenames, you must rename the files before you can use them with MS-DOS Version 2.1 or later. Use a previous version of MS-DOS to rename the files.

SYSTEM CONFIGURATION COMMANDS 7.3

This section describes the commands that you can use in the CONFIG.SYS file. You do not need to change this file unless you have a complex program or you want to change the operating environment to fit your needs.

When MS-DOS loads into memory, it checks the root directory for a system configuration file named CONFIG.SYS. This file contains system initialization routines that tell MS-DOS:

- The number of buffers to allocate for input and output data (BUFFERS)
- The device drivers to be installed (DEVICE)
- The number of files that can be open at the same time (FILES)
- The name of the command processor to be loaded (SHELL)
- When to allow ALT-C to interrupt operations (BREAK)
- The symbol that indicates a command parameter (SWITCHAR)
- Whether a reference to a device filename requires a path name (AVAILDEV)
- The standards associated with particular countries (COUNTRY)

To change the CONFIG.SYS file, use any text editor that operates under MS-DOS, such as EDLIN. You can also create a new CONFIG.SYS file with the COPY CON command. The changes you make to the file become effective the next time you load MS-DOS.

7.3.1 DESCRIPTIONS OF CONFIGURATION COMMANDS

This chapter presents the syntax for the system configuration commands that can appear in CONFIG.SYS.

AVAILDEV

```
AVAILDEV = TRUE(cr)
```

or

```
AVAILDEV = FALSE(cr)
```

The default is AVAILDEV = TRUE. This command controls whether device names (such as CON and LST) refer only to devices or can be used as filenames. (The device files are described in Chapter 2.2.) The default setting means that MS-DOS reads both the following names as the device name LST (LST is used to represent any device name):

```
\DEV\LST
LST
```

If AVAILDEV = FALSE, MS-DOS reads only \DEV\LST as the device name LST. This means that LST by itself refers to a file in the current directory with the same name as the device LST (the file may or may not exist).

BREAK

BREAK = ON(cr)

or

BREAK = OFF(cr)

The default, BREAK = OFF, tells MS-DOS to check for an ALT-C interrupt only during screen, keyboard, printer, or Asynchronous Communication Adapter operations. BREAK = ON tells MS-DOS to check for an ALT-C interrupt during all operations, including whenever MS-DOS is processing a program. You can also change the setting by issuing the interactive BREAK command, described in Chapter 7.4. If you set BREAK = ON, you can abort programs by typing an ALT-C.

BUFFERS

BUFFERS = xx(cr)

where xx is a number between 1 and 99. A buffer is the amount of memory allocated for the reading of data from disk and the writing of data to disk. The default value is 10. If your application does sequential access to the data on disk, you need a small number of buffers. If you are using a database with random access, 10 to 20 buffers is adequate. A larger number of buffers slows system performance.

COUNTRY

```
COUNTRY = nn(cr)
```

where nn is a number between 1 and 99. The COUNTRY command selects configuration standards according to the given country code. Some of the standards affected are the currency separator and the format for the DATE command (such as listing the month first or listing the day first).

The numbers you can use with COUNTRY are the country's telephone code, such as:

1	USA
33	France
44	United Kingdom
49	Germany
81	Japan

DEVICE

```
DEVICE = [path]filename.ext[parameters](cr)
```

This command installs the device driver from the specified file. The default is the device drivers for the standard screen, keyboard, printer, auxiliary device, floppy disk drive, and hard (fixed) disk. DEVICE lets you include device drivers that you may have written. For each device driver, you must include a DEVICE command in the CONFIG.SYS file.

path
: is the path to the device driver file, including drive and subdirectory names where necessary.

filename.ext
is the name of the file containing the device driver.

parameters
is a list of transmission settings for a serial device driver. Do not include these settings for parallel device drivers, which are standardized. The transmission settings include baud rate, parity, stop bits, and bits per character. You can use the values listed for the PORTSET utility settings.

The default device drivers for your input/output ports are:

- SERIALA—serial port A (also known as TTY). If you have only one serial port, it is SERIALA.
- SERIALB—serial port B.
- PARALLEL—parallel port.

You can change the serial port transmission settings (such as baud rate and parity) with the PORTSET utility.

FILES

```
FILES = xx(cr)
```

where xx is a number between 1 and 99. The default value is 8. This command specifies the number of files that can be opened concurrently. For each additional file above 8, the size of the resident portion of MS-DOS increases by 39 bytes per file. The amount of memory available for the application program is reduced by the number of bytes allocated to the number of opened files. **NOTE:** Specifying a value less than 8 may prevent some system operations, such as filtering and piping (see Chapter 6.2).

SHELL

SHELL = [path]filename.ext[comspec path][/P](cr)

This command names the command processor to be loaded when MS-DOS starts. The default is COMMAND.COM. You should not change COMMAND.COM unless you are a systems programmer who has created a command processor for your own applications. If you replace COMMAND.COM with your own command processor, you must provide for the hex 22, 23, and 24 interrupts, the internal commands, the batch processor, and the EXEC function call.

path
: is the path through the directory hierarchy to the file containing the command processor. Include the drive name and subdirectory names where necessary.

filename.ext
: is the name of the file containing the command processor.

/P
: tells the command processor that it is the first program to be run by MS-DOS. The P must be preceded by the correct symbol for command parameters (see SWITCHAR).

comspec path
: is a path through the directory system. This path is appended to the name of the command processor (COMMAND.COM) and SET into the MS-DOS environment as "COMSPEC". COMSPEC is then used by the operating system to locate the command processor. You can display the current COMSPEC path with the SET(cr) command. If omitted from the SHELL command in CONFIG.SYS, the COMSPEC path is the root directory on the default drive.

The following example uses COMMAND within a SHELL command line in a CONFIG.SYS file:

```
shell = a:\bin\command.com a:\bin /p(cr)
```

This SHELL command tells MS-DOS that the command processor is the file COMMAND.COM on drive A in subdirectory BIN. The A:\BIN /P following COMMAND.COM are parameters for the COMMAND portion of the line. These parameters tell MS-DOS that COMMAND.COM is to be loaded permanently into memory from subdirectory BIN on drive A.

SWITCHAR

```
SWITCHAR = x(cr)
```

where x can be any alphanumeric character. This command specifies the symbol that indicates command switches to MS-DOS. The default switch character is /. For example, to add the V switch to the COPY command, you would type /V.

By default, the MS-DOS path-separator character is the backslash (\), while the slash (/) is the switch character. If you set a different switch character with the SWITCHAR command, the path separator becomes the slash (/).

For example, if CONFIG.SYS contains this command:

```
switchar = -
```

you must use the hyphen to introduce command switches (such as -V), and you must use the slash to separate parts of path names (such as /MGR1/TOM).

7.3.2 SAMPLE CONFIGURATION FILE

A typical configuration file might look like this:

```
break = on
buffers = 10
device = \porta.exe, baud 1200, parity none, stop 2, bits 8
files = 10
switchar = /
shell = a:\bin\command.com a:\bin\insure /p
```

This CONFIG.SYS file does the following:

- Sets ALT-C interrupt on
- Adds 10 additional sector buffers to the system list
- Installs the device driver PORTA.EXE from the root directory on the default drive and gives values for the transmission settings
- Allows 10 files to be open at once
- Defines the syntax notation for command parameters as a slash
- Begins execution of the shell (the top-level command processor) from COMMAND.COM in the subdirectory BIN on drive A

The SHELL command in this file invokes COMMAND.COM and includes the /P command parameter, indicating COMMAND.COM is loaded first (see COMMAND in Chapter 7.4). COMSPEC is SET to A:\BIN\INSURE.

COMMAND DESCRIPTIONS 7.4

This chapter presents descriptions of the syntax and examples of usage for the MS-DOS commands.

ALT-C INTERRUPT (BREAK)

BREAK [ON](cr)

or

BREAK [OFF](cr)

BREAK is an internal command that lets you use ALT-C as a system interrupt (BREAK ON) or without the interrupt effect (BREAK OFF) when MS-DOS is processing a program. When BREAK is ON, typing ALT-C interrupts the current operation and redisplays the system prompt. Enter BREAK without parameters to check the status of the interrupt function. MS-DOS responds:

```
BREAK is xx
```

where xx is “on” or “off.”

ON tells MS-DOS to allow an ALT-C interrupt at any time during the processing of a program.

OFF tells MS-DOS to allow an ALT-C interrupt only during the operation of the keyboard, printer, or an auxiliary device. The default is OFF.

CHANGE DIRECTORY (CHDIR)

CHDIR [..][/A][path](cr)

CHDIR is an internal command that changes your position in the directory hierarchy or displays the name of the current subdirectory. You can abbreviate the command name to CD. CHDIR is also described in Chapter 3.

You use this command to:

- ▶ Move to a subdirectory from any other directory.
- ▶ Move to a directory level above this one.
- ▶ Return from a subdirectory to the root directory.
- ▶ Display the name of the directory you are using.
- ▶ Display the names of the working directories on all drives in the system.

CHDIR without parameters displays the directory you are currently using. For example, if you are in a directory called CMD in the default drive and specify:

chdir(cr)

MS-DOS displays:

```
A:\cmd
```

..

Moves you to the directory above the one you are currently using. You can move up two directory levels by specifying:

cd ..\..(cr)

/A

displays the name of the working (default) subdirectory on each drive. This CD command is useful when you are copying files from one drive to another. Entering CD /A tells you the paths to the default directories. When you type your COPY command, you can omit the paths for files you are copying from or to the default directories.

path

is the path through the directory system to the subdirectory level you want. Each directory name is the name of a directory created with the MKDIR command. For example, the following command looks for the EDFILES subdirectory three levels below the root directory:

```
chdir \mgr1.nam\cmd\edfiles(cr)
```

Entering \ without a directory name refers to the directory at the top of the directory hierarchy (the root directory). This CHDIR command returns you to the root directory:

```
chdir \(cr)
```

If you specify a directory name with no \ parameter, MS-DOS looks for the directory name in the current directory. In the previous example, if you are using the CMD subdirectory and you want to access EDFILES, you can enter:

```
chdir edfiles(cr)
```

instead of the path, \MGR1.NAM\CMD\EDFILES. MS-DOS looks in the current directory, CMD, for the EDFILES subdirectory.

CHECK DISK (CHKDSK)

CHKDSK [drivename:][filename.ext][> filename.ext] [/F] [/V](cr)

CHKDSK.COM is an external command. CHKDSK scans the directory hierarchy of the diskette in the specified drive for consistency. If inconsistencies are found, error messages and a status report are displayed.

If you enter CHKDSK without parameters and if the diskette in the default drive contains errors, MS-DOS displays:

```
    Errors found, F parameter not specified.
    Corrections will not be written to disk.

    n lost clusters found in n chains
    Convert lost chains to files (y/n)?
```

where the first *n* is a number.

If you respond with Y, MS-DOS converts the lost chains to files. If you respond with N, MS-DOS displays the amount of memory that would be freed.

drivename
  is the name of the drive containing the disk that you want to check.

filename.ext
  is the name of the file you want to check. Use wild-card characters to refer to more than one file. For example, *.* reports extents (continuous space on disk reserved for a file) for all files in the current directory after a normal consistency check is made. You can copy and rename files with many extents to restore them to a contiguous state.

> filename.ext
redirects the output from the CHKDSK command to the specified file.

/F
automatically fixes the error when:

- ▶ There is an allocation error. The size is adjusted.
- ▶ There is a disk error during the reading of the File Allocation Table (FAT).
- ▶ There is a disk error during the writing of the File Allocation Table (FAT).
- ▶ The file for the specified "filename" contains noncontiguous blocks.
- ▶ The first cluster (file attributes, password, allocation, extent, and statistical information) number is invalid and the file entry is truncated.
- ▶ The command to Change Directories for the specified filename cannot be done because a subdirectory cannot be processed.

You must correct the error when the CHKDSK command encounters:

- ▶ An incorrect MS-DOS version. (You cannot run this version of CHKDSK on MS-DOS versions lower than 2.1.)
- ▶ Insufficient memory. Processing cannot continue. There is not enough memory in the processor unit for CHKDSK to process the disk.
- ▶ An invalid current directory. Processing cannot continue. Restart the system and reissue the CHKDSK command.
- ▶ A Change Directory (CHDIR) command that cannot be made to the root (main) directory. Processing cannot continue. The disk is a bad disk. Try restarting MS-DOS and issue the RECOVER command.
- ▶ Cross-linked files. The same data block is allocated to more than one file. Use the COPY command to copy the files you want to keep. Then delete the files that are cross-linked.

- A non-MS-DOS disk. You are prompted to respond yes (Y) or no (N) if you want CHKDSK to continue processing.
- Insufficient room in the root (main) directory. You are prompted to erase the files in the root directory and reissue the CHKDSK command.
- An unrecoverable error in the directory. You are prompted to respond yes (Y) or no (N) to convert the directory to a file. If you respond Y, you can fix the directory yourself or delete it.
- Lost allocation units. You can either free the "orphan" allocation units or recover them. A CHKDSK recover operation creates "orphan files" in the root directory of the disk. If there is not enough room to create all of the orphan files, run CHKDSK again.

/V

displays a trace (a record of the series of events as they occur) of the files and directories during the CHKDSK processing.

The following example checks the diskette in drive B for consistency and issues a status report:

```
A>chkdsk b:(cr)

  620544 bytes total disk space
   63488 bytes in 2 hidden files
    2048 bytes in 1 directories
  237568 bytes in 16 user files
  317440 bytes available on disk

  472976 bytes total memory
  458576 bytes free

A>_
```

CLEAR SCREEN (CLS)

CLS(cr)

CLS is an internal command that clears the CRT screen and redisplays the system prompt. This command tells MS-DOS to send the VT-52 Escape sequence for clearing the screen to the CRT. You use this command when you want to start over.

RELOAD COMMAND.COM (COMMAND)

COMMAND [path][device] [/D] [/P] [/C string](cr)

COMMAND is an external command that invokes the command processor COMMAND.COM. When you enter COMMAND, MS-DOS reloads the COMMAND.COM file. You can use COMMAND interactively (by entering the command from the keyboard at the system prompt), and you can use COMMAND in batch files to reload COMMAND.COM.

path
  is the path through the directory system to the file containing the command processor. Path includes the drive name and subdirectory names where necessary. No path is necessary if the file resides in the root directory on the default drive.

device
  is the name of the TTY device. If you do not specify a device, MS-DOS uses CON (by default, the keyboard and screen).

/D
  omits prompts for date and time when COMMAND.COM reloads. If you do not specify /D, COMMAND.COM prompts you for date and time.

/P
: places COMMAND.COM permanently in memory (until you reload the operating system). Also tells COMMAND.COM to search for an AUTOEXEC.BAT file and to prompt for the date and time as it loads into memory. If /P is omitted, COMMAND.COM is a transient command.

/C string
: submits the specified string for the command processor to execute after it loads into memory. COMMAND.COM executes this single command string and then exits. The /C switch overrules the /P switch. If present, the /C switch must be the last switch. All text following /C is passed to COMMAND.COM.

CONCATENATE AND OUTPUT FILES (CONCAT)

CONCAT [files or directories](cr)

CONCAT.EXE is an external command (utility) that concatenates (combines) a list of files and displays their contents. The concatenated files are sent to the standard output (the console) but are not saved on disk unless you include the > redirection symbol to send the output to a disk file, as shown in the first example that follows.

files or directories
: is a list of subdirectory names and/or filenames to be operated on. If you do not specify files or directories, input is taken from the keyboard (or redirected input).

For example:

```
concat tod.asm atc.asm > backup(cr)
```

concatenates the TOD.ASM and ATC.ASM files in the current directory. With output redirection, the combined files are sent to the file BACKUP.

You can also use CONCAT to create files. This command:

```
concat > junk(cr)
```

takes any following keyboard input (up to ALT-Z Return) and outputs it to the disk file JUNK.

The following command displays the contents of all files in the \TEMP subdirectory on the default drive and all files with extension .PLM in the current directory:

```
concat \temp *.plm(cr)
```

COPY FILES (COPY)

```
COPY [path]filename.ext [/A] [/B] [+ [path]filename.ext [/A] [/B]...]
   [path][filename.ext] [/A] [/B] [/V](cr)
```

COPY is an internal command that copies one or more files from a source file to a destination file.

path
: is the path through the directory hierarchy to the source or destination file. The path includes the drive name and subdirectory names as necessary.

filename.ext
: is the name defined when the file was created. It can also be the name of a new file created during the COPY operation.

To copy a file to the default drive, specify the name of the drive containing the source file, and the filename. After this file copy, the copy in the default drive has the same name as the source file. For example, this command copies the file from drive B to the default drive (drive A) and gives it the same name:

```
copy b:filename(cr)
```

The copy is made to the default drive and to the current directory.

To copy the file from the default drive to another drive, specify the command name, the name of the file to be copied, and the name of the drive. For example, this command copies the file in the default drive to drive B and gives it the same name:

```
copy filename b:(cr)
```

Using only filename parameters, you can:

- Copy a source file from another drive to a destination file on the default drive and keep the same filename:

  ```
  copy b:chkdisk.bat(cr)
  ```

- Copy a source file from the default drive to a destination file with the same name on a different drive:

  ```
  copy chkdisk.bat b:(cr)
  ```

- Copy a source file to a destination file on the same drive and rename the file:

  ```
  copy chkdisk.bat newfile(cr)
  ```

  The CHKDISK.BAT file is copied on the default drive and renamed NEWFILE.

- Copy a source file from the default drive to another drive and rename the file:

  ```
  copy chkdisk.bat b:newfile(cr)
  ```

- Copy a source file from a specified drive to the default drive and rename it:

  ```
  copy b:chkdisk.bat newfile(cr)
  ```

+

concatenates files (links them together). MS-DOS adds a file or files to the end of the first specified file and puts the result in either the first file or the destination file that you name. For example:

```
copy one.fil + plus.fil(cr)
```

adds the file called PLUS.FIL to the end of ONE.FIL and puts the result in ONE.FIL.

The next examples put the result of the concatenated files into another file:

```
copy one.fil + plus.fil big.fil(cr)
```

or

```
copy one.fil + plus.fil + b:next.fil big.fil(cr)
```

You can concatenate files using wild-card characters:

```
copy *.lst + *.ref combin.fil(cr)
```

This command searches for and combines all the files with extension .LST with the .REF files. The result of the concatenation is put into COMBIN.FIL. For example, FILE1.LST is combined with FILE1.REF and the result is put in COMBIN.FIL.

As another example, the following command combines FILE1.LST with FILE1.REF to form FILE1.FIL. ABC.LST combines with ABC.REF to form ABC.FIL, and so on.

```
copy *.lst + *.ref *.fil(cr)
```

The command:

```
copy *.lst + all.lst(cr)
```

can produce an error if ALL.LST already exists. As each source filename is found, MS-DOS compares the name with the destination filename. If the names are the same, the source file is skipped and MS-DOS displays:

```
Content of destination lost before copy
```

Further concatenation continues normally.

You can, however, use a command such as the following to add all the files with extension .LST to the file ALL.LST:

```
copy all.lst + *.lst(cr)
```

/A

specified with the source file causes the file to be processed as an ASCII (text) file. The file is copied to the first end-of-file character (ALT-Z). The ALT-Z and the rest of the file are not copied. For example:

```
copy file.txt /a memo.txt(cr)
```

If there is an embedded ALT-Z in FILE.TXT, the ALT-Z is removed when FILE.TXT is copied to MEMO.TXT. MEMO.TXT contains only one ALT-Z, the last character of the file.

/A specified with the destination file puts an ALT-Z at the end of the file.

/A is the default when files are being concatenated (designated by the + symbol).

/B

specified with the source file copies the entire file to the physical end-of-file defined by the directory file size. For example:

```
copy b:new.fil /b cop.fil /a(cr)
```

The /B on the source file keeps the end-of-file mark (ALT-Z), and the /A on the destination file inserts an ALT-Z. /B specified with the destination file stops the ALT-Z from being placed at the end of the file.

MS-DOS concatenates files as ASCII (text) files, as the default. An ALT-Z (1A in hexadecimal) in the file is interpreted as an end-of-file mark.

When you want to combine binary files, use /B. /B forces the COPY command to use the physical end-of-file defined in the directory. /B remains in effect until MS-DOS encounters another /A or /B. For example:

```
copy /b bin.fil + cop.fil(cr)
```

concatenates COP.FIL to the end of BIN.FIL and places the result into BIN.FIL as a binary file.

/V

is a command switch that tells MS-DOS to verify that the sectors written on the destination disk are recorded correctly. Use this switch to check that critical data is copied accurately.

You can also use reserved device names (see Chapter 2.2.4) in a COPY command. For example, you can COPY your keyboard input into a disk file with COPY CON:

```
A>copy con text.fil(cr)
This is a text file(cr)
Each time you press the Return key, a line is
created in the file(cr)
The last line ends with an ALT-Z (and a Return)(cr)
^Z(cr)

A>_
```

After you press ALT-Z (^Z) and Return, the COPY ends and the file is saved as TEXT.FIL.

CHANGE INPUT/OUTPUT DEVICES (CAUX, CLST, CTTY)

CAUX [devicename](cr)

CLST [devicename](cr)

CTTY [devicename](cr)

CLST, CAUX, and CTTY are internal commands that define the devices (or device drivers) in your system. CAUX controls the auxiliary input/output device; CLST controls the list device (printer); CTTY controls the console device (keyboard and screen). If you enter one of these commands with no device name, the current device driver is displayed.

devicename

is the name of a device driver. You must use a valid device name for a device driver that was loaded in the CONFIG.SYS file (using the DEVICE command). These are the default drivers installed by the CONFIG.SYS file on your system diskette:

PRN is the list device for CLST
AUX is the auxiliary I/O device for CAUX
CON is the console I/O device for CTTY

To prevent confusion between device names and filenames, precede device names with \DEV\. For example, CTTY = CON is equivalent to CTTY = \DEV\CON.

To select an alternate device driver, type its device name with the correct command, such as:

```
clst fastprtr(cr)

caux cardrdr(cr)

ctty remote(cr)
```

You can change the port used for the list output by using one of these device names with CLST:

SERIALA to select serial port A
SERIALB to select serial port B
PARALLEL to select the parallel port

For more information about your system's device drivers, see the DISKID file on your system diskette.

CURRENT DATE (DATE)

DATE [mm/dd/yy](cr) (in American operating systems)

or

DATE [dd-mm-yy](cr) (in European operating systems)

DATE is an internal command. MS-DOS records the date in the current directory for every file that you create or change. Specifying DATE without parameters displays the day and the numeric representation for the month, the day, and the year last set and the prompt for the new date, as follows:

```
Current date is Wed  6/22/83
Enter new date:_
```

If you do not want to change the displayed date, press the Return key.

Enter DATE with a date to change the date known to the system.

mm
  is the number for the month (1 through 12).

dd
  is the number for the day (1 through 31).

yy
  is the number for the year (80 through 99, or 1980 through 2099).

Separate the month, day, and year with hyphens (-) or slashes (/). If you use any symbols other than these, or if you use letters instead of numbers, the following message and prompt appear:

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

You can change the date from the keyboard or from a batch file. If you are using an AUTOEXEC.BAT file during MS-DOS start-up, however, MS-DOS does not prompt for the date. For this reason, you may want to include the DATE command in that file (see Chapter 2.3.3).

DELETE FILES (DEL)

DEL [path]filename.ext(cr)

DEL is an internal command that deletes all files that match the specified filename.

path

is the path through the subdirectory system to the file you want to delete. Include the drive name in the path if the file is not on the default drive.

Enter DEL *.* to delete all the files. MS-DOS displays:

```
Are you sure (Y/N)?_
```

You must type Y or N followed by a Return to confirm or retract your DEL *.* command.

For example, the command:

```
del b:\bin\text\examp.txt(cr)
```

deletes the file EXAMP.TXT from the TEXT subdirectory.

DIRECTORY (DIR)

```
DIR [path][[filename][.ext]] [/P] [/W](cr)
```

DIR is an internal command. DIR without parameters lists all the files in the current (sub)directory on the default drive.

path
: is the path through the subdirectory system to the subdirectory you want to list, or to the file(s) whose existence you want to verify with DIR. Path includes the drive name and subdirectory names as necessary.

filename
: lists all the files with the given name.

filename.ext
: lists only those files from the root directory with the given name and extension.

/P
: stops listing the directory when the screen is full. MS-DOS prompts you to strike any key to continue viewing the directory.

/W
: displays the filenames across the width of the screen. Only the filenames are displayed (five per line).

DIR lists the filenames and their sizes in bytes, plus the time and the date they were last modified, for all the parameters except /W.

You can use the ? and * wild-card characters (see Chapter 2.5) in the place of parts of a filename. For example:

DIR *.*	is the same as	DIR
DIR FILENAME	is the same as	DIR FILENAME.*
DIR .EXT	is the same as	DIR *.EXT

DUAL DRIVE DISK COPY (DISKCOPY)

(VICTOR mode only)

DISKCOPY [side to side] [/C] [/E] [/Z](cr)

DISKCOPY.COM is an external command or utility program that copies the contents of a diskette in one drive onto a diskette in another drive. The destination diskette does not have to be prepared with FORMAT because DISKCOPY formats the diskette to which the copy is being made. During the copying process, DISKCOPY displays the number of the track being copied. The disk being copied and the new disk must have the same number of physical sectors, and the sectors must be the same size.

If you enter DISKCOPY without parameters, the program prompts you to:

- Name the drive (left or right) containing the diskette to be copied.
- Press the Space bar to start the copy process.
- Repeat the process after the copy is complete.

The following example copies the diskette in the left drive (A) to the diskette in the right drive (B). Screen prompts are shown in the order they appear on the screen, but their position on the screen is not shown in the example.

```
A>diskcopy(cr)

Diskette COPY Utility -- Version ms x.y

Copy from FLOPPY drive? (Left or Right; press RETURN
to end)l

Copy from Left FLOPPY drive to Right FLOPPY drive.
Press space bar when ready.(sp)

Copy from Left FLOPPY drive to Right FLOPPY drive

Copy from Left FLOPPY drive to Right FLOPPY drive complete.

Copy from FLOPPY drive? (Left or Right; press RETURN
to end)(cr)

A>_
```

If you use the steps shown in the previous example, you can remove the system diskette from the left drive after DISKCOPY loads into memory. This also means that once the “Diskette COPY Utility” sign-on message appears, you can use either drive for the source or destination diskette. In this way you can load DISKCOPY from your system diskette and then swap in another diskette that you want to copy.

side to side

is an optional phrase naming the drives containing the source and destination diskettes. The phrase can take either of these two forms:

left to right

or

right to left

You can abbreviate this phrase to letters. You can use L and R to represent left and right, and you can use T instead of TO, or you can omit the T entirely. For example, these three commands copy the diskette in the right drive to the diskette in the left:

```
diskcopy r to l
diskcopy r t l
diskcopy r l
```

If you include the drive sides to be used in a DISKCOPY command, DISKCOPY loads into memory and begins copying immediately, so that you have no opportunity to swap diskettes. For this reason, you should include the drive parameters only if you are copying your system diskette (the diskette containing the DISKCOPY command file).

/C
: displays the total number of soft errors found.

/E
: displays the locations of soft errors (worn or flawed spots on the diskette) and the total number of soft errors found. (/E implies /C.)

/Z
: displays disk zone information—the size of tracks and gaps (the space in bytes between sectors).

You can combine switches. This command copies the contents of the diskette in the right drive to the diskette in the left drive:

```
diskcopy r l /c /z(cr)
```

After the copy is made, the disk zone information and the total number of soft errors found are displayed:

```
Diskette COPY Utility -- Version ms x.y

zone=0, track= 0: tracksize=11056, gap2=30
   .       .          .              .
   .       .          .              .
zone=7, track=71: tracksize= 6981, gap2=30

Diskcopy complete
80 tracks copied: 0 soft errors
```

If there have been many file deletions and creations on the diskette you are copying, the disk is “fragmented.” This condition occurs because disk space is not always allocated sequentially when files are created. Access time on a fragmented disk may be slow. You can consolidate files on a fragmented disk by using the COPY command instead of DISKCOPY.

For example, the following COPY command copies all the files from the diskette in drive A to the diskette in drive B (note that you cannot copy your operating system diskette using *.*):

```
copy a:*.* b:(cr)
```

ECHO MESSAGES TO DISPLAY (ECHO)

ECHO ON(cr)

or

ECHO OFF(cr)

or

ECHO message(cr)

ECHO is an internal command for batch files that controls the display of batch file commands and comments during batch file processing. ECHO without parameters specifies the current setting (ON or OFF).

ON
: displays all batch commands and remarks (REM) during batch file processing. ECHO ON is the default.

OFF
: turns off display of the commands in the batch file. The command (such as A > ECHO OFF) is displayed, but any batch commands following this command are not displayed.

message
: is a string of alphanumeric characters that you want displayed during batch file processing, for example:

```
copy con greet.bat(cr)
echo Welcome aboard!(cr)
echo Enter your name.(cr)
^Z(cr)
```

When you specify GREET, MS-DOS displays:

```
A>echo Welcome aboard!
Welcome aboard!
A>echo Enter your name.
Enter your name.
```

You can include ECHO OFF in a batch file to suppress just the display of the batch commands. And, in the same file, you can specify ECHO with a message to display that message during batch processing. For example, if you add ECHO OFF to the GREET.BAT file above:

```
copy con greet.bat(cr)
echo off(cr)
echo Welcome aboard!(cr)
echo Enter your name.(cr)
^Z(cr)
```

and enter GREET from the keyboard, MS-DOS displays only the messages following each ECHO command:

```
A>echo off
Welcome aboard!
Enter your name.
```

EXIT COMMAND.COM (EXIT)

```
EXIT(cr)
```

EXIT is an internal command. You can use this command to return from COMMAND.COM to the program you were using previously.

When MS-DOS is processing a program and you want to use an MS-DOS command, specify:

```
command(cr)
```

This command invokes the COMMAND.COM program. You can now use any internal command. To return to the previous program, specify:

```
exit(cr)
```

FILE COMPARISON UTILITY (FILCOM)

```
FILCOM [/#] [/B] [/C] [/W] file1 file2(cr)
```

The File Comparison utility (FILCOM.EXE) compares the contents of two files. You may want to compare disk files if there are two copies and you want to see which one is current. FILCOM can output the differences between the two files to the console or to a third file. The files may be source files (files containing source statements of a programming language) or binary files (output by the Macro Assembler, the MS-LINK Linker utility, or a high-level language compiler).

The comparison can be made in either of two ways: on a line-by-line or a byte-by-byte basis. The line-by-line comparison isolates blocks of lines that are different between the two files and prints those blocks of lines. The byte-by-byte comparison displays the bytes that are different between the two files.

FILCOM

matches FILE1 against FILE2 and reports any differences between them. Both filenames can include pathnames. For example:

```
fc b:\mgr1.nam\ted\f1.asm  \mgr2.nam\f2.asm(cr)
```

compares F1.ASM in the \MGR1.NAM\TED subdirectory on drive B with F2.ASM in the \MGR2.NAM subdirectory on the disk in the default drive.

/#

is a number from 1 to 9. This switch specifies the number of lines required to match for the files to be considered as matching again after a difference has been found. If this switch is not specified, it defaults to 3. This switch is used only in source comparisons.

/B

forces a binary comparison of both files. The two files are compared byte-to-byte, with no attempt to re-synchronize after a mismatch. The mismatches are printed as follows:

```
--ADDRS----F1----F2--
xxxxxxxx    yy    zz
```

(where xxxxxxxx is the relative address of the pair of bytes from the beginning of the file). Addresses start at 00000000; yy and zz are the mismatched bytes from FILE1 and FILE2, respectively. If one of the files contains less data than the other, a message is printed out. For example, if FILE1 ends before FILE2, then FILCOM displays:

```
***Data left in F2***
```

/C

ignores the case of letters. All letters in the files are considered uppercase. This switch is used only in source comparisons. For example:

Much MORE data IS NOT FOUND

matches with:

much more data is not found

/W

compresses "whites" (tabs and spaces) during the comparison. Thus, multiple contiguous whites in any line are considered a single white space. Note that although FILCOM compresses whites, it does not ignore them. The two exceptions are beginning and ending whites in a line, which are ignored. This switch is used only in source comparisons. In this example an underscore represents a white:

____More___data_to_be_found_____

matches with:

More_data_to_be_found

and with:

______More____________data_to_be______found________

but will not match with:

Moredata_to_be_found

If both the /W and /C options are specified, then FILCOM compresses tabs and spaces, and ignores case. For example:

____DATA_was_found____

matches with:

data_was_found

file1 file2

are the full filenames and paths for the two files you want to compare. If the file is in the current directory on the default drive, you can omit the drive and directory names.

The File Comparison utility reports the differences between the two files you specify by displaying the first filename, followed by the lines that differ between the files, followed by the first line to match in both files. FILCOM then displays the name of the second file followed by the lines that are different, followed by the first line that matches. For example:

```
          ...
          ...
----------<filename1>
<difference>
<1st line to match file2 in file1>

----------<filename2>
<difference>
<1st line to match file1 in file2>
---------------------------------------------

          ...
          ...
```

FILCOM continues to list each difference.

If there are too many differences (involving too many lines), the program reports that the files are different and stops.

If no matches are found after the first difference is found, FILCOM displays:

```
*** Files are different ***
```

and the MS-DOS prompt reappears.

The differences and matches between the two files are displayed on your screen unless you redirect the output to a file. You can redirect FILCOM output with the > and < symbols as described in Chapter 6. For example, to compare F1.TXT and F2.TXT and then send the FILCOM output to DIFFER.TXT, type:

```
fc f1.txt f2.txt > differ.txt(cr)
```

FILCOM uses a large amount of memory as buffer (storage) space to hold the source files. If the source files are larger than available memory, FILCOM compares what can be loaded into the buffer space. If no lines match in the portions of the files in the buffer space, FILCOM displays only the message:

```
*** Files are different ***
```

For binary files larger than available memory, FILCOM compares both files completely, overlaying the portion in memory with the next portion from disk. All differences are output in the same manner as those files that fit completely in memory.

7

FAST EXPRESSION SEARCH (FGREP)

```
FGREP [options] expression [files or directories](cr)
```

FGREP.EXE is an external command (utility) that searches through a list of files for files which contain a specific text string. FGREP outputs the lines that contain the text string and the names of the files. Options allow you to search subdirectories, list only filenames, and list the number of lines that match.

Options can be any of the following:

/R

recursively searches subdirectories of the directories in the [files or directories] list. If you do not include this option, FGREP searches only the files you name in the command, as well as all files in any directories named in that command.

/I

ignores upper/lowercase when matching the expression.

/L

also prints line numbers for matching text in files.

/N

prints only the filenames of files with matching text.

/V

reverses the match—prints all lines except those with a matching expression.

/C

prints only a count of the lines containing matching text.

/^

matches only text at the beginning of a line.

/$

matches only text at the end of a line.

W

matches only text that is a word (or two words) not preceded by a letter, digit, or underscore.

expression

is the text string to search for. The expression cannot contain spaces or wild-cards. Specify this string exactly as you want it to appear in the files—that is, without delimiters such as quotes. For example:

```
fgrep jack a:chapt1.doc(cr)
```

searches for “jack” in the file CHAPT1.DOC on drive A.

If the expression contains spaces, hyphens, or other MS-DOS command-line characters (such as < or >), you must enclose the string in double quotes. For example:

```
fgrep "a jack-in-the-box" *.pas(cr)
```

searches all files with PAS extension for “a jack-in-the-box”, an expression containing spaces and hyphens.

files or directories

is a list of one or more subdirectory names and/or filenames (which can include wild-card characters). This list specifies the domain on which the FGREP command will operate. If no files are given, FGREP accepts keyboard input (or redirected input) to search.

For example:

```
fgrep /i^$ proc c:\sources d:\project\*.asm(cr)
```

ignores upper/lowercase (/I) and prints a list of files and their lines that begin (/^) and end (/$) with (i.e., consist only of) the string “proc”. Files in the directory C:\SOURCES and files with ASM extension in the D:\PROJECT directory are searched.

This command:

```
fgrep /ri proc c:\sources d:\project\*.asm(cr)
```

acts as the previous example, except that any subdirectories (/R) of C:\SOURCES (and their subdirectories) will also be searched.

This final example:

```
fgrep /vn priority c:\memos d:\*.not(cr)
```

outputs the names of files (/N) which have lines that do not contain (/V) the word "priority". FGREP searches the \MEMOS subdirectory on drive C, and all files with .NOT extension in the root directory on drive D.

FIND ALPHANUMERIC CHARACTERS (FIND)

```
FIND [/C] [/N] [/V] "string" [path]filename.ext...(cr)
```

FIND.EXE is an external command for filtering data. This command searches a file to find a specified string of alphanumeric characters. FIND with only the "string" parameter searches the displayed lines on the screen. MS-DOS then displays all the lines on the screen that contain the specified "string".

/C
displays only the number of matches with the "string". The lines that match the string are not displayed.

/N
displays the relative number of the line(s) that matches the "string". MS-DOS displays the relative line number, then the line.

/V
displays the lines that do not have the "string".

"string"
is a string of alphanumeric characters that you want to find. The characters must be enclosed in quotes. If they are already enclosed in quotes, use double quotes to specify the characters as the string parameter.

path
: is the path through the directory hierarchy to the file(s) to be searched for the "string".

filename.ext
: is the name of the file or files to be searched for the specified string.

The following example searches a mail list file for "Washington, D.C." You can create the mail list file by entering:

```
copy con mail.lst(cr)
George Washington, Washington, D.C.(cr)
Charles de Gaulle, Paris, France(cr)
Pierre Trudeau, Montreal, Canada(cr)
Abraham Lincoln, Washington, D.C.(cr)
Margaret Thatcher, London, England(cr)
Ronald Reagan, Washington, D.C.(cr)
^Z(cr)
```

Then enter:

```
find "Washington, D.C." mail.lst(cr)
```

On the first line MS-DOS displays a dashed line, the path (\) to the file if one was specified, and the name of the file:

```
----------mail.lst
```

MS-DOS then displays every line that contains the string "Washington, D.C."

If MS-DOS displays:

```
Bad command or filename
```

MS-DOS could not locate the FIND command. FIND is an external command that may be in another subdirectory. Enter a PATH command giving the location of the external command FIND.

If MS-DOS displays:

```
FIND: File not found mail.lst
```

the MAIL.LST file is in a subdirectory. You need to specify the path (\) to the file.

The FIND command can also filter data from an input source and pipe it to an output source (see Chapter 6.3.1). For example, you could find the characters “Washington, D.C.” and pipe them to an output file called USPRES.DAT:

```
find "Washington, D.C." mail.lst | sort > uspres.dat(cr)
```

The | tells MS-DOS to use the output from the FIND command as input to the SORT command.

Use the DIR command after you have piped data to verify that the file is in the directory that you want. If it is not, use the COPY command to put the file in the correct directory.

FOR EACH ONE REPEAT DOS COMMAND (FOR–IN–DO)

FOR %%variable IN (set) DO command %%variable(cr)

FOR is an internal command used in batch files. It can also be used interactively. FOR repeats the specified MS-DOS command for each member in the specified set.

%%variable
is a dummy variable where "variable" can be any character except the numbers 0–9 (which are used for the replaceable batch parameters, %0–%9). %%variable is replaced by each sequential member in the specified set. The variable specified at the beginning of the command line must be the same as the last variable in the command line. For example, if you specify %%A for the first variable parameter and %%F for the last variable parameter, MS-DOS displays:

```
Entry error
```

IN
tells MS-DOS to sequentially replace %%variable with each sequential member in the specified set.

(set)
can be any alphanumeric characters, including the replaceable batch parameters %0–%9. The set contains more than one member and must be enclosed in parentheses:

(mgr1 mgr2 mgr3)

You cannot specify a path (\) in the set.

DO
tells MS-DOS to process the command that follows.

command
  is an MS-DOS command.

For example, with FOR you can create a batch file that makes a new directory, changes the directory, and makes subdirectories:

```
copy con mgrdir.bat(cr)
mkdir %1(cr)
chdir %1(cr)
for %%a in (%2 %3 %4) do mkdir %%a(cr)
^Z(cr)
```

To process the file, enter the batch filename (MGRDIR) and the subdirectory names you want to create:

```
mgrdir managers mgr1 mgr2 mgr3(cr)
```

MANAGERS replaces %1, MGR1 replaces %2, MGR2 replaces %3, and MGR3 replaces %4. %%A is replaced by the first member of the set, which is MGR1. The command then tells MS-DOS, "FOR MGR1 IN (MGR1 MGR2 MGR3) DO MKDIR MGR1." MGR1 (%%A) is then replaced by MGR2, and MGR2 is replaced by MGR3.

MS-DOS displays each batch command as it executes:

```
A>mkdir managers
A>chdir managers
A>for %a in (mgr1 mgr2 mgr3) do mkdir %a
A>mkdir mgr1
A>mkdir mgr2
A>mkdir mgr3
A>_
```

FORMAT DISKETTE (FORMAT)
(VICTOR mode only)

```
FORMAT [side:] [/C] [/D] [/E] [/Z] [/Fx](cr)
```

FORMAT.EXE is an external command (utility) that prepares floppy diskettes to receive data. As it formats the storage space on a diskette, FORMAT erases any previous files. You can load the FORMAT program, like the DISKCOPY program, in either of two ways—by typing only the program name, or by typing a full command line. And as with DISKCOPY, FORMAT has a full-screen mode that enables you to process more than one diskette without reloading the program for each one.

If you enter:

```
format(cr)
```

the program loads into memory and displays a screen that asks you to:

- ▶ Select the drive (left or right) containing the diskette to be formatted.
- ▶ Supply an optional Volume ID (an on-disk label of up to 11 alphanumeric characters) to identify the diskette.
- ▶ Select double-sided format if your computer has double-sided drives.

To make selections, use the cursor keys on the right side of the keyboard. Then use the function keys at the top of your keyboard to start or quit the FORMAT process.

The bottom line of the screen shows the current value of the function keys. For example, you can get onscreen HELP by pressing F7 (function key 7).

For a complete description of FORMAT's full-screen mode, see the *Operator's Reference Guide.*

If you include the drive side in your FORMAT command line, the program formats the diskette on the side you name and then exits to the operating system. For example, you can type:

```
format right(cr)
```

The FORMAT program displays its one-line sign-on message and immediately begins formatting the diskette in the right drive (drive B). FORMAT tells you when the process is complete, and then redisplays the system prompt. The drive side can be abbreviated (as for the DISKCOPY command) to L (left) or R (right).

You can expand the FORMAT command by including command switches. FORMAT uses the three switches used by DISKCOPY (/C, /E, and /Z) plus two more (/D and /Fx). The five FORMAT switches operate as follows:

- /C displays the total number of soft errors found and the amount of space available on the diskette. Use this switch to gauge the reliability of a diskette. If 15 or more soft errors appear, repeat the FORMAT. If this number persists, discard the diskette.
- /D formats a double-sided diskette. This switch works only on a computer with double-sided drives.
- /E displays the total number and the locations of soft errors encountered, and the amount of space available on the diskette. The /E switch implies the /C switch.
- /Z displays disk zone information (size of tracks and gaps).
- /Fx names the maximum acceptable number of soft errors, where x is a number from 0–9. When x number of soft errors has been encountered, the format fails.

You can combine switches in one command. For example, the command:

```
format right /e /d(cr)
```

formats both sides of the diskette in drive B and displays the locations and total numbers of soft errors encountered.

GO TO LABEL (GOTO)

```
GOTO label(cr)
```

GOTO is an internal command for batch files. Batch processing continues with the command on the line below the specified label. GOTO is issued as a separate command except when used with the IF command. The IF command uses the GOTO command on the same command line as IF.

label

is a string of alphanumeric characters. The label can be 8 characters. The label in the GOTO command must match a label in the batch file, except that the label in the batch file must be preceded by a colon. A colon starting a command line tells MS-DOS to ignore this line. The line containing the label can precede or follow the GOTO command in the batch file. When the label precedes the GOTO, the batch file goes into a loop. When the label follows the GOTO, batch processing skips the intervening commands. If the batch file contains a GOTO command that does not include a label, the batch file terminates.

In the example of the GOTO command that follows, the double = tells MS-DOS that this is an "equal to," not an assignment command (such as SET PATH = A:\COMDIR).

```
copy con private.bat(cr)
echo off(cr)
if %1 == security goto greet(cr)
echo Access denied(cr)
goto end(cr)
:greet(cr)
echo Welcome! Enter your name!(cr)
:end(cr)
^Z(cr)
```

To process PRIVATE.BAT and to substitute the word SECURITY for the replaceable parameter %1, type:

```
private security(cr)
```

MS-DOS displays:

```
A>echo off
Welcome! Enter your name!
```

If you specify any word other than SECURITY, MS-DOS displays:

```
A>echo off
Access denied
```

The ECHO OFF command stops the display of the batch commands during batch file processing. MS-DOS replaces %1 with the name you entered when you specified the batch file. MS-DOS compares the name to the one in the IF command. When there is a match, the welcome greeting and the prompt message for a name are displayed. If there is no match, the next command processed is the ACCESS DENIED message. MS-DOS then goes to the command following the label :END, which is the end-of-file marker (^Z).

NOTE: The labels are not displayed during batch file processing because MS-DOS ignores lines that start with a colon. You can add nondisplayed comments to a batch file by starting each comment line with a colon.

DISPLAY COMMAND HISTORY (HISTORY)

HISTORY [n](cr)

HISTORY is an internal command that captures console input. It “remembers” the commands you have entered. With HISTORY you can display the command history, and you can select any command from the history to re-enter into the MS-DOS command-line template. Then, using the function keys as described in Chapters 4 and 6, you can edit and/or re-enter that command without re-typing the entire line.

MS-DOS can store a history of any number of command lines, up to an overall limit of 256 characters. Typically, HISTORY retains from 10 to 25 commands. HISTORY commands themselves, however, are not stored in the command history. You can abbreviate HISTORY to HI.

Entering HISTORY with no parameters displays the command history onscreen, like this:

```
n.  <oldest line in HISTORY>
.      .      .      .
.      .      .      .
2.  <command entered before #1>
1.  <command entered before this HISTORY command>
```

Commands too long to fit the display line wrap to the next line.

n

is the number of a command stored in the history. The nth command from the history is placed into the command-line template, where you can edit and/or execute it as shown in Chapter 6.

For example, entering:

```
hi 5(cr)
```

puts the fifth line from the command history into the command-line template. If you want to execute that command without changing it, press F7 (Copy Template).

HISTORY 1 refers to the last command entered prior to any HISTORY command(s). HISTORY 2 is always the non-HISTORY command before that.

IF CONDITION (IF-EXIST)

```
IF [NOT] string1 == string2 command(cr)
```

or

```
IF [NOT] EXIST [path]filename.ext command(cr)
```

or

```
IF [NOT] ERRORLEVEL n command(cr)
```

IF is an internal command for batch files. If a condition is true, MS-DOS processes the command specified in the IF command line. Otherwise, MS-DOS processes the next command in the batch file.

NOT
processes the command when the IF NOT condition is true. (See the example in the "EXIST" description that follows.)

string1 == string2
is a comparison of an alphanumeric string of characters (string1) against another string of alphanumeric characters (string2). If string1 matches string2, the command is processed. To MS-DOS, == means "equal to." String1 is usually a replaceable parameter, such as %1. Command delimiters, such as commas, cannot be used within the string parameters.

For example, this batch file uses the IF command and the string parameters to verify system users:

```
copy con users.bat(cr)
echo off(cr)
if %1  ==  George goto label(cr)
if %1  ==  Jack goto label(cr)
if %1  ==  Mary goto label(cr)
echo You must be an approved user to operate the system.(cr)
goto end
:label(cr)
echo Welcome! Enter your command.(cr)
:end
^Z(cr)
```

When you specify USERS and a name to be processed, MS-DOS replaces the %1 parameter in the first IF command with the specified name. MS-DOS compares that name (string1) to the string of characters in string2. If they are equal, MS-DOS displays:

```
Welcome! Enter your command.
```

If there is not a match in the first command, the next command is processed. If MS-DOS processes the three IF commands in the file without a match, MS-DOS displays:

```
You must be an approved user to operate the
system.
```

EXIST [path]filename.ext

tells MS-DOS to look for the specified file in the specified path. If no path is included, MS-DOS searches the current directory on the default drive. If the file is found, "command" is processed. For example, this command in a batch file:

```
if not exist \mgr\jobs echo Can't find \mgr\jobs.(cr)
```

searches for JOBS in the MGR subdirectory on the disk in the default drive. If JOBS is found, MS-DOS does not process the ECHO command. If the false condition (\MGR\JOBS does not exist) is true, MS-DOS displays:

```
Can't find \mgr\jobs.
```

ERRORLEVEL n

tests for a program failure. This parameter tells MS-DOS to check for an errorlevel number where *n* is the specified number. The next command is processed if the previous program run by MS-DOS had an exit code the same as *n* or higher.

command

can be any of the MS-DOS commands. The most common command used with IF is GOTO.

LIST FILES (LS)

LS [options] [files or directories](cr)

LS.EXE is an external command (utility) that produces a sorted listing (in various directory formats) of a file or list of files. In the LS listing, names of subdirectories are followed by a \ (such as MGR1\).

options can be any of the following switches:

/B
: sorts backwards (Z to A).

/E
: sorts output by file extension.

/G
: lists "toGether." Files are not separated by directory.

/L
: lists the directory entries in long format. The filename, file attributes, date, time, and size are listed for each file. (File attributes, including directory status and read/write status, are described in the *Systems Programmer's Tool Kit, Volume II.*) Long-format listings are not sorted.

/R
: recursively lists all files in subdirectories of any directory specified. You can use this command switch to list the contents of all directories on a drive. LS /R lists each subdirectory followed by all the files in that directory.

/T
: sorts output by date and time of creation, rather than alphabetically.

/1
: lists only one filename per line.

files or directories

is a list of one or more filenames and/or directories. If this parameter is omitted, LS lists the current directory.

The following example lists in alphabetical order the files and subdirectories in the current directory:

```
ls(cr)
```

The next example lists (in long format) the files in the \TEMP directory (and its subdirectories) on drive B, and any file with .ASM extension in the working directory:

```
ls /rl b:\temp *.asm(cr)
```

MAKE DIRECTORY (MKDIR)

```
MKDIR [path](cr)
```

MKDIR is an internal command that makes a new subdirectory beneath the root directory, which MS-DOS created when you formatted your disk. You can abbreviate MKDIR to MD. See Chapter 3 for more information on creating and using subdirectories.

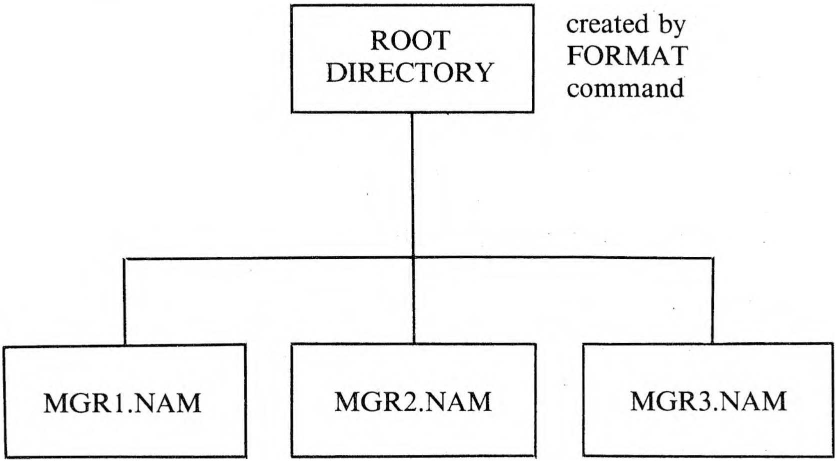
path

is the path through the subdirectory hierarchy to the subdirectory you want to create. The last directory name in the path is the name of the new subdirectory. MS-DOS can create a subdirectory only if a directory above it exists. An initial \ always indicates the root directory. An initial directory name without a \ indicates a subdirectory below the current directory.

For example, if MS-DOS is at the root directory and you enter:

```
mkdir mgr1.nam(cr)
mkdir mgr2.nam(cr)
mkdir mgr3.nam(cr)
```

MS-DOS creates three subdirectories at the first level beneath the root directory:



Each subdirectory is listed in the root directory. You can verify that these subdirectories are in the root directory by entering the DIR command. MS-DOS displays:

```
Volume in drive A is June 8 1983
Directory of A:\

MGR1     NAM <DIR>     6-30-83   9:00a
MGR2     NAM <DIR>     6-30-83   9:00a
MGR3     NAM <DIR>     6-30-83   9:01a
       3 File(s)    370683 bytes free
```

"Directory of A:\" indicates that this is the root directory.

Once a directory exists, you can make a subdirectory below it. For example, this command creates the subdirectory JOBS below the subdirectory MGR1.NAM:

```
mkdir mgr1.nam\jobs(cr)
```

You can enter a directory name without a \ in a MKDIR command to add a subdirectory to the directory you are currently using. For example, if you are working in the MGR1.NAM directory, instead of issuing the preceding command, you can specify:

```
mkdir jobs(cr)
```

Once the JOBS subdirectory has been created, you can create a subdirectory below it:

```
mkdir mgr1.nam\jobs\benefits(cr)
```

or, if the current directory is JOBS:

```
mkdir benefits(cr)
```

You can make as many subdirectories as you want depending on the amount of space on your disk. Subdirectory names can be 1 to 8 characters followed by an extension of 1 to 3 characters. The valid characters for filenames are valid for directory names. You can use duplicate file or directory names as long as they are in separate directories. When you are creating subdirectories, do not use more than 63 characters in the command, including the \.

When you finish making your directory hierarchy, it is a good idea to enter the CHDIR \ command. This command returns you to the root directory. You should always return to the root directory so that you do not accidentally create an entry in the wrong subdirectory.

MODIFY CONSOLE (MODCON)
(VICTOR mode only)

MODCON sourcefile [savefile](cr)

MODCON.EXE is an external command (utility) for modifying the "soft" keyboard and character set used by the console. With MODCON you can select a new keyboard file and/or character set file. You can also save the current set(s), and then restore them later.

sourcefile
is the file or files containing the sets to be made active. Follow these rules for the source filename:

- ▶ If the source file is in a subdirectory, include the path through the directory system to the file (see Chapter 3).
- ▶ To select only a keyboard file (or only a character set file), include the .KB (or .CHR) extension for the source file.
- ▶ If the source filename has no extension, MODCON activates both the .KB and the .CHR files with that filename.
- ▶ If you enter an asterisk (*), no new sets are made active and the configuration remains unchanged.

savefile
is the file or files used to save the current keyboard table and/or character set. Follow these rules for the save file:

- ▶ If you want to save the existing file(s) in a subdirectory, include the path through the directory system to the file or files.
- ▶ If you are saving only a keyboard file (or only a character set file), the saved file should have .KB (or .CHR) extension.
- ▶ To save both keyboard and character set files, omit the extension for the saved filename.

NOTE: The extension for the source file is independent of the extension for the saved file.

Here are some example command lines:

```
modcon b:\dir3\test.chr \chr\save.chr(cr)
```

This command modifies the character set, but does not affect the keyboard. The new character set is TEST.CHR in the \DIR3 subdirectory on drive B. The current character set is saved in SAVE.CHR in the \CHR subdirectory on the default drive.

```
modcon \chr\test.kb(cr)
```

This command selects a new keyboard file (TEST.KB in the \CHR subdirectory on the default drive) but does not save the current keyboard file.

```
modcon germ01 g02save(cr)
```

This command selects GERM01.KB and GERM01.CHR as the new keyboard and character set. The current keyboard table and character set are saved in G02SAVE.KB and G02SAVE.CHR on the default drive.

```
modcon germ01 germ01(cr)
```

This command activates the GERM01.KB and GERM01.CHR files on the default drive. The save operation overwrites the original GERM01.KB and GERM01.CHR files with the current character set and keyboard.

```
modcon aust01(cr)
```

This command makes AUST01.KB and AUST01.CHR the active keyboard table and character set. The original keyboard table and character set are not saved.

```
modcon * brit01.kb(cr)
```

This command saves the active keyboard set in the file BRIT01.KB on the default drive, but leaves it as the active set. The character set is not affected.

```
modcon french.kb b:swedish.chr(cr)
```

This command saves the current character set in B:SWEDISH.CHR and leaves that character set active. It also uses the keyboard from FRENCH.KB on the default drive, overwriting the current keyboard table in memory without saving it.

The following batch file uses MODCON to set up a dedicated keyboard for a WordStar word processing session. The original keyboard is restored when WordStar is terminated.

```
modcon wordamer.kb saved.kb
ws
modcon saved.kb
del saved.kb
```

The first command sets up the American keyboard layout from WORDAMER.KB, and saves the original keyboard in SAVED.KB. The second command invokes WordStar. The third command restores the original keyboard from the file SAVED.KB when the user exits WordStar. The last command deletes the saved keyboard file.

Character Sets and Keyboard Files

To use MODCON, you must have on disk the keyboard or character set files that you want to use in your configuration.

Character sets are obtained by:

- Selecting a set provided on the system selection (SYSGEN) diskette included in the *Applications Programmer's Tool Kit II.*
- Selecting a graphics character set provided with CHARGRAF in the *Graphics Tool Kit II.*
- Modifying the current set (or any available set) using the EFONT character-font editor, available in the *Graphics Tool Kit II* and in the *Applications Programmer's Tool Kit II.*

Keyboard tables are obtained by:

- Selecting a table provided on the system selection (SYSGEN) diskette included in the *Applications Programmer's Tool Kit II.*
- Modifying the current table (or any available table) using the keyboard table editor KEYGEN, included in the *Graphics Tool Kit II* and in the *Applications Programmer's Tool Kit II.*

MORE SCREEN (MORE)

MORE(cr)

MORE.COM is an external filter command that passes through the data in a file and displays one screen of data. At the bottom of the screen, this message is displayed:

```
   -- More --
```

Press Return or Space to display another screen of data. The MORE message is redisplayed with each new screenful until MS-DOS reads the entire file.

MORE is used mostly with other commands to pipe data (see Chapter 6.3). For example, you can display a directory one screen at a time:

dir | more(cr)

The following command uses MORE in a pipe and filter operation:

dir | find "DIR" | more(cr)

The current directory is piped as input to the FIND command. FIND searches the directory for all DIR entries. The output of FIND is piped as input to MORE. The MORE command stops the directory display when the screen is full. When you press the Return key, another screen of directory entries is displayed.

MOVE OR RENAME FILES (MV)

MV [path]filename.ext [path]filename.ext(cr)

or

MV [path]filename.ext... path(cr)

MV.EXE is an external command (utility) that moves files. If you name two files in an MV command, the first file is copied onto the second file, and the original file is deleted. If you name a file and a directory, the file is copied into the directory, and the original file is deleted. If you name multiple files and a directory, all the files are moved into the last named directory.

path
is the path through the subdirectory system to the file(s) you want to move, or to the file or directory where you want to move the file(s).

filename
In the first parameter, filename is the name of the file(s) you want to move. If the second parameter names a file, the first file is copied into the second file, and the original file is deleted. If the two files are on the same drive, MV renames the first file as the second filename, and deletes the second file if it already exists.

You can use wild-card characters in the first filename if you are moving files into a directory. If you use wild-card characters in the first filename of an MV command moving files into another file, however, MV will move only the first matching file into the destination file. Then it displays an error message like this:

```
    I already moved d:FILENAME.1 to d:DESTFILE
    and will not now move d:FILENAME.3 to d:DESTFILE
```

The following example moves all .EXE files to the \BIN directory on drive B:

```
    A>mv *.exe b:\bin(cr)

         mv MV.EXE          to (directory) b:\bin
         mv CONCAT.EXE      to (directory) b:\bin
         mv FORMAT.EXE      to (directory) b:\bin
          .    .    .         .    .     .  .
          .    .    .         .    .     .  .
```

This MV command renames MEMO.MEM to SEPT28.MEM:

mv memo.mem sept28.mem(cr)

You can refer to any number of drives in an MV command. For example, this MV command moves all .EXE files and all files with names starting with SAVE on drive A to the \BACKUP directory on drive B:

mv a:*.exe a:save*.* b:\backup(cr)

PATH TO COMMAND FILE (PATH)

PATH [path][;path](cr)

PATH is an internal command that searches first the current directory then the other specified directories for command or batch files (.COM, .EXE, or .BAT files **only**). PATH allows you to load external commands and batch files from a subdirectory without typing the full path to the subdirectory each time. PATH without parameters displays the previously specified PATH command. For example, if you enter this command:

path \cmd(cr)

and then enter:

path(cr)

MS-DOS displays:

```
PATH=\CMD
```

If a PATH command was not previously specified, MS-DOS displays: "No Path".

;

used with PATH and no other parameters tells MS-DOS to search only the current directory for a command or batch file. A semicolon between directory names in the path specifies more than one path (indicated by the \s) for the command or batch file search.

path

is the path through the subdirectory system to the subdirectory containing the external commands and batch files.

Suppose BIN is the first-level subdirectory and contains all the external commands. You can specify:

```
path \bin(cr)
```

Then you can enter any external command without specifying the \BIN subdirectory, regardless of which directory you are currently using.

If you want MS-DOS to search more than one drive or directory for a command or batch file, you can specify:

```
path a:\bin;a:\bin\utils;c:\pascal\programs;d:\(cr)
```

MS-DOS searches the current directory first, and then the PATH directories in the order that they were specified. In the above example, the directory A:\BIN would be searched before A:\BIN\UTILS. The first .COM, .EXE, or .BAT file found whose name matches the command entered will be loaded.

If your PATH command contains an invalid delimiter or parameter, such as the wrong drive name, MS-DOS does not detect it until it looks for a command or batch file. Then MS-DOS displays either or both of the following error messages:

```
Invalid drive in search path
Bad command or file name
```

PAUSE BATCH PROCESS (PAUSE)

PAUSE [comment](cr)

PAUSE is an internal command for batch files that suspends batch execution and displays: "Strike any key when ready...". Pressing any key except ALT-C resumes execution of the batch file. Pressing ALT-C displays:

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

If you press Y, the remainder of the batch file is not executed, and the MS-DOS command prompt is redisplayed.

At each PAUSE in the file, MS-DOS stops. You can then decide if you want to end processing or do some other action, such as changing diskettes.

comment

is any string of characters (121 maximum) that you specify. The characters are displayed with the "Strike any key when ready..." message. For example, this command stops batch file execution:

pause Insert a diskette in drive b(cr)

MS-DOS displays:

```
Insert a diskette in drive b
Strike any key when ready...
```

CONFIGURE SERIAL PORT(S) (PORTSET)

PORTSET [letter] [baud rate] [parity] [stop bits] [bits per character] [/S](cr)

PORTSET.COM is an external command (utility) that allows you to change the transmission settings (such as the baud rate) for the serial output port(s) A (and B). If your system has one serial port, you can adjust the transmission settings only for port A. Use PORTSET to adjust serial output to match a serial device such as a printer or MODEM. Typing PORTSET with no parameters displays the current configuration of the serial port(s) and displays the command you can use to display help screens:

portset help(cr)

When you load MS-DOS, the values for the serial transmission settings are taken from the DEVICE commands in the CONFIG.SYS file on your system diskette. The default settings, listed in the DISKID file on your system diskette, are:

- Baud rate = 1200
- Parity = none
- Stop bits = 2
- Bits per character = 8

PORTSET temporarily changes these settings during an operating session. Once you reset or power off your system, the temporary settings are lost. When you reboot, the original settings are reloaded from the CONFIG.SYS file. To permanently change your serial output port transmission settings, you must change the DEVICE command(s) in your CONFIG.SYS file.

letter
: is the name of the serial port (A or B) that you are using for your list device. If your system has only one serial port, it is A.

baud rate

is the data transmission or modulation rate, usually defined as "bits per second." Choose a baud rate that matches that of your serial output device (printer or MODEM). The possible values for baud rate range from 50 to 19,200 and are listed in the PORTSET help screens.

parity

is the process of checking data for accuracy by assuring that the number of bits in each character group is ODD or EVEN. The default for your system is no parity checking (NONE).

stop bits

are special data bits that indicate the end of a transmitted character.

bits per character

is the number of data bits in a single character. Options are 5, 6, 7, or 8 (the default).

In a PORTSET command, the values you select for the transmission settings can be separated by spaces, commas, or any of the delimiters listed on the first PORTSET help screen. You can enter the values in any order, and you do not have to enter values for all four transmission settings.

For example, to set the serial output for port B to a baud rate of 300, EVEN parity, 2 stop bits, and 6 bits per character, enter:

```
portset b, 300, even, 2, 6(cr)
```

If you make a mistake typing a command, PORTSET displays an error message pointing out the (first) command element that was incorrect, and giving the correct command syntax.

On a system with two serial ports, you can choose which port (A or B) will receive your printer output by using the CLST command.

PRINT QUEUED FILES (PRINT)

PRINT [drivename:]filename.ext [/C] [/P] [/T] ...(cr)

PRINT.COM is an external command that puts files in a print queue and prints while MS-DOS is doing other tasks. Only files from the current directory can be put in the print queue.

You cannot specify a path (\ with a directory name) with the PRINT command. Use the CHDIR command to move to the directory that contains the file you want to print. PRINT without any parameters displays the names of the files in the queue.

When this command is specified for the first time after MS-DOS starts, MS-DOS displays:

```
Name of list device [PRN]:
```

The list device is the printer. Specify the valid assignment name for your list device, such as SERIALA, SERIALB, or PARALLEL (see the CLST command). If you press the Return key, MS-DOS uses PRN as the list device. PRN is a logical device equivalent to LST.

After you give the device name for your printer, the screen displays:

```
Resident part of PRINT installed
```

If you specify an invalid assignment name for the device, MS-DOS displays:

```
List output is not assigned to a device
```

Be sure that the list device is attached to your computer.

The list device cannot be used when PRINT is being processed. MS-DOS displays an error message until the files are all printed or printing is terminated with the /T parameter.

drivename

is the name of the drive containing the directory of files that you are currently using. If you do not specify a drive, MS-DOS uses the default drive. Do not remove the disk from the drive while the files are being printed. If MS-DOS tries to access the file and the drive is not ready, MS-DOS displays:

```
Drive not ready
```

If MS-DOS finds a disk error, the file being printed is canceled. The printer types a disk error message, advances to the next page, and the printer alarm sounds. MS-DOS then prints the rest of the files in the print queue.

filename

is the name of the file(s) that you want to print. MS-DOS queues and prints the files in the order you specify in the command. You can specify a maximum of 10 files. Entering PRINT and the name of the file(s) without any other parameters:

print chapt1.doc chapt2.doc chapt3.doc(cr)

queues and prints the files. MS-DOS displays the name of the drive and the name of the first file being printed:

```
A:CHAPT1     .DOC is currently being printed
```

The names of the remaining files in the queue are also displayed:

```
CHAPT2     .DOC is in queue
CHAPT3     .DOC is in queue
```

MS-DOS prints a file, moves to the next page, then prints the next file. Tabs are expanded with blanks to the next eighth column.

Wild-card characters can be used in the filename and its extension (see Chapter 2.5).

/C

cancels the named file(s). The filename preceding the /C and subsequent filenames are canceled until MS-DOS finds a /P in the command line. For example, this command cancels the three files CHAPT1, CHAPT2, and CHAPT3 from the print queue and prints CHAPT4.DOC:

print chapt1.doc /c chapt2.doc chapt3.doc chapt4.doc /p(cr)

This command removes all CHAPT?.??? files from the print queue:

print chapt?.* /c(cr)

If there are no other files in the print queue and a file is printing—for example, MORE.TXT—and you enter:

print more.txt /c(cr)

MS-DOS prints:

```
A:MORE     .TXT Canceled by operator
```

The printer advances one page, and the printer alarm sounds once. MS-DOS then displays on your screen:

```
PRINT queue is empty
```

If you specify:

print more.txt echo.txt(cr)

MS-DOS displays:

```
A:MORE      .TXT is currently being printed
ECHO        .TXT is in queue
```

If you then type:

print more.txt /c(cr)

while the MORE file is printing, MS-DOS:

- Stops printing the MORE file
- Prints the operator-canceled message
- Advances the paper one page
- Sounds the printer alarm
- Prints the next file in the queue
- Displays:

```
A:ECHO    .TXT is currently being printed
```

/P

prints the named file(s). The file preceding the /P and subsequent files are printed until MS-DOS finds a /C in the command line. For example:

print chapt5.doc /p chapt6.doc chapt7.doc /c chapt8.doc /p(cr)

prints CHAPT5 and CHAPT6, cancels CHAPT7, and prints CHAPT8. /P is the default when no other parameters are specified with the "filename" parameter and PRINT.

/T

deletes all queued files from the print queue. When you use /T, do not specify any other parameters. For example, this command empties the print queue:

print /t(cr)

MS-DOS displays:

```
PRINT queue is empty
```

MS-DOS prints:

```
All files canceled by operator
```

MS-DOS PROMPT (PROMPT)

PROMPT [prompt-text](cr)

PROMPT is an internal command that changes the MS-DOS default prompt (A >). PROMPT without parameters designates the default prompt (A >).

prompt-text
represents alphanumeric character(s) that you specify.

For example, you can specify the prompt as COMMAND by typing:

prompt COMMAND(cr)

MS-DOS displays the new system prompt instead of the A > prompt:

```
COMMAND
```

If you type the following, the A > prompt returns:

```
COMMANDprompt(cr)
```

The next example changes the prompt from A > to READY, and then reinstates the A > prompt:

```
A>prompt READY(cr)
READYprompt(cr)
A>_
```

You can also insert special characters in the prompt by specifying $c for the prompt-text, where c is:

$ The $ character.

t The current time (set when MS-DOS was loaded or by the last TIME command).

d The current date (set when MS-DOS was loaded or by the last DATE command).

p The name of the current directory of the default drive.

v The MS-DOS version number.

n The default drive.

g The > character.

l The < character.

b The | character.

s A leading space only.

e The ESC character.

_ Carriage return/line feed which returns to the beginning of the next line on the screen.

The special characters inserted with $ must be lowercase. If you specify $ and any other character than the ones described, MS-DOS will not process the command.

To use the current time and date as a system prompt, type:

prompt TIME = t_DATE = $d(cr)

MS-DOS displays, as a two-line command prompt, the current time and date:

```
TIME=10:54:30.50
DATE=Mon 7-04-83
```

READ CPM (RDCPM)

(VICTOR mode only)

[drivename:]RDCPM DIR [drivename:][filename.ext](cr)

or

[drivename:]RDCPM [drivename:]filename.ext(cr)

RDCPM.COM is an external command that can:

- Display the directory of a CP/M-86 diskette while you are using MS-DOS.
- Transfer files from a CP/M-86 diskette to an MS-DOS diskette in the current directory.

DIR

displays the CP/M-86 directory entries, so that you can determine which files to transfer to the MS-DOS diskette.

drivename

specified before the RDCPM command is the name of the drive containing the MS-DOS diskette. If you do not designate a drive, the default drive is used. If you want to transfer files to a specific directory, use the CHDIR command (change directory) before the RDCPM command. You can also specify a path to the RDCPM file if it is in a subdirectory.

drivename:filename.ext

specified after the RDCPM command is the name of the drive containing the CP/M-86 diskette and the name of the file you want to verify (with DIR) or copy to the MS-DOS diskette.

To display a directory, give the drivename. For example, this command displays the CP/M-86 diskette directory in drive B:

```
a:rdcpm dir b:(cr)
```

If you want to list all the CP/M-86 directory entries with the filename ALPHA, enter:

```
a:rdcpm dir b:alpha.*(cr)
```

To transfer a file from a CP/M-86 diskette to an MS-DOS diskette, you must name the CP/M-86 file that you want to transfer. To transfer more than one file, use filename wild-card characters. For example:

```
a:rdcpm b:alpha.*(cr)
```

transfers all the files with the filename ALPHA from the CP/M-86 diskette in drive B onto the MS-DOS diskette in drive A. When the file is transferred properly, the screen displays:

```
Read CP/M Disk Utility -- Version x.y

File transfer complete

A>_
```

RECOVER FILES (RECOVER)

RECOVER [path]filename.ext(cr)

or

RECOVER drivename:(cr)

RECOVER.COM is an external command. Use this command when MS-DOS cannot read a file because of a defective disk. If a file or a diskette directory has a bad sector, this command recovers the file or the entire disk. If there is a bad sector in the directory, this command recovers all the files on the disk.

path
: is the path through the subdirectory system to the file you want to recover.

filename.ext
: is the name of the file that has the bad sector. The file is read sector by sector from the disk. MS-DOS marks the bad sector, skips that sector, and allocates it to a system table. The data in the bad sector is not recovered. MS-DOS does not make any more file allocations to that sector.

The RECOVER command and the drivename without any parameters recovers all the files on the disk. MS-DOS scans the File Allocation Table (FAT) for chains of allocation units (the amount of space used on a disk for a file). MS-DOS creates a new root directory. Each entry in the root directory is renamed as:

```
FILEnnnn.REC
```

where nnnn is a sequential number starting with 0001. Each FILEnnnn.REC points to the beginning of one of the allocation chains.

If the root directory cannot contain all of the allocation unit chains, RECOVER displays a message. The unrecovered chains are left in the File Allocation Table (FAT). Reissue RECOVER and the drive name when you have made more room in the root directory.

REMARK (REM)

```
REM comment(cr)
```

REM is an internal command for batch files. During batch execution, each REM command displays the comment stored on that command line in the batch file.

comment

is a string of characters (123 maximum) that you want displayed when batch execution encounters the REM command in the batch file. The only delimiters you can use in the comment are a blank space, tab, or comma:

```
REM This is a file to display the main directory
PAUSE Insert disk in drive B:
DIR B:
```

You can use a period (.) in place of REM to indicate a comment line. For the first line in the example, you can enter:

```
.This is a file to display the main directory
```

REMOVE DIRECTORY (RMDIR)

```
RMDIR [path](cr)
```

RMDIR is an internal command that removes the last named subdirectory in a directory hierarchy. You can abbreviate RMDIR to RD. Chapter 3 gives more information about RMDIR and the subdirectory system.

path
: is the path through the directory hierarchy to the subdirectory you want to delete.

RMDIR will not delete a subdirectory if it is not empty. If you try to delete the current directory or a subdirectory containing any entries other than the . and .. (notations indicating this directory level and the next higher level), MS-DOS displays:

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

Delete (DEL) all files from the subdirectory you want to remove. Delete all directory entries from the subdirectory with the RMDIR command.

You can use RMDIR and the directory name without a backslash to delete a subdirectory entry in the current directory. Suppose you have a directory hierarchy of:

```
\MGR1.NAM\JOBS\BENEFITS
```

If you are working with the JOBS subdirectory, you can delete BENEFITS from JOBS by entering:

```
rmdir benefits(cr)
```

If you are using any directory except BENEFITS, the command:

```
rmdir \mgr1.nam\jobs\benefits(cr)
```

removes the BENEFITS subdirectory, if BENEFITS is empty. To remove the JOBS subdirectory, return to the root directory with the CHDIR command, make sure JOBS is empty, and then enter RMDIR with the path to JOBS.

RENAME (REN)

```
REN [path]filename.ext filename.ext(cr)
```

REN is an internal command that renames files.

path
: is the path through the subdirectory system to the directory containing the file you want to rename. Path always includes the drive name if the drive is not the default drive.

filename.ext
: The first filename is the name of the file to be renamed. The second filename is the new name. The file itself remains on the diskette where it was originally created.

You can use wild-card characters in either filename. This command changes all the files with .DAT extension to .CMD extension:

```
ren *.dat *.cmd(cr)
```

The following command renames the file PHNLIST on drive B to PHNNUM. The file remains on drive B:

```
ren b:phnlist phnnum(cr)
```

If you rename a file with a name that is already in the directory, MS-DOS displays:

```
Duplicate filename or File not found.
```

SINGLE DRIVE DISK COPY (SDCOPY)
(VICTOR mode only)

```
SDCOPY [drivename:](cr)
```

SDCOPY.EXE is an external command that copies the contents of one diskette onto another diskette in the same disk drive. You insert the source diskette (the diskette to be copied) into the drive, and its contents are read into memory. Then you remove the source diskette, insert a formatted destination diskette into the same drive, and the contents of the source diskette are written from memory onto the destination diskette. SDCOPY is useful for copying diskettes on a single-drive system. SDCOPY does not format the destination diskette; you must use a formatted diskette.

drivename
   is the name of the drive to be used to make the diskette copy.

When you issue the SDCOPY command, screen prompts ask you to insert the source diskette into the drive and then swap the destination diskette. You may have to reinsert the source and destination diskettes more than once depending upon the size of the diskette contents and the amount of available memory. For example:

```
PRINT queue is empty
A>sdcopy(cr)

Single Drive Disk Copy Utility, Ver. x.y

Insert source disk, then press space bar
(return key to exit)(sp)

3 insertions of your source and destination
disks will be required.

Partial read #1 of 3

Insert formatted destination disk, then press
space bar (return key to exit)(sp)

Partial write #1 of 3

Insert source disk, then press space bar
(return key to exit)(sp)

Partial read #2 of 3

Insert formatted destination disk, then press
space bar (return key to exit)(sp)

Partial write #2 of 3

Insert source disk then press space bar (return
key to exit)(sp)

Partial read #3 of 3

Insert formatted destination disk, then press
space bar (return key to exit)(sp)

Partial write #3 of 3

SDCOPY completed

Insert source disk, then press space bar
(return key to exit)(cr)

A>_
```

You can stop the copy process by pressing the Return key in response to a prompt for a diskette. If you stop the copy before it is complete, the destination diskette is unusable.

You can use single- or double-sided diskettes. If you copy a single-sided diskette onto a double-sided diskette, the double-sided diskette becomes a single-sided diskette.

SEARCH FILES (SEARCH)

SEARCH files or directories [matching constraints] [actions](cr)

SEARCH.EXE is a utility (external command) that does these three things:

- Searches through the files and/or directories specified in the command
- Selects matching files according to the specified constraints
- Either lists the matching file(s) or performs specified action(s) on the matching file(s)

You specify the action you want a SEARCH command to perform via command switches (such as /CP for copy files). With the SEARCH utility you can:

- Move, copy, or delete files
- Archive files (by creating a "tape archive" file)
- Change file attributes (such as read-only or read/write status)
- Sum file sizes
- Delete empty subdirectories

Additional switches allow you to request prompting, exclude subdirectories from the search, and copy entire tree structures (subdirectory hierarchies).

Because SEARCH has several types of parameters, the syntax description for this utility is divided into sections listing and describing each type of parameter: "Files or Directories," "Matching Constraints," "Actions," and "Action and Constraint Modifiers."

The first section presents the syntax for "files or directories" and gives examples showing how to use file and directory names with SEARCH. In the other three sections, the command switches for that set of parameters are presented in table form. Examples showing all the SEARCH command parameters are given in a final section, "Hints for Using SEARCH."

Files or Directories—The Domain

The first set of SEARCH parameters—files or directories—immediately follows the keyword SEARCH in a command. This parameter is a list of one or more subdirectory names and/or filenames (which can include the MS-DOS wild-card characters * and ?). The list is the possible domain from which SEARCH selects files to operate on. If only a drive name is given, SEARCH begins at the current working directory of that drive or volume, and searches downward through any subdirectories below. (To display the names of the current working directories, enter the CD /A command.)

If you want SEARCH always to start at the root directory in the domain, use an initial \ in the path. For example, if you specify:

```
search a:\ b:\ other parameters(cr)
```

the SEARCH utility searches downward from the root directories on drives A and B (and only on those drives). Without the \, the search would be "relative," i.e., would start at the current working directory, which may not be the root directory.

You can mix filenames and directories to specify a domain of some specific files and (the files in) some subdirectories. This command:

```
search b:\rtc tod.src tod.asm other parameters(cr)
```

considers the file or directory B:\RTC (and any subdirectories below it) and the two TOD files on the default drive when executing the other parameters.

Matching Constraints

The second set of (optional) parameters in a SEARCH command is matching constraints. These are command switches that further define which files in the specified domain of files and/or directories you want to affect with your SEARCH command. Only files from the domain that meet all the conditions listed as matching constraints are acted on.

You can use any of the switches listed in Table 7-1 only once in a command line.

Table 7-1: Matching Constraint Switches for the SEARCH Command

SWITCH	EFFECT
/NAME filenames	Selects files from the domain that match the filenames following /NAME. To list more than one filename, separate them with exclamation marks, like this: **/name rtc.asm!tod.asm!tod.obj(cr)** Details about including filenames with SEARCH are given in the paragraphs following this table.
/^NAME filenames	Selects files from the domain that do NOT match any of the names following /^NAME. See the examples following this table.
/FNAME pathname	Selects files whose full path names (including subdirectories and filename) match the names following /FNAME. See the examples following this table for a description of what the path names can be.

SWITCH — EFFECT

/^FNAME pathname

Selects files whose full path names do not match any of the names following /^FNAME. See the examples following this table.

/ATTR attribute expression

Selects files whose attributes make the specified attribute expression true. The expression consists of letters representing one of the five "file attributes" described here. Multiple letters can be separated with & (for AND), ! (for OR), and ^ (for unary NOT). You can also use parentheses for grouping subexpressions.

The five file attribute letters are:

A = archive status is set on (i.e., the file has been edited since the last archive)
D = this "file" is a subdirectory
H = this file is a hidden file
R = this file is read-only
S = this file is a system file

To select read/write files (ones that are not read-only) with archive status set on:

```
search a:\ /attr a&^r(cr)
```

To display the status of the five file attributes for your files, use the LS /L command (or SEARCH with the /LS action switch).

/COMP directoryname

Compares the files in the domain with files of the same name in the specified subdirectory. Selects only those files that have the same name as and are equivalent to a file in the specified directory. Any differences between the two files are displayed.

Include the /T switch with /COMP to preserve the directory tree structure of the domain. With /T, the target directory name for the compare is computed as follows:

1. The drive prefix (and any .\ or ..\ prefixes) are deleted from the directory name of the file found (i.e., the drive name is ignored).
2. The remainder of the directory name (including the filename) from the domain is appended to the directory named after the /COMP to produce the file which is to be compared.

For example, SEARCH A: /COMP TEST /T compares files on drive A with files in the relative path TEST. If the file A:\X\Y exists, SEARCH compares it to TEST\X\Y but not to TEST\Y.

SWITCH	EFFECT
	SEARCH A: /COMP TEST without the /T switch compares A:\X\Y to TEST\Y.
/^COMP directoryname	Selects files from the domain if there is not a file of the same name in the specified path OR (if there is a file of the same name) the two files are not equivalent. Any differences between the two files are displayed. The description of /COMP gives more details.
/EXISTS directoryname	Selects files from the domain whose names match filenames in the specified subdirectory. If /T is used, the path is computed as for /COMP.
/^EXISTS directoryname	Selects files from the domain whose names do not match any filenames in the specified path. If /T is included, the path is computed as for /COMP.
/LE filename	Selects files from the domain whose time of creation (date and time) precedes or is the same as that of the named file. (Read LE as "less than or equal to.")
/LT filename	Selects files whose time of creation (date and time) precedes that of the named file. (Read LT as "less than.")
/GE filename	Selects files whose time of creation (date and time) follows or is the same as that of the named file. (Read GE as "greater than or equal to.")
/GT filename	Selects files whose time of creation (date and time) follows that of the named file. (Read GT as "greater than.")
/LEDATE filename	Selects files whose date of creation (regardless of time) precedes or is the same as that of the named file. (Read LE as "less than or equal to.")
/LTDATE filename	Selects files whose date of creation (regardless of time) precedes that of the named file. (Read LT as "less than.")
/GEDATE filename	Selects files whose date of creation (regardless of time) follows or is the same as that of the named file. (Read GE as "greater than or equal to.")
/GTDATE filename	Selects files whose date of creation (regardless of time) follows that of the named file. (Read GT as "greater than.")
/SIZE n	Selects files that are at least *n* bytes in size.
/SIZEB n	Selects files that are at least *n* (512-byte) blocks in size.

File and directory names listed with the /NAME, /^NAME, /FNAME, and /^FNAME switches can have several types of wild-card characters, some of which are themselves switches. The wild cards you can use are as follows:

- The standard MS-DOS ? wild card matches any single character, or none.
- The * matches any number of characters anywhere in the text. In most MS-DOS commands, *A* is equivalent to * because the first * pads to the end of the name. With the /NAME switch, however, *A* matches any text that contains an A.
- The /. switch makes wild cards also match the file extension separator (the . between the filename and extension). For example, without the /. switch, you must specify *.* to match all files; using * with the /. switch matches all files.
- The /\ switch makes wild cards also match the \ in a path name. For example, A**.PAS matches A\X\Y.PAS and A\Z\W.PAS but not A\X\Y\Z\Q\R\X.PAS. Using the * with the /\ switch also matches this last path.
- Multiple-choice selection. If you enclose characters in brackets, SEARCH looks for only the characters listed. For example, [AEIOU]*.ASM matches any filename beginning with a vowel and having an extension of .ASM.

For example, these two /NAME or /^NAME phrases are equivalent:

```
[bv][tc][1k]fd*.*(cr)

[bv][tc][1k]FD* /.(cr)
```

Both match any file with B or V as its first character, T or C as its second, 1 or K as its third, and FD as its fourth and fifth characters. The second example uses the /. switch to match the . between the name and extension.

The next example matches any file in the root directory on drive A (or in any subdirectory on drive A) which does not have the extension .ASM:

```
search a:\ /^name *.asm(cr)
```

The next example matches any file in drive A's root directory (or any subdirectory on A) whose name does not begin with B or V:

```
search a:\ /^name [bv]*.*(cr)
```

The last example uses both /NAME and /^NAME. The files selected from the domain (the root directory of A) must have an extension of .ASM but cannot begin with BT1:

```
search a:\ /name *.asm /^name BT1*.*(cr)
```

The difference between the /FNAME and /NAME switches is that /NAME performs the matching test only on the file's name, while /FNAME performs its test against the full path name for that file (starting at the drive name and the root directory).

For example:

```
search b:\ /fname b:\src*(cr)
```

selects filenames beginning with SRC (and having no filename extension) in the root directory on drive B, or any subdirectory on B that begins with SRC.

Adding the /\ switch matches any subdirectories below the first one. Adding /. matches any file extension. For example, this command matches files in any path beginning with B on drive A:

```
search a:\ /fname a:\b* /\ /.(cr)
```

To match drive designators, remember that the format is D:\. For example:

```
search a:\ /fname a:\t* /\ /.(cr)
```

selects files from the root directory on drive A that begin with T, or any files in subdirectories on A whose directory names begin with T.

The last example selects all files from drive A that are not in subdirectory A:\TEMP:

```
search a:\ /^fname a:\temp\* /\ /.(cr)
```

Actions

The third set of parameters for SEARCH are action switches. These options specify the function to be performed on the files that meet the matching constraints. With action switches you can set and reset file attributes; move, copy, delete, and back up files; print directory entries; and sum file sizes.

Table 7-2 lists and describes the switch settings for actions. Any action marked with an asterisk can be specified only once in a SEARCH command.

Table 7-2: Action Switches for the SEARCH Command

SWITCH	EFFECT
/ATTR + attrs	Sets the specified file attributes on. Specify "attrs" as single-character attributes, as for the /ATTR matching constraint (see Table 7-1). Do not use separators between attribute letters. For example: **/attr + ar(cr)** sets archive and read-only status on.
/ATTR- attrs	Set the specified attributes off.
/BACKUP directoryname *	Is an abbreviation for: **/cp directoryname /attr a /attr- a(cr)** That is, it finds archivable files (those with archive status) and resets archive status off after the copy. To preserve the tree structure of the files you back up, use the /T switch. You cannot use /BACKUP with /CP.
/CP directoryname *	Copies the matching files into the specified directory. This cannot be used with /BACKUP. The /T option can be used as for the /COMP matching constraint.
/LS	Lists a directory of matching files in long format. This switch displays file attribute settings, file size, the time and date of creation, and the full path name. If an attribute is not set for a file, / is printed; if it is set, the attribute is printed. For read/write files, however, w is printed instead of the /. See the matching constraint /ATTR in Table 7-1 for a list of the attributes.
/MV directoryname *	Moves the matching files from the domain into the specified subdirectory. The original files are deleted. If the /T option is specified, the tree structure of the source file is preserved. For example, this command: **search a: /mv z /t(cr)** finds A:\X\Y; the file Z\X\Y would be created, rather than Z\Y (the file created without /T).

SWITCH	EFFECT
	The destination directory name is formed from the source path name as follows: 1. The drive prefix and any .\ or ..\ is removed from the source directory name. 2. The remainder of the source directory name is appended to the directory name after the specified /MV directory name.
/PRINT	Lists the names of the matching files from the domain. PRINT is the default action for a SEARCH command.
/RM	Removes (deletes) the matching files. If both /CP and /RM are specified, the copy is performed before the removal. If /MV is specified with /RM, the /RM has no effect.
/RMDIR	Removes any empty directories encountered. If /MV is specified, the directory remove is attempted after the files are moved. This means that: **search x /mv b: /rmdir(cr)** moves all of X to drive B, and then removes all of subdirectory X.
/SUM	Displays the sum of the sizes of matching files. This is the exact size of the data in the files and does not consider allocation units' round-off.
/TARA filename.ext *	Appends the matching files to an existing TAR (tape archive) file. Otherwise acts same as /TARC.
/TARC filename.ext *	Creates a TAR (tape archive) file named FILENAME.EXT containing all matching files. For example, this command puts all .PAS files on the default drive into the file PASBKUP.TAR: **search *.pas /tarc pasbkup.tar(cr)** See the *Hard Disk User's Guide* for more details on using the /TAR options for archiving and restoring files.

SWITCH	EFFECT
/TARX filename.ext *	Extracts the matching files from the tape archive file FILENAME.EXT. The only action you can perform with this option is to /LS, /PRINT, /SUM, or /CP them. The /RM or /MV switches are ignored. This command lists the files in the tape archive file PASBKUP.TAR: **search /tarx pasbkup.tar /ls(cr)** This command copies the files in the given TAR file into directory XYZ on the default drive: **search /tarx pasbkup.tar /cp xyz(cr)**

*These switches can appear only once in a SEARCH command.

Action and Constraint Modifiers

The switches listed in Table 7-3 can modify both the matching constraints and the actions listed in a SEARCH command. Single-letter options can be grouped after a single switch signal (such as /AV) providing they do not form one of the multiple-letter options. Each multiple-letter option must have a separate signal. For example, you cannot combine the /AV switch with the /NR switch to make /NRAV.

Table 7-3: SEARCH Action and Constraint Modifiers

SWITCH	EFFECT
/A	Matches only files with the archive status on. Short for /ATTR A.
/N	Takes no actions. SEARCH displays what it would have done. The /N switch is equivalent to using /P and always responding negatively.
/NR	Does no recursive searching of subdirectories. SEARCH normally searches any directories you specify and all subdirectories (and their subdirectories, and so on). Using this switch limits searches to only the directories specified.
/P	Prompts (asks yes/no) for /CP, /MV, /RM, /RMDIR, /TARA, /TARC, /TARX, /BACKUP, /ATTR-, or /ATTR + options. Use this switch to verify SEARCH's actions before taking any action that would have a permanent or destructive effect.
/S	Is silent (displays nothing).
/T	Preserves tree structure of directories for /MV, /CP, /COMP, /^COMP, /EXISTS, /^EXISTS, /TARX, and /BACKUP options.
/V	Preserves volume/drive structure. An option for /TARX only, this switch returns files from tape archive to their original volume or drive.
/.	Allows the file extension separator (.) to be wild-carded.
/\	Allows the directory separator (\) to be wild-carded.

Hints for Using SEARCH

- Remember that all matching constraints must be met (they are ANDed together) before a file is processed (listed, deleted, moved, or archived) by SEARCH.
- Mix and match any constraints, action switches, and modifying options switches you want.
- If you are doing anything dangerous, add the /P switch to prompt you for verification of the action, or append the /N switch to only display the actions that would have occurred.

Note also the following:

1. Specifying only a drive name in the domain, such as:

   ```
   search f: /name *.pas(cr)
   ```

   does not necessarily search all files on that drive. SEARCH starts at the current working directory and skips any higher directories. To search drive F for files with the .PAS extension, either do a CD F:\ before the above command, or use:

   ```
   search f:\ /name *.pas(cr)
   ```

2. To list all directories on drive D, type:

   ```
   search d:\ /attr d(cr)
   ```

3. To copy all files that have been modified since the last time you did a backup, use:

   ```
   search a:\ /backup b:(cr)
   ```

   This copies modified files to drive B and resets their archive status off. If the archive status of your files is unknown, enter:

   ```
   search a:\ /attr+ a(cr)
   ```

   to set archive status on for all files (so all files will be archived).

4. Since some commands are shorthand abbreviations, remember that there are some restrictions. For example,

   ```
   search a: /backup b: /attr r(cr)
   ```

   is illegal because /BACKUP B: is an abbreviation for /CP B: /ATTR A /ATTR- A, and only one /ATTR is allowed. Instead you should enter:

   ```
   search a: /cp b: /attr a&r /attr- a(cr)
   ```

5. To find all files created on the same day as SOFTWARE.MEM, on drives A and C, use:

   ```
   search a:\ c:\ /ledate software.mem /gedate software.mem(cr)
   ```

   This command lists all files with dates less than or equal to AND greater than or equal to SOFTWARE.MEM's date (thus, all files created on that date will be listed).

   To copy all of these files to drive B's working directory, enter:

   ```
   search a:\ c:\ /ledate software.mem /gedate software.mem
     /cp b:(cr)
   ```

6. To list all hidden and system files on drives/volumes A, C, D, and E, enter:

   ```
   search a:\ c:\ d:\ e:\ /attr s!h /ls(cr)
   ```

7. Before you archive files with SEARCH /TARC, you can see the amount of space that will be occupied by the archived files. If you are about to archive drives/volumes A, C, and D, enter:

   ```
   search a:\ c:\ d:\ /a /sum(cr)
   ```

   The results tell you the size of the tape archive file that will be created when these files are archived.

8. To archive the files from drives/volumes A, C, and D, enter:

   ```
   search a:\ c:\ d:\ /a /tarc b:backup /attr- a(cr)
   ```

   This command finds all files with archive status on (/A), archives the files into a single file called BACKUP on drive B (/TARC B:BACKUP), and then resets archive status off (ATTR- A).

9. For re-arranging your directory structure, SEARCH can be used to copy entire subdirectories. For example, to move the subdirectory A:\COMMAND\SOURCES and all of its subdirectory structures, including all files, to C, use:

```
search a:\command\sources /mv c:\ /t(cr)
```

   In this example, using the /T option preserves the tree structure (\COMMAND\SOURCES\...). Without the /T switch, all the files of all the directories would be copied into the current working directory of C.

10. To remove any empty directories on drives E and F, use:

```
search e:\ f:\ /rmdir(cr)
```

   The /RMDIR switch can be used with any other actions. Only empty directories will be removed; also, any drive's current working directory cannot be removed.

11. If there are duplicate files in the two directories F:\PROJECTS and G:\MYPROJ, you can delete the redundant files from G:\MYPROJ with this command:

```
search g:\myproj /exists f:\projects /rm(cr)
```

   If you want to delete only files that contain the same data as the similarly named files in F:\PROJECTS, use the following:

```
search g:\myproj /comp f:\projects /rm(cr)
```

SET EQUIVALENT VALUE (SET)

SET [string1 = [string2]](cr)

SET is an internal command that sets one string value equivalent to another string. In subsequent MS-DOS or application program operations, the equivalent value is substituted for the first named string.

With SET, you change the operating environment. You can SET almost anything into the environment. The only syntax requirement is that a single equals sign (=) be in the command line. Entering SET with no parameters displays all the values that have been set with the SET commands. For example:

set(cr)

lists all the values currently set, such as:

```
path=
prompt=$n$g
```

string1

is an environment-name or variable name that you want to set equivalent to the value of string2. For example, you can duplicate the function of the PROMPT command with SET. Enter:

set prompt = YES?(cr)

to change the system prompt to YES?.

To clear any setting, enter the SET command giving only the first string and an equals sign:

```
set prompt = (cr)
```

string2
is the value that you want to substitute for string1 in subsequent operations.

SET is useful if you are doing batch file processing or using an application program that repeats a variable. For example, in a batch file you can create replaceable parameters with names instead of the usual numbers. Replaceable parameters defined as names must have a leading and trailing %, such as %FILE%. You can assign the value to use for that parameter with the SET command:

```
set file = domore(cr)
```

When MS-DOS processes the batch file, %FILE% becomes the filename DOMORE. Using this procedure, you do not have to edit a file to change the parameter names before you run the batch file.

SHIFT REPLACEABLE PARAMETERS (SHIFT)

SHIFT(cr)

SHIFT is an internal command for batch files that moves all replaceable parameters one position to the left:

%1 replaces %0

.

.

.

%10 replaces %9

When there are more than 10 parameters on a command line, each additional parameter shifts into %9.

For example, this batch file named DIRLOOK.BAT uses the SHIFT command to display as many directories as you enter when you load the batch file:

```
dir /p %1
:nexdir
shift
if %1 == stop goto lastone
dir /p %1
goto nexdir
:lastone
cd \
```

If you specify:

```
dirlook managers scheds stop(cr)
```

MS-DOS searches the root directory for the MANAGERS and the SCHEDS directories. During batch processing, MS-DOS displays each batch command (except REM lines or lines beginning with colons) and then performs the command on the line.

MANAGERS is used for %1. Because of the SHIFT command, %1 becomes SCHEDS. PROJ then replaces SCHEDS, and so on. The batch file completes when STOP equals STOP in the IF command.

MS-DOS displays:

```
A>dir /p managers
```

Next the MANAGERS directory is displayed.

```
A>shift
A>if scheds == stop goto lastone
A>dir /p scheds
```

Next the SCHEDS directory is displayed.

```
A>goto nexdir
A>shift
A>if stop == stop goto lastone
A>cd \
```

The CD \ at the end of the file returns you to the root directory (see Chapter 3).

SORT DATA (SORT)

```
[command | ]SORT [/R] [/+n] [< [drivename:]input source]
  [> [drivename:]output source](cr)
```

SORT.EXE is an external command that filters data by rearranging it alphanumerically (using the ASCII collating sequence). The data to be rearranged can be the output from an MS-DOS command.

The command can be any external MS-DOS command whose output you want to sort. The | symbol tells MS-DOS to direct (pipe) the output from the MS-DOS command to the SORT command, as described in Chapter 6.3.

For example, this command displays the current directory sorted alphabetically:

```
dir | sort(cr)
```

The command:

```
dir | find "< DIR >" | sort(cr)
```

uses the output from the DIR command as input to the FIND command. The output of FIND (the names of all the subdirectories in the current directory) is sorted alphabetically and displayed onscreen.

/R

tells MS-DOS to reverse the alphanumerical sort (Z to A, or *n* to 0 where *n* is an integer).

/+n

tells MS-DOS to sort the data by columns. *n* is the number of the column where the SORT is to start. For example:

```
dir | sort /+1(cr)
```

sorts the directory alphanumerically by filename, which is the first column in the directory display. The command:

```
dir | sort /r /+1(cr)
```

does a reverse alphanumerical sort of the directory.

<

tells MS-DOS to use the parameters following the < symbol as input.

drivename

is the name of the drive containing either the input data to be sorted or the sorted output data. The drive name is optional when the input or output source is on the default drive.

input source

is the data you want to sort. It can be a disk file or data from a MODEM. For a disk file, specify the name of the file (and the path to the file if it is not in the current directory). For a MODEM or any other device, you must specify the physical device name, such as UL1 for serial port A (see the CLST and PORTSET commands).

>

tells MS-DOS to use the parameters following the > symbol as output. You must include an output source if you include the > symbol. For example, if you enter:

```
dir | sort >(cr)
```

MS-DOS displays:

```
File creation error
```

output source

is the destination of the sorted data from the input source. It can be files, printers, or other devices. For files you must include the path to the file unless it is in the current directory. This command:

dir | sort > lst(cr)

sorts the directory alphabetically and sends the output to the LST device (your printer). The default output is > CON. If you do not include an output source, the output from the SORT command is sent to CON (the CRT screen).

The command:

find "95066" mail.lst | sort > zip.cod(cr)

pulls all the records containing the zipcode 95066 from the MAIL.LST file and sorts them alphabetically into the ZIP.COD file. ZIP.COD is created by this command, but if ZIP.COD already exists, its contents are overwritten by the output of SORT. To see the sorted data records in ZIP.COD, enter:

type zip.cod(cr)

MS-DOS displays:

```
---------- mail.lst
```

followed by an alphabetical list of each 95066 record.

SYSTEM COPY (SYS)

(VICTOR mode only)

SYS source [destination](cr)

SYS.EXE is an external command or utility program that copies an operating system from an existing system diskette or from a file created during system configuration (see the *Applications Programmer's Tool Kit II*). The default operating system file is MSDOS.SYS, a hidden file. SYS copies MSDOS.SYS, COMMAND.COM, CONFIG.SYS, CONFIG.BAT, and any device drivers named in DEVICE commands in CONFIG.SYS.

source

is drivename: or [path]filename.ext, where drivename is the name of the drive containing the existing system diskette. Path is the path through the subdirectory hierarchy to the operating system file. Filename.ext is the specified operating system file created during system configuration. If you give a filename without an extension or path, MS-DOS assumes the extension .SYS and the current directory or the root directory on the default drive. The source must contain the operating system and the COMMAND.COM file, which is copied with the operating system.

destination

is the name of the drive containing the diskette onto which the operating system is to be copied. If you do not specify the name of the drive for the system copy, SYS asks you for the copy destination. After the copy is complete, you are prompted for another destination.

/O

is a command switch that copies only the hidden file MS-DOS.SYS.

For example, the following SYS command names only the source. This example uses a dual-drive system; if you are using a different system, the SYS prompts will indicate the drives (and volumes) in your system.

```
A>sys msdos.new(cr)

SYStem copy utility -- Version msx.y

Destination drive? (A-B; press RETURN to end.)b

Copy to B: (FLOPPY). Press space bar when
ready.(sp)

Copying...
```

When the copy is complete, the message “Copying...” disappears and this message appears:

```
Copy completed.
```

The operating system file, MSDOS.NEW, COMMAND.COM, and the CONFIG files are copied from drive A to drive B.

If you specify both source and destination with your SYS command, the copy is made without prompts. The following command copies the operating system (MSDOS.SYS, COMMAND.COM, and the CONFIG files) from drive A to drive B:

```
A>sys a: b:(cr)

SYStem copy utility -- Version msx.y

Copying...
```

When the copying completes, this message appears:

```
Copy completed.
```

The system command prompt reappears.

Do not attempt to SYS onto the default drive or onto the drive from which SYS is loading.

PRINT END OF FILE (TAIL)

TAIL [/n] [files or directories](cr)

TAIL.EXE is an external command (utility) that displays the last n lines of a file or list of files.

/n

is the number of lines at the end of the file(s) you want to display onscreen. If you do not include a number, TAIL prints the last 10 lines. The maximum value is around 450. If a line exceeds 130 characters, TAIL truncates it.

files or directories

is a list of subdirectory names and/or filenames to be operated upon. If you do not include files or directories, TAIL gets its input from the keyboard (or redirected input).

For example, this command lists the last 3 lines of every file with .LST extension in the current directory:

tail /3 *.lst(cr)

Either of the next two commands lists the last 10 lines of the file INPUT. The < symbol in the first command indicates that INPUT is being redirected as input to the TAIL command.

```
tail < input(cr)

tail input(cr)
```

The following command prints the last 50 lines of every file in the \TEMP subdirectory on drive A:

```
tail /50 a:\temp(cr)
```

TIME

```
TIME [hh:mm:ss](cr)
```

TIME is an internal command that sets the time for the operating system. If you enter the command without parameters, the current time is displayed and you are prompted for a new time. Press the Return key if you do not want to change the time. For example:

```
A>time(cr)
Current time is 13:01:10.00
Enter new time:9:05(cr)
```

The new time known to the operating system is now 09:05:00.00. The colon separates the hours, minutes, and seconds. The period separates the hundredths of a second.

hh
  represents the hour. Use 00 to 23.

mm
  represents the minutes. Use 00 to 59.

ss
  represents the seconds. Use 00 to 59. You do not have to type the seconds or hundredths of a second.

If you enter the TIME command with valid parameters (numbers, not letters, in the indicated range for each parameter), MS-DOS displays the system prompt. If the parameters are invalid, such as letters instead of numbers or a larger than acceptable range of numbers, MS-DOS displays:

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

MS-DOS also displays this message if you use invalid delimiters, such as a hyphen instead of a colon or a period.

TYPE FILE (TYPE)

```
TYPE [path]filename.ext(cr)
```

TYPE is an internal command that displays the contents of the specified file. Some files may not be readable because they contain non-alphanumeric characters, such as Alternate or Escape sequences. Tabs expand to 8 characters (column 8, 16, and so on through column 80).

path
: is the path through the subdirectory system to the file you want to display.

filename.ext
: is the name of the file to be displayed.

For example, to display the contents of the JOBS.LST file in the \MGR1.NAM\JOBS subdirectory, enter:

```
type \mgr1.nam\jobs\jobs.lst(cr)
```

VERSION OF MS-DOS (VER)

VER(cr)

VER is an internal command that displays the version of MS-DOS that you are using. For example:

ver(cr)

displays:

```
MS-DOS Version  x.y
```

where x.y is the version number of the operating system.

VERIFY WRITE TO DISK (VERIFY)

VERIFY [ON](cr)

or

VERIFY [OFF](cr)

VERIFY is an internal command that verifies whether subsequent data written by MS-DOS onto the disk can be read by MS-DOS when requested. VERIFY has the same function as the V switch on the COPY command. VERIFY without parameters displays the current setting (ON or OFF) of the command. For example, if you type:

verify(cr)

MS-DOS displays:

```
   VERIFY is xx
```

where xx is “on” or “off”.

ON
tells MS-DOS to verify the data each time it writes data to the disk. MS-DOS issues an error message if it cannot do the write operation.

OFF
turns the verify instruction off.

VOLUME LABEL DISPLAY (VOL)

VOL [drivename:] [/C](cr)

VOL is an internal command that displays the Volume ID of a diskette or hard disk volume, and allows you to change the ID. A Volume ID is a name or label you give to a diskette during formatting, or to a hard disk volume during initialization. A Volume ID can be up to 11 characters long, using the valid characters for filenames. MS-DOS displays the Volume ID for a diskette or volume when you enter a DIR command, a CHKDSK command, or a VOL command. You can also use VOL to change a Volume ID without reformatting or re-initializing your diskette or hard disk.

drivename
: is the name of the drive containing the diskette (or the name of the hard disk volume) whose ID you want to check or change. If you do not specify a drive, MS-DOS displays the ID for the default drive or volume.

/C
: is a command switch you use to request a change in the Volume ID.

If you try to display the Volume ID for a diskette or volume that does not have an ID, MS-DOS displays:

```
Volume ID for drive x: has not been set.
```

where x is the name of the drive.

If you type:

vol b: /c(cr)

MS-DOS displays the Volume ID for drive/volume B and prompts you for a new Volume ID, like this:

```
Volume ID for drive B: is ARCHIVE1031
Enter new volume ID:_
```

After you type in the new Volume ID and press Return, MS-DOS writes the new ID to the disk or volume. You can display the new Volume ID for drive B by entering:

vol b:(cr)

WORD AND LINE COUNT (WC)

WC [/R] [/W](cr)

or

WC [/R] [/W] [path...](cr)

or

WC [/R] [/W] [filename...] (cr)

WC.EXE is an external command (utility) that counts words and/or lines in a file or files. WC can count all the files in a directory or in a hierarchy of directories. WC lists the number of lines, the number of words, the filename, and the total count of lines and words. Entering WC with no parameters counts the lines and words in the keyboard input (or redirected input) that follows, up to end-of-file (ALT-Z Return).

/R

is a command switch that expands the count to include files in all subdirectories below the directory named in the WC command. To use this switch, you must list a directory, not a file, in your WC command.

The following example uses the /R switch with the root directory (\) to count the words and lines in all the files in every directory on the default drive:

```
A>wc /r \(cr)

          21 L      386 W   \MSDOS.SYS
          43 L      495 W   \COMMAND.COM
           5 L       16 W   \CONFIG.SYS
           1 L      385 W   \FORMAT.DAT
          49 L      297 W   \SUBDIR1\FINAL.MEM
          86 L      455 W   \SUBDIR2\CHAP2.DOC
           . .        . .     .   .   .
           . .        . .     .   .   .
         563 L     6511 W    total
```

/W

is a command switch that omits the word count from the WC operation. Counting only lines is faster than counting words and lines.

path

is the path through the subdirectory system to the file or subdirectory you want to count. If you list a directory name (or the single \ to indicate the root directory) without a filename, WC counts all the files in that directory.

filename

is the name of the file you want to count. You can use the ? and * wild-card characters to refer to more than one file.

The first example below counts lines and words in the FINAL.MEM file, and the second example counts lines in the FINAL.MEM and FINAL.BAK files:

```
wc final.mem(cr)

wc /w final.mem final.bak(cr)
```

You can list multiple files and directories in a WC command, and the files and directories can be on different drives. The following command counts the number of lines in all files with .DAT extension on the default drive as well as all files in the subdirectory \DRAFT1 on drive B:

```
wc /w *.dat b:\draft1(cr)
```

8

ERROR MESSAGES AND CODES

When the operating system encounters a problem, it displays an error message onscreen. Different error messages are displayed for different types of errors. Error messages are displayed in situations like these:

▶ You have entered a command that the OS does not recognize.

▶ You have misspelled a filename, or you have entered an incorrect drive name.

▶ You have inserted a diskette that does not contain the file or OS that the computer is looking for.

Error messages that may appear during the booting process are listed in Chapters 8.1 and 8.2. These messages are displayed by the boot ROM in response to diagnostic errors and loading errors. Chapter 8.3 lists the error messages that the operating system may display during normal operations. Operating error messages are listed in alphabetical order. Error codes that are displayed with operating error messages are listed and defined in Chapter 8.4.

Some error messages contain two-digit error codes. Error codes displayed by the boot ROM are listed in Table 8-1. Error codes displayed by the operating system are defined in Table 8-2.

8.1 DIAGNOSTIC ERRORS

At power-up the boot ROM tests the size of the installed memory (RAM) and runs diagnostic tests on parts of the system. If a test fails, you will see an error message, as shown here.

8.1.1 MEMORY TEST AND ERROR CODES

The boot ROM tests memory size when you power up the computer and when you press the reset button. If an error is found in the first 16K of memory, the boot process cannot continue. If a memory error occurs between 16K and 128K (or between 128K boundaries), a memory error message is displayed. The message tells you the error's location and the size of usable memory, in this format:

```
M nnnK mm
```

where: M stands for memory test
nnnK is the size of usable memory in Kbytes
mm is a code identifying the memory error as:
03—Unreproduceable error
3x—Error at bit x
4x—Error at bit x and other bits

8

A computer with a memory error may still operate, depending on the size of usable memory. Contact field service for help in correcting memory errors.

After you insert a system diskette, the boot ROM compares the size of the OS on the diskette with the size of RAM. If the installed memory is insufficient to load the OS, the following error message appears with the diskette symbol:

```
M X nnnK
```

where nnnK is the size of usable memory in Kbytes. If this message appears, you must either load an OS that requires less memory or acquire more memory.

DIAGNOSTIC TEST AND ERROR CODES 8.1.2

The boot ROM runs diagnostic tests on the hardware in the system and tells you if any of the tested components are nonfunctional. If any diagnostic test fails, you will see this error message:

Xnxy0

- If n = 1, the PIC (Programmable Interrupt Controller) is nonfunctional.
- If x = 2, the parallel port is nonfunctional.
- If y = 3, the keyboard is nonfunctional.
- If n, x, or y is 0, that component is functional.

Contact your dealer or your field service representative for assistance in correcting any hardware problem.

8.2 LOADING ERRORS

When an error occurs while the boot ROM is loading the OS, the screen displays:

- An X
- A symbol representing the device performing the boot operation
- A two-digit error code defining the type of error

The symbol displayed represents either a floppy disk, the hard disk, or a network. (All boot ROM display symbols are illustrated in Section II of the *Operator's Reference Guide.*) Table 8-1 lists and describes the boot ROM error codes for each device.

If you get a floppy disk error when you try to load the OS from diskette, remove the diskette to escape from the error situation. Check that the diskette is a system diskette and is not write-protected. Try inserting it again or use another system diskette.

If you get a network error when you try to load the network OS, note the server number and the error code displayed. Give this information to your network system administrator, who can tell you how to correct the error.

If you get a hard disk error when booting up from a hard disk, make sure your hard disk is configured to have a boot volume, and the boot volume contains an operating system. You may have to run SYS to install an OS in the boot volume.

Table 8-1: Boot ROM Error Codes

ERROR CODE	MEANING
FLOPPY DISK ERRORS	
01	No sync
02	No header ID after sync
03	Checksum error in header
04	Requested track doesn't match track header
05	No match on sector, error count expired
06	No data ID after header
07	Checksum error in data
08	Sync present too long
09	Door opened during operation
0A	GCR error
20	Internal error: Invalid device unit requested
21	Internal error: Invalid sector requested
99	Floppy label indicates no OS on diskette
NETWORK ERRORS (SERVER NETWORK)	
Boot ROM-Detected Errors	
40	Transporter took too long to send message
41	Could not recover from retry of error 80 or 82
42	Transporter took too long to set up to receive
43	No response from server
44	Bad server number in response
45	Bad sequence number in response
46	Bad message type received
47	Server cannot send a boot
69	Internal error
Transporter-Detected Errors	
80	Message not acknowledged by server
81	Message too large for server's buffer
82	Server's socket not initialized (or busy)
83	Control length mismatch between station and server
84	Invalid socket number (consider this an X69 error)
85	Server's socket is in use
86	Invalid destination address (consider this an X69 error)

ERROR CODE	MEANING
Server-Detected Errors	
10	No boot OS on server's system volume
11	I/O error in server seeking for label
12	I/O error in server reading label
13	Bad ID on label on server's boot OS file
14	Bad blocking information in server's boot OS file
15	Station requested a bad block size (consider this an X69 error)
16	I/O error in server seeking to read an OS block
17	I/O error in server reading an OS block
18	Unexpected end of file in reading OS
19	Bad handle received from station (consider this an X69 error)
1A	Bad request received from station (consider this an X69 error)
HARD DISK ERRORS	
01	No index detected from disk drive
02	No seek-complete signal from disk drive
03	Write fault from disk drive
04	Drive not ready after it was selected
06	Track 00 not found
10	ID field read error
11	Uncorrectable data error
12	Address mark not found
14	Target sector not found
15	Seek error (incorrect cylinder and/or track)
18	Correctable data error
19	Bad-track flag detected
1A	Format error
20	Invalid command
21	Illegal disk address
30	RAM diagnostic failure
31	Program memory checksum error
32	ECC diagnostic failure
3D	Internal error: invalid command
3F	Time out on disk operation
40	Internal error: bad operation
41	Hard disk I/O error: bad region information in label
4F	Internal error: invalid block count
98	Bad hard disk label
99	Label indicates no operating system on disk

error description
Abort, Retry, Ignore?

Type A (to abort) the command, I (to ignore) the error, or R (to retry) the command that generated this error. These are some of the error descriptions that may precede this message:

Drive not ready error
Non-DOS disk error reading drive X
Write protect error writing drive X

Allocation error for file filename

CHKDSK error: The named file had a data block allocated to it that does not exist (that is, a data block number larger than the largest possible block number). CHKDSK truncates the file short of the bad block.

Bad Command

Emulator error: The specified command contained invalid characters. The Emulator terminates.

Bad command or file name

You typed an incorrect command or an invalid delimiter, parameter, or filename. Retype the command or filename correctly. Or, you specified an external command that is not in the current directory. Use the PATH command then enter the external command.

Bad DISKCOPY command

DISKCOPY error: The command was typed incorrectly. Retype the command.

BAD MEDIA. PRESS RETURN KEY TO START OVER OR ALT-C TO EXIT.

SDCOPY error: Utility encountered a bad sector while reading or writing a diskette. Re-FORMAT or replace the diskette.

Bad or missing filename

CONFIG.SYS error: The filename specified in the "DEVICE = filename" line of the CONFIG.SYS file was not found. Be sure that the name of the device driver file agrees with the filename specified in CONFIG.SYS.

Bad parameter on command line -- bad param

SYS error: The indicated parameter "bad param" is not correct. Correct the command and redo it. The most common parameter errors are:

- ▶ No parameter specified. You must specify a source drive or file in the command.
- ▶ Invalid drive. The drive specified must be assigned.
- ▶ Bad destination. The destination specified is not a drive.
- ▶ Too many parameters. You typed more than the source and destination.

CAN FORMAT FLOPPY DISKS ONLY

You have asked FORMAT to format the hard disk. Change your FORMAT command.

Cannot COPY double-sided diskettes

DISKCOPY error: The computer does not have double-sided disk drives. You can copy only single-sided diskettes on single-sided drives.

Cannot Create MSDOS.SYS

or

Cannot Create COMMAND.COM

SYS error: There is no room left in the directory of the destination drive. Delete some files, run CHKDSK, and try again. Or increase the number of open files allowed (change the "files = nn" command in CONFIG.SYS to at least 20.)

Cannot find command

Emulator error: The specified command was not on the default drive. The Emulator terminates.

Cannot FORMAT double-sided diskettes

FORMAT error: Your computer does not have double-sided disk drives. You cannot use the /D command switch with FORMAT unless you are using double-sided drives.

CANNOT FORMAT TRACK

FORMAT error: You may have a bad diskette. Run FORMAT again. If the error persists, discard the diskette.

Cannot Open COMMAND.COM

SYS error: COMMAND.COM is not on the source drive. Use the COPY command to copy COMMAND.COM onto the source drive.

Cannot Open MSDOS.SYS

SYS error: The specified source drive does not contain a bootable diskette. Either change to the directory that contains the operating system or specify a source filename.

Cannot Open source filename

SYS error: The specified source filename cannot be opened. Make sure you typed the name correctly and that the file is on disk.

CANNOT READ TRACK

DISKCOPY error: You may have a bad diskette. Run the program again. If the error persists, discard the diskette.

CANNOT SIZE TRACK

DISKCOPY or FORMAT error: You may have a bad diskette. Run the program again. If the error persists, discard the diskette.

Content of destination lost before copy

COPY error: During file concatenation (COPY with the + parameter), MS-DOS found a source filename with the same name as the destination filename. MS-DOS skips the source file and displays this message.

CLOSE DRIVE DOOR

SDCOPY error: Drive door open during attempted read or write of sector.

Command not found

You typed an incorrect command or an invalid delimiter, parameter, or filename. Retype the command or filename correctly.

Copy aborted at track nn

DISKCOPY error: The diskette may have a bad track, and the program is unable to continue. Run the program again, or try moving to another drive. If the error persists, discard the diskette.

Direct bios call

Emulator error: This message occurs if a direct BIOS call is made to an unsupported function.

Directory error-file: filename

CHKDSK error: No valid data blocks are allocated to the named file. CHKDSK deletes the file.

Disk not initialized

CHKDSK error: No directory or file allocation table was found. If files exist on the disk, and the disk has been physically harmed, it may still be possible to transfer files from this disk to recover data.

Divide by Zero

Calculator error: You have tried to divide a number by zero. Type two shifted CALCs (or CLEAR ENTRY) to abandon the calculation.

Divide overflow

Your program has crashed. Because of a logical error, the program has jumped out of the code segment.

DOUBLE-SIDED DISK CANNOT BE COPIED ON SINGLE-SIDED DRIVE.

DISKCOPY or SDCOPY error: You have tried to copy a double-sided diskette on a single-sided disk drive. You can, however, use COPY to copy the files one at a time onto the single-sided diskette.

DOUBLE-SIDED SOURCE DISK CANNOT BE COPIED TO SINGLE-SIDED DESTINATION.

SDCOPY error: You have tried to copy a double-sided diskette to a single-sided diskette. SDCOPY will again prompt you to insert a destination diskette. Replace the destination diskette with a double-sided one.

Drive = n Track = n Error = MM
Abort, Retry, Ignore?

Type A (to abort) the command, I (to ignore) the error, or R (to retry) the command that generated this error. See Chapter 8.4 for a list of error codes (MM in this message).

Drive not available for CP/M reading

RDCPM error: You gave an invalid drive specifier with the CP/M-86 filename. Retype the command with a valid drive specifier.

DRIVE DOOR OPENED

DISKCOPY or FORMAT error: Close the drive door and restart the program.

Drive not ready

During a print operation, the diskette containing the file being printed was removed.

Duplicate filename or File not found

You renamed a file with a name that is already in the directory.

Emul error - invalid select

Emulator error: The selected drive is invalid. The Emulator terminates.

Entry error

The %% parameters specified in the FOR command must match: FOR %%A IN (%%02 %%03 %%04) DO MKDIR %%A.

Error: Bad Label on drive x

SYS error: The diskette in drive x is unreadable. Swap diskettes or reinsert the diskette.

Error reading CP/M disk

RDCPM error: A disk error occurred while trying to read the CP/M-86 file. Try to run the program again.

Error reading CP/M disk label

RDCPM error: The disk label on the CP/M-86 diskette cannot be read. The CP/M-86 diskette is probably unformatted or bad.

ERROR WHILE READING ON DRIVE x
Abort,Ignore,Retry:

Type A (abort), I (ignore), or R (retry). Retry the operation before ignoring or terminating (aborting) the program. Ignoring the error could result in bad data. Check the resulting file carefully.

ERROR WHILE WRITING ON DRIVE x
Abort,Ignore,Retry:

Type A (abort), I (ignore), or R (retry). Retry the operation before ignoring or terminating (aborting) the program. Ignoring the error could result in bad data. Check the resulting file carefully.

EXEC failure

The FILES = line in CONFIG.SYS specifies too low a number to accommodate system needs. Edit the file to specify a larger number, and then reboot.

FILE ALLOCATION TABLE BAD FOR DRIVE x

Check that the diskette is formatted. (The message indicates that the copy in memory of one of the allocation tables has pointers to nonexistent blocks.)

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

COPY error: The destination file in the COPY command is also the source file. The file copy must be renamed if it is on the same drive, or it must be on a different drive than the source file.

File Creation Error

The directional output symbol (>) was used in the SORT command without an output source.

Or MS-DOS could not complete the copy of a hard disk volume to a diskette because the volume directory was larger than the diskette directory.

File not found

The file you specified does not exist. Check that you typed the filename correctly and try again.

File size error for file filename

CHKDSK error: The size of the file in its directory differs from its actual size. CHKDSK adjusts the directory to indicate the actual file size. (The amount of useful data may be less than the size shown because the last data block may not be used fully.)

Files cross-linked: filename and filename

CHKDSK error: The same data block is allocated to both files. CHKDSK does not try to correct this. To correct the problem, first COPY both files; then delete the originals. Review each file for validity and edit as necessary.

FIND: File not found xxxx.xxx

where x is a character in the filename. You entered the FIND command for a string of characters in a file. MS-DOS could not locate the file, because the file is not in your current directory. You need to simplify a path through the directory hierarchy to the subdirectory containing the correct filename.

Format aborted at track nn

FORMAT error: The diskette has a bad track, and the program is unable to continue. Run the program again, or try moving to another drive. If the error persists, discard the diskette.

Format failure

FORMAT error: The FORMAT was unsuccessful. This message occurs only with one of the other FORMAT error messages.

Insufficient disk space

RDCPM error: There is not enough room on the MS-DOS diskette to transfer the CP/M-86 file. Use a different MS-DOS diskette with more available space.

MS-DOS error: MS-DOS copied as many files as possible. MS-DOS issues this message when the diskette is full and there are more files to be copied. Remove the diskette that is full, insert another formatted diskette, and copy the remaining files.

Insufficient memory to load command

Emulator error: The CP/M-86 program to be loaded required more memory than was available. The Emulator terminates.

Invalid CP/M disk type

RDCPM error: The diskette that is being read from is not a CP/M-86 diskette. Try again with a CP/M-86 diskette.

Invalid date
Enter new date:

You typed the date incorrectly. Retype the date correctly, as month/day/year or day-month-year. Use:

1 through 12 for the month
1 through 31 for the day
80 through 99
or
1980 through 2099 for the year

Invalid directory

The specified subdirectory is not in the current directory or does not exist.

Invalid drive in search path

An invalid drive was specified in a PATH command. Retype the PATH command using valid drive and directory specifications.

Invalid drive specification

You specified an invalid drive. Retype the command.

Invalid parameter

FORMAT error: You named an invalid command parameter or switch. Retype the command with valid parameters or switches.

Invalid switch

Emulator error: The command line contains an invalid switch (optional parameter) character. The Emulator terminates.

Invalid time
Enter new time:

MS-DOS error: You typed the time incorrectly. Retype the time correctly using colons to separate the numbers:

00 to 23 for the hours
00 to 59 for the minutes
00 to 59 for the seconds

List output is not assigned to a device

PRINT error: Only the four physical devices assigned to the logical device name can be specified as a list device. (See the PORTSET command in Chapter 7.4.)

nnnnnn bytes of disk space freed

CHKDSK error: Disk space shown as allocated was not actually allocated and has been freed.

NO DISKETTE OR BAD TRACK

FORMAT error: If you have not inserted a diskette in the drive to be formatted, do so and restart the program. If the drive contains a diskette, take it out and retry with another diskette.

No Path

Entering the PATH command without parameters displays the previously entered PATH command. If no PATH command has been given, this message is displayed.

No room in directory to create files

RDCPM error: The directory on the MS-DOS diskette is full. Use a different MS-DOS diskette with fewer files, or delete files from the original diskette.

No space for COMMAND.COM

SYS error: There is not enough space on the destination disk for COMMAND.COM. Check the size of COMMAND.COM, using the DIR command, and remove enough files from the destination drive to make room.

No space for MSDOS.SYS

SYS error: There is not enough space on the disk for MSDOS.SYS (the system image). Delete some files and try again. Note that MSDOS.SYS must reside on contiguous disk space. Thus if the diskette is "checker boarded" (i.e., small, noncontiguous disk spaces available) you may have to delete many files. In this case, it is probably best to start with a newly formatted diskette; first do the SYS and then use the COPY command to copy the rest of the files.

Not emulated - function: hh

Emulator error: When a BDOS function which is not emulated is executed, this message appears. Press any key to continue the emulation. The function is hh in hexadecimal.

Not fully emulated - function: hh

Emulator error: This message is issued when a BDOS function which is not fully emulated is executed. The function is hh in hexadecimal.

Operating system version mismatch

FORMAT error: The operating system being used is not compatible with the version of FORMAT being used. Use the correct version of the program or load the correct operating system.

Overflow

Calculator error: Your calculation is longer than 14 digits. Type a shifted CALC (or CLEAR ENTRY) twice to abandon the calculation.

Read error: D = nn, T = nn, S = nn, E = nn
CANNOT READ TRACK

DISKCOPY or FORMAT error: Retry the program; discard the diskette if the error persists. (D stands for drive, T for track, S for sector, and E for error code. Error codes are defined in Table 8-2.)

Sector write error: D = nn, T = nn, S = nn, E = nn
CANNOT WRITE DISK LABEL

DISKCOPY or FORMAT error: The program is unable to write the disk label to the diskette. The disk label is part of the internal identification system for diskettes. The location and type of the problem are indicated by the drive, track, sector, and error code numbers. Error codes are defined in Table 8-2.

Soft format error: D = nn, T = nn, S = nn, E = nn

FORMAT error: A soft format error has occurred. The drive, track, sector, and error code numbers are indicated. Error codes are defined in Table 8-2.

If there are more than 9 soft errors on a track, the program aborts. Soft errors in moderate numbers do not hurt diskette performance.

Source and destination drives must not be the same

RDCPM error: The CP/M-86 file cannot be moved onto the same diskette in MS-DOS format. Specify separate drives in the command.

Source file name missing

RDCPM error: You typed the RDCPM command without a CP/M-86 file specifier. Retype the command with a CP/M-86 file specifier.

Source file not found

RDCPM error: The CP/M-86 filename given cannot be found on the CP/M-86 diskette. Retype the command with the correct filename or insert the correct CP/M-86 diskette.

Syntax error

You did not follow the command syntax. For example, you cannot specify a \ in the "(set)" parameter with the FOR command. Check your entry against the command documented in this manual.

Unable to create directory

MKDIR error: You already have an existing subdirectory with the same name. Re-enter the MKDIR (Make Directory) command.

MS-DOS cannot create a subdirectory unless a previous one exists. The new subdirectory is an entry in an already existing directory. Re-enter the MKDIR command.

VOLUME IN DRIVE IS NONEXISTENT OR IS NOT A FLOPPY.

SDCOPY error: There is no floppy in the disk drive.

WARNING - DOUBLE-SIDED DESTINATION DISK WILL BECOME SINGLE-SIDED AS A RESULT OF THIS OPERATION. DO YOU WISH TO CONTINUE? (y/n)

SDCOPY error: You have tried to copy a single-sided diskette to a double-sided diskette. Type y for yes, or n for no.

WRITE PROTECTED DISKETTE

FORMAT error: The diskette being formatted has a write-protect tab on it. Remove the tab and try again. **CAUTION:** Files already on the diskette will be erased if you do this.

WRONG DESTINATION DISK

SDCOPY error: The current destination diskette is different from the original one. This message appears in multiple insertions of the destination diskette. Insert the correct destination diskette.

WRONG SOURCE DISK

SDCOPY error: The current source diskette is different from the original one. This message applies to multiple insertions of the source diskette. Insert the correct source diskette.

8.4 ERROR CODES

Several MS-DOS error messages contain two-digit error codes. This section defines the error codes you may see in operating system error messages. Error codes displayed by the boot ROM are defined in Table 8-1 (see Chapter 8.2).

There are five general types of error codes:

1x	Cannot find sector
2x	Cannot read sector
3x	Sector verify error
4x	Formatting error
Fx	Miscellaneous

where x can be any digit 1–9 or letter A–E.

Table 8-2 lists and defines the error codes.

Table 8-1: Error Codes

ERROR CODE NUMBER	DEFINITION
COULD NOT COMPLETE READ-RELATED COMMAND	
11	Noise encountered on sync line
12	Bad header block ID
13	Checksum error in header
14	Header GCR error
15	Wrong track
16	Wrong sector
17	Bad job code
INVALID DATA ON DISKETTE	
21	Bad data block ID
22	Checksum error in data
23	GCR error
24	Sync time out

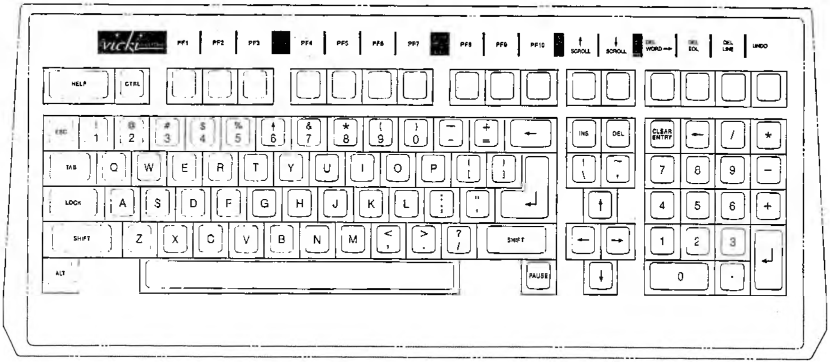
ERROR CODE NUMBER	DEFINITION
DEFECTIVE DRIVE OR DISKETTE	
31	Bad data block ID
32	Verify error
33	Checksum error
34	GCR error
FORMAT PROGRAM ERROR CODES	
41	No sync found (bad or missing diskette)
42	Bad header ID
43	Wrong track
44	Wrong sector
45	Bad header checksum
46	Gap error
47	GCR error
48	No data sync
49	Bad data ID
4A	Data verify error
4B	Data checksum
4C	Gap 2 error
4D	GCR error
MISCELLANEOUS	
F1	Cannot address second side of diskette
F2	Step error—cannot find track
F3	Data not written due to disk change
F4	Cannot write to disk until logged with ^C
F5	Wrong diskette type (wrong OS)
F6	Cannot start disk operation
F7	Illegal track number
F8	Illegal drive number
F9	Illegal disk operation
FA	Door open
FB	Drive motor not up to speed
FC	Write-protected diskette
FD	Bad track on diskette
FE	Cannot complete disk operation
FF	Bad or unformatted diskette

Appendixes

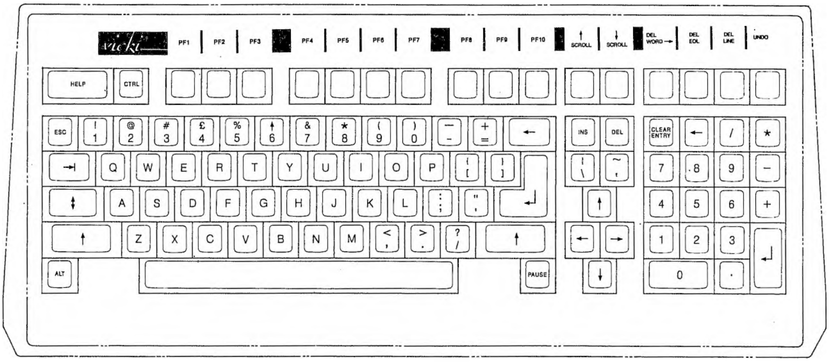
CHAPTERS

KEYBOARDS

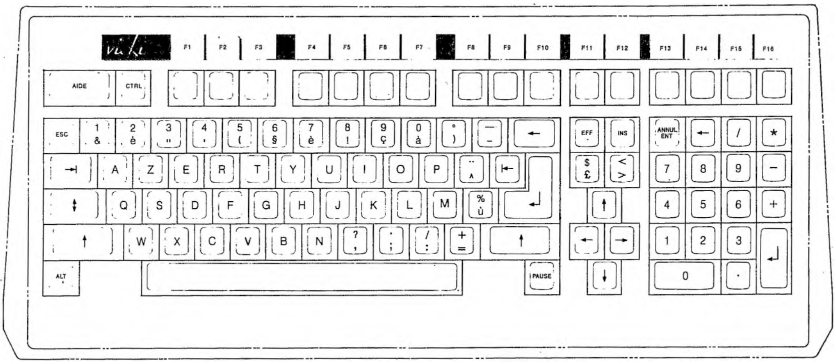
American Standard Keyboard (Portable)


vicki
PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10
↑ SCROLL
↓ SCROLL
DEL WORD →
DEL EOL
DEL LINE
UNDO
HELP CTRL
ESC ! 1 @ 2 # 3 $ 4 % 5 ↑ 6 & 7 * 8 (9) 0 — - + = ←
INS DEL
CLEAR ENTRY ← / *
TAB Q W E R T Y U I O P { [}]
| \ ~ ,
7 8 9 −
LOCK A S D F G H J K L : ; " '
4 5 6 +
SHIFT Z X C V B N M < , > . ? / SHIFT
1 2 3
ALT PAUSE
0 .

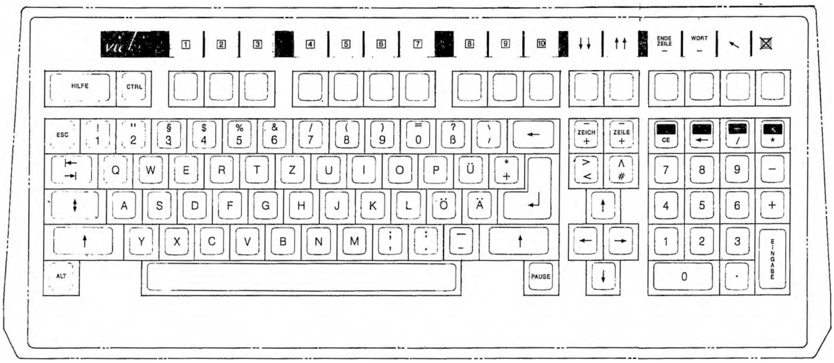
British Standard Keyboard (Portable)


vicki
PF1 PF2 PF3 PF4 PF5 PF6 PF7 PF8 PF9 PF10
↑ SCROLL
↓ SCROLL
DEL WORD →
DEL EOL
DEL LINE
UNDO
HELP CTRL
ESC ! 1 @ 2 # 3 £ 4 % 5 ↑ 6 & 7 * 8 (9) 0 — - + = ←
INS DEL
CLEAR ENTRY ← / *
→| Q W E R T Y U I O P { [}]
| \ ~ ,
7 8 9 −
A S D F G H J K L : ; " '
4 5 6 +
↑ Z X C V B N M < , > . ? / ↑
1 2 3
ALT PAUSE
0 .

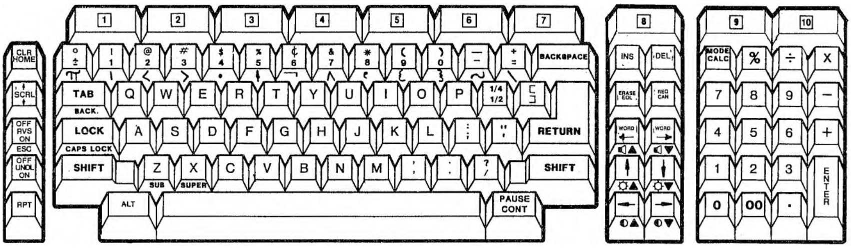
French Standard Keyboard (Portable)



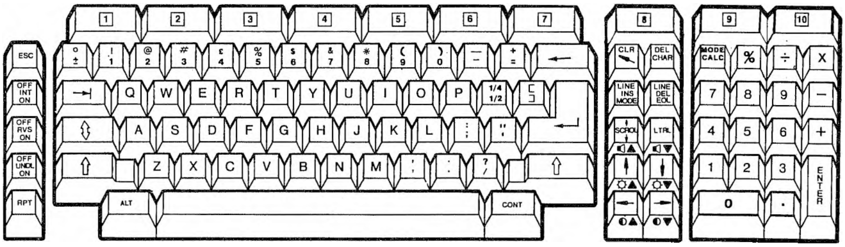
German Standard Keyboard (Portable)



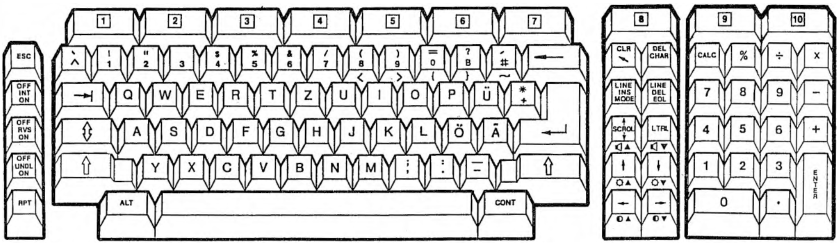
American Standard Keyboard (S1/V9000)



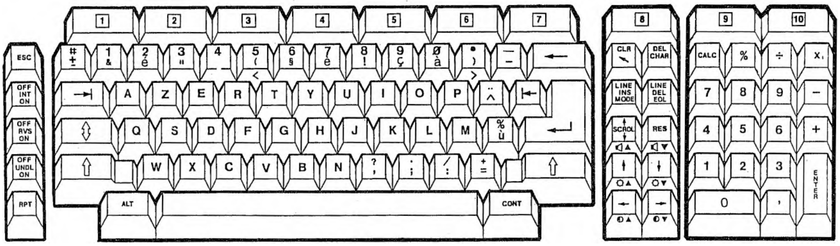
British Standard Keyboard (S1/V9000)



German Standard Keyboard (S1/V9000)



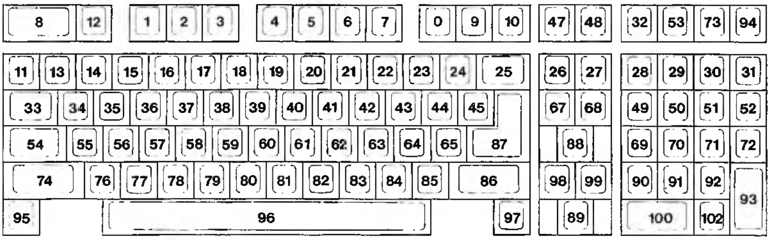
French Standard Keyboard (S1/V9000)



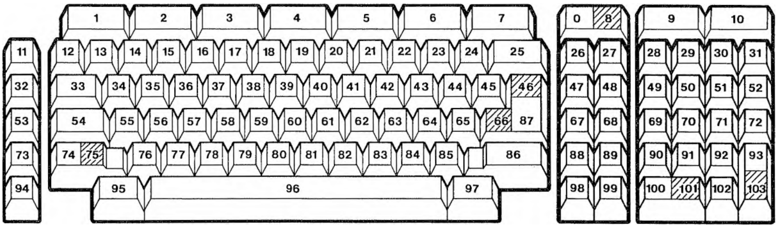
B

DIFFERENCES BETWEEN STANDARD KEYBOARDS

Logical Key Numbers for Standard Portable Keyboards



Logical Key Numbers for Standard S1/V9000 Keyboards



CHARACTER SETS

International Character Set

HIGH NIBBLE

LOW NIBBLE ➧	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	⌂
8:	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	£	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

British Standard Character Set

HIGH NIBBLE

LOW NIBBLE ➧	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	£	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	⌂
8:	Ç	ü	é	â	ä	à	å	ç	ê	ë	è	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ù	ÿ	Ö	Ü	¢	#	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

French Standard Character Set

LOW NIBBLE ➧

HIGH NIBBLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	£	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	à	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	°	ç	§	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	é	ù	è	¨	⌂
8:	~	ü	{	}	â	ä	å	ê	ë	[	]	ï	î	ì	Ä	Å
9:	É	æ	Æ	ô	ö	ò	û	ÿ	Ö	Ü	\	¢	#	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E:	α	β	Γ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

German Standard Character Set

LOW NIBBLE ➧

HIGH NIBBLE

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0:		☺	☻	♥	♦	♣	♠	•	◘	○	◙	♂	♀	♪	♫	☼
1:	►	◄	↕	‼	¶	§	▬	↨	↑	↓	→	←	∟	↔	▲	▼
2:		!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3:	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4:	§	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5:	P	Q	R	S	T	U	V	W	X	Y	Z	Ä	Ö	Ü	^	_
6:	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7:	p	q	r	s	t	u	v	w	x	y	z	ä	ö	ü	ß	⌂
8:	~	{	}	â	à	å	ê	è	é	ë	[	]	î	ì	ï	Å
9:	É	æ	Æ	ô	ò	ù	û	ÿ	ç	°	\	¢	£	¥	₧	ƒ
A:	á	í	ó	ú	ñ	Ñ	ª	º	¿	⌐	¬	½	¼	¡	«	»
B:	░	▒	▓	│	┤	╡	╢	╖	╕	╣	║	╗	╝	╜	╛	┐
C:	└	┴	┬	├	─	┼	╞	╟	╚	╔	╩	╦	╠	═	╬	╧
D:	╨	╤	╥	╙	╘	╒	╓	╫	╪	┘	┌	█	▄	▌	▐	▀
E:	α	Γ	λ	π	Σ	σ	µ	τ	Φ	Θ	Ω	δ	∞	φ	ε	∩
F:	≡	±	≥	≤	⌠	⌡	÷	≈	°	∙	·	√	ⁿ	²	■	

D

DISPLAY DRIVER SPECIFICATIONS

INTRODUCTION D.1

The software in the BIOS display interface receives ASCII characters and either displays them or uses them for display control. The characters that are not displayed are the ASCII control characters (hex 00 through hex 1F and hex 7F), except ESC 8. These characters may, however, affect the display. Most of the control characters act independently. But for some actions, more than one control character is required to specify the action. This is done by using the ASCII escape code (hex 1B), followed by one or more characters. The control characters and the escape sequences are described in D.2 and D.3.

The display controls are accessed from high-level languages by using the PRINT or equivalent statements. For example, in MS-BASIC this command turns on reverse video:

```
PRINT CHR$(27) + CHR$(112)
```

NOTE: Decimal 27 equals hex 1B (Escape) and decimal 112 is hex 70. (See Table D-7.)

D.2 CONTROL CHARACTERS

- ▶ Bell (ALT-G, hex 07): Sends a series of signals that make the sound of a bell. Not really a display control character. Can also be used with the optional CODEC.
- ▶ Backspace (ALT-H, hex 08): Moves the cursor one column to the left. If cursor is at column 1, it moves to column 80 of the previous row (unless cursor is at column 1, row 1, in which case it moves to column 80, row 1).
- ▶ Horizontal Tab (ALT-I, hex 09): Moves the cursor to the next tab stop. Tab stops are fixed at columns 9, 17, 25, 33, 41, 49, 57, 65, and 72 through 80. If the cursor is at column 80, it remains there.
- ▶ Line Feed (ALT-J, hex 0A): Moves the cursor down one row. If cursor is at row 24, then the display scrolls up one row. (Line feed can also be treated as a return. See ESC x9.)
- ▶ Carriage Return (ALT-M, hex 0D): Moves the cursor to column 1 of the current row. (Return can also be treated as a line feed. See ESC x8.)
- ▶ Shift In (ALT-N, hex 0E): Shifts to character set 1 (G1).
- ▶ Shift Out (ALT-0, hex 0F): Shifts to character set 0 (G0).

Table D-1: Alternate Sequences

ALTERNATE SEQUENCE	HEX CODE
ALT-@	00
ALT-A	01
ALT-B	02
ALT-C	03
ALT-D	04
ALT-E	05
ALT-F	06
ALT-G (Bell)	07
ALT-H (Backspace)	08
ALT-I (Tab)	09
ALT-J (Line Feed)	0A
ALT-K	0B
ALT-L	0C
ALT-M (Carriage Return)	0D
ALT-N	0E
ALT-O	0F
ALT-P	10
ALT-Q	11
ALT-R	12
ALT-S	13
ALT-T	14
ALT-U	15
ALT-V	16
ALT-W	17
ALT-X	18
ALT-Y	19
ALT-Z	1A
ALT-[(Escape)	1B
ALT-\	1C
ALT-]	1D
ALT-^	1E

D.4 ESCAPE SEQUENCES

Table D-2: Cursor Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
ESC A	1B, 41	Moves the cursor up one line.
ESC B	1B, 42	Moves the cursor down one line without changing columns.
ESC C	1B, 43	Moves the cursor one character position to the right.
ESC D	1B, 44	Moves the cursor one character position to the left.
ESC H	1B, 48	Moves the cursor to the home position (upper left corner of screen).
ESC I	1B, 49	Moves the cursor up one line, and stays in the same column.
ESC n	1B, 6E	Reports the cursor position.
ESC j	1B, 6A	The display driver saves the cursor position.
ESC k	1B, 6B	Returns the cursor to the previously saved cursor position.
ESC Y[l][c]	1B, 59	Moves the cursor via direct cursor addressing, where [l] is the line number (from 1 to 25) and [c] is the column number (from 1 to 80). No movement occurs if l and/or c are invalid. Table D-3 gives the printing character you can use to represent the cursor coordinates in this escape sequence. You use two characters—one for the line number (1–25) and one for the column number (1–80)—to address a cursor position on the screen. You can also use hex values: lines are 1 to 19 hex; columns are 1 to 50 hex. You must add the proper line and column numbers to 1F hex.

Table D-3: Characters for Direct Cursor Addressing

LINE OR COLUMN	CHARACTER	LINE OR COLUMN	CHARACTER
1	(space)	41	H
2	!	42	I
3	"	43	J
4	#	44	K
5	$	45	L
6	%	46	M
7	&	47	N
8	'	48	O
9	(	49	P
10	)	50	Q
11	*	51	R
12	+	52	S
13	,	53	T
14	-	54	U
15	.	55	V
16	/	56	W
17	0	57	X
18	1	58	Y
19	2	59	Z
20	3	60	[
21	4	61	\
22	5	62	]
23	6	63	^
24	7	64	_
25	8	65	`
26	9	66	a
27	:	67	b
28	;	68	c
29	<	69	d
30	=	70	e
31	>	71	f
32	?	72	g
33	@	73	h
34	A	74	i
35	B	75	j
36	C	76	k
37	D	77	l
38	E	78	m
39	F	79	n
40	G	80	o

Table D-4: Editing Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
ESC E	1B, 45	Erases the entire screen.
ESC b	11B, 62	Erases from the start of the screen up to (and including) the cursor position.
ESC J	1B, 4A	Erases from the cursor position to the end of the page.
ESC l	1B, 6C	Erases entire line.
ESC o	1B, 6F	Erases from the beginning of line up to (and including) the cursor position.
ESC K	1B, 4B	Erases from cursor position to the end of the line.
ESC L	1B, 4C	Inserts a blank line. The current line and all following lines scroll down one line. The cursor moves to the beginning of the blank line.
ESC M	1B, 4D	Moves cursor to beginning of line, deletes the line, and then scrolls all following lines up one line. A blank line is inserted at line 24.
ESC N	1B, 4E	Deletes character at cursor position and shifts the rest of the line one character position to the left.
ESC @	1B, 40	Enters the Insert Character mode. This lets you insert characters into screen text. As each new character is inserted, the character at the end of the line is lost.
ESC O	1B, 4F	Exits Insert Character mode.

Table D-5: Configuration Functions

<table>
<tr><th>ESCAPE
SEQUENCE</th><th>ASCII CODE
GENERATED
(HEXADECIMAL)</th><th>SEQUENCE DEFINITION</th></tr>
<tr><td>ESC x[P1...Px]</td><td>1B, 78</td><td>Sets mode(s) according to the parameters you name (P1...Px), as follows:
P Mode
1 Enable 25th line.
3 Hold screen mode on.
4 Use block cursor.
5 Cursor off.
8 Automatic line feed after a Return.
9 Automatic Return after a line feed.
A Turn beeper on.
B Increase CRT brightness.
C Increase CRT contrast.</td></tr>
<tr><td>ESC y[P1...Px]</td><td>1B, 79</td><td>Resets mode(s) according to the parameters you name (P1...Px), as follows:
P Mode
1 Disable 25th line.
3 Hold screen mode off.
4 Use underscore cursor.
5 Cursor on.
8 No auto line feed.
9 No auto Return.
A Turn beeper off.
B Decrease CRT brightness.
C Decrease CRT contrast.</td></tr>
<tr><td>ESC ^</td><td>1B, 5E</td><td>Toggle hold mode.</td></tr>
<tr><td>ESC [</td><td>1B, 5B</td><td>Set hold mode.</td></tr>
<tr><td>ESC \</td><td>1B, 5C</td><td>Clear hold mode.</td></tr>
<tr><td>ESC |</td><td>1B, 7C</td><td>Activate user-defined console.</td></tr>
</table>

Table D-6: Operation Mode Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
ESC p	1B, 70	Enters reverse video mode.
ESC q	1B, 71	Exits reverse video mode.

Table D-7: Special Functions

ESCAPE SEQUENCE	ASCII CODE GENERATED (HEXADECIMAL)	SEQUENCE DEFINITION
ESC }	1B, 7D	Disables the keyboard.
ESC {	1B, 7B	Enables the keyboard.
ESC v	1B, 76	Enables wrap-around at the end of the line.
ESC w	1B, 77	Disables wrap-around at the end of the line.
ESC z	1B, 7A	Resets terminal to power-on configuration.
ESC $	1B, 24	Transmits the character at cursor location.
ESC]	1B, 5D	Transmits the 25th line.
ESC #	1B, 23	Transmits the page.
ESC (	1B, 28	Sets high intensity.
ESC)	1B, 29	Sets low intensity.
ESC +	1B, 2B	Clears the foreground. (High-intensity displayed characters.)
ESC Z	1B, 5A	Identifies display as emulating VT52 (the terminal responds with an ESC K).
ESC 0	1B, 30	Sets the underline mode.
ESC 1	1B, 31	Resets the underline mode.
ESC 2	1B, 32	Enables cursor blink.
ESC 3	1B, 33	Disables cursor blink.
ESC 8	1B, 38	Sets the test (literally) mode for the next single character.
ESC i[n]	1B, 69	Displays the system sign-on banner, as follows (n is the ASCII numeric character): n Display 0 Entire banner. 1 Company logo only. 2 Product name only. 3 Configuration information only.

Table D-8: ASCII Conversion Table

In the column headings, Dec means decimal, Hex means hexadecimal (H) and CHR means character.

Dec	Hex	CHR	Dec	Hex	CHR	Dec	Hex	CHR
000	00H	NUL	043	2BH	+	086	56H	V
001	01H	SOH	044	2CH	,	087	57H	W
002	02H	STX	045	2DH	-	088	58H	X
003	03H	ETX	046	2EH	.	089	59H	Y
004	04H	EOT	047	2FH	/	090	5AH	Z
005	05H	ENQ	048	30H	0	091	5BH	[
006	06H	ACK	049	31H	1	092	5CH	\
007	07H	BEL	050	32H	2	093	5DH	]
008	08H	BS	051	33H	3	094	5EH	^
009	09H	HT	052	34H	4	095	5FH	_
010	0AH	LF	053	35H	5	096	60H	`
010	0BH	VT	054	36H	6	097	61H	a
012	0CH	FF	055	37H	7	098	62H	b
013	0DH	CR	056	38H	8	099	63H	c
014	0EH	SO	057	39H	9	100	64H	d
015	0FH	SI	058	3AH	:	101	65H	e
016	10H	DLE	059	3BH	;	102	66H	f
017	11H	DC1	060	3CH	<	103	67H	g
018	12H	DC2	061	3DH	=	104	68H	h
019	13H	DC3	062	3EH	>	105	69H	i
020	14H	DC4	063	3FH	?	106	6AH	j
021	15H	NAK	064	40H	@	107	6BH	k
022	16H	SYN	065	41H	A	108	6CH	l
023	17H	ETB	066	42H	B	109	6DH	m
024	18H	CAN	067	43H	C	110	6EH	n
025	19H	EM	068	44H	D	111	6FH	o
026	1AH	SUB	069	45H	E	112	70H	p
027	1BH	ESCAPE	070	46H	F	113	71H	q
028	1CH	FS	071	47H	G	114	72H	r
029	1DH	GS	072	48H	H	115	73H	s
030	1EH	RS	073	49H	I	116	74H	t
031	1FH	US	074	4AH	J	117	75H	u
032	20H	(sp)	075	4BH	K	118	76H	v
033	21H	!	076	4CH	L	119	77H	w
034	22H	″	077	4DH	M	120	78H	x
035	23H	#	078	4EH	N	121	79H	y
036	24H	$	079	4FH	O	122	7AH	z
037	25H	%	080	50H	P	123	7BH	{
038	26H	&	081	51H	Q	124	7CH	|
039	27H	′	082	52H	R	125	7DH	}
040	28H	(	083	53H	S	126	7EH	~
041	29H	)	084	54H	T	127	7FH	DEL
042	2AH	*	085	55H	U			

Note: LF = Line feed, FF = Form feed, CR = Carriage return, and DEL = Delete.

D.5 132-COLUMN UTILITY

DESCRIPTION

The 132-column utility (132C) provides a simulated 132-column display in the 800-dot by 400-line HIRES display mode of your computer. The characters are displayed in a 5-by-7 dot matrix in a 6-by-10 cell to give the 132-column display. In addition, a standard display of 80 columns by 25 lines is simulated with an 8-by-11 dot matrix in a 10-by-16 cell. The 132-column mode is enabled when the normal BIOS display interface recognizes the appropriate escape sequence.

INSTALLATION

Enter this sequence to activate the 132-column feature for use with an application program:

```
132C(cr)
132ON(cr)
```

You can combine these commands in a batch (.BAT) file if you like.

The 132C feature is turned off with the command:

```
132OFF(cr)
```

The 132C program installs itself into the BIOS. The program requires about 50K of memory: the 800-by-400-line HIRES screen requires 40K; the character sets and code use the rest. The 132C program copies itself into the lowest 64K block of memory and removes its 50K required bytes from the system until you press the reset button or turn off your computer.

You can access 132C, if installed from an application program, by sending an ESC | (1B 7C) to the display interface. This is done from MS-BASIC with:

```
PRINT CHR$(27) + CHR$(124)
```

From MS-BASIC, 132C is turned off with an ESC z (1B 7A):

```
PRINT CHR$(27) + "z"
```

The display control sequences described above are also available in 132C mode.

When you've finished and saved your work, press the reset button to release the memory allocated to 132C and return to normal character mode.

ATTACHING PERIPHERAL DEVICES

This appendix gives step by step instructions for attaching peripheral devices to your portable computer. Instructions are also included for constructing a cable to connect a device to your computer.

GENERAL PROCEDURE E.1

To attach a peripheral device to your computer, use the procedure described here and in Section I, Chapter 1.4 of your *Operator's Reference Guide.* This procedure uses a printer as an example of a peripheral device.

1. Obtain a serial or parallel cable and connectors that enable your computer to "talk to" the printer. A standard Centronics cable works for almost any parallel interface device. Serial device cables (RS232C), however, may have different wiring patterns. If you do not have a working cable, a cable must be constructed to match the requirements of the printer and the computer (see Appendix E.2).
2. Connect the cable to the printer and to the computer. Serial cables and some parallel cables have distinct ends for the computer and for the device. IEEE-488 cables can piggy-back to connect several devices to the parallel port.
3. Adjust the settings on the printer to match the requirements of your computer. Printer settings are usually made via dip switches on the front or inside of the printer.
   a. You may have to set the parity, stop bits, and baud rate. These settings control the rate and type of data transmission. The values to use for these settings are given in the DISKID file on your operating system distribution diskettes.

b. If the baud rate required for the printer does not match the computer's, you can temporarily change the computer's baud rate using the PORTSET utility (see Chapter 7.4). You can change the baud rate assigned at boot time using the DEVICE command in the CONFIG.SYS file (see Chapter 7.3).

4. Configure the operating system to send output to the port where the device is connected. To do this, use the CLST command, described in Chapter 7.4. With CLST you can check the current assignments for the output ports, and you can assign output to a different port.

E.2 CONNECTORS AND PIN ASSIGNMENTS

This section describes the connectors for serial and for parallel peripheral devices. Step by step instructions describe how to construct serial and parallel cables. The pin assignments that determine the wiring pattern in the cable are listed for serial, parallel, IEEE-488, and MODEM devices. If your dealer can provide you with a working cable, you do not need to refer to this section.

E.2.1 SERIAL INTERFACE CONNECTION

Your dealer can supply you with a tested cable for several standard serial printers. If you do not have a working serial cable, you can construct one as follows:

1. For the cable, you need a length of multiconductor cable to reach from your computer to the device (maximum length 50 feet). Depending on your application, you need from 5- to 25-conductor cable. If you are uncertain about your needs, use a 25-conductor cable.

2. For the connector to your computer, you need one D25 male connector. See Figure E-1.
3. For the connector to the printer or other device, you need a D25 connector (male or female, to fit the connector on the device).
4. Locate the port layout in your printer manual and compare it to the pin assignments for your computer's serial port, listed in Table E-1.

Figure E-1: Serial Connector

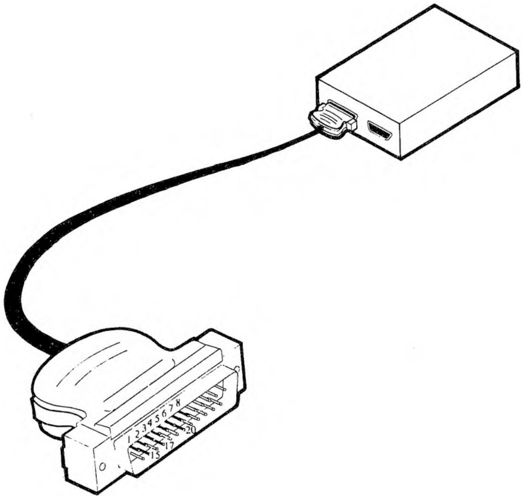


Table E-1: Serial Interface Pin Numbers

PIN	SIGNAL		SOURCE	CODE	V.24
1	FG	Frame Ground		AA	101
2	TD	Transmitted Data	out	BA	103
3	RD	Received Data	in	BB	104
4	RTS	Request To Send	out	CA	105
5	CTS	Clear To Send	in	CB	106
6	DSR	Data Set Ready*	in	CC	107
7	SG	Signal Ground		AB	102
8	DCD	Data Carrier Detect	in	CF	109
15	TC	Transmitter Clock*	in	DB	114
17	RC	Receiver Clock*	in	DD	115
20	DTR	Data Terminal Ready	out	CD	108.2
22	RI	Ring Indicator*	in	CE	125
24	TC	External Transmit Clock	out	DA	113

*These pins are used for connecting a MODEM device and are not usually required for printers.

CODE refers to the EIA interface standard (Electronics Industries Association).

V.24 refers to the CCIT,T interface standard (International Telephone and Telephone Consultative Committee).

5. If pin 3 of the serial device is Received Data (RD or RXD), and if pin 20 is Data Terminal Ready (DTR), then construct the cable as follows. Label one connector "Computer" and the other connector "Printer." Then connect the cable wires from the pins on the Computer connector to the pins on the Printer connector, as shown here:

Computer		Serial Printer
1	——	1
2	——	3
3	——	2
7	——	7
8, 5	——	20

6. If pin 3 of the serial device is Received Data (RD or RXD), and pin 20 is not Data Terminal Ready (DTR), construct the cable as follows. Label one connector "Computer" and the other "Printer." Connect the cable wires from the pins on the Computer connector to the pins on the Printer connector, as shown here:

Computer		Serial Printer
1	——	1
3	——	2
2	——	3
7	——	7
8, 5	——	4

PARALLEL INTERFACE CONNECTION E.2.2

If you do not have a working parallel (Centronics) cable, you can construct one as follows:

1. For the cable, you need a length of 488 cable to reach from the computer to the printer.
2. For the end of the cable that attaches to your computer, you need a male Centronics-compatible Amphenol 57-30360 type connector.
3. For the end of the cable that attaches to your printer or other device, use the type of connector suggested by the manufacturer. The connector is likely to be the same at both ends of the cable.
4. Locate the port layout and pin assignments in your printer manual, and compare them with the pin numbers and signal assignments for the parallel port of your computer, listed in Table E-2.

Table E-2: Parallel Interface Pin Numbers

PIN	SIGNAL
1	Data Strobe
2	Data 1
3	Data 2
4	Data 3
5	Data 4
6	Data 5
7	Data 6
8	Data 7
9	Data 8
10	ACK
11	Busy
17	Pshield
12, 18, 30, 31	Not connected
Remaining	GND

5. If the pin numbers and signal requirements for the printer are the same as for the computer, construct the cable as follows:

Computer		Device
1	——	1
2	——	2
3	——	3
4	——	4
5	——	5
6	——	6
7	——	7
8	——	8
9	——	9
10	——	10
11	——	11
16	——	16

Because this cable is constructed with one-for-one pin assignments, you can attach either end to your computer or printer.

6. If the printer has the same signals as your computer, but on different pins, follow these steps to construct a cable:

   a. Label one connector "Computer" and the other "Printer."

   b. Connect pin 1 at the Computer connector to the Data strobe pin on the Printer connector.

   c. Connect pins 2 through 9 at the Computer connector to Data 1 through Data 8 (which may be labelled Data 0 through Data 7) at the Printer connector.

   d. Connect pin 10 at the Computer connector to the ACK pin at the Printer connector.

   e. Connect pin 11 at the Computer connector to the Busy pin at the Printer connector.

7. Attach the "Computer" end of the parallel cable to your computer, and attach the "Printer" end to the parallel device.

IEEE-488 INTERFACE CONNECTION E.2.3

The IEEE-488 cable attaches to the parallel port on your computer. Use the pin assignments listed in Table E-3 to construct an IEEE-488 cable. In the table, the pin number refers to the actual computer port connector. The IEEE-488 pin number refers to the standard IEEE-488 pin-out as it must attach to the parallel port.

Table E-3: IEEE-488 Pin Assignments

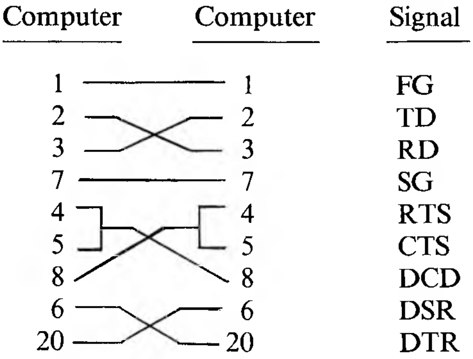
PIN NUMBER		IEEE SIGNAL		IEEE PIN NUMBER
1	———	DAV	———	6 **a
19	———	GND	———	18 **a
2	———	DIO1	———	1
3	———	DIO2	———	2
4	———	DIO3	———	3
5	———	DIO4	———	4
6	———	DIO5	———	13
7	———	DIO6	———	14
8	———	DIO7	———	15
9	———	DIO8	———	16
10	———	NRFD	———	7 **b
28	———	GND	———	19 **b
11	———	SRQ	———	10 **c
29	———	GND	———	22 **c
13	———	NDAC	———	8 **d
33	———	GND	———	20 **d
15	———	EOI	———	5
17	———	shield	———	12
34	———	REN	———	17
35	———	ATN	———	11 **e
16	———	GND	———	23 **e
36	———	IFC	———	9 **f
27	———	GND	———	21 **f
20	———	GND	———	24

**These wires are to be bound together as twisted pairs. The letter following the ** indicates the twisted pair where the wire goes.

E.2.4 MODEM DEVICE CONNECTION

A MODEM (modulator/demodulator) device attaches to the serial port of your computer. Your dealer can supply you with a tested cable for use with some types of MODEM devices. This section gives the pin assignments for three types of MODEM cables: a null MODEM cable for connecting one computer to another (async only); a MODEM cable using asynchronous transmission; and a MODEM cable using synchronous transmission.

Null MODEM Cable



(Signals are crossed in cable because no MODEM is used.)

Asynchronous MODEM Cable

Computer		MODEM	Signal
1	——	1	FG
2	——	2	TD
3	——	3	RD
4	——	4	RTS
5	——	5	CTS
6	——	6	DSR
7	——	7	SG
8	——	8	DCD
20	——	20	DTR

Synchronous MODEM Cable

Computer		MODEM	Signal
1	——	1	FG
2	——	2	TD
3	——	3	RD
7	——	7	SG
4	——	4	RTS
5	——	5	CTS
8	——	8	DCD
6	——	6	DSR
16	——	15	TC
17	——	17	RC
20	——	20	DTR

F

KEYBOARD CONSIDERATIONS

This appendix gives you the character or function generated by each key position of the keyboards available with your computer. Table F-1 lists each key position in shifted, unshifted, and Alternate modes. The key positions (logical key numbers) for the standard keyboards are shown in Appendix B.

For each key position, Table F-1 lists one of the following:

- The actual character (letter, number, or symbol) generated by that key. You can look up the hexadecimal code for the character in the appropriate character set illustration in Appendix C.
- The hexadecimal code of the character generated by that key. You can look up the character represented by this code in the appropriate character set illustration in Appendix C.
- The function generated by that key (such as BKSP, DEL, ALT, and ESC). Appendix D contains a list of all the Escape sequences and Alternate sequences.

Some listings include both the code and the character generated (such as F8 ° for key number 8).

Table F-1: Keycodes and Characters

<table>
<tr><th>KEY POSITION</th><th>STAND.</th><th>DOMESTIC</th><th>FRENCH</th><th>BRITISH</th><th>GERMAN</th><th>ITALIAN</th></tr>
<tr><td>0 unshifted</td><td>F8 °</td><td>F8 °</td><td>F8 |</td><td>F8 °</td><td>|</td><td>F8 °</td></tr>
<tr><td>0 shifted</td><td>F8 °</td><td>F8 °</td><td>F8 |</td><td>F8 °</td><td>|</td><td>F8 °</td></tr>
<tr><td>0 alternate</td><td>F8 °</td><td>F8 °</td><td>F8 |</td><td>F8 °</td><td>|</td><td>F8 °</td></tr>
<tr><td>1 unshifted</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td></tr>
<tr><td>1 shifted</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td></tr>
<tr><td>1 alternate</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td><td>F1 ±</td></tr>
<tr><td>2 unshifted</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td></tr>
<tr><td>2 shifted</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td></tr>
<tr><td>2 alternate</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td><td>F2 ≥</td></tr>
</table>

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
3 unshifted	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤
3 shifted	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤
3 alternate	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤	F3 ≤
4 unshifted	F4	F4	F4	F4	F4	F4
4 shifted	F4	F4	F4	F4	F4	F4
4 alternate	F4	F4	F4	F4	F4	F4
5 unshifted	F5	F5	F5	F5	F5	F5
5 shifted	F5	F5	F5	F5	F5	F5
5 alternate	F5	F5	F5	F5	F5	F5
6 unshifted	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷
6 shifted	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷
6 alternate	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷	F6 ÷
7 unshifted	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃
7 shifted	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃
7 alternate	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃	F7 ≃
8 unshifted	F8 °	F8 °	F8 |	F8 °	|	F8 °
8 shifted	F8 °	F8 °	F8 |	F8 °	|	F8 °
8 alternate	F8 °	F8 °	F8 |	F8 °	|	F8 °
9 unshifted	F9	F9	F9	F9	F9	F9
9 shifted	F9	F9	F9	F9	F9	F9
9 alternate	F9	F9	F9	F9	F9	F9
10 unshifted	FA	FA	FA	FA	FA	FA
10 shifted	FA	FA	FA	FA	FA	FA
10 alternate	FA	FA	FA	FA	FA	FA
11 unshifted	ESC	ESC H	ESC	ESC	ESC	ESC
11 shifted	ESC	ESC E	ESC	ESC	ESC	ESC
11 alternate	ESC	ESC E	ESC	ESC	ESC	ESC
12 unshifted	\	\	23 £	\	'	*
12 shifted	F8 °	F8 °	$	*Null*	'	A8 §
12 alternate	E3	E3	$	*Null*	'	A8 §
13 unshifted	¡	¡	&	!	1	9C
13 shifted	!	!	1	!	!	1
13 alternate	¦	¦	1	¦	!	¦
14 unshifted	2	2	7B é	2	2	82 é
14 shifted	@	@	2	@	"	2
14 alternate	<	<	2	<	"	<
15 unshifted	3	3	"	3	3	"
15 shifted	#	#	3	3	40 §	3
15 alternate	>	>	3	>	40 §	>
16 unshifted	4	4	'	4	4	'
16 shifted	$	$	4	23 £	$	4
16 alternate	F9	F9	4	23 £	$	#
17 unshifted	5	5	(	5	5	(
17 shifted	%	%	5	%	%	5
17 alternate	ALT-X	ALT-X	<	%	%	@
18 unshifted	6	6	5D §	6	6	_
18 shifted	9B ¢	9B ¢	6	$	&	6
18 alternate	AA	AA	6	$	&	[

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
19 unshifted	7	7	7D è	7	7	8A è
19 shifted	&	&	7	&	\	7
19 alternate	~	~	7	~	7	]
20 unshifted	8	8	!	8	8	^
20 shifted	*	*	8	*	(	8
20 alternate	·	·	*	·	<	·
21 unshifted	9	9	5C ç	9	9	87 ç
21 shifted	(	(	9	(	)	9
21 alternate	{	{	9	{	>	{
22 unshifted	0	0	40 à	0	0	85 à
22 shifted	)	)	0	)	=	0
22 alternate	}	}	0	}	=	}
23 unshifted	-	-	)	-	7E	)
23 shifted	_	_	5B °	_	?	F8 °
23 alternate	`	`	>	`	?	~
24 unshifted	=	=	-	=	#	-
24 shifted	+	+	_	+	'	+
24 alternate	\	\	–	\	`	\
25 unshifted	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H
25 shifted	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H
25 alternate	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H
26 unshifted	ESC H	BB	ESC H	ESC H	ESC H	ESC H
26 shifted	ESC E	CB	ESC E	ESC E	ESC E	ESC E
26 alternate	ESC z	DB	ESC z	ESC z	ESC z	ESC z
27 unshifted	DEL	DEL	DEL	DEL	DEL	DEL
27 shifted	DEL	CC	DEL	DEL	DEL	DEL
27 alternate	DEL	DC	DEL	DEL	DEL	DEL
28 unshifted	=	=	=	=	=	=
28 shifted	FD	FD	FD	FD	FD	FD
28 alternate	FD	=	FD	FD	FD	FD
29 unshifted	%	%	%	%	%	%
29 shifted	%	%	%	%	%	%
29 alternate	%	%	%	%	%	%
30 unshifted	/	/	/	/	/	/
30 shifted	/	/	/	/	/	/
30 alternate	/	/	/	/	/	/
31 unshifted	*	*	*	*	*	*
31 shifted	*	*	*	*	*	*
31 alternate	*	*	*	*	*	*
32 unshifted	ESC (	BD	ESC (	ESC (	ESC (	ESC (
32 shifted	ESC)	CD	ESC)	ESC)	ESC)	ESC)
32 alternate	ESC)	DD	ESC)	ESC)	ESC)	ESC)
33 unshifted	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I
33 shifted	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I
33 alternate	ALT-I	E9	ALT-I	ALT-I	ALT-I	ALT-I
34 unshifted	q	q	a	q	q	q
34 shifted	Q	Q	A	Q	Q	Q
34 alternate	ALT-Q	ALT-Q	ALT-A	ALT-Q	ALT-Q	ALT-Q

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
35 unshifted	w	w	z	w	w	z
35 shifted	W	W	Z	W	W	Z
35 alternate	ALT-W	ALT-W	ALT-Z	ALT-W	ALT-W	ALT-Z
36 unshifted	e	e	e	e	e	e
36 shifted	E	E	E	E	E	E
36 alternate	ALT-E	ALT-E	ALT-E	ALT-E	ALT-E	ALT-E
37 unshifted	r	r	r	r	r	r
37 shifted	R	R	R	R	R	R
37 alternate	ALT-R	ALT-R	ALT-R	ALT-R	ALT-R	ALT-R
38 unshifted	t	t	t	t	t	t
38 shifted	T	T	T	T	T	T
38 alternate	ALT-T	ALT-T	ALT-T	ALT-T	ALT-T	ALT-T
39 unshifted	y	y	y	y	z	y
39 shifted	Y	Y	Y	Y	Z	Y
39 alternate	ALT-Y	ALT-Y	ALT-Y	ALT-Y	ALT-Z	ALT-Y
40 unshifted	u	u	u	u	u	u
40 shifted	U	U	U	U	U	U
40 alternate	ALT-U	ALT-U	ALT-U	ALT-U	ALT-U	ALT-U
41 unshifted	i	i	i	i	i	i
41 shifted	I	I	I	I	I	I
41 alternate	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I	ALT-I
42 unshifted	o	o	o	o	o	o
42 shifted	O	O	O	O	O	O
42 alternate	ALT-O	ALT-O	ALT-O	ALT-O	ALT-O	ALT-O
43 unshifted	p	p	p	p	p	p
43 shifted	P	P	P	P	P	P
43 alternate	ALT-P	ALT-P	ALT-P	ALT-P	ALT-P	ALT-P
44 unshifted	AB	AB		*Null*	7D ü	8D ì
44 shifted	AC	AC	7E ¨	*Null*	5D Ü	=
44 alternate	AC	AC	7E ¨	*Null*	ALT-]	=
45 unshifted	]	]	~	]	+	$
45 shifted	[	[	~	[	*	&
45 alternate	ALT-[(Esc)	ALT-[(Esc)	~	ALT-[(Esc)	*	$
46 unshifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
46 shifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
46 alternate	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
47 unshifted	ESC @	BE	ESC @	ESC @	ESC @	ESC @
47 shifted	ESC L	CE	ESC L	ESC L	ESC L	ESC L
47 alternate	ESC O	DE	ESC O	ESC O	ESC O	ESC O
48 unshifted	ESC K	BF	ESC K	ESC K	ESC K	ESC K
48 shifted	ESC M	CF	ESC M	ESC M	ESC M	ESC M
48 alternate	ESC M	DF	ESC M	ESC M	ESC M	ESC M
49 unshifted	7	7	7	7	7	7
49 shifted	7	7	7	7	7	7
49 alternate	7	7	7	7	7	7
50 unshifted	8	8	8	8	8	8
50 shifted	8	8	8	8	8	8
50 alternate	8	8	8	8	8	8

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
51 unshifted	9	9	9	9	9	9
51 shifted	9	9	9	9	9	9
51 alternate	9	9	9	9	9	9
52 unshifted	-	-	-	-	-	-
52 shifted	-	-	-	-	-	-
52 alternate	-	-	-	-	-	-
53 unshifted	ESC p	ESC p	ESC p	ESC p	ESC p	ESC p
53 shifted	ESC q	ESC q	ESC q	ESC q	ESC q	ESC q
53 alternate	ESC q	ESC	ESC q	ESC q	ESC q	ESC q
54 unshifted	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*
54 shifted	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*	*Caps Lock*
54 alternate	*Shift Lock*	*Shift Lock*	*Shift Lock*	*Shift Lock*	*Shift Lock*	*Shift Lock*
55 unshifted	a	a	q	a	a	a
55 shifted	A	A	Q	A	A	A
55 alternate	ALT-A	ALT-A	ALT-Q	ALT-A	ALT-A	ALT-A
56 unshifted	s	s	s	s	s	s
56 shifted	S	S	S	S	S	S
56 alternate	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S
57 unshifted	d	d	d	d	d	d
57 shifted	D	D	D	D	D	D
57 alternate	ALT-D	ALT-D	ALT-D	ALT-D	ALT-D	ALT-D
58 unshifted	f	f	f	f	f	f
58 shifted	F	F	F	F	F	F
58 alternate	ALT-F	ALT-F	ALT-F	ALT-F	ALT-F	ALT-F
59 unshifted	g	g	g	g	g	g
59 shifted	G	G	G	G	G	G
59 alternate	ALT-G	ALT-G	ALT-G	ALT-G	ALT-G	ALT-G
60 unshifted	h	h	h	h	h	h
60 shifted	H	H	H	H	H	H
60 alternate	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H	ALT-H
61 unshifted	j	j	j	j	j	j
61 shifted	J	J	J	J	J	J
61 alternate	ALT-J	ALT-J	ALT-J	ALT-J	ALT-J	ALT-J
62 unshifted	k	k	k	k	k	k
62 shifted	K	K	K	K	K	K
62 alternate	ALT-K	ALT-K	ALT-K	ALT-K	ALT-K	ALT-K
63 unshifted	l	l	l	l	l	l
63 shifted	L	L	L	L	L	L
63 alternate	ALT-L	ALT-L	ALT-L	ALT-L	ALT-L	ALT-L
64 unshifted	;	;	m	;	7C ö	m
64 shifted	:	:	M	:	5C Ö	M
64 alternate	:	ALT-ˆ	ALT-M	:	ALT-\	ALT-M
65 unshifted	'	'	7C ù	'	7B ä	97 ù
65 shifted	"	"	%	"	5B Ä	%
65 alternate	"	"	%	"	ALT-[	%
66 unshifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
66 shifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
66 alternate	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*

KEY POSITION	STAND	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
67 unshifted	ALT-J	E4	ALT-J	ALT-J	ALT-J	ALT-J
67 shifted	ESC I	E6	ESC I	ESC I	ESC I	ESC I
67 alternate	ESC xA	ESC xA	ESC xA	ESC xA	ESC xA	ESC xA
68 unshifted	ESC 8	E5	ESC 8	ESC 8	ESC 8	ESC z
68 shifted	ESC 9	E7	ESC 9	ESC 9	ESC 9	ESC z
68 alternate	ESC yA	ESC yA	ESC yA	ESC yA	ESC yA	ESC yA
69 unshifted	4	4	4	4	4	4
69 shifted	4	4	4	4	4	4
69 alternate	4	4	4	4	4	4
70 unshifted	5	5	5	5	5	5
70 shifted	5	5	5	5	5	5
70 alternate	5	5	5	5	5	5
71 unshifted	6	6	6	6	6	6
71 shifted	6	6	6	6	6	6
71 alternate	6	6	6	6	6	6
72 unshifted	+	+	+	+	+	+
72 shifted	+	+	+	+	+	+
72 alternate	+	+	+	+	+	+
73 unshifted	ESC 0	ESC 0	ESC 0	ESC 0	ESC 0	ESC 0
73 shifted	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1
73 alternate	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1	ESC 1
74 unshifted	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*
74 shifted	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*
74 alternate	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*	*Left shift*
75 unshifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
75 shifted	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
75 alternate	*Null*	*Null*	*Null*	*Null*	*Null*	*Null*
76 unshifted	z	z	w	z	y	w
76 shifted	Z	Z	W	Z	Y	W
76 alternate	ALT-Z	ALT-Z	ALT-W	ALT-Z	ALT-Y	ALT-W
77 unshifted	x	x	x	x	x	x
77 shifted	X	X	X	X	X	X
77 alternate	ALT-X	ALT-X	ALT-X	ALT-X	ALT-X	ALT-X
78 unshifted	c	c	c	c	c	c
78 shifted	C	C	C	C	C	C
78 alternate	ALT-C	ALT-C	ALT-C	ALT-C	ALT-C	ALT-C
79 unshifted	v	v	v	v	v	v
79 shifted	V	V	V	V	V	V
79 alternate	ALT-V	ALT-V	ALT-V	ALT-V	ALT-V	ALT-V
80 unshifted	b	b	b	b	b	b
80 shifted	B	B	B	B	B	B
80 alternate	ALT-B	ALT-B	ALT-B	ALT-B	ALT-B	ALT-B
81 unshifted	n	n	n	n	n	n
81 shifted	N	N	N	N	N	N
81 alternate	ALT-N	ALT-N	ALT-N	ALT-N	ALT-N	ALT-N
82 unshifted	m	m	,	m	m	,
82 shifted	M	M	?	M	M	?
82 alternate	ALT-M	ALT-M	?	ALT-M	ALT-M	?

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
83 unshifted	,	,	;	,	,	;
83 shifted	,	,	.	,	;	.
83 alternate	,	ALT-\	.	,	;	.
84 unshifted	.	.	:	.	.	:
84 shifted	.	.	/	.	:	/
84 alternate	.	ALT-]	/	.	:	/
85 unshifted	/	/	=	/	-	95 ò
85 shifted	?	?	+	?	..	!
85 alternate	?	?	+	?	_	!
86 unshifted	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*
86 shifted	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*
86 alternate	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*	*Rt. shift*
87 unshifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
87 shifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
87 alternate	ALT-M	ALT-J	ALT-M	ALT-M	ALT-M	ALT-M
88 unshifted	ESC A	ESC A	ESC A	ESC A	ESC A	ESC A
88 shifted	ESC A	ESC A	ESC A	ESC A	ESC A	ESC A
88 alternate	ESC xB	ESC xB	ESC xB	ESC xB	ESC xB	ESC xB
89 unshifted	ESC B	ESC B	ESC B	ESC B	ESC B	ESC B
89 shifted	ESC B	ESC B	ESC B	ESC B	ESC B	ESC B
89 alternate	ESC yB	ESC yB	ESC yB	ESC yB	ESC yB	ESC yB
90 unshifted	1	1	1	1	1	1
90 shifted	1	1	1	1	1	1
90 alternate	1	1	1	1	1	1
91 unshifted	2	2	2	2	2	2
91 shifted	2	2	2	2	2	2
91 alternate	2	2	2	2	2	2
92 unshifted	3	3	3	3	3	3
92 shifted	3	3	3	3	3	3
92 alternate	3	3	3	3	3	3
93 unshifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
93 shifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
93 alternate	ALT-M	ALT-J	ALT-M	ALT-M	ALT-M	ALT-M
94 unshifted	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*
94 shifted	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*
94 alternate	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*	*Repeat*
95 unshifted	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*
95 shifted	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*
95 alternate	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*	*ALT*
96 unshifted	*Space*	*Space*	*Space*	*Space*	*Space*	*Space*
96 shifted	*Space*	*Space*	*Space*	*Space*	*Space*	*Space*
96 alternate	*Space*	*Space*	*Space*	*Space*	*Space*	*Space*
97 unshifted	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S
97 shifted	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S
97 alternate	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S	ALT-S
98 unshifted	ESC D	ESC D	ESC D	ESC D	ESC D	ESC D
98 shifted	ESC D	ESC D	ESC D	ESC D	ESC D	ESC D
98 alternate	ESC xC	ESC xC	ESC xC	ESC xC	ESC xC	ESC xC

KEY POSITION	STAND.	DOMESTIC	FRENCH	BRITISH	GERMAN	ITALIAN
99 unshifted	ESC C	ESC C	ESC C	ESC C	ESC C	ESC C
99 shifted	ESC C	ESC C	ESC C	ESC C	ESC C	ESC C
99 alternate	ESC yC	ESC yC	ESC yC	ESC yC	ESC yC	ESC yC
100 unshifted	0	0	0	0	0	0
100 shifted	0	0	0	0	0	0
100 alternate	0	0	0	0	0	0
101 unshifted	00	00	00	00	ALT-(a	00
101 shifted	00	00	00	00	ALT-(a	00
101 alternate	00	000	00	00	ALT-(a	00
102 unshifted	.	.	.	.	.	.
102 shifted	.	.	.	.	.	.
102 alternate	.	.	.	.	.	.
103 unshifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
103 shifted	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M
103 alternate	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M	ALT-M

G

DIFFERENCES BETWEEN MS-DOS AND PC-DOS

The first column in Table G-1 shows all of the PC-DOS commands documented by IBM. The second column shows the comparable MS-DOS 2.1 commands for your computer. For MS-DOS commands that have no comparable PC-DOS command, the first column is empty. Likewise, for PC-DOS commands for which there is no comparable MS-DOS command, the second column is blank.

Table G-1: MS-DOS/PC-DOS Commands

PC-DOS 2.0	MS-DOS 2.1
INTERNAL COMMANDS	
BREAK	BREAK
	CAUX, CLST, CTTY
CHDIR (CD)	CHDIR (CD)
CLS	CLS
COPY	COPY
DATE	DATE
DIR	DIR
ERASE (DEL)	ERASE (DEL)
	EXIT
	HISTORY (HI)
MKDIR (MD)	MKDIR (MD)
PATH	PATH
	PROMPT
REN	REN
RMDIR (RD)	RMDIR (RD)
	SET
TIME	TIME
TYPE	TYPE
VER	VER
VERIFY	VERIFY
VOL	VOL

PC-DOS 2.0	MS-DOS 2.1
BATCH COMMANDS	
ECHO	ECHO
FOR	FOR
GOTO	GOTO
IF	IF
SHIFT	SHIFT
PAUSE	PAUSE
REM	REM
EXTERNAL COMMANDS	
ASSIGN	
BACKUP	
CHKDSK	CHKDSK
COMP	FILCOM
	CONCAT
DISKCOMP	
DISKCOPY	DISKCOPY
	FGREP
	FIND
FORMAT	FORMAT
GRAPHICS	
	LS
	MODCON
MODE	
	MORE
	MV
	PORTSET
PRINT	PRINT
RECOVER	RECOVER
	SEARCH
	SORT
SYS	SYS
	TAIL
TREE	
	WC

NOTE: For the PC-DOS commands BACKUP and TREE, you can use SEARCH. For GRAPHICS, substitute GRAFIX in the *Graphics Tool Kit II.*

H

USING THE CALCULATOR (CALC)

You can use the calculator pad on your keyboard in two ways:

- ▶ Like the typewriter number keys (to enter numbers into a file or command line)
- ▶ Like a calculator attached to your computer (to add, subtract, multiply, and divide)

Using the calculator pad to type numbers requires no special preparation. To use the calculator pad as a calculator, however, you must follow these two steps:

1. Load the CALC utility as described in Section H.1.
2. To make a calculation, you must turn the calculator on, as described in Section H.2.

Once you have loaded the calculator program, you can load an application program or work in the operating system. Then, at any time during that work session, you can turn on the calculator, perform a calculation, and return to your work exactly where you left off.

If you reload the operating system by powering down or by pressing the reset button, you must reload CALC to use the calculator again.

H.1 LOADING THE CALCULATOR PROGRAM

The calculator program is CALC.COM for MS-DOS and CALC.CMD for CP/M-86. To load the calculator program:

1. Load MS-DOS or CP/M-86, or get to the system command prompt. If you are using another program, its user's guide should explain how to exit from the program.
2. Type:

```
calc(cr)
```

3. The screen may display a message telling you the calculator program has been installed (loaded).

H.2 TURNING THE CALCULATOR ON AND OFF

Turn on the calculator by pressing a shifted CALC (press the CALC key while holding down the Shift key). The CALC key (located at the top left of the calculator pad) may also be labeled MODE/CALC or CLEAR ENTRY.

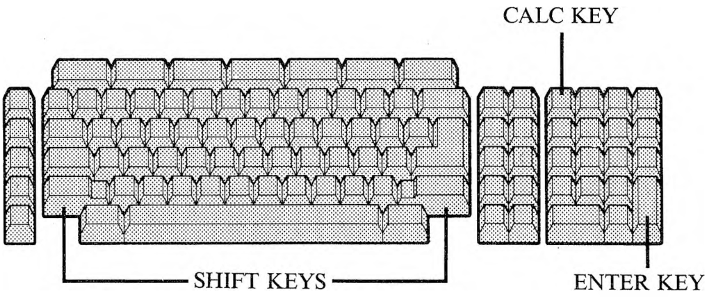
There are two ways to turn off the calculator:

- ▶ Press another shifted CALC.
- ▶ Press the Enter key.

Both methods return you to your work exactly where you left off. The Enter key, however, also places the result of your calculation in your work, as if you had typed the number.

Figure H-1 shows the CALC key, the Enter key, and the Shift keys.

Figure H-1: CALC, Enter, and Shift Keys



When you turn on the calculator, the **calculator field** appears at the bottom of the screen:

```
Calculator Functions On                  !       0. !
```

The exclamation points mark the area where your calculation will appear. The calculator field shows each step in your calculation one digit at a time, with a maximum of 14 digits.

Table H-1 summarizes the methods for turning the calculator on and off.

Table H-1: Turning the Calculator On and Off

FUNCTION	ACTION	RESULT
Turn calculator on	Shifted CALC key	Calculator field appears.
Turn calculator off	Shifted CALC key	1. Calculator field disappears. 2. You return to your work where you left off. 3. Any result or unfinished calculation is discarded.
	Enter key	1. Calculator field disappears. 2. You return to your work where you left off. 3. The result of your calculation is entered in your work.

H.3 USING THE CALCULATOR

Add and Subtract

1. Type the numbers you are adding or subtracting in any order. If the first number in the calculation is a positive number, press the Add key (+) after entering the number. If the first number is a negative number, press the Subtract key (−) after the number.
2. Press the Add key (+) after each number you are adding to the first number (and any following numbers). Press the Subtract key (−) after each number you are subtracting.
3. The calculator field shows the result of each step of the calculation as you enter the + or − indicating Add or Subtract.
4. To clear the calculator field after adding or subtracting, press the CALC key once or twice (see the note following these instructions).

Multiply

1. Type the numbers you are multiplying. Press the Multiply key (×) after each one.

2. Press the × or CALC to make the calculation. The × multiplies but does not clear the product from the calculator field. CALC finishes the multiplication and clears the field (see the note following these instructions).

Divide

1. Type the numbers you are dividing (in order). Press the Divide key (÷) after each one.

2. Press the ÷ or the CALC key to complete the calculation. The ÷ divides but does not clear the dividend from the calculator field. CALC performs the division and clears the field (see the note following these instructions).

Note: Remember that pressing the CALC key once (without the Shift key) completes a multiply/divide operation. For an incomplete add/subtract operation, however, pressing CALC once clears the calculation and returns the calculator field to zero. For a completed add/subtract operation, pressing CALC once clears the calculation without changing the calculator field. (See the examples in Section H.5.)

Pressing CALC twice always returns the calculator field to zero. You can start a new calculation, however, after the first or second CALC.

H.4 CORRECTING ERRORS

You can use the Backspace key and the CALC key to correct errors.

- ▶ Pressing the Backspace key deletes the last number you typed.
- ▶ Pressing the CALC key once cancels the calculation; pressing CALC twice returns the calculator field to zero.

H.5 EXAMPLES

```
KEYSTROKES          SCREEN DISPLAY

To Add:

25
          Calculator Functions On       !        25.   !
+
          Calculator Functions On       !        25.   !
42
          Calculator Functions On       !        42.   !
+
          Calculator Functions On       !        67.   !
CALC
          Calculator Functions On       !        67.   !
CALC
          Calculator Functions On       !         0.   !


To Subtract:

632.50
          Calculator Functions On       !    632.50    !
+
          Calculator Functions On       !     632.5    !
42.45
          Calculator Functions On       !     42.45    !
```

KEYSTROKES	SCREEN DISPLAY
–	Calculator Functions On ! 590.05 !
CALC	Calculator Functions On ! 590.05 !
CALC	Calculator Functions On ! 0. !

To Multiply:

KEYSTROKES	SCREEN DISPLAY
78	Calculator Functions On ! 78. !
×	Calculator Functions On ! 78. !
5	Calculator Functions On ! 5. !
CALC or ×	Calculator Functions On ! 390. !
CALC	Calculator Functions On ! 0. !

To Divide:

KEYSTROKES	SCREEN DISPLAY
873.6	Calculator Functions On ! 896. !
÷	Calculator Functions On ! 896. !
42	Calculator Functions On ! 16. !
CALC or ÷	Calculator Functions On ! 56. !
CALC	Calculator Functions On ! 0. !

I

TROUBLESHOOTING

This appendix can help you locate and solve problems you have with your computer. Section I.1 describes how to solve general problems, in particular those that are not accompanied by onscreen error messages. Sections I.2 and I.3 tell how to replace the fuse and how to prepare for a service call.

GENERAL PROBLEM SOLVING I.1

When you have a problem, don't become alarmed. You can solve most problems yourself in a few minutes. First examine the situation and consider the following questions (keep your answers in mind if you call your service representative):

▶ Was an error message displayed?

If so, look up the error message and/or error code in your program user's guide. For example, Chapter 8 of the *MS-DOS User's Guide* defines the MS-DOS error messages and codes and tells how to recover from the problem. If no error message is displayed, refer to Table I-1.

▶ Is your computer making unusual noises?

Unexplained noises can mean problems with the computer's power supply, the fan motor, or the disk drives. Call your service representative.

▶ Are the in-use lights functioning, and are they on or off?

If the lights do not work (i.e., do not light up when disk access is attempted), your system may need service. If a light is on, the system may be "hung up"; try pressing the reset button.

- What were you doing when the problem began?

  Note the type of operation you attempted and the program you were using (name and version number, if possible). Refer to the program's user guide; it should explain how to respond to any error messages displayed by the program.

Table I-1 describes how to solve problems when no error message is displayed on the screen.

Table I-1: Problems and Solutions

PROBLEM	SOLUTION
Screen is blank and there is no fan noise.	1. Check that the power cord is connected properly. 2. Check that the wall socket has power (plug in another device, such as a lamp). 3. Replace the fuse (as described in Section I.2).
Screen is blank or very dim, but there is fan noise.	1. Make sure all connections are secure. 2. Try to increase CRT brightness (see the *Operator's Reference Guide*). If you cannot increase brightness, the system may be "hung up" and require resetting.
The operating system does not load and there is no drive noise.	Call your service representative.
The operating system does not load, but there is drive noise.	1. Check that screen brightness is adjusted properly. 2. Try to boot with a backup system diskette.
When you load an application program, there is drive activity, but the computer "hangs up" before the first frame of the program.	Some application programs require that you you first load a programming language module. Check that you loaded the proper module.

PROBLEM	SOLUTION
The display freezes and the computer does not respond to your commands.	1. Type ALT-C once or twice. 2. Check that the keyboard is connected properly. 3. Press the reset button.
Repeated inconsistent or erratic results.	You may have a worn or damaged diskette. Retry the operation or try your backup diskette.
Bell sounds.	You typed a key that cannot be used in the current program. May also indicate a "lock up" situation; press the reset button.

REPLACING THE FUSE I.2

If the fuse burns out, your computer will not power up, and will power down if it is on at the time. Your service representative can change the fuse for you, or you can change it yourself. The fuse holder for both the desktop and portable computers is located on the rear of the main cabinet (see Figures I-1 and I-2), just above the power cord receptacle.

Figure I-1: Replacing the Fuse on the Desktop Computer

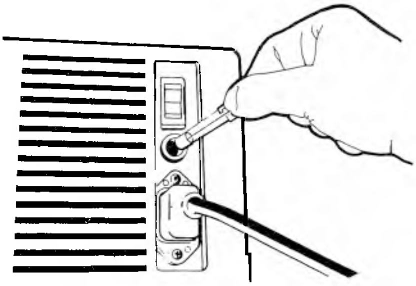
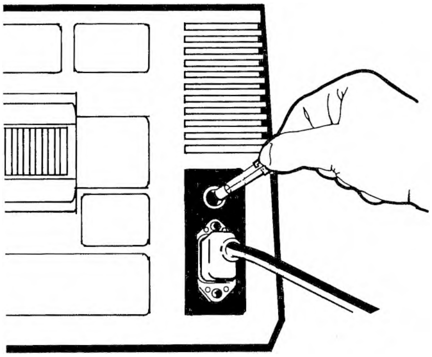


Figure I-2: Replacing the Fuse on the Portable Computer



To replace the fuse, follow these steps:

1. Turn off the power for the main unit, and for the external hard disk if present.
2. Disconnect the power cord.
3. Open the fuse holder with a coin or screwdriver.
4. Replace the old fuse with a new fuse. Use only fuses obtained from your dealer for your computer.
5. Close the fuse holder.
6. Reconnect the power cord.
7. Turn on the power.

PREPARING FOR A SERVICE CALL I.3

If you have a problem that persists, or if your problem is not described in Table I-1, you may need the services of a computer technician. Before requesting help, make sure you have the following information:

- A complete description of the problem, including error messages.
- The program and data diskettes you were using when the problem occurred. Do not make any changes to these diskettes.
- Model and serial numbers of your:

  —Keyboard

  —CRT (if separate from main unit)

  —CPU

  —External hard disk (if you have one)
- Your business hours and phone number.
- The name of the person to contact. If this person is not you, give him or her your diskettes and your notes on the problem.

INDEX