NCR

Personal Computer

NCR-DOS

MS and GW (MS™-MACRO Assembler and GW™-BASIC) are trademarks of Microsoft Corporation; The INSTRUCTOR and Professor DOS are registered trademarks of Individual Software Inc. IBM is a registered trademark of International Business Machines Corporation. Dr. Graph and GSX are trademarks of Digital Research Inc.

Third Edition, May 1985

It is the policy of NCR Corporation to improve products as new technology, components, software, and firmware become available. NCR Corporation, therefore, reserves the right to change specifications without prior notice.

All features, functions, and operations described herein may not be marketed by NCR in all parts of the world. In some instances, photographs are of equipment prototypes. Therefore, before using this document, consult your nearest dealer or NCR office for information that is applicable and current.

NCR

CUSTOMER PROGRAM LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THE DISKETTE(S) PACKAGE. OPENING THIS DISKETTE(S) PACKAGE INDICATES YOUR ACCEPTANCE OF THESE TERMS AND CONDITIONS. IF YOU DO NOT AGREE WITH THEM, YOU SHOULD PROMPTLY RETURN THE PACKAGE UNOPENED; AND YOUR MONEY WILL BE REFUNDED.

NCR provides this Program(s) and licenses its use under these terms and conditions and under Copyright Law. You assume responsibility for the selection of the Program(s) to achieve your intended results, and for the installation, use and results obtained from the Program(s). This program is confidential, proprietary to and a trade secret of the owner, and should be safeguarded by you as such.

LICENSE

You may:

a. use the Program(s) only on a single machine at a single location;

b. copy the program into any machine readable or printed form for backup or modification purposes only, to support your use of the Program(s) on the single machine (Certain programs, however, may include mechanisms to limit or inhibit copying. They are marked "copy protected");

c. modify the Program(s) and/or merge it into another program for your use on the single machine (Any portion of this Program(s) merged into another program will continue to be subject to the terms and conditions of this Agreement); and

d. transfer the Program(s) and license to another party only if the other party agrees to accept the terms and conditions of the Agreement. You must advise NCR of the name and address of the other party and the other party must sign a copy of the NCR Customer Program License Agreement and have the same received by NCR. If you transfer the Program(s), you must at the same time either transfer all copies whether in printed or machine readable form to the same party or destroy any copies not transferred; this includes all modifications and portions of the Program(s) contained or merged into other programs.

You must reproduce and include any copyright notice and serial number on any copy, modification or portion merged into another program.

TERM

The license is effective until terminated. You may terminate it at any time by destroying the program together with all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this Agreement or if you fail to comply with any term or condition of this Agreement. You agree upon such termination to destroy the Program(s) together with all copies, modifications and merged portions in any form.

YOU MAY NOT USE, COPY, MODIFY, OR TRANSFER THE PROGRAM(S), OR ANY COPY, MODIFICATION OR MERGED PORTION, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE.

IF YOU TRANSFER POSSESSION OF ANY COPY, MODIFICATION OR MERGED PORTION OF THE PROGRAM TO ANOTHER PARTY, YOUR LICENSE IS AUTOMATICALLY TERMINATED.

EXCLUSION OF WARRANTY

THE PROGRAM(S) IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY OF FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU SHOULD THE PROGRAM(S) PROVE DEFECTIVE, YOU (AND NOT NCR OR ITS DEALER OR DISTRIBUTOR) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. NCR does not warrant that the functions contained in the Program(s) will meet your requirements or that the operation of the program will be uninterrupted or error free.

LIMITED WARRANTY

NCR warrants the diskette(s) on which the program is furnished, to be free from defects in materials and workmanship under normal use for a period of ninety (90) days from the date of delivery to you as evidenced by a copy of your receipt.

NCR's entire liability and your exclusive remedy shall be:

1. the replacement of any diskette(s) not meeting NCR'S "Limited Warranty" and which is returned to NCR or an authorized NCR dealer or distributor, with a copy of your receipt, or

2. if NCR or its authorized dealer or distributor is unable to deliver a replacement diskette(s) and repair is not practicable or cannot be timely made, you may terminate this Agreement by returning the program and your money will be refunded.

IN NO EVENT WILL NCR BE LIABLE TO YOU FOR ANY DAMAGES, INCLUDING ANY LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE DISKETTE(S) EVEN IF NCR OR AN AUTHORIZED NCR DEALER OR DISTRIBUTOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, OR FOR ANY CLAIM BY ANY OTHER PARTY.

Some states do not allow limitations on how long an implied warranty lasts, so the above exclusion may not apply to you.

Some states do not allow the limitation or exclusion of liability for incidental or consequential damages so the above limitation or exclusion may not apply to you.

This warranty gives you specific legal rights and you may also have the other rights which vary from state to state.

HOW TO USE THIS MANUAL

Before you use this manual you should be familiar with the information contained in the *OWNER'S MANUAL*. Specifically, you should know how to insert a flexible disk (diskette) in the disk drive, you should know the names and purposes of all the components of your NCR PERSONAL COMPUTER, and you should have the system installed and operating.

Now you are ready to learn about NCR-DOS. (NCR-DOS is NCR's version of Microsoft®'s MS™-DOS). To actually perform the backup copy of your master NCR-DOS diskette (one of the diskettes included in back of this manual) when this procedure is described in the first chapter, you will want a supply of blank 5-1/4" flexible diskettes.

NOTE: 5 1/4" flexible disks are made by several different manufacturers. Some are labeled "Diskettes"; some are labeled "Mini-Disks"; others may have different markings on the box. The important thing is the indication that the disks are double-density, preferably double-sided (though single-sided is acceptable for everything except for your DOS operating system copy), and soft-sectored. When buying from either a dealer or a mail-order catalog, be sure to specify that they are for use on the NCR Personal Computer.

The terms "disk" and "diskette" are used interchangeably in these manuals when referring to flexible disks. However, to refer to a fixed ("hard") disk, only the term "disk" is appropriate. When either flexible or fixed disk is intended, the term "disk" is used generically.

This manual describes NCR-DOS and how to use it. Chapter 0 tells you how to begin using NCR-DOS and how to format and make a backup copy of your NCR-DOS diskette. Chapter 1 discusses disk drive configurations and identification, and leads you through some frequently performed operations. It also describes how to use the HELP facility for guidance in using DOS commands.

Chapter 2 tells you about files and directories – what they are and how to use them. Chapter 3 teaches you about commands – types, options, entry conventions (standard formats and procedures that apply to all commands), and batch files. Chapter 4 describes each

NCR-DOS command individually, giving you its purpose, its syntax (the precise format, or optional format variations, that can be recognized by NCR-DOS as legitimate instructions), and comments which include examples of the use of the command.

Chapter 5 defines the use of the special editing keys in the entry of NCR-DOS commands and Chapter 6 describes the line editor, EDLIN, which also uses the special editing keys. With practice, you will become quite adept at the use of the editing keys and the other EDLIN functions. Chapter 7 describes the use of NCR-LINK, which must be used to produce final machine-readable versions of any programs you write, whether you are a novice or an experienced programmer.

Appendices to this manual include instructions on the use of the advanced configuration feature, guidelines for buying, installing and using applications software, and information on setting up a disk for a different keyboard, the GSX-86 CRT driver, and on disk errors and messages and how you can respond to them.

Finally, for easy access to NCR-DOS information, a glossary of terms and an index follow the text portion of this manual.

Always remember that you have two readily available sources of information to assist you in learning more about the use of NCR-DOS and its commands and features, and in finding your way out of unexpected situations:

- This NCR-DOS manual

- The on-line HELP facility which you can call upon while you are using NCR-DOS

If you plan to progress to the point of writing your own programs and need more in-depth technical assistance, or if you are already experienced in programming, you may want to obtain the *NCR-DOS PROGRAMMER'S MANUAL* from NCR or from your NCR Personal Computer Dealer.

COMPATIBILITY WITH IBM

The NCR PC is operationally compatible with the IBM PC-XT. It is possible to create NCR PC programs that can run on the IBM PC-XT without modification and vice versa. This will need some further explanation.

A program cannot be run on two different systems unless they use the same microprocessor family to execute the language of the program. This requirement is met (the NCR PC uses the Intel 8088 microprocessor). However, to make an application program written for one system run on a different system, the BIOS and operating system interrupts must be compatible. The NCR PC has maintained all IBM BIOS system interrupts, and its operating system is operationally compatible with IBM-DOS. This means that an application created on the NCR PC operates the same on the IBM PC-XT and vice versa.

NOTE: The BIOS microcode of the NCR PC differs from that of the IBM PC-XT. As a result, accessing routines and /or storage locations by absolute address may produce different results.

Furthermore, NCR has taken care to provide the hardware video memory required for operational compatibility (CRT, keyboard, hard/flex disk interfaces, etc.)

Operational compatibility ensures that your NCR PC can accommodate all application and language software designed for the IBM PC-XT.

The topic of compatibility with IBM and other systems is discussed in detail in Appendix B, “How to Obtain and Install Software”.

NCR-DOS

Contents

Chapter 0 Getting Started

Getting Started 0-1
Copying Your Master Software Disk 0-1
2 Flexible Disk Drives 0-3
1 Flexible Disk Drive 0-4

Chapter 1 Introduction

Understanding the Software's Role 1-1
What is NCR-DOS? 1-1
What is an Operating System? 1-1
Identifying Your Disk Drives 1-2
Disk Drive Designations 1-2
Flexible/Fixed Disk Systems Considerations 1-3
Copying Your Master Software Disk 1-3
Copying with Two Disk Drives 1-3
Copying with One Disk Drive 1-4
Frequently Performed Operations 1-5
Loading DOS 1-5
Entering Date and Time 1-6
Displaying the Disk Directory 1-8
Formatting a Blank Disk 1-9
Changing the Default Drive 1-10
Backing Up Your Disks 1-10
Using the Programmable Function Keys 1-11
Running Programs Automatically 1-11
Dual-Operating Systems Considerations 1-11
Turning the System Off 1-12
If You Have a Fixed Disk 1-12
Formatting Your Fixed Disks 1-12
Partitioning the Fixed Disk 1-12
Defining the Fixed Disk 1-13
Re-Formatting a Fixed Disk Drive 1-16
Defining a Serial Printer 1-17
The Help Tutorial 1-17

Chapter 2 Files and Directories

Files 2-1
How DOS keeps track of your Files 2-1

The DIR (show directory) Command 2-2
Naming your File 2-2
Universal Characters 2-3
The ? Character 2-4
The * Character 2-4
Naming Conventions 2-5
Common Extensions 2-6
Copying Files 2-7
Protecting Files 2-8
Directories 2-9
Filenames and Paths 2-11
Pathnames 2-11
Pathing and External Commands 2-12
Pathing and Internal Commands 2-13
Displaying your Working Directory 2-13
Creating a Directory 2-14
Changing your Working Directory 2-15
Removing a Directory 2-15

Chapter 3 Learning about Commands

Types of DOS Commands 3-1
Internal Commands 3-1
External Commands 3-2
The Concept of Syntax 3-2
DOS Syntax Notation 3-2
Command Options 3-4
Common Entry Conventions 3-4
Batch Processing 3-6
The AUTOEXEC.BAT File 3-8
Creating an AUTOEXEC.BAT File 3-10
Creating .BAT Files with Replaceable Parameters 3-10
Executing a .BAT File 3-11
Input and Output 3-12
Redirecting your output 3-12
Filters 3-13
Command Piping 3-13

Chapter 4 Commands

ASSIGN 4-5
BACKUP 4-7
BATCH COMMANDS 4-9
ECHO 4-11
FOR .. IN .. DO 4-12
GOTO 4-13

  IF 4-14
  PAUSE 4-15
  REM 4-16
  SHIFT 4-17
CHDIR 4-18
CHKDSK 4-19
CLS 4-22
COMMAND 4-23
COMP 4-24
CONFIG.SYS COMMANDS 4-26
  BREAK 4-27
  BUFFERS 4-28
  DEVICE 4-29
  FILES 4-32
  SHELL 4-33
COPY 4-34
CTTY 4-37
DATE 4-38
DEL 4-40
DIR 4-41
DISKCOMP 4-43
DISKCOPY 4-44
ERASE 4-46
ESC 4-46a
EXE2BIN 4-47
EXIT 4-48
FDISK 4-49
FIND 4-55
FORMAT 4-57
GRAFTABL 4-58a
GRAPHICS 4-59
HELP 4-60
KEYB 4-60a
MKDIR 4-61
MODE 4-62
MORE 4-67
PATH 4-68
PRINT 4-69
PROMPT 4-72
RECOVER 4-74
RENAME 4-75
RESTORE 4-76
RMDIR 4-78
SET 4-79
SORT 4-80

SYS 4-82
TIME 4-84
TREE 4-86
TYPE 4-87
VER 4-88
VERIFY 4-89
VOL 4-90

Chapter 5 Editing and Function Keys

Special Editing Keys 5-1
Programming of Editing Keys 5-1
Editing of Commands 5-3
Function Keys 5-9
Programming of Function Keys 5-9
Keyboard Key Reassignment 5-10

Chapter 6 Line Editor

General Information 6-1
How to Start EDLIN 6-1
Special Editing Keys 6-2
EDLIN Commands 6-3
Format Conventions 6-4
Command Options 6-6
line-number 6-7
A(PPEND) 6-9
C(OPY) 6-10
D(ELETE) 6-12
(E)ND 6-14
I(NSERT) 6-15
L(IST) 6-18
M(OVE) 6-21
P(AGE)) 6-22
Q(UIT) 6-23
R(EPLACE) 6-24
S(EARCH) 6-27
T(RANSFER) 6-30
W(RITE) 6-31
Error Messages 6-32

Chapter 7 NCR Link

General Information 7-1
Program Overview 7-1
Definitions You'll Need to Know 7-3

Files that LINK Uses 7-6
Input File Extensions 7-6
Output File Extensions 7-6
VM.TMP (Temporary) File 7-6
Using LINK 7-7
Starting LINK 7-7
Method 1: Prompts 7-7
Method 2: Command Line 7-8
Method 3: Response File 7-9
COMMAND CHARACTERS 7-10
Command Characters 7-10
Command Prompts 7-11
LINK Options 7-13
Sample Link Session 7-15
Error Messages 7-17

Appendix A

Advanced Configuration Feature A-1

Appendix B

How to Obtain and Install Software B-1

Appendix C

Messages C-1

Appendix D

RAM-Disk Demonstration D-1

Appendix E

Keyboard Considerations E-1

Appendix F

GSX-86 CRT Driver F-1

GETTING STARTED

The NCR-DOS master disk (or diskette), the one you received with this book, contains all operating system software files and all commands. You should therefore start using the NCR Personal Computer by making a backup copy of your NCR-DOS disk. This way you will not be "shut down" if it becomes damaged. Let's start now by loading DOS.

Insert the DOS disk in drive A, turn the drive access lever (door) a quarter turn to the right, and turn on your computer. Press either of the (<CR>) keys. Depending on the size of memory, loading DOS can take up to 25 seconds.

COPYING YOUR MASTER SOFTWARE DISK

When you load DOS, the software on your disk displays some general information and then starts the formatting and copying processes automatically.
Following is a program that will establish the correct keyboard format for your language on your DOS master disk.

The term "target diskette", which is used during these processes, refers to the new (copy) disk. When you are instructed to "press any key", you must understand that to mean "almost any key". There are several keys which do not transmit a character to DOS and should not be used in this manner. They include the Ctrl, Num Lock, Alt and Shift keys.

While the displays are self-explanatory, you may prefer following along with printed text. Your actions are marked with a ● .

The program displays:

CONGRATULATIONS

... on your selection of the NCR Personal Computer and the NCR-DOS operating system, a powerful, yet easy-to-use combination of hardware and software.

You've already loaded the operating system into memory. Now, you will make a backup copy of the master diskette, the one in drive A, and establish the correct keyboard format for your language on the target diskette.

The complete procedure is performed by DOS - with some help from you. DOS asks a question, and you answer; DOS gives a simple direction, and you follow it.

Are you a little worried about using your computer for the first time? Don't be. The NCR Personal Computer is sturdy and your DOS master diskette is write protected: You can't destroy anything.

Have a new diskette ready for the backup copy. If you have two flexible disk drives, follow the instruction for copying with two disk drives; if you have one flexible disk drive (whether or not you also have one or more fixed disk drives). follow the instructions for copying with one disk drive. Press any key to begin.

- **Press any key!**

2 FLEXIBLE DISK DRIVES

* ATTENTION *

If an error message is displayed during the following procedure, try again. Turn off the computer and then load the DOS master diskette again. Use another target diskette.

NOTE: DOS formats the diskette automatically. When the formatting process is completed, you are informed about the number of bytes total disk space and the number of bytes available on disk. You are also asked whether you want to format another disk. Enter N (for no)!

```
Insert new diskette for drive B:
and strike any key when ready
```

- **Insert a new diskette and press any key!**

```
Formatting ... Format/Complete
xxxxxx bytes total disk space
xxxxxx bytes available on disk

Format another (Y/N)?
```

- **Press N.**

DOS is now copying the master diskette to the formatted diskette. Wait until the - Copy complete - message is displayed.

```
Copying...

Copy complete
```

```
Remove the master diskette from drive A: and save it for system pro-
tection. Take the diskette from drive B: and place it in drive A:. Press
any key to continue.
```

- **Follow this instruction**

(To continue, see page 0-5)

1 FLEXIBLE DISK DRIVE

* ATTENTION *

If an error message is displayed during the following procedure, try again. Turn off the computer, and then load the DOS master diskette again. Use another target diskette.

NOTE: DOS formats the diskette automatically. When the formatting process is completed, you are informed about the number of bytes total disk space and the number of bytes available on disk. You are also asked whether you want to format another disk. Enter N (for no)!

```
Insert new diskette for drive A:
and strike any key when ready
```

- **Remove the master diskette form drive A:, insert the target diskette, and press any key!**

```
Format ... Format complete
xxxxxx bytes total disk space
xxxxxx bytes available on disk

Format another (Y/N?)
```

- **Press N (for no).**

```
During the following copy procedure, change diskettes as instructed
until the - Copy complete - message appears.
Press any key to continue.
```

- Press any key

```
Insert master diskette into drive A;
Press any key when ready.
```

- **Change diskettes and press any key!**

```
Copy complete
```

Please save the master diskette for system protection. Make sure that the target diskette is in drive A:, and press any key to continue.

- **Follow this instruction**

SETTING UP THE KEYBOARD FORMAT

Next you will establish the correct keyboard format for your language.

1 = US-English
2 = UK-English
3 = French
4 = German
5 = Italian
6 = Spanish

0 = exit

- **Enter your selection**

Your target diskette will now be set up to use the correct keyboard format for your language when it is loaded. When the system prompt A> appears, you may end your first computer session.

Remove the diskette from drive A: and label it as the new master diskette. Always use this diskette for processing.
Now you may turn your NCR PC off to learn more about DOS.

As you have already made a copy of your original DOS disk, you may skip the section "Copying your Master Software Disk" in chapter 1. If you have a fixed disk, complete setup procedures as described in "If you have a Fixed Disk" at the end of that chapter.

COPYING GW-BASIC

The NCR-DOS master diskette as well as every copy created with the MCOPY or DISKCOPY utilities contain the files GWBASIC, GW-BASIC, BASIC and BASICA. These can be used interchangeably to call GW-BASIC. They are marked read-only to ensure protection against deletion. When issuing a DISKCOPY command, these directory entries cannot subsequently be deleted from the destination disk.

Should you wish to create a disk that does not contain all of the GW-BASIC program files, use the COPY command. If you use the COPY *.* command, the system attempts to make four copies of the GW-BASIC program (one copy for each directory entry). As there is not enough disk space on the destination disk to accommodate these additional copies, the following message will be returned:

```
Insufficient disk space
```

At this point all software files and commands (including GWBASIC) will have been transferred. Missing are GW-BASIC, BASIC and BASICA which represent the last directory entries.

NOTE: When running CHKDSK the system informs you about the clusters on which the directory entries are cross-linked.

Working with DOS

UNDERSTANDING THE SOFTWARE'S ROLE

WHAT IS NCR-DOS?

NCR-DOS is a Disk Operating System for the NCR Personal Computer. An operating system, along with program development software and applications software, comprise the three primary types of "software" that are used with the physical computer components (the "hardware"). The term *Disk* Operating System is used when the operating system software itself, and most or all of the program development and applications software, are stored on (flexible or hard) disks and are called into the computer's memory from the disks as they are needed.

The DOS master disk (diskette), the one you received with this book, contains all the operating system software files and all commands. In this chapter, you learn how to install the software, switch processing from one disk to another, protect your master disk, and format other new disks.

WHAT IS AN OPERATING SYSTEM?

An operating system provides a link between you and your computer. It is a software program, or collection of programs that controls all the programs in your computer. An operating system also controls the allocation of the computer's resources (memory, disk space, printer, etc.), and prepares your computer to accept the commands of other software packages, particularly when more than one task is to be performed at a time.

The operating system provides the communication channel or link between you and the computer by means of prompts and commands. A prompt is a message or other signal displayed by the operating system to you, indicating that you should make a decision or perform some action before the system can continue. In some cases, you are given a specific set of choices, such as Y/N (yes/no). In other cases, the system has completed a task, or portion of a task, and you enter one of the set of commands recognized by the particular operating system. These commands direct the system to perform tasks like the following:

- Start the next program.
- Edit a program source file.
- Display the directory (list of contents) of a disk.
- Copy a file or an entire disk.
- Compile a program.

IDENTIFYING YOUR DISK DRIVES

Before actually loading your software, you may need to know a little more about NCR PC and those all-important disks. Depending on your computer model, you have either a flexible disk system or a flexible/fixed disk system. The types of disks are not important to DOS; the software only wants to know where to get and put information.

DISK DRIVE DESIGNATIONS

When you copy your master diskette, or load almost any software or data, you are directed to insert or change a disk. You are instructed to do this by drive designation. For example, "Insert a disk into drive A."

The drive designation, which is always an alphabetic character, tells DOS where to get and put information. No matter what types of disk units you have, each drive always has its own designation. Consider the following disk configuration examples:

- You have two flexible disk drives. One drive is designated (and labeled) A; the other drive is designated B.

- You have one flexible disk drive. The drive is labeled A, but is designated as both A and B by DOS.

- You have one flexible disk drive and one fixed disk unit. The flexible disk drive is labeled A, but is designated as both A and B by DOS. The fixed disk unit is designated C.

- Now, assume you have two flexible disk drives and one fixed disk unit. What are the drive designations? The flexible disk drive designations are A and B; the fixed disk unit drive designation is C. Note that in this case you are using a freestanding fixed disk drive unit since the two drive positions in the cabinet are used for the two flexible disk drives.

Drive designations are assigned by DOS when it is loaded into memory. If you have two flexible disk drives attached, they are designated A and B. If you have one flexible disk drive attached, it is designated both A and B, since DOS must have both letters assigned to flexible disk. If you have a fixed disk drive, it is designated C. Any additional freestanding disk units you may add will be designated D, E, and so on.

FLEXIBLE/FIXED DISK SYSTEMS CONSIDERATIONS

To simplify explanations in this manual, examples are shown based on a two-drive, flexible disk system. However, if you have a flexible/fixed disk system, you will use only the flexible disk drive to format and make copies of flexible disks. In these situations, DOS always "prompts" you to change disks and waits for you to insert the new disk. You then continue processing by pressing any key.

COPYING YOUR MASTER SOFTWARE DISK

You start using DOS by making a backup copy of your master software disk. If you have two flexible disk drives, follow the instructions for copying with two disk drives; if you have one flexible disk drive (whether or not you also have one or more fixed disk drives), follow the instructions for copying with one disk drive.

The only action required by you in this procedure is to load the NCR-DOS master diskette and a blank diskette when instructed, and to strike a key when necessary. When you are instructed by DOS to "strike any key", you must understand that to mean "*almost* any key". There are several keys which do not transmit a character to DOS and should not be used in this manner. They include the Ctrl, Num Lock, Alt and Shift keys. You should learn to respond with the space bar, the carriage return key (↵), or any alphabetic or numeric key.

NOTE: After you make a backup copy of your master diskette and if you have a fixed disk, complete setup procedures as described in "If you have a Fixed Disk" at the end of the chapter.

COPYING WITH TWO DISK DRIVES

The program to copy your NCR-DOS master diskette executes automatically. Load the master disk into drive A and turn your computer on. After several seconds (for the operating system to load), a series of instructions will appear on your console.

Soon the program displays:

```
Insert source diskette in drive A:
Insert target diskette in drive B:
Strike any key when ready
```

Since you already have the DOS disk (the one you want to copy from) in drive A all you need to do is to have a blank disk that is labeled as double-density, double-sided to copy to. Insert the blank diskette in drive B, close the "door", and you are ready. Now, press the space bar or any alphabetic or numeric key. The DISKCOPY program will automatically format your new diskette and copy all of the DOS operating system files. You will see the display:

```
Copying 9 sectors per track, 2 side(s)
```

then, when finished:

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

Enter **N**. You have successfully copied your master disk and now have a new master.

NOTE: ALWAYS USE THIS NEW MASTER FOR PROCESSING!

Remove the master diskette from drive A and save for system protection. Take the diskette from drive B and place it in drive A. Good Luck with your NCR PC and NCR-DOS 2.11.

COPYING WITH ONE DISK DRIVE

The program to copy your NCR-DOS master diskette executes automatically. Load the master diskette into drive A and turn your computer on. After several seconds (for the operating system to load), a series of instructions will appear on your console.

The DISKCOPY program will automatically format your new diskette and copy all of the DOS operating system files.

You have one flexible disk drive. To copy the master disk using one drive requires a change of disks. DOS tells you when to do this.

(Insert SOURCE diskette means insert the DOS master disk.)
(Insert TARGET diskette means insert the new (copy) disk.)

During the following copy procedure, change disks as instructed until the 'copy complete' message appears.

```
Insert source diskette into drive A
```

```
Strike any key when ready
```

You already have the DOS master disk in drive A. Press any key. After DOS reads as much into memory from the source disk as it can, it displays:

```
Insert target diskette into drive A
Strike any key when ready
```

DOS writes out what it stored in memory. You will also see the message **"Formatting target while copying"**.

NOTE: Depending on the memory space of your computer, these messages will continue to alternate until the entire disk has been copied. The copy procedure is not complete until the following message appears:

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

Enter **N**.

You have successfully copied your master diskette and now have a new master.

NOTE: ALWAYS USE THIS NEW MASTER FOR PROCESSING!

Save the original master diskette for system protection. Good Luck with your NCR PC and NCR-DOS 2.11.

FREQUENTLY PERFORMED OPERATIONS

LOADING DOS

You are now ready to load your operating system, having made a copy of your master DOS software disk. But, first, you must understand the instructions you will be given. To do this, you must know the syntax notation used. The concept of command syntax, and a complete list of the syntax notations used in NCR-DOS, are defined in the *Learning About Commands* chapter. For now, there are two specific notations you must recognize.

First, there are two keys on your keyboard that are used to process an entry you just made. They both have a short vertical line on the right connected to a left-pointing arrow ↵. To avoid confusion with the actual left-pointing arrow (backspace) keys, and to relate to a commonly-used term both within and outside the realm of computers, these keys are thought of as carriage returns (or, sometimes, ENTER keys). The notation <CR> is used when either one of these keys is to be pressed.

Second, when two keytop names are enclosed in a set of angle brackets, such as <Ctrl-Break>, press the first key and HOLD IT DOWN while pressing the second indicated key.

You can produce on the screen a character for which there is no key on your keyboard by entering its decimal code with the numeric key pad keys while the Alt key is pressed. The decimal codes for the entire character set are listed in Appendix B of your GW-BASIC manual.

Regardless of which disk system you have, you always start processing from flexible disk drive A or from fixed disk drive C. Let's do that now by loading DOS into memory.

Once NCR-DOS is loaded, the system searches the DOS disk for the COMMAND.COM file and loads it into memory. The COMMAND.COM file is a program that processes the commands you enter and then runs the appropriate programs. It is also called the Command Processor.

When the command processor is loaded, you see a copyright and software identification message on your screen. When using the supplied DOS master diskette, you will immediately begin the Disk Copy procedure. Otherwise, the command processor always asks you what it should do next by displaying a "prompt". A prompt is a signal to you that input from you is needed. The system default prompt is "A>". The term "default" defines what the system is programmed to do when you receive it, and what it will do unless you tell it otherwise.

Normally, when a default is listed, you do have the ability to change it. The system prompt can be changed using the PROMPT command, described in the *Commands* chapter. Changing the default drive designation is defined later in this chapter.

NOTE: In the following sections, the entries you make are shown in **boldface** type to distinguish them from operating system messages and prompts, and from text descriptions.

ENTERING DATE AND TIME

Whenever you load DOS into memory or restart your computer, the first thing DOS will do is ask you to tell it the current date and time by displaying the Date and Time prompts. You should adopt the practice of entering this information accurately. It is extremely useful in helping to keep track of when you created or updated data or programs on your disks since the date and time are recorded for each disk file. The program displays:

```
Current date is Tue   1-01-1980
Enter new date: _
```

Enter today's date in the form mm-dd-yy (or mm/dd/yy) where:

- mm is a 1- or 2-digit number from 1 to 12 (month)
- dd is a 1- or 2-digit number from 1 to 31 (day)
- yy is a 2-digit number from 80-99 (the 19 is assumed) or a 4-digit number from 1980-2099 (year)

Any date is acceptable in answer to the new date prompt as long as it follows the format. Separators between the numbers can be hyphens (-) or slashes (/). For example:

4-4-84 or **04/04/1984**

are both acceptable answers. DOS calculates forward from its default date (1-01-1980) to determine today's day of the week (in this case, Wed). You can just enter <CR> without a date entry, but it is better to have the actual date created recorded on your new disks.

If you enter an invalid date or form of date, the system prompts you again with "Enter new date."

After you respond to the new date prompt, DOS asks you for the time:

```
Current time is 0:00:00.00
Enter new time: _
```

Enter the current time in the hh:mm format, where:

- hh is a 1- or 2-digit number from 0 to 23 (hours)
- mm is a 1- or 2-digit number from 0 to 59 (minutes)
- ss is a 1- or 2-digit number from 0 to 59 (seconds)
- xx is a 1- or 2-digit number from 0 to 99 (hundredths)

DOS uses this time value to keep track of when you last updated and/or created files on the system. Notice that DOS uses military time (24 hour clock); for instance, 3:30 p.m. is written 15:30.

Example:

```
Current time is 0:00:00.00
Enter new time: 9:05
```

Use only the colon(:) to separate hours and minutes. If you enter an invalid number separator, DOS repeats the prompt.

NOTE: If you make a mistake while typing, press the <Ctrl> key on your keyboard, HOLD IT DOWN, and then press the <Break> key. (If you have trouble finding it, the legend Break is on the front of the key labeled Scroll Lock in the upper right-hand corner of the keyboard.) The <Ctrl-Break> function aborts your current entry and displays ↑C. You can then re-answer the prompt or type another command. To correct a line before you press <CR>, use the backspace (left-pointing arrow) key to erase one letter at a time, or press Escape to delete the current entry to reenter from next line.

DISPLAYING THE DISK DIRECTORY

To check your new DOS operating system disk, place it in the default drive A. Type in:

A>**DIR**

and press <CR>. You will receive information concerning your new disk, similar to the following:

```
   Volume in drive A has no label
   Directory of  A:\

COMMAND    COM      15957     3-11-84    3:51p
CHKDSK     COM       6468     3-11-84   11:47a
DISCOMP    COM       1857    11-22-83    9:12a
DEBUG      COM      12223    10-19-83    7:52p
RECOVER    COM       2295    10-19-83    7:51p
SYS        COM       2640     4-19-84    4:40p
MORE       COM       4364     4-19-84    4:40p
LINK       EXE      42330     7-07-84    8:18a
EXE2BIN    EXE       1647     4-19-84    4:40p
   .         .         .          .          .
   .         .         .          .          .
   .         .         .          .          .
   .         .         .          .          .
ANSI       SYS       1581     4-19-84    4:40p
EDLIN      COM       8080    12-19-83    1:22p
       31 File(s)
```

Figure 1.1 Sample directory listing

As you can see, the directory tells you the volume label (if any), and the filename, extension, size (in bytes), and the date and time last modified for each file. The directory listing also tells you how many files are on the disk and the amount of space not being utilized.

To display a directory listing of a disk in drive B, type

A>**DIR B:**

FORMATTING A BLANK DISK

Every new disk must be formatted to allow the disk to accept data for storage. To format a blank diskette in drive B, respond to the system prompt "A>" by entering:

FORMAT B:

or

FORMAT B:/1 if you are formatting a single-sided disk

or

FORMAT B:/V/S if you want to include a Volume label (/V) and System files (/S).

or

FORMAT B:/1/V/S if you are using a single-sided disk and wish to use all options.

The program displays:

```
Insert new diskette for drive B:
and strike any key when ready _
```

NOTE: Remember that "strike any key", means "*almost* any key". Ctrl, Num Lock, and the Alt and Shift keys should not be used in this manner. Learn to respond with the space bar, the <CR> key, or any alphabetic or numeric key.

Press any key; the program displays:

```
Formatting... _

(then, when finished)
Format complete
System transferred

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

You now enter any 11-character unique name or number by which you want to identify this new disk, and then press <CR> (or just press <CR> if you want the volume label to be blank). The program then displays the amount of space used and available on the disk, and asks if you want to format another disk:

```
        362496 bytes total disk space
         40960 bytes used by system
        321536 bytes available on disk

    Format another (Y/N)?
```

When you answer **N** (no <CR> is required), the system prompt A> reappears, and the formatting step is complete.

CHANGING THE DEFAULT DRIVE

The A> is the DOS prompt from the command processor. It tells you that DOS is ready to accept commands. The A in the previous prompt is the default disk drive. This means that DOS searches only the disk in drive A for any filenames you enter and writes files to that disk unless you specify a different drive. You can ask DOS to search a disk in another drive by changing the drive designation or by specifying it in a command. To change the disk drive designation, enter the new drive letter followed by a colon. For example:

```
    A>
    A>B:<CR> (you have typed B: and <CR> in response to
    the prompt)
    B>
```

The system prompt B> appears and drive B is now the default drive. DOS searches only the disk in drive B until you specify a different default drive. To move back to drive A, simply specify A: <CR>. (Don't forget the colon.)

```
    B>A:<CR>
    A>
```

BACKING-UP YOUR DISKS

You've made a backup copy of your master software disk; you should make backup copies of all your disks. If a disk becomes damaged or if files are accidentally erased, you will still have all of the information on your backup disk.

You make backup copies of flexible disks with the DISKCOPY command; you make backup copies of fixed disks with the BACKUP command. (Both of these commands are discussed in detail in the *Commands* chapter.)

USING THE PROGRAMMABLE FUNCTION KEYS

Your NCR PC has two columns of special keys. These keys are labeled F1 through F10 and are located on the left side of the keyboard. They are special because you can define (program) them to do any function you want.

Like the automatic-program-execution feature (see next section), the programmable function keys are convenient, especially for performing an often-used or difficult function. For example, you may always want to check the contents of a disk before you access it. You could assign the directory display (DIR) command to a function key. Then, to use the command, you could simply press the key instead of typing the command through the keyboard.

Within the DOS operating system, special keys F1 thru F6 have been pre-defined for the user's convenience. For further information concerning the use of these keys, or altering and expanding the special function keys, refer to the DOS *Editing and Function Keys* chapter.

RUNNING PROGRAMS AUTOMATICALLY

If you want to run a specific program automatically each time you start DOS, you can do so with Automatic Program Execution. For example, you may want to have DOS display the names of your files each time you load DOS.

When you start DOS, the command processor searches for a file named AUTOEXEC.BAT on the DOS disk. This file is a program that DOS will run each time DOS is started. The *Learning About Commands* chapter tells you how to create an AUTOEXEC.BAT file.

MULTIPLE OPERATING SYSTEMS CONSIDERATIONS

You may use your NCR Personal Computer with more than one operating system. If you are planning to use DOS and one or more other operating systems, you are responsible for protecting data files and disk software. Generally, data protection is simply a matter of keeping the flexible disks properly labeled so that you always process using compatible disk files with their proper operating systems. You must also be sure that only compatible software and disk files are located on the same flexible disk. On the fixed disk you can partition the available space using your "hard" disk utilities. These commands are described in chapter 4.

TURNING THE SYSTEM OFF

There is no "logoff" command in DOS. To end your terminal session, open the disk drive doors and remove the disks. Then, simply turn your terminal off in response to a default drive prompt.

IF YOU HAVE A FIXED DISK

If you have one or more fixed disk drives (also called hard disks), you must continue setup procedures by defining your disk configuration. (If you have only flexible disks, you can skip this section.) Before you actually setup your fixed disk as described in "Defining the Fixed Disk", read the next two sections that contain some general comments about formatting and partioning the disk.

FORMATTING YOUR FIXED DISKS

You've already learned that each new disk must be formatted. Formatting a fixed disk is similar to formatting a flexible diskette, although some comments about fixed disks should be noted.

- To format a fixed disk unit takes approximately 2 minutes.

- Before you format a fixed disk, always use the DIR command to check that the disk drive hasn't already been used.

- If you change partition size using FDISK you must FORMAT the new partition. Use BACKUP to save any needed files in the old partition before altering the partition size, because FORMAT will destroy the files in the old partition.

PARTITIONING THE FIXED DISK

If you want DOS to share a fixed disk drive with one or more other operating systems, you must establish up to 4 "partitions" by allocating portions of the disk's storage space to each operating system. You can make the partitions any size (the total of partition sizes can not exceed the storage capacity of the disk). The DOS partition is allocated and formatted for use by DOS by the FDISK command. The other partition(s), if any, are similarly handled by the appropriate utilities in those other operating system(s). When the fixed disk is shared, you should put a label on the unit clearly showing the partition size for each operating system.

Properly used, dual operating systems significantly increase your processing capabilities. Just remember the guidelines:

- Keep compatible operating system software, application software, and data on the same disks or in the same partition of the fixed disk.

- Never use a command that could destroy the contents of a disk without first finding out what's on the disk. If you are not sure of the exact contents, use a command to identify the contents (DIR for example).

- Finally, be sure you always have backup copies of all important software and data.

DEFINING THE FIXED DISK

You define your disk configuration with the FDISK command. With self-explanatory screens, this routine is easy to use: You simply enter the name and follow the displays, carrying on a conversation with DOS.

The following paragraphs summarize the procedure for formatting and allocating the entire fixed disk to DOS. If you later decide to partition the fixed disk between DOS and one or more other operating systems, you should copy all your DOS files to flexible disks using the BACKUP command; then, reconfigure your fixed disk using other options of the FDISK command.

What you say (enter) is in bold type; what DOS responds is in normal type. Actions you must perform, and comments for guidance when performing the sequence, are discussed in the text.

```
    A>
```

You're using a flexible/fixed disk system, but DOS thinks it has only flexible disk drives available.

```
    A>FDISK
```

```
The FDISK Disk utility allows you to configure your fixed disk(s). s).


     Fixed Disk Setup Program

     Current Fixed Disk Drive

     FDISK Options

     Choose one of the following:
         1. Create MS-DOS Partition
         2. Change Active Partition
         3. Delete MS-DOS Partition
         4. Display Partition Information
         5. Select next fixed Disk Drive

   Enter choice: ..... [ ]

      Press ESC to return to DOS ..... [     ]
```

Study this main menu for a minute. These functions can all be performed with FDISK. However, you can not change either your current fixed disk drive or select your next fixed disk drive unless you have two or more fixed disk drives. If you have only one fixed disk drive, neither option 5 nor the Current fixed disk drive number [1] will be displayed.

Enter choice: 1

You now want to create your DOS partition (in this case, the entire disk). The program displays:

```
   Fixed Disk Setup Program

   Create MS-DOS Partition

   No partition defined

   Total disk space is xxxx cylinders.

   Do you wish to use the entire
   fixed disk for MS-DOS (Y/N)? ... [Y]

   Press ESC to return to FDISK Options. [     ]
```

You answer:

[Y] or <CR>

By telling DOS you want the entire fixed disk available you give yourself all the data space located on the fixed disk.

Press ESC to return to the FDISK options menue and press it again to return to DOS. When the system prompt appears enter

```
A>FORMAT C:/S/V
```

This will format drive C.

NOTE: If you do not want your fixed disk to contain the system, do not use the /S parameter (i.e. **FORMAT C:/V**). However, without the system you will not be able to boot from the fixed disk.

```
PRESS ANY KEY TO BEGIN FORMATTING DRIVE C:
```

Press any key. The program displays:

```
Formatting...
```

(then, when finished)

```
Format complete.
System transferred. (If /S is used)

Volume label (11 characters, ENTER for none)? _ (If /V is used)
```

You now enter any unique name or number up to 11 characters by which you want to identify this new disk, and press <CR> (or just press <CR>). You should give your fixed disk a volume name now, because you will never get another opportunity. The program then displays the amount of space used and available on the disk, and asks if you want to format any more disks:

```
10462496 bytes total disk space
   40960 bytes used by system
10421536 bytes available on disk

Format another (Y/N)?
```

When you answer **N** (no <CR> is required), the system prompt A> reappears, and the formatting step is complete.

```
A>COPY *.* C:
```

Use this if you want all DOS files to reside on your fixed disk. If you decide to do this, from now on your DOS files will be constantly ready and your default drive can be C. If the system has been transferred, next time when you power up or ROM Boot, and drive A is either empty or has the door "open", the system will boot from drive C.

RE-FORMATTING A FIXED DISK DRIVE

In the unlike event that an unrecoverable error should occur in a fixed disk drive, or following certain service procedures, it may be necessary to re-format the drive.

CAUTION

Re-formatting a fixed disk drive destroys all data stored on it.

Re-formatting consists of three separate procedures:

1. Pre-formatting
2. Disk partitioning using FDISK
3. Final formatting using FORMAT

1.

Place an Operating System containing the file DEBUG.COM in drive A and restart the system.

In response to the system prompt enter

```
DEBUG <CR>
```

A new prompt appears in the form of a hyphen. Now enter

```
g=C800:5 <CR>
```

The following message appears:

```
WX2 FORMAT Revision 2.0 (C) Copyright Western Digital Corp. 1984

(AH) = relative drive number (0-7)
(AC) = interlace factor (3 is standard)

Press "y" to begin formatting drive X with interleave YY
```

The information provided here is sometimes required by programmers. All you have to do is press the "y" key to start the preformatting procedure. A confirmation appears on the screen as soon as the procedure has been completed, and the system prompt is issued again.

If you decide that there might still be information on the fixed disk you need to retrieve, simply press any key other than "y" in response to the above request.

2. and 3.

You can now proceed to partition and format the fixed disk drive in accordance with the descriptions (FDISK, FORMAT) given in this Manual.

DEFINING A SERIAL PRINTER

Are you using a printer? DOS assumes it is a parallel printer. If using a serial printer, you must define it to NCR-DOS with the MODE command described in the *Commands* chapter.

THE HELP TUTORIAL

For your convenience while learning to use DOS, the HELP tutorial accompanies the operating system on the master DOS diskette. When you need information on a specific command, you can type:

```
A>HELP command<CR>
```

and find a description of the command and how to use it. If you are unsure of which command you want displayed you can type:

```
A>HELP<CR>
```

to see all the commands included in the HELP tutorial. When this "Help Menu" is displayed you can position the cursor using the cursor control keys (up, down, right, left) immediately after the the name of the command to be selected and press any key. To return to DOS press <Ctrl-Break>.

It is strongly suggested that all but the most experienced programmers read the descriptions and usages of each command in the *Commands* chapter before attempting to use them.

Files and Directories

In the following section you will learn that directories contain the names of your files. Later in this chapter, you'll learn how to name and copy your files. You'll also learn more about the NCR-DOS hierarchical directory structure that makes it easy for you to organize and locate your files.

FILES

A file is a collection of related information. A file on your disk can be compared to a file folder in a desk drawer. For example, one file folder might contain the names and addresses of the employees who work in the office. You might name this file the Employee Master File. A file on your disk could also contain the names and addresses of employees in the office and could also be named Employee Master File.

All programs, text, and data on your disk reside in files and each file has a unique name. You refer to files by their names. Later in this chapter you will find out how to name your files.

You create a file each time you enter and save data or text at your terminal. Files are also created when you write and name programs and save them on your disks.

HOW DOS KEEPS TRACK OF YOUR FILES

The names of files are kept in directories on a disk. These directories also contain information on the size of the files, their location on the disk, and the dates that they were created and updated. The directory you are working in is called your current or working directory.

An additional system area is called the File Allocation Table. It keeps track of the location of your files on the disk. It also allocates the free space on your disks so that you can create new files.

These two system areas, the directories and the File Allocation Table, enable DOS to recognize and organize the files on your disks. The File Allocation Table is copied to a new disk when you format it with the

DOS FORMAT command; also, one empty directory is created, called the root directory.

THE DIR (SHOW DIRECTORY) COMMAND

If you want to know what files are on your disk, you can use the DIR command. This command tells DOS to display all the files in the current directory on the disk that is named. For example, if your DOS disk is in drive A and you want to see the listing for the current directory on that disk, type:

DIR [A:]<**CR**>

DOS responds with a directory listing of all the files in the current directory on your DOS disk. To stop the screen to study the files, press the <Ctrl> key, and while holding it down, press the <Num Lock> key. To continue the display, press any key.

NOTE: **Two DOS system files, IO.SYS and MSDOS.SYS are "hidden" files and do not appear when you issue the DIR command.**

You can also get information about any file on your disk by typing DIR and a filename. For example, if you have created a file named MYFILE.TXT, the command

DIR MYFILE.TXT<**CR**>

gives you a display of all the directory information (name of file, size of file, date last edited) for the file MYFILE.TXT.

For more information on the DIR command, refer to Chapter 4, *Commands.*

NAMING YOUR FILES

The name of a typical DOS file looks like this:

NEWFILE.EXE

The name of a file consists of two parts. The filename is NEWFILE and the filename extension is .EXE.

The filename can be from 1 to 8 characters long. The filename extension can be three or less characters. You can type any filename in small or capital letters and DOS will translate these letters into uppercase characters.

In addition to the filename and the filename extension, the name of your file may include a drive designation. A drive designation tells DOS to look on the disk in the designated drive to find the filename typed. For example, to find directory information about the file NEWFILE.EXE which is located on the disk in drive B (drive B is NOT the default drive), type the following command:

```
DIR B:NEWFILE.EXE
```

Directory information about the file NEWFILE.EXE is now displayed on your screen.

If drive A is the default drive, DOS will search only the disk in drive A for the filename NEWFILE and so the drive designation is not necessary. A drive designation is needed if you want to tell DOS to look on another drive to find a file.

Your filenames will probably be made up of letters and numbers, but other characters are also allowed. Legal characters for filename extensions are the same as those for filenames. Here is a complete list of the characters you can use in filename and extensions:

```
a-z   0-9   $   &   #

%     !     (   )   -
```

All of the parts of a filename comprise a file specification. The term file specification (or filespec) is used in this manual to indicate the following filename format:

```
[drive designation:]filename[filename.extension]
```

Brackets indicate optional items. The drive designation is not required unless you need to indicate to DOS which disk drive to search for a file. You do not have to give your filename a filename extension.

Examples of file specifications are:

```
B:MYPROG.COB
A:YOURPROG.EXT
A:NEWFILE
TEXT
```

Universal Characters

Two special characters (called universal characters) can be used in filenames and extensions: the question mark (?) and the asterisk (*).

These special characters give you greater flexibility when using filenames in DOS commands.

The ? Character — A question mark (?) in a filename or filename extension indicates that any character can occupy that position. For example, the command:

```
DIR TEST?RUN.EXE
```

lists all directory entries on the default drive that have eight (8) characters, begin with TEST, have any next character, end with the letter RUN, and have a filename extension of .EXE. Here are some examples of files that might be listed by the previous DIR command:

```
TEST1RUN.EXE
TEST2RUN.EXE
TEST6RUN.EXE
```

The command:

```
DIR Test????.???
```

will get

```
TEST1RUN.EXE
TEST2RUN.EXE
TEST3RUN.EXE
TESTALL.EXE
```

However, the command:

```
DIR TEST1RUN.??
```

will not get .EXE

The * Character — An asterisk (*) in a filename or filename extension indicates that any character can occupy that position or any of the following positions in the filename or extension. For example:

```
DIR TEST*.EXE
```

lists all directory entries on the default drive with filenames that begin with the characters TEST and have an extension of.EXE. Here are some examples of files that might be listed by this DIR command:

```
TEST1RUN.EXE
TEST2RUN.EXE
TEST6RUN.EXE
TESTALL.EXE
```

The universal designation *.* refers to all files on the disk.

NOTE: This designation can be very powerful and destructive when used in DOS commands. For example, the command ERASE *.* deletes ALL files on the default drive.

Examples:

To list the directory entries for all files named NEWFILE on drive A (ignoring filename extensions), simply type:

DIR A:NEWFILE.*

To list the directory entries for all files with filename extensions of .TXT (Regardless of filenames) on the disk in drive B, type:

DIR B:????????.TXT or **DIR B:*.TXT**

This command is useful if, for example, you have given all your text programs a filename extension of .TXT. By using the DIR command with the universal characters, you can obtain a listing of all your text files even if you do not remember all of the filenames.

Naming Conventions

DOS treats some device names specially, and certain three letter names are reserved for the names of these devices. The following names can not be used as filenames or extensions:

AUX

Used when referring to input from or output to an auxiliary device (such as a printer or disk drive).

CON

Used when referring to keyboard input or to output to the terminal console (screen).

LST
or
PRN

Used when referring to the printer device. These may be used as file extensions but never as filenames (unless you want to print or list the entire extension).

NUL

Used when you do not want to create a particular file, but the command requires an input or output filename.

Even if you add device designations or filename extensions to these filenames, they remain associated with the devices listed above. For example, A:CON.XXX always refers to the console and can not be the name of a disk file.

The filename and especially the filename extension are used to meaningfully identify the file. If you are an experienced computer user, you probably already know that certain extension names generally describe the type or use of the file. The following list includes some of the more common extensions. Some of these are created by the operating system or DOS compatible software, where others are common usage in the data processing industry.

COMMON EXTENSIONS

ASM	— Assembler output
BAK	— Line Editor backup copy
BAS	— BASIC program (source)
BAT	— An executable batch program file
BIN	— EXE2BIN default
COB	— Compiler output for Cobol programming language
COM	— Command file on system disk
C	— Compiler output for "C" programming language
DAT	— Common usage for data files
DCT	— Documentation for software
EXE	— Any executable file
LIB	— Linker library file input
MAC	— Macro filename

MAP — Linker listing default

OBJ — Compiler output for most programming language (object)

OVR — File overlay

SYS — A system file

TXT — Any text file

TMP — A temporary file

COPYING FILES

Just as with paper files, you often need more than one copy of a disk file. The COPY command allows you to copy one or more files to another pre-formatted disk. You can also give the copy a different name if you specify the new name in the COPY command.

The COPY command can also make copies of files on the same disk. In this case, you must assign a different filename or you will get the error message "FILE CANNOT BE COPIED ONTO ITSELF". You cannot make a copy of a file on the same disk unless you specify a different filename for the new copy.

The format of the COPY command is:

```
    COPY filespec [filespec]
```

For example:

```
    COPY A:MYFILE.TXT B:MYFILE.TXT
          or
    COPY A:MYFILE.TXT B:
```

copies the file MYFILE.TXT on disk A to a file that will be named MYFILE.TXT on disk B. A duplicate copy of MYFILE.TXT now exists on drive B.

Figure 2.1 illustrates what a COPY command does.

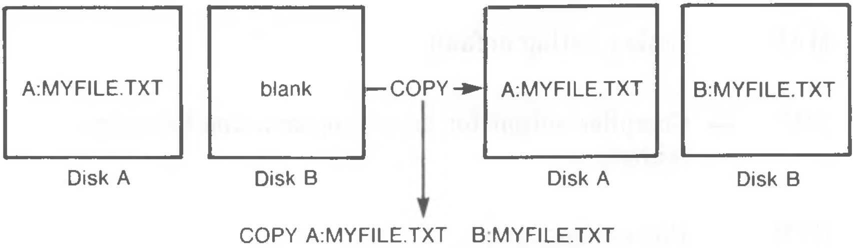


Figure 2.1 Copying files to another disk

If you want to duplicate the file named MYFILE.TXT on the same disk, type:

```
COPY A:MYFILE.TXT A:NEWNAME.TXT
```

You now have two copies of your file on disk A, one named MYFILE.TXT and the other named NEWNAME.TXT. Figure 2.2 illustrates this example.

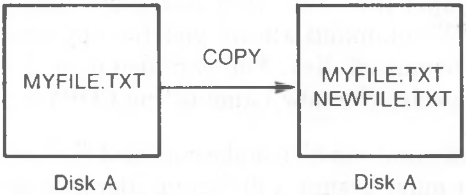


Figure 2.2 Copying files on the same disk

You can also copy all files on a disk to another disk (that is, make a backup copy) with the COPY command by using the universal character * (for example: COPY A:*.* B:). Refer to the *Commands* chapter for more information on this process.

PROTECTING FILES

DOS is a powerful and useful tool in processing your personal and business information. As with any information system, inadvertent errors may occur and information may be misused. If you are processing information that cannot be replaced or that requires a high level of security, you should take steps to ensure that your data and programs are protected from accidental or unauthorized use, modification, or destruction. Simple measures you can take, such as removing your disks when they are not in use, keeping backup copies of valuable information, using the write-protect tabs supplied with your diskettes, and installing your equipment in a secure facility can help you maintain the integrity of the information in your files.

DIRECTORIES

As you now know, the names of your files are kept in a directory on each disk. The directory also contains information on the size of the files, their locations on the disk, and the dates that they were created or updated.

If multiple users share your computer, or if you are working on several different projects, the number of files in the directory can become large and unwieldy. You may want your own files kept separate from a co-worker's, or you may want to organize your programs into categories that are convenient for you.

In an office, you can separate files by putting them in different filing cabinets, creating, in effect, different directories of information. DOS allows you to organize the files on your disks into directories. Directories are a way of dividing your files into convenient groups of files. For example, you may want all of your accounting programs in one directory and text files in another. Any one directory can contain any reasonable number of files, and it may also contain other directories (referred to as sub-directories). This method of organizing your files is called a *hierarchical directory structure.*

A hierarchical directory structure can be thought of as a "tree" structure: directories are branches of the tree and files are the leaves, except that the "tree" grows downward; that is, the "root" is at the top. The root is the first level in the directory structure. It is the directory that is automatically created when you format a disk and start putting files in it. You can create additional directories and sub-directories by following the instructions in the *Learning About Commands* chapter.

The "tree" of file structure grows as you create new directories for groups of files or for other people on the system. Within each new directory, files can be added, or new sub-directories can be created.

It is possible for you to "travel" around this tree; for instance, you can find any file in the system by starting at the root and traveling down any of the "branches" to the desired file. Conversely, you can start where you are within the file system and travel towards the "root".

The filenames discussed earlier in this chapter are relative to your current directory and do not apply system-wide. Thus, when you turn on your computer, you are "in" your directory. Unless you take special action when you create a file, the new file is created in the directory in which you are currently working. Users can have files of the same name that are unrelated because each is in a different directory.

Figure 2.3 illustrates a typical hierarchical directory structure

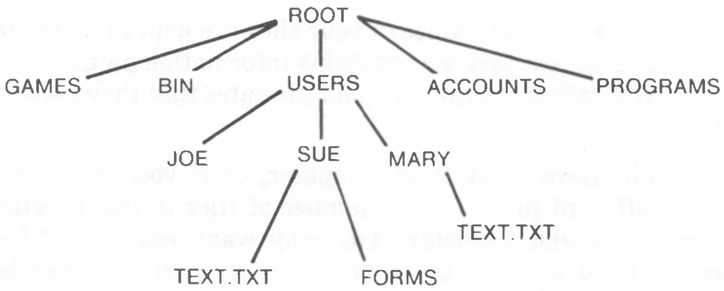


Figure 2.3 A sample hierarchical directory structure

The "ROOT" directory is the first level in the directory structure. You can create sub-directories from the "ROOT" by using the MKDIR (Make Directory) command (refer to the *Commands* chapter for information on MKDIR). In this example, five sub-directories of "ROOT" have been created. These include:

- A directory of games, named GAMES.
- A directory of all external commands, named BIN (refer to the chapter, *Learning About Commands,* for more information on the BIN directory.).
- A USERS directory containing separate sub-directories for all users of the system.
- A directory containing accounting information, named ACCOUNTS.
- A directory of programs, named PROGRAMS.

Joe, Sue and Mary each have their own directories which are sub-directories of the USERS directory. Sue has a sub-directory under the \USERS SUE sub-directory named FORMS. Sue and Mary have files in their sub-directories, each named TEXT.TXT. Notice that Mary's text file is unrelated to Sue's.

This organization of files and directories is not important if you only work with files in your own directory, but, if you work with someone else or on several projects at one time, the hierarchical directory structure becomes extremely useful. For example, you could get a list of the files in Sue's FORMS directory by typing:

```
DIR \USERS\SUE\FORMS
```

Note that the back slash mark (\ ) is used to separate directories from other directories and files.

To find out what files Mary has in her directory, you could type:

```
DIR \USERS\MARY
```

FILENAMES AND PATHS

When you use hierarchical directories, you must tell DOS where the files are located in the directory structure. Both Mary and Sue, for example, have files named TEXT.TXT. Each will have to tell DOS in which directory her files resides if she wants to access it. This is done by giving a pathname to the file.

Pathnames

A simple filename is a sequence of characters that can optionally be preceded by a drive designation and followed by an extension. A pathname is a sequence of directory names followed by a simple filename, each separated from the previous one by a backslash (\ )

The syntax of pathnames is:

```
[d:] [directory]\[directory...]\[filename] [.ext]
```

If a pathname begins with a backslash, DOS searches for the file beginning at the root (or top) of the "tree"; otherwise, DOS begins at the user's current directory, known as the working directory, and searches downward from there. The pathname of Sue's TEXT.TXT file is \USERS\SUE\TEXT.TXT.

When you are in your working directory, a filename and its corresponding pathname may be used interchangeably. The following list shows some sample names:

\
: Indicates the root directory.

\PROGRAMS
: Sample directory under the root directory containing program files.

\USERS\SUE\FORMS\1A
: A typical full pathname. This one happens to be a file named 1A in the directory named FORMS belonging to the USERS named Sue.

USERS\MARY

A relative pathname; it names the file or sub-directory MARY in sub-directory USERS of the working directory. If the working directory is the root(\), it names \BIN\MARY.

TEXT.TXT

Name of a file or directory in the working directory.

DOS provides special shorthand notations for the working directory and the parent directory (one level up) of the working directory:

.(single period)

This shorthand notation indicates the name of the working directory in all hierarchical directory listings. This entry is created when a directory is made.

..(double period)

The shorthand name of the working directory's parent directory. If you type:

```
DIR ..
```

then the files in the parent directory of your working directory are listed. If you type:

```
DIR ..\..
```

Then the files in the parent's PARENT directory are listed.

Pathing and External Commands — External commands reside on disk as program files. They must be read from the disk before they execute. (For more information on external commands, refer to the chapter, *Learning About Commands.)*

When you are working with more than one directory you may find it convenient to put all DOS external commands into a separate directory so they do not clutter your other directories. When you issue an external command, DOS immediately checks your working directory to find that command. You must tell DOS in which directory these external commands reside. This is done with the PATH command.

For example, if you are in a working directory named \BIN\PROG, and all external commands are in \BIN, you must specify the \BIN path to find the FORMAT command. The command

```
PATH\BIN
```

tells DOS to search in your working directory and the \BIN directory for all commands. You only have to specify this PATH once during your terminal session. DOS will then search in \BIN for the external commands. If you want to know what the current PATH is, type the word PATH and the current value of PATH will be displayed.

For more information on the PATH command, see the *Commands* chapter.

Pathing and Internal Commands — Internal commands are the simplest, most commonly used commands. They execute immediately because they are incorporated into the command processor. (For more information on internal commands, refer to the chapter, *Learning About Commands.)*

Some internal commands can use PATHS. The four commands, COPY, ERASE, DIR, and TYPE, have greater flexibility when you specify a pathname after the command.

COPY pathname pathname

> If the second pathname to COPY is a directory, all files are copied into that directory.

ERASE pathname

> If the pathname is a directory, all the files in that directory are deleted. NOTE: The prompt "Are you sure (Y/N)?" is displayed if you try to delete a path. Type Y to complete the command, or type N for the command to abort.

DIR pathname

> Displays the directory for a specific path.

TYPE pathname

> You must specify a file in the pathname for this command.

DISPLAYING YOUR WORKING DIRECTORY

All commands are executed while you are in your working directory. You can find out the name of the directory you are in by issuing the CHDIR (Change Directory) command with no options. For example if your current directory is \USERS\JOE, when you type:

```
CHDIR
```

you will see:

```
A: \USERS\JOE
```

This is your current drive designation plus the working directory (\USERS\JOE).

If you now want to see what is in the \USER\JOE directory, you can use the DIR command. The following is an example of the display you might receive from the DIR command for a sub-directory:

```
Volume in drive A has no label
Directory of A: \ USERS \JOE

.                   (DIR)        7-04-84     10:09a

..                  (DIR)        7-04-84     10:09a

TEXT                (DIR)        7-04-84     10:09a

FILE1     COM             5243   7-04-84      9:30a

        4#File(s)       250518   bytes free
```

A volume label for this disk was not assigned when the disk was formatted. Note that both files and directories are in this output. As you can see, Joe has another directory in this "tree" structure named TEXT. The '.' indicates the working directory \USERS\JOE, and '..' is the shorthand notation for the parent directory \USERS. FILE1.COM is a file in the \USERS\JOE directory. All of these directories and files reside on the disk in drive A.

Because files and directories are listed together (see previous display), DOS does not allow you to give a sub-directory the same name as a file in that directory. For example, if you have a path \BIN\USERS\JOE where JOE is a sub-directory, you cannot create a file in the USERS directory named JOE.

CREATING A DIRECTORY

To create a sub-directory in your working directory, use the MKDIR (Make Directory) command. For example, to create a new directory named NEWDIR under your working directory, simply type:

```
MKDIR NEWDIR
```

After this command is executed, a new directory exists in your "tree" structure under your working directory. You can also make directories anywhere in the "tree" structure by specifying MKDIR and then a pathname. DOS automatically creates the . and .. entries in the new directory.

To put files in the new directory, use the method described in Chapter 6, *Line Editor (EDLIN)*, or use the COPY command.

CHANGING YOUR WORKING DIRECTORY

Changing from your working directory to another directory is very easy. Simply issue the CHDIR (Change Directory) command and supply a pathname. For example:

```
CHDIR \USERS
```

changes the working directory from \USERS\JOE to \USERS. You can specify any pathname after the command to "travel" to different branches and leaves of the directory tree. The command "CHDIR.." will always put you in the parent directory of your working directory.

REMOVING A DIRECTORY

To delete a directory in the "tree" structure, use the RMDIR (Remove Directory) command. For example, to remove the directory NEWDIR from the working directory, type:

```
RMDIR NEWDIR
```

Note that the directory NEWDIR must be empty except for the . and .. entries before it can be removed; this connection prevents you from accidentally deleting files and directories. You can remove any directory by specifying its pathname. To remove the \BIN\USERS\JOE directory, make sure that it has only the . and .. entries, then type:

```
RMDIR \BIN\USERS\JOE
```

To remove all the files in a directory (except for the . and .. entries), type DEL and then the pathname of the directory. For example, to delete all files in the \BIN\USERS\SUE directory, type:

```
ERASE \BIN\USERS\SUE
```

You can not delete the . and .. entries. They are created by DOS as part of the hierarchical directory structure.

In the next chapter, you learn more about DOS commands.

Chapter 3

Learning About Commands

Commands are a way of communicating with the computer. By entering NCR-DOS commands at your terminal, you can ask the system to perform useful tasks:

- Compare, copy, display, delete, and rename files
- Copy and format disks
- Execute system programs such as EDLIN, as well as your own programs
- Analyze and list directories
- Enter date, time, and remarks
- Set various printer and screen options
- Copy DOS system files to another disk
- Request DOS to wait for a specific period of time

TYPES OF DOS COMMANDS

There are two types of DOS commands: internal commands and external commands.

INTERNAL COMMANDS

Internal commands are the simplest, most commonly used commands. You cannot see these commands when you do a directory listing of your DOS disk; they are part of the command processor. When you type these commands, they execute immediately. The following internal commands are described in the next chapter:

BREAK
CHDIR (CD)
CLS
COPY
CTTY
DATE
DIR
ECHO
ERASE DEL
EXIT
FOR
GOTO
IF
MKDIR (MD)
PATH
PAUSE
PROMPT
REM
REN (RENAME)
RMDIR (RD)
SET
SHIFT
TIME
TYPE
VER
VERIFY
VOL

EXTERNAL COMMANDS

External commands reside on disk as program files. They must be read from disk before they can execute. If the disk containing the command is not in the drive, DOS will not be able to find and execute the command.

Any filename with a filename extension of .COM, .EXE, or .BAT is considered an external command. For example, the program FORMAT.COM is an external command. Because all external commands reside on disk, you can create commands and add them to the system. Programs that you create with most languages (including assembly language) will be .EXE (executable files).

When you enter an external command, you need not include its filename extension. The following external commands are described in the next chapter.

ASSIGN
BACKUP
BUFFERS
CHKDSK
COMMAND
COMP
DEVICE
DISKCOMP
DISKCOPY
ESC
EXE2BIN
FDISK
FILES
FIND
FORMAT
GRAPHICS
HELP
MODE
MORE
PRINT
RECOVER
RESTORE
SHELL
SORT
SYS
TREE

THE CONCEPT OF SYNTAX

Before you begin to use the DOS commands, you must understand the syntax notation. Syntax refers to the precise format, or optional format variations, that DOS recognizes as legitimate commands. Some errors in syntax will just cause DOS to respond with a question mark, indicating that it does not know what you want. Other errors, more serious, can cause DOS to do something other than what you meant to tell it to do.

DOS SYNTAX NOTATION

The following notation is used throughout this manual in descriptions of command and statement syntax. Don't be overwhelmed by this list; after you use the commands a few times, the notation becomes quickly familiar.

[] Square Brackets

Indicates that the enclosed entry is optional.

<> Angle Brackets

Indicates that you supply the text or specific key(s) for this entry. When the angle brackets enclose lowercase text, type in an entry defined by the text; for example, <filename>.

When the angle brackets enclose uppercase or initial uppercase text, such as <Ins>, <F3>, or <Del>, press the specific key whose keytop has that value printed on it.

When the angle brackets enclose two keytop names connected with a -, such as <Ctrl-Break> or <Ctrl-Z>, press the first key and HOLD IT DOWN while pressing the second indicated key.

There are two special exception cases. The notation <CR> is used when the (←) key is to be pressed. Second, since there is no universally-accepted substitute for a space (blank) character, the notation <space> is used in some instances where the entry of a space character is critical in the syntax.

{} Braces

Indicate that you have a choice between two or more entries. At least one of the entries enclosed in braces must be chosen unless the entries are also enclosed in square brackets.
A bar separates the entries.

| A bar when used with a filter indicates a pipe. When used within braces, a bar separates the entries.

(. . .) ellipses

Indicate that an entry may be repeated as many times as needed or desired.

CAPS Capital Letters

Indicated portions of statements or commands that must be entered exactly as shown. Capital letters also indicate specific keys, such as <CR>.

All other punctuation, such as commas, colons, slash marks, and equal signs must be entered exactly as shown.

COMMAND OPTIONS

Options can be included in your commands to specify additional information to the system. If you do not include some options, DOS provides a default value. (Refer to individual command descriptions in the next chapter for the default values.)

The following is the format of all commands:

```
Command [options...]
```

where options can be:

d:
    Refers to the disk drive designation.

filename
    Refers to any valid name for a disk file. The filename option does not refer to a device or to a disk drive designation.

.ext
    Refers to an optional filename extension consisting of a period and 1-3 characters. When used, filename extensions immediately follow filenames.

filespec
    Refers to an optional drive designation, a filename, and an optional three letter filename extension in the following format:

```
[d:]filename[.ext]
```

pathname
    Refers to a pathname of filename in the following format:

```
[directory]\[<directory...>]\[filename.ext]
```

option parameter
    These control DOS commands. They are preceded by a forward slash (for example, /P), or allow a choice (ON or OFF)

COMMON ENTRY CONVENTIONS

The following information applies to all commands:

1. Commands are usually followed by one or more options.

2. Commands and options may be entered in uppercase of lowercase, or a combination of the two.

3. Commands and options must be separated by delimiters. Because they are easiest, you will usually use the space and comma as delimiters. For example:

```
DEL MYFILE.OLD NEWFILE.TXT
RENAME THISFILE THATFILE
```

You can also use the semicolon (;), the equal sign (=), or the tab key as delimiters in DOS commands. (In this manual, we use a space as the delimiter in commands.)

4. Do not separate a file specification with delimiters, since the colon and the period already serve as delimiters.

5. When instructions say "Press any key," you can press any alpha (A-Z) or numeric (0-9) key or the space bar or <CR>.

6. You must include the filename extension when referring to a file that already has a filename extension.

7. You can abort commands when they are running by pressing (Ctrl-Break).

8. Commands take effect only after you have pressed the <CR> key.

9. Universal characters and device names (for example, PRN or CON) are not allowed in the names of any commands.

10. When commands produce a large amount of output on the screen, the display automatically scrolls to the next screen. You can press <CR> with <Num Lock> key to suspend the scrolling. Press any key to resume scrolling.

11. Editing and function keys can be used when entering commands. Refer to the DOS *Editing and Function Keys* chapter for a complete description of these keys.

12. The prompt from the command processor is the default drive designation plus a greater-than (>) sign; for example, A>.

13. Disk drives will be referred to as source drives and destination drives. A source drive is the drive you transfer information from; a destination drive is the drive you transfer information to.

BATCH PROCESSING

Often you may find yourself typing the same sequence of commands over and over to perform some commonly used task. With DOS, you can put the command sequence into a special file, called a batch file. "Batches" of your commands in such files are processed as if they were from the keyboard. Each batch file must be named with the .BAT extension, but can be executed by typing the filename without its extension.

You can create a batch file by either using the Line Editor (EDLIN) or by using the method shown in the "Creating an AUTOEXEC.BAT File" section later in this chapter.

Two DOS commands are available for use expressly in batch files: REM and PAUSE. REM permits you to include remarks and comments in your batch files without these remarks being executed as commands. PAUSE prompts you with an optional message and permits you to either continue or abort the batch process at a given point. REM and PAUSE are described in detail in the *Commands* chapter under "Batch Commands".

Batch processing is useful if you want to execute several commands with one batch command, such as when you format and check a new disk. For example, a batch file called NEWDISK.BAT might look like this:

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

To execute this .BAT file, simply type the filename with or without the .BAT extension:

NEWDISK [.BAT]

The result is the same as if each of the lines in the .BAT file was entered at the terminal as individual commands.

Figure 3.1 shows the steps used to write, save, and execute a .BAT file.

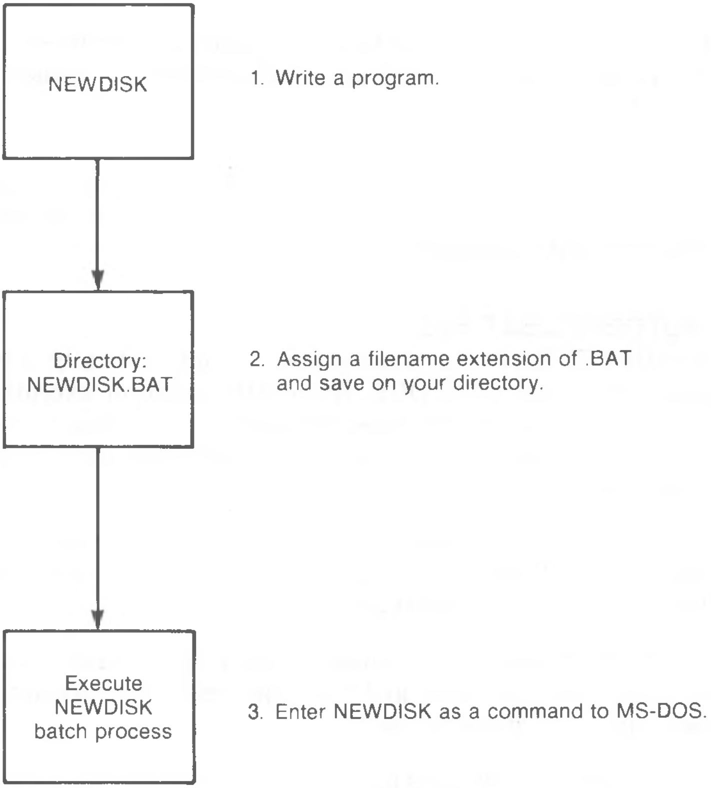

NEWDISK
1. Write a program.
Directory:
NEWDISK.BAT
2. Assign a filename extension of .BAT and save on your directory.
Execute
NEWDISK
batch process
3. Enter NEWDISK as a command to MS-DOS.

Figure 3.1

The following list contains information that you should read before executing a batch file.

1. Do not enter the filename BATCH (unless the name of the file you want to execute is BATCH.BAT)

2. Enter only the filename to execute the batch file; do not enter the filename extension.

3. The commands in the file named <filename>.BAT are executed.

4. If you press <Ctrl-Break> while in batch mode, this prompt appears:

   Terminate batch job (Y/N)?

   If you press Y, the remainder of the commands in the batch file are ignored and the system prompt appears.

   If you press N, only the current command ends and batch processing continues with the next command in the file.

5. If you remove the disk containing a batch file being executed, DOS prompts you to insert it again before the next command can be read.

6. The last command in a batch file may be the name of another batch file. This allows you to call one batch file from another when the first is finished.

THE AUTOEXEC.BAT FILE

An AUTOEXEC.BAT file allows you to automatically execute programs when you start DOS. Automatic program execution is useful when you want to run a specific application package, or when you want to execute a batch program automatically each time you start the system.

When you start DOS, the command processor searches the disk for a file named AUTOEXEC.BAT. The AUTOEXEC.BAT file is a batch file that runs each time you start the system.

If the AUTOEXEC.BAT file is found, the file is immediately executed by the command processor and the date and time prompts are bypassed, unless included in the file.

If the AUTOEXEC.BAT file is not found when you first load the disk, then the date and time prompts are issued. Figure 3.2 shows how DOS uses the AUTOEXEC.BAT file.

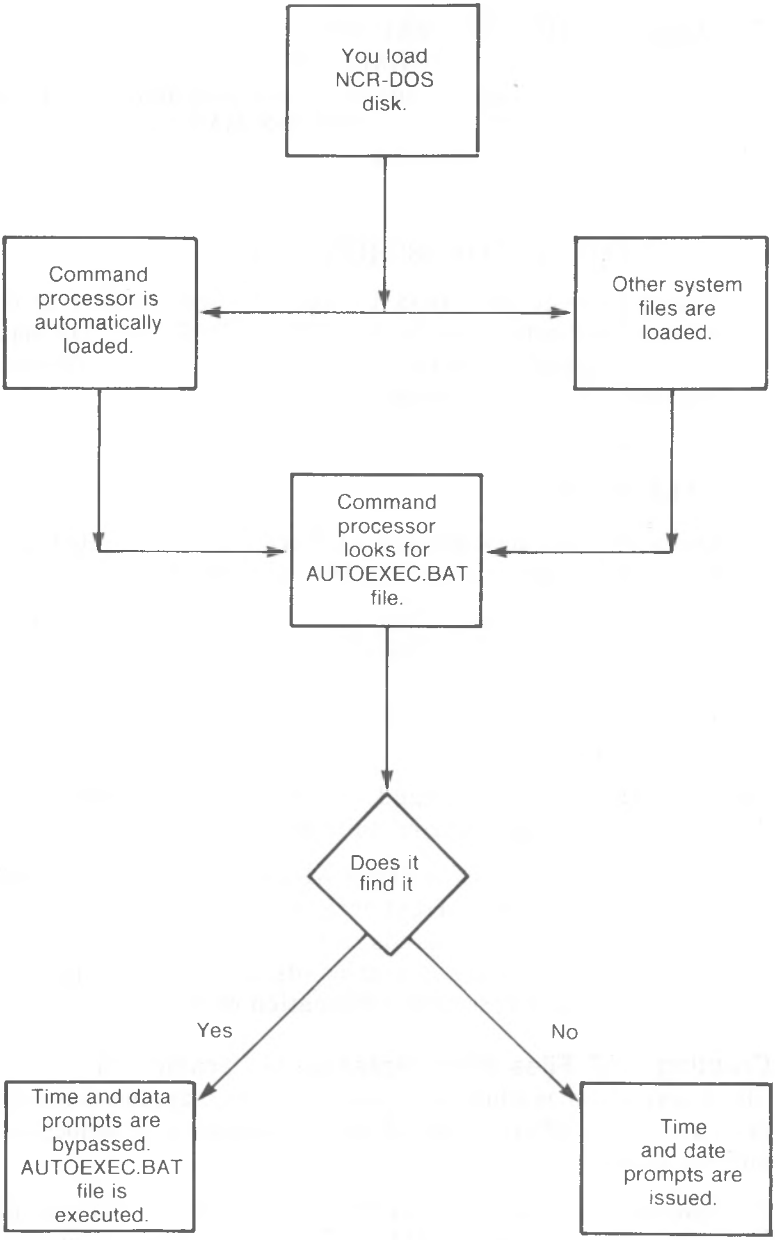

You load NCR-DOS disk.
Command processor is automatically loaded.
Other system files are loaded.
Command processor looks for AUTOEXEC.BAT file.
Does it find it
Yes
No
Time and data prompts are bypassed. AUTOEXEC.BAT file is executed.
Time and date prompts are issued.

Figure 3.2

Creating An AUTOEXEC.BAT File

To see how to create an AUTOEXEC.BAT file, assume that each time you start DOS, you want to automatically load and run a program called CALENDAR within the AUTOEXEC.BAT. You could create an AUTOEXEC.BAT file as follows:

1. Type:

   COPY CON: AUTOEXEC.BAT<CR>

   This statement tells DOS to copy the information from the console (keyboard) into the AUTOEXEC.BAT file (CON replaces the drive identifier). Note that the AUTOEXEC.BAT file must be located in the root directory of your disk.

2. Now type:

   CALENDAR

   This statement goes into the AUTOEXEC.BAT file. It tells DOS to run the program whenever DOS is started.

3. Press the <F6> key; then press the <CR> key to put the command in the AUTOEXEC.BAT file.

4. The CALENDAR program will now run automatically whenever you start DOS.

You can enter any command or series of commands in the AUTOEXEC.BAT file; one command per line.

NOTE: Remember that if you use an AUTOEXEC.BAT file, DOS does not prompt you for a current date and time unless you include the DATE and TIME commands in the AUTOEXEC.BAT file. You should include these two commands in your AUTOEXEC.BAT file to keep your directory information current.

Creating .BAT Files With Replaceable Parameters

There may be times when you want to create an application program and run it with different sets of data. This data may be located in a number of files.

When used in commands, a parameter is an option that you define. You can create a batch (.BAT) file with dummy (replaceable) parameters. These parameters, named %0-%9, can be replaced by values supplied when you run the .BAT file.

For example, if you have a number of BASIC source files to compile, you could write the following .BAT file:

```
A>COPY CON:BASICOMP.BAT
COPY %1.BAS %2.BAS
B:
A:GWBCOM %2.BAS,%2.OBJ,%2.OBJ
A:
TYPE %2.LST
TYPE %0.BAT
```

Now, press <F6> and then press <CR>. DOS responds with the message:

```
1 File(s) copied
A>
```

The file BASICOMP.BAT, which consists of six commands, is now on the disk in the default drive.

The dummy parameters %1 and %2 are replaced sequentially by the parameters you supply when you run the file. The dummy parameter %0 is always replaced by the drive designator, if specified, and the filename of the batch file (BASICOMP).

NOTE: Up to 10 dummy parameters (%0-%9) can be specified. (Refer to the SHIFT command if you wish to specify more than 10 parameters.) Also, if you use the percent sign as part of a filename within a batch file, you must type it twice. For example, to specify the file ABC%, you must type it as ABC%% in the batch file.

Executing a .BAT File

To execute a batch file with replaceable parameters, you must enter the batch filename (without its extension) followed by the parameters you want DOS to substitute for %1, %2, etc.

To execute the BASICOMP.BAT file shown in the previous section, type:

```
BASICOMP A:PROG1 B:PROG2 <CR>
```

BASICOMP is substituted for %0, A:PROG1 for %1, and B:PROG2 for %2.

The result is the same as if you had typed each of the commands in BASICOMP with their parameters, as follows:

```
Copy A:PROG1.BAS B:PROG2.BAS <CR>
B:<CR>
```

```
A:GWBCOM B:PROG2.BAS, B:PROG2.OBJ, B:PROG2.LST <CR>
A:<CR>
TYPE B:PROG2.LST <CR>
TYPE BASICOMP.BAT <CR>
```

The following table show DOS replaces each of the above parameters:

BATCH FILENAME	PARAMETER1 (%0)	PARAMETER2 (%1) (PROGNAME)	PARAMETER3 (%2) (PROGNAME)
BASICOMP	BASICOMP	PROG1	PROG2

Remember that the dummy parameter %0 is always replaced by the drive designator (if specified) and the filename of the batch file.

INPUT AND OUTPUT

DOS always assumes that input comes from the keyboard and output goes to the terminal screen. However, the flow of command input and output can be redirected. Input can come from a file rather than a terminal keyboard, and output can go to a file or to a line printer instead of to the terminal. In addition, "pipes" can be created that allow output from one command to become the input to another. Redirection and pipes are discussed in the next sections.

REDIRECTING YOUR OUTPUT

Most commands produce output that is sent to your terminal. You can send this information to a file by using a greater-than sign (>) in your command. For example, the command

```
DIR<CR>
```

displays a directory listing of the disk in the default drive on the terminal screen. The same command can send this output to a file named MYFILES by designating the output file on the command line:

```
DIR >MYFILES<CR>
```

If the file MYFILES does not already exist, DOS creates it and stores your directory listing in it. If MYFILES already exists, DOS overwrites what is in the file with the new data.

If you want to append your directory or a file to another file (instead of replacing the entire file), two greater-than signs (>>) can be used to append the output of the command (such as directory listing) to the end of a specified file. The command

```
DIR >>MYFILES<CR>
```

appends your directory listing to a currently existing file named MYFILES. If MYFILES does not exist, it is created.

REDIRECTING YOUR INPUT

It is often useful to have input for a command come from a file rather than from a terminal. This is possible by using a less-than sign (<) in your command. In the following example, PROG1 does not wait for input from the keyboard but receives input from the file PARAFIL.TXT:

```
PROG1 < PARAFIL.TXT
```

In the following example, the command

```
SORT <NAMES>LIST1<CR>
```

sorts the file NAMES and sends the sorted output to a file named LIST1.

FILTERS

A filter is a command that reads your input, transforms it in some way, and then outputs it, usually to your terminal or to a file. In this way, the data is said to have been "filtered" by the command. Since filters can be put together in many different ways, a few filters can take the place of a large number of more specific commands.

DOS filters include FIND, MORE, and SORT and perform the following functions:

FIND
: Searches for a constant string of text in a file.

MORE
: Takes a standard terminal output and displays it, one screen at a time.

SORT
: Sorts text in ascending or descending order and starting on a chosen column.

You can see how these filters are used in the next section.

COMMAND PIPING

If you want to give more than one command to the system at a time, you can "pipe" commands. For example, you may occasionally need to have the output of one program sent as the input to another program. A typical case would be a program that produces output in columns. It could be desirable to have this columnar output sorted.

Piping is done by separating commands with the pipe separator, which is the vertical bar symbol (¦). For example, the command

```
DIR ¦ SORT<CR>
```

gives you an alphabetically sorted listing of your directory. The vertical bar causes all output generated by the left side of the bar to be sent to the right side of the bar for processirg.

Piping can also be used when you want to output to a file. If you want your directory sorted and sent to a new file (for example, DIREC.FIL), you would type:

```
DIR ¦ >SORT>DIREC.FIL
```

DOS creates a file named DIREC.FIL on your default drive. DIREC.FIL contains a sorted listing of the directory on the default drive, since no other drive was specified in the command. To specify drive other than the default drive, type:

```
DIR ¦ SORT>B:DIREC.FIL<CR>
```

This command sends the sorted data to a file named DIREC.FIL on drive B.

A pipeline may consist of more than two commands. For example,

```
DIR ¦ SORT ¦ MORE<CR>
```

sorts your directory, shows it to you one screen at a time, and puts "--MORE--" at the bottom of your screen when there is more output to be seen.

You will find many uses for piping commands and filters. You will also find more information on using filters in the next chapter, *Commands* .

Chapter 4

Commands

This chapter describes each of the NCR-DOS commands, arranged in alphabetical order for quick reference. Certain commands are used only if you are writing batch files. These commands, ECHO, FOR, GOTO, IF, PAUSE, REM and SHIFT, are noted as batch processing commands and can be found in the BATCH COMMANDS section of this chapter. Similarly, the commands BREAK, BUFFERS, DEVICE (with RAM-Disk facility), FILES, and SHELL are located in the CONFIG.SYS section of this chapter. The individual command descriptions are preceded by a table summarizing the complete set.

Before studying or using any of the commands, be sure to become familiar with the notations that indicate how to format a command. (The notations were explained in an earlier chapter, but are important enough to repeat.)

CAPS

Words shown in capital letters are required entries. These words are called keywords and must be entered exactly as shown. You can enter these keywords in any combination of upper/lower case; DOS converts all keywords to uppercase.

<>

You supply the keystroke(s) for items enclosed in angle brackets (<>). For example, you should press the Ctrl (Control) key and the Scroll Lock (Break) key simultaneously for <Ctrl-Break>.

filespec-

The word "filespec" asks for the entire name of the file (i.e. disk drive letter, filename, and file extension — d:filename.ext)

[]

Items in square brackets ([]) are optional. If you include optional information, do not include the square brackets, only the information within the brackets.

{ }

Items contained in braces ({ }) are identical in usage and any one may be used, but only one may be used. The braces should not be included.

|

A bar when used with a filter indicates a pipe.

When used within braces, a bar separates the entries.

...

An ellipses indicates that you may repeat an item as many times as you want.

<CR>

To execute any command you must press the <CR> (<-) key.

You must include all punctuation where shown (with the exception of square brackets and braces), such as commas, equal signs, question marks, colons, or slashes. You must include all blanks where spaces are shown.

Name	Purpose	Syntax
ASSIGN	Assigns a disk drive	ASSIGN [d1=d2...]
BACKUP	Copies fixed disk to flexible disks	BACKUP d1:[pathname] [filename[.ext]] d2:[/S] [/M] [/A] [/D:mm-dd-yy]
BATCH	Series of commands for batch processing: ECHO, FOR, GO-TO, IF, PAUSE, REM, SHIFT	
BREAK	Sets Ctrl-C check (described under CONFIG.SYS)	BREAK ON BREAK OFF
BUFFERS	Tells DOS how many disk buffers to use (described under CONFIG.SYS)	BUFFER=n
CHDIR (CD)	Changes directories; displays working directories	CHDIR [[d:]pathname]
CHKDSK	Scans the directory of the default or designated drive and checks for consistency	CHKDSK [d:] [filename.ext] [/F] [/V]
CLS	Clears the screen	CLS
COMMAND	Invokes secondary command processor	COMMAND d:pathname[/P] [/C string]
COMP	Compares two files	COMP [filespec1] [filespec2]
CONFIG	Configures operating system with BREAK, BUFFER, DEVICE, FILES, SHELL commands	
COPY	Copies specified file(s) to disk	COPY [filespec1] [/A] [/B] [filespec2] [/V] [/A] [/B]
CTTY	Changes console TTY	CTTY DEV
DATE	Displays and sets date	DATE [mm-dd-yy]
DEL	Deletes file(s) specified	DEL [d:] [filespec] [pathname]
DEVICE	Allows peripheral connection (described under CONFIG.SYS)	DEVICE=[DEV]
DIR	Lists requested directory entries	DIR [d:] [filespec] [/P] [/W]
DISKCOMP	Compares flexible disks	DISKCOMP [d1:] [d2:] [/1] [/8]

Name	Purpose	Syntax
DISKCOPY	Copies disks	DISKCOPY [d1:] [d2:] [/1]
ECHO	Turns batch file echo feature on/off (described under BATCH)	ECHO ON message ECHO OFF message ECHO message
ERASE	Deletes specified files	ERASE [filespec]
ESC	Enables input of escape sequences through the keyboard	see description
EXE2BIN	Converts file to binary format	EXE2BIN [filespec1] [filespec2]
EXIT	Exits command and returns to lower level	EXIT
FDISK	Formats fixed disk	FDISK
FILES	Increases number of files (described under CONFIG.SYS)	File = n
FIND	Searches for a constant string of text	FIND [N] [/C] [/N] "string" [filespec]
FOR..IN..DO..	Batch command extension (described under BATCH) ● for batch processing ● for interactive processing	 FOR %% variable IN list DO command FOR % variable IN list DO command
FORMAT	FORMATS disk for NCR-DOS files ● flexible disks ● fixed disks	 FORMAT [d:] [/8] [/1] [/V] [/B] [/S] FORMAT [d:] [/V] [/S]
GOTO	Batch command extension (described under BATCH)	GOTO label
GRAFTABL	Supports character display in graphics mode	GRAFTABL
GRAPHICS	Prints display on matrix printer	GRAPHICS
HELP	Display on-line information about commands	HELP [command]
IF	Batch command extension (described under BATCH)	If condition command
KEYB	Changes the keyboard format	See description
MKDIR (MD)	Makes a directory	MKDIR [d:] [pathname]
MODE	Defines mode operation	See description
MORE	Displays output screen by screen	MORE
PATH	Sets a command search path	PATH [[d1:]pathname1][:[d2:pathname2]

Name	Purpose	Syntax
PAUSE	Pauses for input (described under BATCH)	PAUSE [message]
PRINT	Background print feature; prints a text file on a line printer	PRINT [[d:] [pathname1]filename1[.ext1]] [/T] [/C] [/P]...
PROMPT	Designates command prompt	PROMPT [prompt-string]
RAMDISK	Allocates Ram-Disk memory (described under CONFIG.SYS/DEVICE)	DEVICE=RAMDISK.SYS
RECOVER	Recovers a bad disk	RECOVER [d:] [filespec] RECOVER d:
REM	Displays a comment in a batch file (described under BATCH)	REM [comment]
REN (RENAME)	Renames a file	REN [d1:] [pathname1] [filename1[ext1]]...
RESTORE	Restores backed-up files from a flexible disk to the hard disk	RESTORE [d1:] {C:D:} [filespec] [/S] [/P)
RMDIR (RD)	Removes a directory	RMDIR [d:] pathname
SET	Sets one string value to another	SET [string1=[string2]]
SHELL	Moves command processor out of root directory (described under CONFIG.SYS	SHELL = filename.ext [d:pathname] [/P] [/C string]
SHIFT	Increases number of replaceable parameters in batch file (described under BATCH)	SHIFT
SORT	Sorts data forward or backwards	SORT [/R] [/+n]
SYS	Transfers NCR-DOS system files from drive A to specified drive	SYS d:
TIME	Displays and sets time	TIME [hh[:mm[:ss[:cc]]]]
TREE	Displays all directories	TREE [d:] [/F] [>PRN]
TYPE	Displays contents of file on the screen	TYPE [d:] [pathname]filename[.ext]

Name	Purpose	Syntax
VER	Prints DOS version number	VER
VERIFY	Verifies data written to disk	VERIFY ON VERIFY OFF
VOL	Displays disk volume identification number	VOL [d:]

ASSIGN

TYPE

External

PURPOSE

Instructs DOS to use a disk drive other than one specified by a command or program.

SYNTAX

```
ASSIGN [d1=d2[...]]
```

where:

d1 = the letter of the disk drive that the program or DOS normally uses.
d2 = the letter of the disk drive replacing d1

COMMENTS

The command ASSIGN allows you to replace accesses of one valid drive with another. If you want to remove the reassignment just type ASSIGN with no options. When using this command remember:

- A colon is not necessary after the drive designation and should never be used.
- A disk drive can never be reassigned to itself.
- Multiple reassignments may be set at one time. (i.e. ASSIGN A=C B=C), assigns files from A and B to fixed disk C.
- This command should be used primarily when you run applications (developed by others) which have been programmed to operate on specific drives. Your NCR PC may not have the same set of drives, or you may want to run applications programmed for flexible disks on your fixed disk.

ADVANCED INFORMATION

When developing applications for possible sale or for use by others, you should minimize the need for the ASSIGN function by asking the user to specify drive assignments.

CAUTION

When a program asks the operating system to read or write a certain drive at a specific location, the ASSIGN command cannot change the access to another disk.

It is not advisable to use the ASSIGN command in normal processing because some commands and programs, such as PRINT, DISKCOPY and DISKCOMP, require the actual drive designation.

BACKUP

CD/Directory Name.
BACKUP␣C:*.*␣A:Z

TYPE
External

PURPOSE
Copies one or more files from the fixed disk (or the DOS partition of the fixed disk, if you are using multiple operating systems) in the source drive to flexible disks.

SYNTAX

?

```
BACKUP d1:[pathname] [filename[.ext]]
d2:[/S] [/M] [/A] [/D:mm-dd-yy]
```

where:

d1 = fixed disk to be backed up
d2 = destination of backup
pathname = directory path for backup
filename.ext = the file(s) to be backed up (the universal characters "*" and "?" are allowable in this field)
/S = copy all files in all of the subdirectories in addition to the files in the designated directory
/M = copy all files that have changed since the last Backup
/A = Add the files to the destination flexible disk. Otherwise, the destination disk is first purged of existing files and programs.
/D: = Back up all files created or modified since the given date mm/dd/yy.

COMMENTS
The BACKUP utility is essential if you have a fixed disk. This command lets you easily copy programs and data files. Some of the things to remember when executing BACKUP are:

- The source of a BACKUP can be the entire fixed disk, the DOS partition, a file — with or without universal characters, a directory path, or any combination of the above.
- The destination can be any flexible disk drive.
- The default source drive is the current drive in use. The same applies to the directory pathname; in other words, if you want to use a different path or drive than the one currently in use you must explicitly state the source.

- The destination flexible disks must have been formatted previously.

- The BACKUP utility will delete all files currently on the destination disk if you do not specify the /A parameter. When using the /A parameter, the destination disk must already be in the correct disk drive.

- The utility will prompt you to change the disk in the destination disk drive when the total size of the file(s) being copied is greater than the capacity of one or more flexible disks.

After completing the BACKUP utility, DOS sets a number. Response code "0" denotes successful completion, "1" means there are no eligible files to backup on the designated source drive, and any other number means the backup was unsuccessful. If the response code is greater than 1, you should check syntax, make sure the destination drive is correctly loaded and retry the utility.

These response codes can be used with the IF command (batch processing).

To retrieve a file that has been backed up, the RESTORE command must be used. To check the accuracy of the copying during creation of the new files, make sure you have VERIFY set to ON before beginning the BACKUP command entry. (See RESTORE and VERIFY later in this chapter.)

BATCH COMMANDS

A batch command is an operational word for use in a Batch file. A batch file is a series of DOS commands, comments, and acceptable statements that have been put in a disk file. The advantage of using a batch file is that the user can execute a whole series of seemingly complex commands and retain the use of the operating system for input, printing, etc.

To execute a batch file, just type in the filename if it's on the default drive, or the drive, path, and filename if its on a non-default drive. Since all batch files have the file extension .BAT, DOS will locate it and begin executing it automatically. Do not use a .BAT filename that is identical to a .COM or a .EXE filename since the system will not know whether to execute the batch file, the command, or the program.

You can create a batch file using EDLIN, the DOS COPY command, or any other text editor. There are several valid batch commands along with all of the other DOS commands. The BATCH sub commands are ECHO, FOR, GOTO, IF, PAUSE, REM, and SHIFT, and all are thoroughly explained later in this section.

THE AUTOEXEC.BAT FILE

The only batch filename that does not have to be typed for the batch file to execute is AUTOEXEC.BAT. This batch file, when placed in the root directory of your DOS disk, executes automatically whenever the system is booted. This is convenient if you wish to bypass the date and time prompts, or execute a series of commands every time DOS is loaded. Remember that the file must be called AUTOEXEC.BAT and must reside in the root directory of the boot diskette.

PARAMETERS

A batch file parameter is similar to a normal DOS command parameter; that is, it is the additional information associated with a command to further define an operation. For example, if you type

```
DIR B:MYFILE.TXT
```

you are requesting the directory command to use the filespec parameters B:MYFILE.TXT to further define the directory information. Batch file parameters work similarly, but in a different way. When typing the Batch filename you may append up to ten additional parameters on the same line. These parameters are picked up by the batch file through the use of the percent sign (%).

When you wish to use a parameter (a filename, a number, an extension, etc.) in a batch file, you use the "%1" symbol to identify that variable. ("%0" is reserved for the filename within the batch file.) A second parameter is represented by %2, a third by %3, and so on up to and including %9. Naturally, if you wish to use a percent sign as a percent sign, you must indicate this to DOS so it will not mistake it as a parameter symbol. The convention used by DOS is %% (i.e. 100%.TXT is listed in a Batch file as 100%%.TXT)

When substituting a parameter for a %n the user should remember that the batch program will substitute the parameter for %n throughout the file. Therefore, if within the same batch file you have %1.TXT and %1.LST, the batch program will substitute the same filename for both file extensions. A universal character is a valid para- meter and may be used as such.

If you create the batch file TEST.BAT that looks like this:

```
COPY A:%1 %2:%3
DEL A:%1
DIR %2: >PRN
TYPE %0.BAT
```

and type

```
A> TEST FILE1 B FILE2
```

the batch file you execute will actually look like this:

```
COPY A:FILE1 B:FILE2
DEL A:FILE1
DIR B: >PRN
TYPE TEST.BAT
```

(Remember that TEST = %0)

You have just copied a file (FILE1), renamed it (FILE2), put it on a different disk drive (B), deleted the original file, printed the new directory of the B drive, and listed your batch program.

ECHO

PURPOSE

Turns batch echo feature on and off.

SYNTAX

```
ECHO      ON message
          OFF message
          message
```

COMMENTS

Normally, commands in a batch file are displayed ("echoed") on the console when they are seen by the command processor. ECHO OFF turns off this feature. ECHO ON turns the echo back on ECHO message will display the rest of the line if ECHO is currently off.

If ON or OFF are not specified, the current setting is displayed.

FOR..IN..DO

PURPOSE

Command extension used in batch and interactive file processing.

SYNTAX

- For batch processing:

  FOR %% variable IN list DO command

- For interactive processing:

  FOR % variable IN list DO command

where:
variable = any non-capitalized single letter
command = any DOS command or batch sub command

COMMENTS

<variable> can be any character except 0,1,2,3,..,9 (to avoid confusion with the %0-%9 batch parameters.)

The %% variable letter is set sequentially to each member filename.ext and then command is evaluated. If a member of filename.ext is an expression involving universal characters (* and/or ?), then the variable is set to each matching pattern from disk. In this case, only one such item may be in the filespec and any item besides the first is ignored.

NOTE: The words FOR, IN, and DO must be in uppercase.

Consider these examples:

FOR %%f DO IF EXIST A:%%f DISKCOPY A: C:
FOR %%f IN (*.ASM) DO MASM %%f;
FOR %%f IN PAYROLL.DAT (ACCTPAY.DAT) DO COPY *.DAT *.PAY

The '%%' is needed so that after batch parameter (%0-%9) processing is done, there is one '%' left. If only '%f' were there, the batch parameter processor would see the '%', look at 'f', decide that '%f' was an error (bad parameter reference) and throw out the '%f', so that the command FOR would never see it. If the FOR is not in a batch file, then only one '%' should be used.

NOTE: The FOR..IN..DO command must execute in the current directory.

GOTO

PURPOSE

Command extension used in batch file processing.

SYNTAX

GOTO<label>

COMMENTS

GOTO causes commands to be taken from the batch file beginning with the line after the label definition. A label definition is recognized by the system by a colon followed by a label name. If no label has been defined, the current batch file terminates. Label can either precede or follow the GOTO.

For example:

```
:abc
REM looping...
GOTO abc
```

produces an infinite sequence of messages: REM looping....

ADVANCED INFORMATION

Starting a line in a batch file with 'Try Again' causes the line to be ignored by batch processing. The characters following GOTO define a label, but this procedure may also be used to put in comment lines.

IF

PURPOSE

Command extension used in batch file processing.

SYNTAX

IF condition command

COMMENTS

The parameter condition is one of the following:

ERRORLEVEL n

True if and only if the previous program executed by COMMAND had an exit code of "n" or higher. Only the RESTORE and BACKUP commands set a value that can be tested.

string1= =string2

True if and only if string1 and string2 are identical after parameter substitution. Strings may not have embedded separators.

EXIST filename

True if and only if filename exists.

NOT condition

True if and only if condition is false.

The IF statement allows conditional execution of commands. When the condition is true, then the command is executed; otherwise, the command is ignored.

Consider the following examples:

```
IF NOT EXISTTMP.BID ECHO Can't find file

IF NOT ERRORLEVEL 3 PRINT B:DATA.* /P
```

PAUSE

PURPOSE

Suspends execution of the batch file and displays a message.

SYNTAX

PAUSE [message]

COMMENTS

During the execution of a batch file,you may need to change disks or perform some other action. PAUSE suspends execution until you press any key (except <Ctrl-Break> or <Ctrl-C>).

When the command processor encounters PAUSE, it prints the message:

```
Strike any key when ready...
```

If you press any key your message is displayed. If you press <Ctrl-Break> the following prompt will be displayed:

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

If you type Y in response to this prompt, execution of the remainder of the batch command file is aborted and control is returned to the operating system command level. Therefore, PAUSE is used to break a batch file into pieces, allowing you to end the batch command file at an intermediate point.

The message is optional and is entered on the same line as PAUSE. You may want to prompt the user of the batch file with some meaningful message when the batch file pauses. For example, you may want to change disks in one of the drives. An optional prompt message may be given in such cases. The comment prompt is displayed before the "Strike any key" message.

REM

PURPOSE

Displays remarks that are on the same line as the REM command in a batch file during execution of that batch file.

SYNTAX

REM [comment]

COMMENTS

The only separators allowed in the comment are the space, tab, and comma. Consider the following example:

```
1: REM This file checks new disks

2: REM It is named NEWDISK.BAT

3: PAUSE Insert new disk in drive B:

4: FORMAT B:/S

5: DIR B:

6: CHKDSK B:
```

SHIFT

PURPOSE

Allows access to more than 10 replaceable parameters in batch file processing.

SYNTAX

SHIFT

COMMENTS

Usually, command files are limited to handling 10 parameters, %0 through %9. To allow access to more than ten parameters, use SHIFT to change the command line parameters. For example, if

%0 = "PROG1"
%1 = "PROG2"
%2 = "PROG3"
%3...%9 are empty

then a SHIFT results in the following:

%0 = "PROG2"
%1 = "PROG3"
%2...%9 are empty

If there are more than 10 parameters given on a command line, those that appear after the 10th (%9) are shifted one at a time into %9 by successive shifts.

CHDIR

TYPE
Internal

PURPOSE
Changes current (working) directory to a different path; displays current (working) directory.

SYNTAX

CHDIR [[d:]pathname]

where:
    d: = disk drive

COMMENTS
If your working directory is \BIN\USER\JOE and you want to change to another directory, such as \BIN\USER\JOE\FORMS, type:

CHDIR \BIN\USER\JOE\FORMS

and DOS puts you in the new directory.

A shorthand notation is also available with this command:

CHDIR...

This command always puts you in the parent directory of your working directory.

CHDIR used without a pathname displays your working directory. If your working directory is \BIN\USER\JOE on drive B, and you type CHDIR<CR>, DOS displays:

```
B:\BIN\USER\JOE
```

This command sometimes referred to as CD, is useful if you forget the name of your current working directory.

CHKDSK
(Check Disk)

TYPE

External

PURPOSE

Scans the directory of the specified disk drive, checks it for consistency, reports memory capacity, and fixes any disk errors.

SYNTAX

```
CHKDSK [d:] [filename.ext] [/F] [/V]
```

where:
d: = disk drive
/F = request to correct disk errors
/V = display messages while running

COMMENTS

CHKDSK should be run occasionally on each disk to check for errors in the directory. If any errors are found, CHKDSK displays error messages, and then displays a status report similar to the one below.

```
 362496    bytes total disk space
  22528    bytes in 2 hidden files
    512    bytes in 2 directories
 335872    bytes in 38 user files
   4096    bytes available on disk

 131072    bytes total memory
 106624    bytes free
```

CHKDSK does not correct the errors found in your directory unless you specify the /F (fix) option. Typing /V causes CHKDSK to display messages while it is running.

You can redirect the output from CHKDSK to a file. Simply type:

```
CHKDSK A:>filename
```

The errors are sent to the filename specified, so do not use the /F switch if you redirect CHKDSK output.

CHKDSK

ERRORS, MESSAGES, AND REMEDIES

First cluster number is invalid entry truncated
Truncated if /F specified

Allocation error, size adjusted
Adjusted if /F specified

Has invalid cluster, file truncated
Truncated if /F specified

Disk error reading FAT
Copy or BACKUP all files on disk and reformat

Disk error writing FAT
Copy or BACKUP all files on disk and reformat

<filename> contains
non-contiguous blocks
Use copy to better utilize space, disk is fragmented

All specified file(s) are contiguous
Good disk

Invalid drive specification
Re-enter command

"entry" Invalid parameter
Re-enter command

Invalid sub-directory entry
Re-enter command

Incorrect DOS version
You cannot run CHKDSK on versions of DOS that are not 2.0 or higher.

Insufficient memory
Processing cannot continue
There is not enough memory in your machine to process CHKDSK for this disk. You must obtain more memory to run CHKDSK.

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

You must specify the /F switch if you want the errors corrected by CHKDSK.

Invalid current directory
Processing cannot continue

Restart the system and rerun CHKDSK.

Cannot CHDIR to root
Processing cannot continue

The disk you are checking is bad. Try restarting DOS and RECOVER the disk.

<filename> is cross linked on cluster

Make a copy of the file you want to keep, and then delete both files that are cross linked.

X lost clusters found in y chains
Convert lost chains to file (Y/N)?

If you respond Y to this prompt, CHKDSK creates a directory entry and a file for you to resolve this problem (files created by CHKDSK are named FILEnnnn.

CHKDSK then displays:

X bytes disk space freed

Probable non-DOS disk
Continue (Y/N)?

The disk you are using is a non-DOS disk. You must indicate whether or not you want CHKDSK to continue processing.

Insufficient room in root directory
Erase files in root and repeat CHKDSK

CHKDSK cannot process until you delete files in the root directory.

Unrecoverable error in directory
Convert directory to file (Y/N)?

If you respond Y to this prompt, CHKDSK converts the bad directory into a file. You can then fix the directory yourself using the DUBUG utility, or you can delete it.

CLS
(Clear Screen)

TYPE
Internal

PURPOSE
Clears the terminal screen.

SYNTAX
CLS

COMMENTS
CLS (clear screen) removes all displayed characters from the screen. The cursor is returned to the upper left-hand corner of the screen. The CLS command does not affect disk files or memory in any way.

If you have used the ANSI control codes to set the foreground and background colors, they will remain the same.

If the foreground/background color control codes have not been set, the display screen will revert to light foreground and dark background.

ADVANCED NOTES
The CLS command causes DOS to send the (ASCII) escape sequence ESC 2J (that clears the screen) to the console.

COMMAND

Type

Internal

PURPOSE

Utilizes a second version of the command processor

SYNTAX

COMMAND d:pathname [/P] [/C string]

where:

/P = retains the second command processor permanently (until the system is reset.

/C = places the following string in the command processor

string = command to be added to the command processor.

COMMENTS

COMMAND allows the user to copy the COMMAND.COM file to the disk and the directory of the user's choice. The COMMAND.COM file within DOS handles your housekeeping activities, your internal DOS commands, your system prompt, and your error handling routines. Because of this, changing your command processor should not be done lightly, nor by inexperienced users. For more information on the COMMAND command consult the *NCR-DOS PROGRAMMER'S MANUAL* .

COMP
(COMPARE)

TYPE

External

PURPOSE

Compares two files and reports their differences.

SYNTAX

```
COMP  [d1:] [pathname1] [filename] [.ext1] [d2:]
[pathname2] [filename2] [.ext2]
```

where:

d1:pathname1\filename1.ext1 is the source file to be compared
d2:pathname2\filename2.ext2 is the target file the first file is compared to

COMMENTS:

The compare utility is useful to check a recently copied file, or to compare a text file to a .BAK file. The compare utility does not work, however, on files that have been BACKed UP from the fixed disk without first using a RESTORE. Some things to remember about the COMP command:

- Files of different lengths cannot be compared, and, if attempted, the utility will issue an error message.
- Upon exceeding ten mismatched characters when comparing two files, the COMP utility will abort.
- System files cannot be successfully compared.
- Directory paths cannot be successfully compared
- If you enter COMP without the associated filespecs, the utility will prompt you for the filenames to be compared.

If your files are identical, the message:

```
Files compare ok
Compare more files (Y/N)?
```

will appear. However, if there is an inequality with the files the message

```
Compare error at offset nnnn
File1 = xx
File2 = yy
```

will be displayed. The offset number refers to the number of bytes into the file where the error occurred (using base 16). The next two lines show the hexadecimal code of the offending characters. For example if, at the beginning of a file (file1), the word "the" was transposed as "hte" in file2, the first error message would read:

```
Compare error at offset 01
File1 = 74
File2 = 68
```

and the second would look like:

```
Compare error at offset 02
File1 = 68
File2 = 74
```

CONFIG. SYS COMMANDS

The term configure means to set up for operation in a particular way. Similarly, to configure your operating system is to set up your computer for operation in a particular way.

CONFIGURATION (CONFIG. SYS) FILE

NCR-DOS has a major feature that assists in the configuration process: CONFIG.SYS. This special text file lies in the root directory of your boot disk and contains commands that can improve or alter the performance and flexibility of your computer. Some of these improved functions include adding and changing your function keys and expanding or changing memory. The CONFIG.SYS functions are loaded at the same time as DOS and cannot be changed without rebooting.

The following commands can be located in the configuration file: BREAK, BUFFERS, DEVICE, FILES, and SHELL.

MAKING A CONFIG.SYS FILE

There are two major commands you may want to try initially with NCR-DOS: BUFFERS and DEVICE. You can use the text editor EDLIN, or the COPY CON: command to create CONFIG.SYS.

```
BUFFERS = 10
DEVICE = ANSI.SYS
```

Place the above file in the root directory of your DOS boot diskette, calling it CONFIG.SYS . If you are using a fixed disk, put the file in the root directory of the hard disk. If you have included the second line, DEVICE=ANSI.SYS, copy the ANSI.SYS file to the disk you will use for booting.

After you have saved the file, reboot the system using your new disk. DOS will read the CONFIG.SYS file and alter itself to accommodate any commands in the file.

If you change the CONFIG.SYS file, the changes will implemented the next time you boot the system from the disk. Remember that CONFIG.SYS is used only when DOS starts up.

BREAK

PURPOSE

Allows user to specify a more-frequent test for a Ctrl-Break prior to executing a long program.

SYNTAX

BREAK $\left\{\begin{matrix}\text{ON}\\\text{OFF}\end{matrix}\right\}$

COMMENTS

If you are running an application program that uses Ctrl-Break function keys, you will want to turn off the function so that when you press <Ctrl-Break> you affect your program and not the operating system. Since DOS is set with BREAK OFF every time the operating system is loaded, there is no need to set the BREAK OFF unless it has been previously set to BREAK ON.

BUFFERS

PURPOSE

To increase or decrease the number of buffer storage areas.

SYNTAX

BUFFERS=n

where:
n = any integer between 1 and 99 inclusive

COMMENTS

A buffer is an area of the system memory used to store a block (usually several records) of data read from or to be written to a disk. Without using the BUFFER command, DOS defaults to two buffers with the capacity of 528 bytes each. Normally this is sufficient for most programs. However, if you have applications requiring a lot of reading and writing to and from the disk, a greater number of disk buffers will speed up execution. This allows DOS to access the information directly instead of searching and reading the disk for the data.

You should remember that when you are defining the number of buffers there can be too few or too many buffers for any given program. If the number of buffers is too small, the operating system spends too much time accessing the disk since less data is stored in RAM memory. On the other hand, too many buffers can slow down DOS due to the time spent handling buffers instead of executing a program. For larger programs, accounting applications, and word processing the optimal number of buffers will fall between 12 and 24.

A minimum of 3 buffers should be assigned to memory if you are using a fixed disk drive.

DEVICE

PURPOSE

This command allows the user to assign a new device and it's driver to the system, allowing DOS to recognize the device.

SYNTAX

DEVICE = [d:] [pathname]filename[.ext]

COMMENTS

The device driver is the software used to control a peripheral. Your DOS comes equipped with the drivers needed for your existing hardware. When you start the system, the existing driver software is loaded so you can control the devices. However, if you add devices to your system, you will need to add the appropriate software to your CONFIG.SYS file so that DOS can talk with the new hardware device. This is done through the DEVICE command.

One of the more common device driver programs is the ANSI.SYS file provided with DOS. This allows the user to expand the usage of a color terminal by controlling the color through programs or reprogramming the keyboard. To place the ANSI.SYS file within CONFIG.SYS, you may use EDLIN, another text editor, or the COPY CON: command. Type the command

```
DEVICE = ANSI.SYS
```

when the DOS disk is in the default disk driver. If you are using a fixed disk, you should add the directory path as well

```
DEVICE = A:pathname\SYS\ANSI.SYS
```

whereas flexible diskette users will place the device drivers in the root directory.

RAM-DISK CONSIDERATIONS

Your DOS flexible disk contains a file RAMDISK.SYS which you can use to maintain a "RAM-Disk". A RAM-Disk is an area of computer memory set aside for data storage. You can store files in and read files from a RAM-Disk as if it were a disk drive. The difference is that access to RAM-Disk is much faster than to a "real" disk drive. However, before switching off your computer, you must save RAM-Disk files you wish to keep on a real disk.

To set up one or more RAM-Disks, first make sure that the file **RAMDISK.SYS is present on the disk which you use to load** (boot) DOS. If necessary, copy these files from your original system disk. If your original DOS flexible disk already has a CONFIG.SYS file, copy this file onto your new system disk, too.

Using EDLIN or another editor, add information to the file CONFIG.SYS in the following format (if the file was not present on your original DOS flexible disk, start a new file with this filename):

DEVICE=RAMDISK.SYS
[/<sides>] [/<sectors>] [/Snnn] [/Dnnn] [/Lnnn]

A space must be present before the initial slash. The brackets indicate the optional nature of the information and are not to be typed when writing the line for CONFIG.SYS.

<sides> can be digit 1 or 2 according to whether you wish your RAM-Disk to be regarded as single or double sided; if you do not specify a value, NCR- DOS assumes 2 (double sided). <sectors> can be a digit 8 or 9 according to whether your RAM-Disk is to be regarded as having 8 or 9 sectors per track; if you do not specify a value, 9 is assumed. Your choice of <sides> and <sectors> influences the amount of space and the maximum number of directory entries of your RAM-Disk:

DEVICE=RAMDISK.SYS	/2/9	allows a maximum of 360 KB and 112 directory entries
DEVICE=RAMDISK.SYS	/2/8	allows a maximum of 320 KB and 112 directory entries
DEVICE=RAMDISK.SYS	/1/9	allows a maximum of 180 KB and 64 directory entries
DEVICE=RAMDISK.SYS	/1/8	allows a maximum of 160 KB and 64 directory entries

The creation of a RAM-Disk with one of the previously listed dimensions is realized only if total computer memory is at least 64 KB greater than the specified RAM-Disk size combined with the memory requirements of the operating system. (The operating system needs about 32KB.) If this is not the case, DOS sets up the largest possible RAM-Disk within memory constraints.

Instead of using one of the standard four RAM-Disk definitions, you may specify RAM-Disk size and the maximum number of directory entries using the /S and /D options, respectively.

Using /S, nnn represents the size of the RAM-Disk in kilobytes. The minimum RAM-Disk size created by DOS is 2 KB. The considerations regarding available memory mentioned earlier also apply when using the /S switch.

/D defines the number of directory entries which are to be allowed on the RAM-Disk. DOS accommodates up to 752 such entries.

/L defines the size of memory in KB that RAM-Disk must not use.

The following entry in the CONFIG.SYS file has the special effect of setting up a RAM-Disk with the maximum possible size; that is, total memory space minus operating system requirements minus 64 KB. The maximum number of directory entries is 112, if the RAM-Disk size is greater than 180 KB; otherwise, the maximum is 64.

```
DEVICE=RAMDISK.SYS /
            or
DEVICE=RAMDISK.SYS /64
```

If your computer has enough memory and you wish to use more than one RAM-Disk, all you have to do is add similar lines to the CONFIG.SYS file.

If a RAM-Disk entry in CONFIG.SYS contains more than one definition for size or directory entries (for example, you specify both <sides> and the /S switch), DOS recognizes only the <sides> and <sectors> definition.

When you have finished editing CONFIG.SYS for the new RAM-Disk system, save the new version of CONFIG.SYS on the appropriate disk. When you load the new system, the RAM-Disk(s) are assigned drive letters following those of the real disk drives. For example, if your system has two flexible disk drives, the first (or only) RAM-Disk is referred to as drive C. If you already have a C drive, RAM Disk will be labeled as the D disk.

NOTE: A demonstration of RAM-Disk is included on your NCR-DOS diskette. Refer to *Appendix D* for information on how to run the program. For copying diskettes to RAM-disk use the **DISKCOPY** utility.

FILES

PURPOSE

The FILES command permits the user to expand the number of files that can be opened concurrently.

SYNTAX

FILES = n

where:
n is any number between 1 and 99 inclusive

COMMENTS

The NCR DOS default number of files that can be opened at any given time is 8. If you specify a number below 5, DOS will allow you to open 5 anyway. If you write assembler language programs, see the section on handles in the *NCR-DOS PROGRAMMER'S MANUAL* .

SHELL

PURPOSE

To permit the users to create their own command processor

SYNTAX

SHELL = filename.ext [d:pathname] [/P] [/C string]

where:

filename.ext is the name of the new command processor
/P = the new command processor is to be permanent (as opposed to this current session only)
/C = places the following command string in the command processor
string = Command to be executed immediately

COMMENTS

The SHELL command converts the standard command processor, COMMAND.COM, to one of the users choosing. The user should be totally conversant with assembler language before attempting to use this command.

COPY

TYPE

Internal

PURPOSE

Copies one or more files to another disk. If you prefer, you can give the copies different names. This command can also copy files on the same disk.

SYNTAX

COPY [CON:] [d1:] [pathname1] [filespec1] [/A] [/B]
[d2:] [pathname2] [filespec2] [/V] [/A] [/B]

where:

CON: = console, to allow user to copy the screen contents to a file. Used exactly like a drive designation (i.e. A: or B:). Exit copy made with <Ctrl-Z>.
filespec1 = source file
filespec2 = destination file
pathname1 = source source directory path
pathname2 = destination
/V = verification option
/A (source) = treats file as an ASCII (text) file, and copies up to, but not including end-of-file marker (Ctrl-Z). The remainder is not copied. May also be used with pathname.
/A (destination) = adds an end of file character. May also be used with pathname
/B (source) = treats file as if it were binary, including end-of-file character. May also be used with pathname; copies entire file
/B (destination) does not add end-of-file character to binary file. May also be used with pathname.

COMMENTS

Before using this command, be sure the destination disk contains sufficient space for the copy.

If filespec2 is not given, the copy is to the default drive and has the same name as the original file (filespec1). If filespec1 is on the default drive and filespec2 is not specified, the COPY is aborted (copying files to themselves is not allowed) and DOS returns the error message:

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

Filespec2 may have three forms:

1. If only a drive designation (d2:) is entered, the original file is copied with the original filename, to the designated drive.

2. If a filename only is entered, the original file is copied to a file on the default drive with the filename specified.

3. If a full filespec is entered, the original file is copied to a file on the designated drive with the filename specified.

NOTE: If the destination file already exists it will be overwritten, with the new copy of the file replacing the original.

The /V switch causes DOS to verify that the sectors written on the destination disk are recorded properly. This option causes the COPY command to run more slowly because DOS must check each entry recorded on the disk.

The COPY command also allows file concatenation (joining) while copying. Concatenation is accomplished by simply listing any number of files as options to COPY, separated by "+". For example,

```
COPY FILE1.XYZ+FILE2.COM+B:FILE3.TXT
BIGFILE.CRP<CR>
```

This command concatenates files named FILE1.XYZ, FILE2.COM, and FILE3.TXT and places them in the file on the default drive called BIGFILE.CRP.

To combine several files utilizing universal characters into one file you could type:

```
COPY *.TXT COMBIN.XMT
```

This command would take all files with a filename extension of .TXT and combine them into a file named COMBIN.XMT.

In the following example, for each file found matching *.WKD, that file is combined with the corresponding .REF file. The result is a file with the same filename but with the extension .WK. Thus, MON.WKD will be combined with MON.REF to form MON.WK, then TUR.WKD and TUR.REF will form TUR.WK, and so on.

```
COPY *.WKD+*.REF *.WK
```

The following COPY command combines all files matching *.FST, then all files matching *.SEC, into one file named COMBIN.TOT

```
COPY *.FST+*.SEC COMBIN.TOT
```

Do not enter a concatenation COPY command where one of the source filenames has the same extension as the destination. For example, the following command is an error if ALL.LST already exists:

```
COPY *.LST ALL.LST
```

The error would not be detected, however, until ALL.LST is appended. (At this point it could have already been destroyed.)

COPY compares the filenames of the input file with the filename of the destination. If they are the same, that one input file is skipped, and the error message “Content of destination lost before copy” is printed. Further concatenation proceeds normally. This allows “summing” files, as in this example:

```
COPY ALL.LST+*.LST
```

This command appends all *.LST files, except ALL.LST itself, to ALL.LST. This command does not produce an error message and is the correct way to append files using the COPY command.

CTTY
(CHANGE CONSOLE)

TYPE

Internal

PURPOSE

Change the device from which commands are issued (TTY represents the console).

SYNTAX

```
CTTY DEV
```

COMMENTS

DEV stands for "device", which is the device from which you are giving commands to NCR-DOS. This command is useful if you want to change the device on which you are working. The command:

```
CTTY AUX
```

moves all command I/O (input/output) from the current device (the console) to the AUX port, such as a teleprinter. The command:

```
CTTY CON
```

moves I/O back to the original device (here, the console).

CAUTION

The user should note that it is a flagrant error to move I/O from the terminal to a printer without a keyboard. If this happens, NCR-DOS will not be able to respond (since there is no input from an output- only device), and the system will be inoperative until it is reset.

DATE

TYPE

Internal

PURPOSE

Enter or change the date known to the system. This date is recorded in the directory for any files you create or alter.

You can change the date from your terminal or from a batch file. (DOS does not display a prompt for the date if you use an AUTOEXEC.BAT file, unless you include a DATE command in that file.)

SYNTAX

DATE [mm-dd-yy]

COMMENTS

If you type DATE, DATE responds with the message:

```
Current date is TUE 1-1-80
Enter new date: _
```

Press <CR> if you do not want to change the date shown.

You can also type a particular date after the DATE command on the same line, as in:

```
DATE 7-4-84
```

In this case, you do not have to answer the Enter new date: prompt.

The new date must be entered using numerals only; letters are not permitted. The allowed options are:

mm=1-12
dd=1-31
yy=80-99
or 1980-2099
(if not included, the 19 is assumed)

The date, month, and year entries may be separated by hyphens (-), periods (.), or slashes(/).

If the options or separators are not valid, DATE displays the message:

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

DATE then waits for you to enter a valid date.

DEL
(DELETE)

TYPE
Internal

PURPOSE
Deletes all files with the designated filespec.

SYNTAX

DEL [d:] [pathname] [filespec]

COMMENTS
If the filespec is *.*, the prompt "Are you sure?" appears. If a "Y" or "y" is typed as a response, then all files are deleted as requested. You can also type ERASE for the DELETE command. If you enter only pathname, you will delete all files within the directory. To delete the directory pathname, see the RMDIR command.

NOTE: If only a pathname has been specified, the utility assumes all the files in that directory are to be deleted (*.*). Hidden files cannot be deleted without reformatting the disk.

DIR

TYPE

Internal

SYNTAX

DIR [d:] [pathname] [filespec] [/P] [/W]

where:
/P = Pause when screen is full before continuing
/W = Wide (80 column) display of file names only. Does not include FILE SIZE and other information.

PURPOSE

Lists the files in a directory.

COMMENTS

If you just type DIR, all directory entries on the default drive are listed. If only the drive specification is given (DIR d:), all entries on the disk in the specified drive are listed. If only a filename is entered with no extension (DIR filename), then all files with the designated filename on the disk in the default drive are listed. If you designate a file specification (for example, DIR d:filename.ext), all files with the filespec specified on the disk and in the drive specified are listed. In all cases, files are listed with their size in bytes and with the time and date of their last modification.

The universal characters ? and * (question mark and asterisk) may be used in the filename option. For example, the following table shows equivalent command designations.

COMMAND	EQUIVALENT
DIR	DIR *.*
DIR FILENAME	DIR FILENAME.*
DIR .EXT	DIR *.EXT

Three optional parameters options may be specified with DIR. The /P option selects Page Mode. With /P, the display of the directory pauses after the screen is filled. To resume display of output, press any key.

The /W option selects Wide Display. With /W, only filenames are displayed, without other file information. Files are displayed five per line. When using the wide option without a filename parameter, you may have difficulty in distinguishing between directories and files' subdirectories.

The third option is the >PRN parameter. This allows you to redirect the directory listing to your printer. If you would like to have your directory in a file, use the option ">filespec".

DISKCOMP
(COMPARE DISKETTES)

TYPE

External

PURPOSE

This allows the user to compare two identical flexible disks.

SYNTAX

```
DISKCOMP   [d1]: [d2:] [/1] [/8]
```

where:

```
    d1,d2 = two drives containing diskettes to be compared
       /1 = 1st. side of diskettes compared only (on double-sided
            diskette)
       /8 = first 8 sectors/track compared only (diskette containing
            9 sectors/track)
```

COMMENTS

Usually the DISKCOMP utility is executed after having created the second diskette with the DISKCOPY utility. DISKCOMP runs a comparison that is sector-to-sector and track-to-track. A message is issued if the tracks are not equal, which indicates the track and side of the unequal track. Since the COPY utility copies files instead of sectors or tracks, it is unlikely that a copy of a diskette will be identical to the original. Similarly, due to the differences in size, a fixed disk cannot be compared to a flexible disk.

If you have a single-drive unit, the command DISKCOMP without the disk drive designations will be acceptable. DOS will prompt you to insert/remove diskettes as necessary.

DISKCOPY

TYPE

External

PURPOSE

Copies the contents of the disk in the source drive to the disk in the destination drive.

SYNTAX

```
DISKCOPY   [d1]: [d2:] [/1]
```

where:
d1 = source disk drive
d2 = distination disk drive
/1 = Copies only the first side of a diskette

COMMENTS

The first option you specify is the source drive; the second option is the destination drive. The destination disk will be formatted if the formats do not match. You can specify the same drive or you may specify different drives. If the designated drives are the same, or if no drive is designated, a single-drive copy operation is performed. You are prompted to insert the correct disks at the appropriate times. DISKCOPY waits for you to press any key before continuing.
After copying, DISKCOPY prompts:

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

If you press Y, the next copy is performed on the same drives that you originally specified, after you have been prompted to insert the proper disks. To end the copying, press N.

Before using DISKCOPY, consider the following command characteristics:

- If you omit both options, a single-drive copy operation is performed on the default drive.
- If you omit the second option, the default drive is used as the destination drive.

- Disks that have had a lot of file creation and deletion activity can become fragmented, because disk space is no longer allocated sequentially. The first free sector found is the next sector allocated, regardless of its location on the disk. If this is the case, you may want to use the COPY command, in stead of DISKCOPY, to copy your disk file for file and eliminate fragmentation.

  COPY A:*.* B:

  copies all files from the disk in drive A to the disk in drive B.

- DISKCOPY automatically determines the number of sides to copy, based on the source drive and disk, unless you use /1 option.

- If disk errors are encountered during a DISKCOPY, DOS displays something similar to:

  error writing to B:

Refer to the appendix on Messages for information on this error message.

ERASE

TYPE

Internal

PURPOSE

Erases all files with the designated filespec

SYNTAX

ERASE [d]: [pathname] [filename] [.ext]

where:

d: = the disk drive where the file or directory path resides

COMMENTS

If the filespec is *.*, the prompt "Are you sure?" appears. If a Y or y is typed as a response, then all files are erased as requested. You can also type DEL or DELETE for the ERASE command.

NOTE: If no filename has been specified in association with a directory path, this command erases all the files within that directory.

ESC

TYPE

External

PURPOSE

Enables input of ESC sequences through the keyboard

SYNTAX

ESC <delimiter> <ESC sequence>

The keyword ESC must always be seperated from the sequence by one delimiter, e.g., a space. An escape sequence is entered through the keyboard and may consist of any character with the exception of > and ¦.

If the ESC sequence is to be processed by the CRT driver it must be entered as an ANSI escape sequence, that is, the first character must be a [.

The syntax of an ANSI escape sequence is as follows:

[{[=]¦[?]}{<#>¦“<string>“}[;...]<end-of-sequence sign>

- The left-bracket character is part of the ANSI escape sequence and must be entered as the first character.
- Spaces are allowable only in strings.
- A string may consist of any characters except for quotation marks. These are entered as a decimal value (34).
- <#> represents a decimal value the meaning of which varies. It may, for example, stand for a certain function or function key, or for the decimal value of a character which usually cannot be entered as part of a string (e.g.,the <CR> key).
- The end-of-sequence sign is a special character. The ESC sequence for a function key definition, for example, is terminated by lowercase p.

COMMENTS

The ANSI escape sequences, supported by your NCR PC, are listed in Appendix B.

The ESC sequence is usually sent to the CRT controller. It can also be sent to another control program (e.g., printer interface LPTx) or be stored in a file (see NCR-DOS I/O Redirection, Chapter 3). The latter is of advantage in the case of long ESC sequences. When stored in a file, they can be sent to any control program with a TYPE command.

NOTE: No error message is displayed if an incorrect ESC command is entered. In the case of an incorrect ANSI escape sequence, the part of the sequence that cannot be processed by the CRT controller is displayed on the screen.

Example 1: Changing graphic attributes

```
ESC  [7m
```

The screen is switched to inverse video.
(ESC [0m restores original settings.)

Example 2: Sending a control code sequence to the printer

```
ESC  90;44 G>LPT1
```

In the case of the NCR 6442 printer, this ESC sequence (hexadecimal 1B 39 30 3B 34 34 20 47) sets the line spacing to 8 LPI and the character width to 16.5 CPI.

Example 3: Storing an ESC sequence in a file

```
ESC  [0;103;"DIR A:/P";13p > SETF10
ESC  [0;113;"DIR B:/P";13 p >> SETF10
TYPE  SETF10
```

The ESC sequences which assign the DIR command to the function keys <Ctrl-F10> and <Alt-F10> are stored in the file SETF10; they are activated with the TYPE command. (For detailed information and further examples of how to use ESC for key definitions see Chapter 5).

EXE2BIN

TYPE

External

PURPOSE

Changes an .EXE file into a .COM or .BIN file

SYNTAX

EXE2BIN [d1]: [pathname1] [filename1] [.ext1]
[d2]: [\pathname2] [\filename2] [.ext2]

where:

d1 = source disk drive containing the .EXE file
pathname1 = source directory containing the .EXE file
filename1 = file name to be changed
.ext1 = .EXE (executable file)
d2 = destination disk drive for newly created .BIN or .COM file
pathname2 = destination directory for newly created .BIN or .COM file
filename2 = filename to be created (if different than filename1)
.ext2 = .BIN (binary image) or .COM (command module)

COMMENTS

EXE2BIN is a programming utility that converts executable files to a binary form, allowing faster loading and execution by compressing the coded file. If the user does not specify a disk drive or pathname, the default drive/pathname is assumed. The user should ensure that the .EXE file is properly formatted within the guidelines of NCR-DOS.

NOTE: The resident, or actual, code should not exceed 64K.

EXIT

TYPE

Internal

PURPOSE

Exits the program COMMAND.COM (the command processor) and returns to a previous level, if one exists.

SYNTAX

```
EXIT
```

COMMENTS

This command can be used when you are running an application program and want to start the DOS command processor, then return to your program. For example, to look at a directory on drive B while running an application program, you must start the command processor by typing COMMAND in response to the default drive prompt:

```
    A>COMMAND
```

You can now type the DIR command and DOS displays the directory for the default drive. When you type EXIT, you return to the previous level (your application program).

FDISK

TYPE

External

PURPOSE

Sets up your fixed disk to handle DOS and optionally to include other operating systems.

SYNTAX

FDISK

COMMENTS

This command will display and alter your fixed disk partioning or add a new fixed disk. This command is completely menu driven.

When you type FDISK and press <CR>, the following screen will appear:

```
Fixed Disk Setup Program

Current Fixed Disk Drive

FDISK Options

Choose one of the following:

      1. Create MS-DOS Partition
      2. Change Active Partition
      3. Delete MS-DOS Partition
      4. Display Partition Information
      5. Select next fixed Disk Drive

Enter choice: ..... [    ]

Press ESC to return to DOS ..... [    ]
```

(NOTE: The current fixed disk drive and option 5 are only shown if you have more than 1 fixed disk.)

INITIAL FDISK USE

If you have never partitioned your fixed disk, type 1 and press <CR> to create your DOS partition. The following screen appears:

```
Fixed Disk Setup Program

Create MS-DOS Partition

No partition defined

Total disk space is xxxx cylinders.

Do you wish to use the entire
fixed disk for MS-DOS (Y/N)? ... [Y]

Press ESC to return to FDISK Options. [     ]
```

If you intend to use DOS as your only operating system, type "Y" and press <CR>. Make sure your DOS diskette is loaded in your default disk drive. Your entire fixed disk will be dedicated to DOS and will require formatting after you reboot. (Read the FORMAT command in this chapter.) After formatting, and creating a volume label, you can copy your DOS commands by typing COPY *.* C: while your DOS diskette still resides in the A drive.

On the other hand, if you wish to include more than one operating system, you should enter an "N" and press <CR>. The following screen appears:

```
Fixed Disk Setup Program

Create MS-DOS Partition

No partition defined


Total disk space is xxxx cylinders.

Maximum available space is xxxx
cylinders at cylinder xxxx.

Enter partition size .....: [xxxx]

Press ESC to return to FDISK Options. [     ]
```

This screen shows the total number of cylinders available on your fixed disk. The partition size (in cylinders) should be entered now. The minimum practical size is 10. (Roughly the size of a flexible diskette.) If you want, you may leave this blank and DOS will default to the largest amount of cylinders allowable.

The next prompt is:

Enter starting cylinder number ....: [xxxx]

If this is the first time you have used this command, the default will be zero. If you have previously used FDISK, DOS will find the first area on your disk large enough to accommodate your requested partition size.

When you press <CR>, the screen will change to show your new, fixed disk partition statistics.

If you wish to make this newly created partition your "active" partition, go to option 2 at the menu.

Subsequent FDISK Use

After you use FDISK to establish disk partitions, you may use the other options of the command.

Changing the Active Partition (Option2) — Select this option when you want to start a different operating system in another partition. You will see a screen similar to the following:

Fixed Disk Setup Program

Change Active Partition

```
Partition    Status    Type      Start    End    Size
    1          N      non-DOS     000     049      50
    2          N      non-DOS     050     149     100
    3          A        DOS       150     304     155
```

Total disk space is xxxx cylinders.

Enter the number of the partition you
want to make active .....: [ ]

Press ESC to return to FDISK Options. [ ]

Enter the number of the partition whose operating system you want to get control when the system is started from the fixed disk.

If you want to start the operating system in the partition you just made active, perform the following steps:

1. Open the diskette drive A door.
2. Press and hold Ctrl and Alt, and then press Del.

The operating system in the active partition should then start.

Deleting the MS-DOS Partition (Option 3) – Select this option when you want to delete a DOS partition.

NOTE: This option could result in loss of the data in the DOS partition so make sure you have backed up all of your files before you proceed.

You will need to insert a DOS diskette and restart the system from diskette drive A if you want to continue processing under DOS.

If you want to start a system in another fixed disk partition, you should change the active partition to that partition number before you delete the DOS partition.

You will see a screen similar to the following:

```
    Fixed Disk Setup Program

    Delete MS-DOS Partition


    Partition    Status    Type       Start    End    Size
        1          N      non-DOS      000     049     50
        2          N      non-DOS      050     099     50
        3          N        DOS        100     249    150
        4          A      non-DOS      250     304     55


Total disk space is xxxx cylinders.


Warning! All MS-DOS partition data
will be DESTROYED. Do you wish to
continue .....? [N]

Press ESC to return to FDISK Options. [     ]
```

If you have backed up all of your files and are ready to continue, type Y and press <CR>. If you decide to cancel the operation, press either the <CR> key or the Esc key to return to the FDISK options menu.

Displaying Partition Information (Option 4) – You can use this option to display fixed disk status information. Your screen appears similar to the following:

```
     Fixed Disk Setup Program

     Display Partition Information

     Current Fixed Disk Drive: 1

     Partition   Status   Type     Start   End   Size
         1         A       DOS      000    199   200
         2         N     non-DOS    200    304   105

 Total disk space is xxxx cylinders

 The current active partition is


 Press Esc to return to FDISK Options [    ]
```

The line with the current fixed disk drive appears only if you have more than one fixed disk drive.

One line is shown for each assigned partition.

The Partition column shows the relative number of the partition (in the order it appears on the fixed disk).

The Status column shows which partition's system gets control when the system unit is started from the fixed disk. That partition's status is shown as A (for active); the others are shown as N (non-active).

The Type column shows which partition, if any, is the DOS partition.

The Start and End columns show the starting and ending cylinder numbers for a partition, and the Size column shows its size in cylinders.

The next line shows you the total amount of space on the fixed disk.

Press the Esc key when you are ready to return to the FDISK options menu.

Selecting Next Fixed Disk Drive (Option 5) — Select this option when you want to use the DOS Fixed Disk Setup Program with the next fixed disk drive.

After you have entered the option, you will see the current fixed disk drive number change on the FDISK options menu.

This option is available only if your system has more than one fixed disk drive.

FIND

TYPE

External

PURPOSE

This command searches for a specific string of text in a file or files and displays them.

SYNTAX

```
FIND [/V] [/C] [/N] “string” [d:] [pathname] [filename]
[.ext]
```

where:

/V = Displays all lines that do not contain string

/C = Counts the number of occurrences of a string in the file without displaying them

/N = Displays the line number in front of the string

“string” = a group of characters within double quotes that the user wishes to find

pathname and filename = The directory or file in which the string is located

NOTE: The universal characters “*” and “?” are unacceptable to the FIND command

COMMENTS

FIND is a filter that takes as options a string and a series of filenames. It displays all lines that contain a specified string from the files specified in the command line.

If no files are specified, FIND takes the input on the screen and displays all lines that contain the specified string.

Three options can be used with FIND:

/V

This option causes FIND to display all lines not containing the specified string.

/C

This option causes FIND to display only the count of lines that contain the string in each of the files.

/N

This option causes each line to be preceded by its relative line number in the file.

The string should be enclosed in quotes. For example:

FIND "Fool's Paradise" BOOK1.TXT BOOK2.TXT

NCR DOS displays all lines from BOOK1.TXT and BOOK2.TXT (in that order) that contain the string "Fool's Paradise." The command

DIR B: ¦ FIND /V "DAT"

causes NCR-DOS to display all names of the files on the disk in drive B that do not contain the string "DAT". Type double quotes around a string that already has quotes in it.

When an error is detected, FIND responds with one of the following error messages:

Incorrect DOS version

FIND only runs on DOS versions 2. and higher.

FIND: Invalid number of parameters

You did not specify a string when issuing the FIND command.

FIND: Syntax error

You typed an illegal string when issuing the FIND command.

FIND: File not found <filename>

The filename you have specified does not exist or FIND cannot find it.

FIND: Read error in <filename>

An error occurred when FIND tried to read the file specified in the command.

FIND: Invalid parameter <option-name>

You specified an option that does not exist.

FORMAT

TYPE
External

PURPOSE
Formats the disk in the specified drive to accept NCR-DOS files.

SYNTAX
```
FORMAT [d:] [/8] [/1] [/V] [/B] [/S]          (flexible diskettes)
FORMAT [d:] [/V] [/S]                         (fixed disks)
```

COMMENTS
This command formats flexible and fixed disks and initializes the directory and file allocation tables. If no drive is specified, the disk in the default drive is formatted.

The /V and /S switches can be specified when formatting flexible or fixed disks.

/V

Causes FORMAT to pause in the formatting process and to display a message asking for a volume label (useful in disk identification).

/S

Causes FORMAT to copy the operating system files from the disk in the default drive to the newly formatted disk. The files are copied in the following order: IO.SYS, MSDOS.SYS, COMMAND.COM. (Other files can then be selectively copied with the COPY command.)

Formatting fixed disks
If you are formatting your fixed disk, the /V switch is recommended. Identifying the disk with a 1- to 11-character volume label is good computer operational practice, and will prove useful later in the use of your C disk drive. Prior to formatting you must use the FDISK utility to partition the fixed disk.

Fixed disks are formatted at 17 sectors per track.

Formatting flexible disks

The next group of switches allows various formats to be created on flexible disk. If no switch is specified, the default format is assumed: ~~9 sectors per track, double-sided, double density (360 KB disk capacity).~~ 1.2Mb.

/1

Formats at 9 sectors per track, single-sided, double density (180 KB disk capacity).

/8

Formats at 8 sectors per track, double-sided, double density (320 KB disk capacity).

/1/8

Formats at 8 sectors per track, single-sided, double density (160 KB disk capacity).

NOTE: These 4 formats are IBM compatible.

The /B switch may not be used simultaneously with either the /V or /S options. /B formats at 8 sectors per track and leaves room for the DOS operating system without copying it, in case you wish to put another version of NCR-DOS on the disk. The /V switch may not be used simultaneously with the /8 option.

/4 = 360 Kb

GRAFTABL

TYPE

External

PURPOSE

Enables full character support in graphics mode.

SYNTAX

```
GRAFTABL
```

COMMENTS

In graphics mode, the GRAFTABL.COM file must be loaded to ensure full character support. Depending on the display mode selected (low, medium, or high resolution), the file initializes an 8x8 or an 8x8 and an 8x16 character generator. Interrupt vector 1FH is set up to point to an 8 by 128 byte table or to an 8 by 128 byte and a 16 by 128 byte table. These tables contain the hex values 128 through 255 for some non-United States characters.

When the command is entered, the program returns:

```
Display Mode

1 Low or Medium Resolution
2 High Resolution

*Enter Selection
```

If you enter "1" for low or medium resolution, the 8x8 character generator is loaded and interrupt vector 1FH set up to point to the 8 by 128 byte table, If you enter "2" for high resolution, the program loads both the 8x8 and the 16x8 character generator and interrupt vector 1FH points to the 8 by 128 byte table as well as to the 16 by 128 byte table.

The following message will be displayed:

```
    GRAPHICS CHARACTERS LOADED
```

Control is returned to DOS. The file will remain in memory until the system is reset or turned off. If you operate in low or medium resolution (1) it uses approximately 1K of user memory, in high resolution (2) approximately 3K.

GRAPHICS
(Screen Print)

TYPE

External

PURPOSE

Prints graphics on an acceptable printer

SYNTAX

GRAPHICS

COMMENTS

To print screen graphics on a printer, simply invoke the command by typing GRAPHICS. After the command is invoked, you can "dump" a screen by simultaneously pressing the <Shift> and the <Prt Sc> (print screen) keys. Once invoked, do not use the command again until after your system has been reset, since it increases the size of DOS by 688 bytes every time it's used. The GRAPHICS command is retained by the operating system until the system is reset.

ADVANCED INFORMATION

If you use this command frequently, it may prove helpful to add this to your AUTOEXEC.BAT file. If you want an assembler program to execute GRAPHICS, insert the following in your code:

```
PUSH BP
INT 5
POP BP
```

If the GRAPHICS command has been placed in the CONFIG.SYS file prior to booting, the <F3> key will invoke GRAPHICS.

HELP

TYPE

External

PURPOSE

Allows access to on-line information concerning internal and external commands.

SYNTAX

HELP [command]

COMMENTS

The Help tutorial can aid users in syntax, usage, and error routines for DOS commands. To exit from this utility, press the END key. The following commands are included in the tutorial

ASSIGN
BACKUP
BATCH COMMANDS
BREAK
CHDIR (CD)
CHKDSK
CLS
COMP
CONFIG.SYS
COPY
CTTY
DATE
DEL
DIR
DISKCOMP
DISKCOPY
ERASE
FDISK
FIND
FORMAT
MKDIR
MODE
PATH
PRINT
PROMPT
RECOVER
REN
RESTORE
RMDIR
SORT
SYS
TIME
TREE
TYPE
VER
VERIFY
VOL

KEYB

TYPE

External

PURPOSE

Loads a keyboard program into memory which disables the keyboard program resident in ROM BIOS and enables setup of another keyboard format.

SYNTAX

KEYBUK

or

KEYBGR

or

KEYBFR

or

KEYBIT

or

KEYBSP

COMMENTS

The KEYB utility makes it possible to change to a keyboard format other than the one resident in ROM BIOS by loading one of the programs listed above. A keyboard program remains in memory until the system is reset or turned off.

Another keyboard program can be loaded which will replace the earlier program. Although the eralier program still remains in memory, it cannot be returned to. Loading several keyboard programs without performing a system reset results in a considerable loss of memory space.

The following table shows the keyboard programs, the keyboard formats they support, and the memory allocated to each of them:

Program	Keyboard	Memory Space
KEYBUK	U.K. English	1712 Bytes
KEYBGR	German	2096 Bytes
KEYBFR	French	2080 Bytes
KEYBIT	Italian	1776 Bytes
KEYBSP	Spanish	2032 Bytes

You may return to the United States keyboard format, as is resident in ROM BIOS, by pressing the Ctrl and Alt keys together with the F1 key. Press the F2 key together with the Ctrl and Alt keys to return to the keybaord program last loaded into memory.

For additional information on keyboard formats, turn to Appendix E "Keyboard Considerations".

MKDIR
(Make Directory)

TYPE
Internal

PURPOSE
Creates a subdirectory on a disk.

SYNTAX

```
MKDIR [d:] [pathname]
```

COMMENTS
This command, which can also be called with the name MD, is used to create a hierarchical directory structure. When you are in your root directory, you can create subdirectories by using the MKDIR command. The command

```
    MKDIR \USER
```

creates a subdirectory \USER in your root directory. To create a directory named JOE under \USER, type:

```
    MKDIR \USER\JOE
```

MODE

TYPE

External

PURPOSE

Defines the characteristics of a Color/Graphics Adapter and converts a Monochrome Display to a Color Display on a screen or a printer with both capabilities

SYNTAX

See "Formats" in the following section.

COMMENTS

If you find yourself issuing a MODE command each time you boot DOS, it's time to put this command in the AUTOEXEC.BAT batch file.

MODE handles the printers, serial adapters, and display adapters.

Format 1

For printing, MODE sets the lines per inch, characters per inch, and time-out handling, and indicates whether you are using a serial printer. Remember that you can have up to three printers. The first printer is called LPT1: It is the primary printer. (PRN is the pseudonym for whatever printer is LPT1: When you use PRN, you are telling DOS to use whatever has been assigned as LPT1:.)

The syntax for this part of MODE is

```
    MODE LPT#: [cpl] [,lpi] [, P]
```

where:

    # = the number of the printer (1, 2, or 3);

    cpl = the number of characters per line, based on an 8-inch wide line. The number can be either 80 (10 characters per inch) or 132 (16.5 characters per inch).

    lpi = lines per inch, which can be either 6 or 8.

    P = "keep trying" when DOS finds that the printer is not ready. (Time-out is the technical name.)

DOS starts with the printer at 80 characters per line and 6 lines per inch.

When using MODE, always include the colon with the device name. This means that the first line printer must be called LPT1:, not just LPT1. Because MODE also does not accept pathed device names, phrases such as \dev\Lpt1 cannot be used.

The P is the "keep trying" option. If you want DOS to keep trying to send characters to the printer when something is wrong, you must give the full command with the ,P every time you use MODE LPTx: The first time you give MODE LPTx: without the ,P, DOS will not retry continuously but will give an error message when the printer "times out."

If you give the ,P and your printer hangs up, your computer will lock up. You can get the computer out of this by typing a control-break sequence (<Ctrl-Break> or <Ctrl-C>). You'll need to wait almost a minute more for DOS to recognize the control-break.

If you don't want to change the characters per line, the lines per inch, or the continuous retries, either drop the element from the command line or just use the comma and leave out the number.

The following command line

```
    MODE LPT1: 132,8
```

changes the characters per line to 132 and the lines per inch to 8.

The command line

```
    MODE LPT1: 80
```

sets the characters per line to 80, but leaves the lines per inch the same.

The command line

```
    MODE LPT1: ,6
```

leaves the characters per line unchanged and changes the lines per inch to 6. In all three examples, DOS will not retry on a time-out because the ,P has not been included.

Format 2

The second format of MODE also relates to the printer. The syntax is

```
MODE LPT#:=COMx
```

where:

# = the number of the printer you are reassigning
x = the communications adapter for your serial printer.

This command tricks DOS into using the serial printer connected to a communications adapter instead of the normal parallel printer. You may need to check what the correct number is for the communications adapter because the number can be either 1 or 2. Use 1 if you have two adapters or check with your dealer.

The following command

```
MODE LPT1:=COM1
```

redirects DOS to use the printer that is connected to the first communications adapter.

To reverse this setting, use format 1 of the MODE command.

Before you use this version of MODE, you should set up the asynchronous adapter, using format 3 of the MODE command covered in the next section.

Format 3

This format of MODE sets the communications adapter's characteristics: baud rate, parity, data bits, stop bits, and retries. The syntax is

```
MODE COMx: baud rate, [parity, data bits, stop bits, P]
```

where:

x = the number of the communications adapter, 1 or 2.
baud rate = 110, 300, 600, 1200, 2400, 4800, or 9600. (You may abbreviate and give just the first two numbers of the rate.)
parity = Odd or Even. The default value is Even.
data bits = 7 or 8. The default value is 7.
stop bits = either 1 or 2. If you use 110 baud, 2 stop bits is the default. Any baud rate other than 110 defaults to 1 start bit.
P = "keep trying".

"baud rate" is the only required parameter; all other parameters are optional, but, when included, must be in the sequence shown in the syntax line. Use a comma to indicate an omitted parameter.

If you don't set P or forget to, then it is off. Retry is good for printers but not as good when the adapter is being used with a modem. Use a Ctrl-Break or Ctrl-C to get out of a time-out retry loop.

Either one of the following two command lines

```
MODE COM1: 1200
MODE COM1: 12
```

sets the baud rate at 1200 for the first communications adapter. Everything but retry (,P) remains the same. (Even if retry was on, it is now off because the P is not included.)

The command line

```
MODE COM1: 96,,8
```

sets the first adapter to 9600 baud and 8 data bits. Everything else but the retry is unchanged. Retry is turned off.

The command line

```
MODE COM1: 48,,,,P
```

sets the adapter to 4800 baud and continuous retries. As before, nothing else is changed.

Format 4

The last format of MODE handles the video displays. If you use only the Monochrome Display, this command will not work.

The syntax of this command is

```
MODE display type, shift, T
```

The shift shifts the display left or right by one character position. Enter either L or R. MODE can be used with either a television set or a monitor to adjust the display. With some monitors, characters "fall off" the edge of the screen. This command lets you shift the line back into position.

T stands for test pattern. When you request it, MODE shows a line of either 40 or 80 characters. Then, MODE asks whether the display is okay. You enter Y if it is. If you enter N, MODE again shifts the display by one character and repeats the process. In this way, you can adjust your screen without having to re-enter the MODE command.

The display type can be any one of the following. (The C/G display refers to the monitor or television attached to the Color/Graphics Adapter.)

40 40-column lines for the Color/Graphics display

80 80-column lines for the Color/Graphics display

BW40 Makes the C/G display the active display, turns off color, and uses 40-character lines

BW80 Makes the C/G display the active display, turns off color, and uses 80-character lines

CO40 Makes the C/G display the active display, turns on color capabilities, and uses 40- character lines

CO80 Makes the C/G display the active display, turns on color capabilities, and uses 80- character lines

MONO Makes the Monochrome Display the active display

An active display is the display that DOS and your programs use to show information. If you have two displays, such as a Monochrome Display and another monitor or TV attached to the C/G board, either one can be used with this command.

Color is arbitrary. Some programs use color; others do not. Specifying CO40, or C080 does not guarantee that your programs will be in color; the selection merely lets DOS display things in color. It's up to you and your programs to make color appear.

MODE also doesn't seem to affect BASIC. If you use a color command in BASIC, it makes no difference whether you specify BW40, BW80, CO40, or CO80. BASIC still tries to send colors. Whether you see colors or not depends on the monitor or TV you use. However, the size of the screen (40 or 80 columns) will remain in effect.

MORE

TYPE

External

PURPOSE

Sends output to console at one screen of information at a time, allowing operator to see the displayed information before continuing.

SYNTAX

MORE

COMMENTS

MORE is a filter that reads from standard input (such as a command from your terminal) and displays one screen of information at a time. The MORE command then pauses and displays the --MORE--message at the bottom of your screen.

Pressing any key displays another screen of information. This process continues until all the input data has been read. To stop the execution of the command, press <Ctrl-Break> or <Ctrl-C>.

The MORE command is useful for viewing a long file or directory one screen at a time. If you type

```
TYPE MYFILES.COM ¦ MORE
```

NCR-DOS displays the file MYFILES.COM (on the default drive) one screen at a time. Notice that there is be a space both before and after the pipe. These spaces are optional.

PATH

TYPE

Internal

PURPOSE

Sets a command path.

SYNTAX

PATH [[d1:]pathname1] [;[d2:]pathname2]
[;[d3:]pathname3];[...]

COMMENTS

This command allows you to tell DOS which directories to search for external commands after DOS has searched your working directory. The default value is \BIN, where \BIN is the name of the directory in which all DOS external commands reside.

To tell DOS to search your \BIN\USER\JOE directory for external commands (in addition to a search of the \BIN directory), type:

```
PATH\BIN\USER\JOE
```

DOS now also searches the \BIN\USER\JOE directory for external commands until you set another path or shut down DOS.

You can tell DOS to search more than one path by specifying several pathnames separated by semicolons. For example,

```
PATH\BIN\USER\JOE;\BIN\USER\SUE;\BIN\DEV
```

tells DOS to search the directories specified by the above pathnames to find external commands. DOS searches the pathnames in the order specified in the PATH command.

The command PATH with no options prints the current path. If you specify PATH;, DOS sets the NUL path, meaning that only the working directory is searched for external commands.

PRINT

TYPE

External

PURPOSE

Prints a text file on a line printer while you are processing other NCR-DOS commands (usually called "background printing").

SYNTAX

PRINT [[d:] [pathname1]filename1[.ext1]] [/T] [/C] [/P]]...

where:
/T = Terminate
/C = Cancel
/P = Print

COMMENTS

You use the PRINT command only if you have a line printer attached to your computer. The following options are provided with this command:

/T — TERMINATE

This option deletes all files in the print queue (those waiting to be printed). A message to this effect is printed.

/C — CANCEL

This option turns on cancel mode. The preceding filespec and all following filespecs are suspended in the print queue until you type a /P option.

/P — PRINT

This option turns on print mode. The preceding filespec and all following filespecs are added to the print queue until you issue a /C option.

PRINT with no options displays the contents of the print queue on your screen without affecting the queue.

Consider the following examples:

```
PRINT /T
```

empties the print queue.

PRINT /T*.ASM

empties the print queue and queues all.ASM files on the default drive.

PRINT A:TEMP1.TST/C A:TEMP2.TST A:TEMP3.TST

removes the three files indicated from the print queue.

PRINT TEMP1.TST /C TEMP2.TST /P TEMP3.TST

removes TEMP1.TST from the queue, and adds TEMP2.TST and TEMP3.TST to the queue.

If an error is detected, PRINT displays one of the following error messages:

Name of list device [PRN:]
This prompt appears when PRINT is run the first time. Any current device may be specified and that device then becomes the PRINT output device. As indicated in the [ ], simply pressing <CR> results in the device PRN being used.

List output is not assigned to a device
This message is displayed if the "Name of list device" specified to the preceding prompt is invalid. Subsequent attempts return the same message until a valid device is specified.

PRINT queue is full
There is room for 10 files in the queue. If you attempt to put more than 10 files in the queue, this message appears on the console.

PRINT queue is empty
There are no files in the print queue.

All files cancelled by operator

If the /T (TERMINATE) option is used, the message "All files cancelled by operator" is output on your printer. If the current file being printed is cancelled by a /C, the message "<filename> cancelled by operator'' is printed.

PROMPT

TYPE
Internal

PURPOSE
Changes the NCR-DOS command prompt.

SYNTAX

PROMPT [prompt-string]

COMMENTS
This command allows you to change the NCR-DOS system prompt. If no text is typed, the prompt is set to the default prompt, which is the default drive designation. You can set the prompt to a special prompt by using the characters indicated below.

The following characters can be used in the prompt command to specify special prompts. They must all be preceded by a dollar sign ($) in the prompt command:

Use This Character		**To Get This Prompt:**
$	-	The '$' character
t	-	The current time
d	-	The current date
p	-	d:\current path
v	-	The version number
n	-	The default drive
g	-	The '>' character
l	-	The '<' character
b	-	The '¦' character
←	-	A <CR>, line feed sequence

s	-	A space
h	-	A backspace, erasing the previous character
e	-	ASCII code X'1B' (escape)
q	-	The equal sign

For example, the command PROMPT $g will give the user the prompt ">".

Consider the following example:

PROMPT $n:

Sets the prompt to the default drive followed by a colon.

RECOVER

TYPE

External

PURPOSE

Recovers a file or an entire disk containing bad sectors or a bad directory.

SYNTAX

```
RECOVER [d:] [pathname] [filename] [.ext]
RECOVER d:
```

COMMENTS

If a sector on a disk is bad, you can recover either the file contained in that sector (without the bad sector) or the entire disk (if the bad sector was in the directory).

To recover a particular file, type:

RECOVER filename

This causes NCR-DOS to read the file sector-by-sector and to skip the bad sector(s). When NCR-DOS finds the bad sector(s), the sector(s) are marked and NCR-DOS no longer allocates your data to that sector.

To recover a disk, type:

RECOVER d:

where d: is the letter of the drive containing the disk to be recovered.

When recovering an entire disk, the directory will be rebuilt, and all files will be renamed FILEnnnn.REC where nnnn is a four digit number. It is generally a good idea to print your old directory, making the renaming process easier. Sub-directories are also renamed in the FILennnn.REC format.

If there is not enough room in the root directory, RECOVER prints a message and stores information about the extra files in the File Allocation Table. You can run RECOVER again to regain these files when there is more room in the root directory.

RENAME (REN)

TYPE

Internal

PURPOSE

Changes the name of the first option (filespec) to the second option (filename)

SYNTAX

```
RENAME   [d1:] [pathname1] [filename1 [.ext1]] [d2:] [filename2]
[.ext2]
```

COMMENTS

The first option (filespec) must be given a drive designation if the disk resides in a drive other than the default drive. Any drive designation for the second option (filename) is ignored. The file remains on the disk where it currently resides.

The universal characters may be used in either option. All files matching the first filespec are renamed. If universal characters appear in the second filename, corresponding character positions are not changed.

For example, the following command changes the names of all files with the .LST extension to the same names with a .PRN extension:

```
REN *.LST *.PRN
```

In the next example, REN renames the file ABODE on drive B to ADOBE:

```
REN B:ABODE ?D?B?
```

The file remains on drive B.

An attempt to rename a file to a name already present in the directory results in the error message "Duplicate file name or file not found".

restore a: c:\ranger\*.*

RESTORE

TYPE

External

PURPOSE

Restores backed-up files from a flexible disk to the fixed disk.

SYNTAX

RESTORE d1: {C: | D:} [pathname] [filename.ext.] [/ S] [/P]

[/P]

where:

d1: = the disk drive containing the flexible disk

{C: | D:} = the fixed disk drive

pathname = the directory path that will receive the RESTORED file

filename.ext = the name of the file to be restored

/S = restores files in current directory and all of its subdirectories

/P = supplies prompts for the user to determine which recently altered files to restore. It also prompts for read-only (hidden) files.

COMMENTS

The RESTORE command replaces a previously backed-up file from a flexible disk to the fixed disk. This command only restores files that have been saved through the DOS BACKUP command. As in other DOS commands, the absence of a stated pathname defaults to the current directory; the lack of a filename in the command implies the request for all files (*.*) in the directory.

The error codes for exiting this command will be one of the following

0 NORMAL COMPLETION

1 NO FILES FOUND TO RESTORE

3 TERMINATE BY THE OPERATOR (through a <Ctrl-Break> or <Esc>)

4 TERMINATED DUE TO ERROR

These codes can be used with the IF command (batch processing).

If the user decides to include the /S parameter within the RESTORE command, the subdirectories will be re-created prior to restoring the files into them.

RMDIR
(REMOVE DIRECTORY)

TYPE

Internal

PURPOSE

Removes a (sub) directory from a hierarchical directory structure.

SYNTAX

RMDIR [d:]pathname

COMMENTS

This command, which can be abbreviated RD, removes a (sub) directory that is empty except for the . and .. shorthand symbols.

To remove the \BIN\USER\JOE directory, first issue a DIR command for that path to ensure that the directory does not contain any important files that you do not want deleted. If no such files are present, but files which can be deleted do exist, use DELETE or ERASE to delete all except the . and .. entries in the directory. Then type:

```
    RMDIR \BIN\USER\JOE
```

The directory is deleted from the directory structure.

SET

TYPE

Internal

PURPOSE

Sets one string value equivalent to another string for use in later programs.

SYNTAX

```
SET [string1=[string2]]
```

where:
    string1 = the actual string you are adding.
    string2 = the parameter you want string1 to equal.

COMMENTS

This command is meaningful only if you want to set values that will be used by programs you have written.

Set without an accompanying string parameter, the command displays the current values. These values will initially only consist of PATH=pathnames, COMSPEC=A:\COMMAND.COM. (the location of your command processor), and PROMPT= for any prompt variations. Using SET with string1 but not string2 resets string1 to its default value.

ADVANCED INFORMATION

An application program can check all values that have been set with the SET command by issuing SET with no options. For example, SET TTY = VT52 sets your TTY value to VT52 until you change it with another SET command (SET TTY=).

The SET command can also be used in batch processing. In this way, you can define your replaceable parameters with names instead of numbers. If your batch file contains the statement "LINK %FILE%", you can set the name that NCR-DOS will use for that variable with the SET command. The command SET FILE=DOMORE replaces the %FILE% parameter with the filename DOMORE. Therefore, you do not need to edit each batch file to change the replaceable parameter names. Note that when you use text (instead of numbers) as replaceable parameters, the name must end with a percent sign.

SORT

TYPE

External

PURPOSE

Reads input from a file or your terminal, sorts the data, then writes it to your output device.

SYNTAX

```
SORT [<filespec1] [>filespec2] [/R] [/+n]
```

where:

<filespec1 = source for sort
>filespec2 = output for sorted data.
/R = Reverse order of sort
/+n = Starting column of sort key

COMMENTS

SORT can be used, for example, to alphabetize a file by a certain column. There are two options that you are allowed to select.

/R

Reverses the sort; that is, sorts from Z to A.

/+n

Sorts starting with column n where n is some number. If you do not specify this switch, SORT begins sorting from column 1.

Consider the following examples. In the first one, the command reads the file UNSORT.TXT, reverses the sort, and then writes the output to a file named SORT.TXT:

```
SORT/R <UNSORT.TXT>SORT.TXT
```

The next command pipes the output of the directory command to the SORT filter. The SORT filter sorts the directory listing starting with column 14 (this is the column in the directory listing that contains the file size), then sends the output to the console. Thus, the result of this command is a directory sorted by file size:

```
DIR ¦ SORT /+14
```

The command

```
DIR ¦ SORT/+14 ¦ MORE
```

does the same thing as the command in the previous example, except that the MORE filter gives you a chance to read the sorted directory one screen at a time.

ADVANCED INFORMATION

Although the SORT command can be a very useful tool, it has some limitations. Since it sorts according to ASCII values, certain irregularities may appear within your sorted data.

- In ASCII, uppercase letters precede lowercase letters in the ascending order of a sort. Therefore, a lowercase "a" will come after all uppercase letters, and uppercase "Z" will precede any lowercase letters.

- Unless numbers have the same decimal placement and the same amount of significant digits, a sorted list of numbers may prove inaccurate. Since sort judges all characters as letters, the number 1 precedes the number 2, but so do the numbers 10, 186, 1984, etc. However, if the numbers are listed as 0001, 0002, 0010, 0186, and so on, the sort will perform correctly.

- Special characters and control characters are sorted according to their ASCII value. Therefore, an end-of-file marker (Ctrl-Z) or a Tab (Ctrl-I) may destroy the integrity of your sort. Removing the special characters will remedy that situation.

SYS
(SYSTEM)

TYPE
External

PURPOSE
Transfers the NCR-DOS system files from the disk in the default drive to the disk in the drive specified by d:

SYNTAX

SYS d:

COMMENTS
SYS is normally used to update the system or to place the system on a formatted disk that contains no files. An entry for d: is required. There must be enough room on the destination disk, preferably through the use of the /S or /B options with the FORMAT command.

If IO.SYS and MSDOS.SYS are on the destination disk, they must take up the same amount of space on the disk as the new system will need.

The destination disk must be completely blank or already have the system files IO.SYS and MSDOS.SYS.

The transferred files are copied in the following order:

IO.SYS
MSDOS.SYS

IO.SYS and MSDOS.SYS are both hidden files that do not appear when the DIR command is executed. COMMAND.COM (the command processor) is not transferred. You must use the COPY command to transfer COMMAND.COM and any other external DOS commands you may want on the new disk.

If SYS detects an error, one of the following messages will be displayed:

Not enough room for MS-DOS on this disk
There is not enough room on the destination disk for the IO.SYS and MSDOS.SYS files.

SYS cannot install MS-DOS on this disk

The system files IO.SYS and MSDOS.SYS will not take up the same amount of space on the destination disk as was allocated for them.

TIME

TYPE

Internal

PURPOSE

Displays and sets the time.

SYNTAX

TIME [hh[:mm[:ss[:cc]]]]

where:

hh = hour
mm = minute
ss = seconds
cc = hundredths of seconds

COMMENTS

If the TIME command is entered without any parameters, the following message is displayed:

```
Current time is hh:mm:ss.cc
Enter new time:_
```

Press the <CR> key if you do not want to change the time shown. A new time may be given as an option to the TIME command as in:

```
TIME 8:20
```

The new time must be entered using numerals only; letters are not allowed. The allowed options are:

hh = 00-24
mm = 00-59

The hour and minute entries must be separated by colons or periods. You do not have to type the mm (minutes) ss (seconds) or cc (hundredths of seconds) options.

DOS uses the time entered as the new time if the options and separators are valid. If the options or separators are not valid, DOS displays the message:

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

DOS then waits for you to type a valid time.

ADVANCED INFORMATION

If you are using the AUTOEXEC.BAT file, the command TIME must be included if you wish to enter the correct time at the start of your session.

TREE

TYPE

External

PURPOSE

Displays all directories and sub-directories on a disk or all files within a directory

SYNTAX

TREE [d:] [/F] [>PRN]

where:

d: = the name of the disk drive to be displayed
/F = displays all the files in the current directory
>PRN = displays to printer

COMMENTS

If the user doesn't specify a disk drive the default drive will be used. The TREE command displays all sub-directories. With the /F option it will display all files with their corresponding pathnames. If there are more than a few files and sub-directories on the disk, you may want to route the TREE display to a printer through the use of >PRN.

TYPE

TYPE
Internal

PURPOSE
Displays the contents of the file on the console screen.

SYNTAX

```
TYPE [d:] [pathname]filename[.ext]
```

COMMENTS
Use this command to examine a file without modifying it. (Use DIR to find the name of a file and EDLIN to alter the contents of a file.) You cannot type a directory, nor can you use the universal characters * and ?.

ADVANCED INFORMATION
The only formatting performed by TYPE is that tabs are expanded to spaces consistent with tab stops every eighth column. Note that a display of binary files causes control characters (such as Ctrl-Z) to be sent to your computer, including bells, form feeds, and escape sequences.

VER
(VERSION)

TYPE
Internal

PURPOSE
Prints MS-DOS version number.

SYNTAX

```
VER
```

COMMENTS
If you want to know what version of MS-DOS you are using, type VER. The version number is displayed on your screen.

VERIFY

TYPE

Internal

PURPOSE

Turns the verify switch on or off when writing to disk.

SYNTAX

```
VERIFY  {ON  }
        {OFF }
```

COMMENTS

This command has the same purpose as the /V switch in the COPY command. If you want to verify that all files are written correctly to disk, you can use the VERIFY command to tell DOS to verify that your files are intact (no bad sectors, for example). DOS performs a VERIFY each time you write data to a disk. You receive an error message only if DOS was unable to successfully write your data to disk.

VERIFY ON remains in effect until you change it in a program (by a SET VERIFY system call), or until you issue a VERIFY OFF command to DOS.

If you want to know what the current setting of VERIFY is, type VERIFY with no options.

VOL

TYPE
Internal

PURPOSE
Displays disk volume identifier, if it exists.

SYNTAX

VOL [d:]

COMMENTS
This command prints the volume ID of the disk in drive d:. If no drive is specified, DOS prints the volume ID of the disk in the default drive.

Editing and Function Keys

SPECIAL EDITING KEYS

The special editing keys allow you to save some keyboard entry effort by repeating all or part or the last previous line you created on the display screen. Each line entered is copied by NCR-DOS into a special one-line storage area; characters needed for the next line can be selectively copied from the stored line into the next line with additional characters inserted or added as needed.

NCR-DOS makes this capability available in two different places in your operations:

- Use of special editing keys in the entry of NCR-DOS operating system control commands is defined in this chapter.

- Use of special editing keys in the editing of lines of data within source program or text files is defined in the *Line Editor (EDLIN)* chapter.

The keys used for special editing functions are the Delete (Del) key, the Escape (Esc) key, the Insert (Ins) key, and the first five of the programmable function keys at the left side of the keyboard (F1 through F5). Special editing keys and their functions are described in Figure 5-1.

PROGRAMMING OF EDITING KEYS

For editing purposes, NCR-DOS automatically programs the meanings shown in Figure 5-1 to function keys F1 through F5. F6 is also programmed to signal the end of certain keyboard entry operations, as defined in the description of the creation of a batch file in the *Learning About Commands* chapter and in the description of the COPY command in the *Commands* chapter.

However, F1 through F6 can be programmed to perform other functions, and other function keys can be programmed to perform the editing functions. This procedure is described in the "Programming of Function Keys" section of this chapter. If any editing function is programmed to a different key, or eliminated by reprogramming of any F1-F5 key, the descriptions of the use of that function key as given in this chapter and in the Line Editor (EDLIN) chapter will no longer be accurate.

Function	**Key**	**Description**
Copy one character	F1 or ->	Copies one character from the stored line to the command line.
Copy up to character	F2	Copies characters up to the character specified from the stored line to the command line.
Copy stored line	F3	Copies all remaining characters from the stored line to the command line.
Skip one character	Del	Skips over (does not copy) a character in the stored line.
Skip up to character	F4	Skips over (does not copy) the characters in the stored line up to the character specified.
Quit input	Esc	Voids the current input line; leaves the stored line unchanged.
Insert mode	Ins	Enters insert mode.
Replace mode	Ins (2nd time)	Turns off insert mode; this is the default mode.
Store new line	F5	Makes the new line the stored line.

Figure 5.1 Special editing functions

EDITING OF COMMANDS

Using the stored line and special editing keys, you can take advantage of the following DOS features:

- A command line can be instantly repeated by pressing two keys, <F3> (copy remainder of stored line) and <CR>.
- If you make a mistake in the command line, you can edit it and retry without having to retype the entire command line.
- A command line that is similar to a preceding command line can be edited and executed with a minimum of typing.

The relationship between the command line and the special line storage area is shown in Figure 5-2.

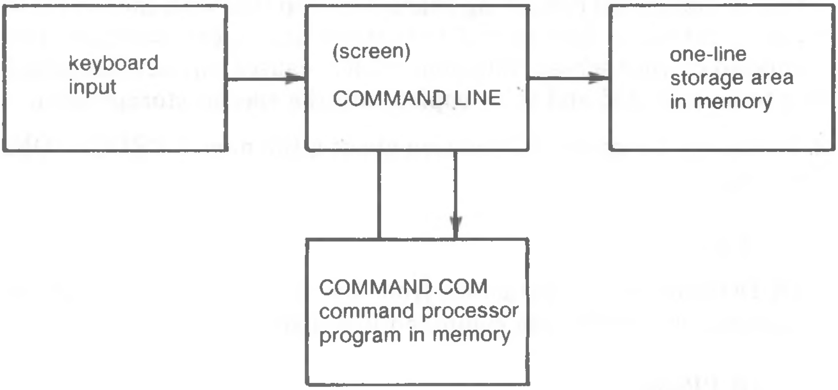

keyboard input
(screen)
COMMAND LINE
one-line storage area in memory
COMMAND.COM command processor program in memory

Figure 5.2 Command line usage

You type a command to NCR-DOS on the command line. When you press the <CR> key, the command is automatically sent to the command processor (COMMAND.COM) for execution. At the same time, a copy of this command is stored in the special one-line storage area (replacing the previous stored line). You can now recall the command or modify it with the NCR-DOS special editing keys.

The examples in the following section illustrate the use of the editing keys. Each of these keys is more fully described in the next chapter, where they can be used to edit your source program or text files.

EXAMPLES

If you type the command:

DIR PROG.COM<CR>

NCR-DOS displays information from the current directory about the file PROG.COM on your screen. The command line is also copied into the special storage area. This stored line can be edited in different ways:

1. If you want to repeat the command with no change, type:

<F3><CR>

The display now reads:

```
DIR PROG.COM
```

The function of <F3> is to copy all remaining characters from the stored line to the command line. Since this is the first editing key used, the copy begins with the first character of the stored line. The effect of copying all remaining characters starting with the first is to create a complete duplicate of the stored line in the command line displayed on your screen. Entering <CR> causes this new command line to be executed and to be copied into the special storage area.

2. To display directory information about a file named PROG1.COM, first type:

<F2>.

NCR-DOS copies all characters from the stored line up to, but not including, the period. The command line display now reads:

```
DIR PROG _
```

Note that the underline is your cursor. Now type:

<Ins>1

Insert mode is entered; a "1" is inserted. The command line now reads:

```
DIR PROG1 _
```

Now type:

><F3><CR>

Any function key and <CR> turns off insert mode (the Del key does not); the <F3> key copies the remainder of the stored line; the display is now:

```
DIR PROG1.COM _
```

<CR> causes the command line to be executed and to be copied into the special storage area.

Note that the first use of the <Ins> key to enter insert mode was required to permit insertion of the "1" ahead of the period. If the <Ins> key had not been used, the default condition, replacement mode, would have been in effect. In replacement mode, the "1" entry would have been taken as a replacement for the period; the subsequent <F3> copy would have begun with the character following the period and the resulting command line would have read DIR PROG1COM.

3. To next display directory information about a file named PROG.ASM, first type:

<F2>1

NCR-DOS copies all characters from the stored line up to, but not including, the "1" inserted during the previous example. The display now reads:

```
DIR PROG _
```

Now type:

<Del>

This skips over the "1" in the stored line. The displayed command line does not change.

Next type:

<F1> (or ->)

Either key causes the copy of one character, the period, from the stored line. The display now reads:

```
DIR PROG. _
```

(The same result can be obtained by typing:

<F2>C

This entry (copy up to the "C") can be used instead of <F1> because in this particular instance only the period is copied.)

Now type:

ASM

The result is:

```
DIR PROG.ASM _
```

Pressing <**CR**> causes this command line to be sent to the command processor for execution and to be copied into the special storage area.

Note that the "COM" extension previously in the stored line is not copied, skipped or affected by any of the editing functions; it is ignored. It is then destroyed when the revised command is copied into the storage area replacing the old contents.

Note also that another alternate sequence of entries which can be used to accomplish the same results as in this example. This sequence, is shown last because it illustrates the use of fewer special editing keys, is to type:

<**F2**>**1**

as before. After the display:

```
DIR PROG _
```

type:

.ASM

The display is now:

```
DIR PROG.ASM _
```

To execute and copy this command, type:

<**CR**>

4. Now assume that you want to display the contents of the file whose directory information was displayed in the previous step. You want to modify the previous command to read:

```
TYPE PROG.ASM
```

To do this, type:

TYPE<**Ins**><**space**><**F3**><**CR**>

(where <**space**> represents entry of a single space character)

Notice that when you type the letters "TYPE", these characters go directly into the command line. Since you are in replacement mode (rather than insert mode), these characters effectively overwrite, or replace, the corresponding characters ("DIR<space>") in the stored line. The next editing function which references the stored line will now begin with the "P".

The first <Ins> and the <space> character are needed now to insert a space after "TYPE" before copying "PROG.ASM" into the command line. The second <Ins> is required to turn off insert mode and return to replacement mode. <F3> copies the remainder of the stored line (PROG.ASM) into the command line. <CR> causes the command line to be executed and to be copied into the special storage area.

5. If at this point you decide that you want to see directory information for all assembler language source programs in the subdirectory PAYROLL of the subdirectory PROGSRCE on this disk, you may type:

<F4><space><Del><Ins>DIR<space>
PROGSOURCE\PAYROLL<space><Ins><F2>.
<Ins>*<F3>

<F4> skips over the characters in the stored line up to the space. The other editing keys function as previously described. The command line now displays:

```
DIR \PROGSOURCE\PAYROLL\PROG*.ASM _
```

6. You realize that you forgot to abbreviate SOURCE in the first subdirectory name. To fix this, first type:

<F5>

This copies the new command line into the special storage area without executing it. NCR-DOS identifies the results of this function by leaving the command line (the new stored line) displayed on the screen, followed by an "@". The cursor is then positioned to start a second line for the new command line. The display now appears:

```
DIR \PROGSOURCE\PAYROLL PROG*.ASM@
_
```

Now type:

<F2>S<F1><Del><Del><F3><CR>

F2>S copies up to the"S". (Your first inclination would be to copy up to the "O" in "SOURCE". However, a single <F2> would only copy up to the "O" in "PROG", which is not what you want.) <F1> copies the "S" (note that the sequence <F2>O<F2>O would be equivalent to <F2>S<F1>). The <Del><Del> sequence skips the "OU" characters. <F3> copies the remainder of the latest stored line into the command line. The display is now:

```
DIR \PROGSOURCE\PAYROLL PROG*.ASM@
DIR \PROGSRCE\PAYROLL PROG*.ASM _
```

<CR> causes the new command (bottom) line to be executed and to be copied into the special storage area.

7. Next you may decide to display the contents of one of the payroll source programs found in the directory display during the previous example.

To do this, you type:

**<Ins>TYPE<F4><space><F2>*<Del><Ins>W2
<Ins><F3>**

The display now reads:

```
TYPE \PROGSRCE\PAYROLL PROGW2.ASM_
```

8. Before you press <CR>, you change your mind and decide you would rather see the directory display of your general ledger source programs first. You type:

<Esc>

<Esc> voids the current input line and leaves the special one-line storage area unchanged. NCR-DOS identifies the result of this action by showing a "\" following the line you voided. The display now reads:

```
TYPE \PROGSRCE\PAYROLL PROGW2.ASM\
_
```

Since the stored line still contains the last "DIR ..." entry, you now type:

<F2>P<F2>P<Del>LEDGER<F3>

The two <F2>P functions copy up to the word "PAYROLL" in the "DIR ..." command; the <Del> skips the letter "P"; the letters "LEDGER" replace the letters "AYROLL"; and the <F3> function copies the remainder ("<space>PROG*.ASM") of the stored line. The display now reads:

```
TYPE \PROGSRCE\PAYROLL PROGW2.ASM\
DIR \PROGSRCE\LEDGER PROG*.ASM _
```

<CR> now causes the new command (bottom) line to be executed and to be copied into the special storage area.

FUNCTION KEYS

This section tells you how to program individual keys of your keyboard. You will probably want to use this facility for programming the Function Keys or the Alt mode of particular keys.

PROGRAMMING OF FUNCTION KEYS

First, you have to install in CONFIG.SYS the ANSI driver which is provided on your NCR-DOS disk. The filename of this driver is ANSI.SYS. Using EDLIN or another editor, add the following line to the file CONFIG.SYS (if CONFIG.SYS does not yet exist, start a new file with that filename):

```
DEVICE=ANSI.SYS
```

Now all you have to do is type an NCR-DOS ESC command. The general format of the command is

```
ESC  [0;<key position>;{"<string>"¦<#>}[;...]p
```

- <key position> is the scan code of a function key or of another programmable key.
- < string> is a sequence of any characters from the keyboard with the exception of the quotation marks (")
- <#> is the decimal value of a character which cannot be displayed (e.g. 34 for ", 13 for <CR>, etc.)
- p is the end-of-sequence sign (hexadecimal 70 = ASCII character p).

Table of Function Key Scan Codes

<F1>	...	<F10>	:	59,... , 68
<Shift-F1>	...	<Shift-F10>	:	84,... , 93
<Ctrl-F1>	...	<Ctrl-F10>	:	94,... ,103
<Alt-F1>	...	<Alt-F10>	:	104,... ,113

You can also define the alphanumeric keys of your keyboard in the Alt mode. The scan codes of these keys are

<Alt-1> , <Alt-2>, ...	:	120, ... ,131	(top row of the keyboard)
<Alt-Q> ... <Alt-P>	:	16, ... , 25	(Q,W,E,R,T,Y,U,I,O,P)
<Alt-A> ... <Alt-L>	:	30, ... , 38	(A,S,D,F,G,H,J,K,L)
<Alt-Z> ... <Alt-M>	:	44, ... , 50	(Z,X,C,V,B,N,M)

In addition, individual keys can be defined in the base, shifted and Ctrl modes:

<Shift-TAB>	: 15,	<Home>	: 71,
<Arrow up>	: 72,	<PgUp>	: 73,
<Arrow left>	: 75,	<Arrow right>	: 77,
<End>	: 79,	<Arrow down>	: 80,
<PgDown>	: 81,	<Ins>	: 82,
<Del>	: 83,	<Ctrl-PrtSc>	:114,
<Ctrl-Arrow-left>	:115,	<Ctrl-Arrow-right>	:116,
<Ctrl-End>	:117,	<Ctrl-PgDn>	:118,
<Ctrl-Home>	:119,	<Ctrl-PgUp>	:132,
<Ctrl-2>	: 3.	Ctrl and key no. 3 (char. 2 on alphanumeric keyboard) create NUL character	

Example:
Definition of function key F10 with the CHKDSK<CR> command.

```
ESC  [0;68;“CHKDSK”;13p
```

For further information and examples refer to the description of the ESC command, Chapter 4.

KEYBOARD KEY REASSIGNMENT

You can assign a new definition to an already defined key.
The command

```
ESC  [{<#>¦<string>}[;...]p
```

defines the key specified by the first character of the string or by the first decimal value <#> with the value of the string/decimal values.

To restore the original assignment, type

```
ESC  [<#>p
```

or reload the system. # represents the value of the reassignment.

Example:

```
ESC  [65;81p
ESC  [“AQ”p
ESC  [“A”;“Q”p
```

Each of these commands assigns the value of “Q” to the alphabetical key A.

The original assignment is restored with the command

```
ESC  [65p
```

Line Editor (EDLIN)

GENERAL INFORMATION

In this chapter, you learn how to use the Line Editor (EDLIN). You can use EDLIN to create, change, and display files, whether they are source program or text files. Specifically, you can use EDLIN to perform the following:

- Create new source files and save them.
- Update existing files and save both the updated and original files.
- Delete, edit, insert, and display lines.
- Search for, delete, or replace text within one or more lines.

The text in files created or edited by EDLIN is divided into lines, each up to 253 characters long. Line numbers are generated and displayed by EDLIN during the editing process, but are not actually present in the saved file.

When you insert lines, all numbers following the inserted text advance automatically by the number of lines being inserted. When you delete lines in a file, all line numbers following the deleted text decrease automatically by the number of lines deleted. As a result, lines are always numbered consecutively in your file.

HOW TO START EDLIN

To start EDLIN, type:

EDLIN [d:] [pathname] filespec [/B]

where /B means to ignore embedded
Ctrl-Z characters (end-of-file markers)

If you are creating a new file, the filespec should be the name of the file you wish to create. If EDLIN does not find this file on a drive, EDLIN creates a new file with the name you specify. The following message and prompt are displayed:

```
New file
*
_
```

SPECIAL EDITING KEYS

As you learned in the preceding chapter, there are special editing keys that are also useful when you are in EDLIN. Figure 6.1 contains a brief summary of what these keys do.

KEY	**FUNCTION PERFORMED**
Del	Skips one character (does not copy) one character in the stored line.
Esc	Voids the current input line; leaves the stored line unchanged.
Ins	Enters insert mode. Keytip light comes on. Press again to exit mode when done.
F1 or ->	Copies one character from the stored line to the new changed line, each time it is pressed.
F2x	Copies the stored line up to the character (x).
F3	Copies the remaining characters in the stored line to the new line.
F4x	Does not copy (skips) characters up to the (x).
F5	Makes the new line the stored line, moves the cursor to the beginning of the next line, ready for more changes.

Figure 6.1 Special Editing Keys

Notice that the prompt for EDLIN is an asterisk (*). You can now type lines of text into your new file. To begin entering text, you must enter an I (Insert) command to insert lines. The I command is discussed later in this chapter.

If you want to edit an existing file, <filespec> should be the name of the file you want to edit. When EDLIN finds the file you specify on the designated or default drive, the file is loaded into memory. If the entire file can be loaded, EDLIN displays the following message on your screen:

```
End of input file
*
```

You can then edit the file using EDLIN editing commands.

If the file is too large to be loaded into memory, EDLIN loads lines until memory is 3/4 full, and then displays the * prompt. You can then edit the portion of the file that is in memory.

To edit the remainder of the file, you must save some of the edited lines on disk to free memory; then EDLIN can load the unedited lines from disk into memory. Refer to the Write and Append commands in this chapter for the procedure.

When you complete the editing session, you can save the original and the updated (new) files by using the End command. The End command is discussed in this chapter in the section EDLIN Commands. The original file is renamed with an extension of .BAK, and the new file has the filename and extension you specify in the EDLIN command. The original .BAK file is not erased until the end of the editing session, or until disk space is needed by EDLIN.

CAUTION

> Do not try to edit a file with a filename extension of .BAK because EDLIN assumes that any .BAK file is a backup file. If you find it necessary to edit such a file, rename the file with another extension (using the NCR-DOS RENAME command discussed in the *Commands* chapter); then start EDLIN and specify the new <filespec>.

EDLIN COMMANDS

This section describes the individual EDLIN commands that perform editing functions on lines of text. These commands are to be used after the * prompt, not while in Edit mode. (Edit mode is entered with the Edit, line-number, command. See "line-number" in this chapter.) Before using an EDLIN command, read the conventions and options that apply to all commands.

FORMAT CONVENTIONS

1. Pathnames are acceptable as options to commands. For example, typing EDLIN \BIN\USER\JOE\TEXT.TXT allows you to edit the TEXT.TXT file in the subdirectory JOE.

2. You can reference line numbers relative to the current line (the line with the asterisk). Use a minus sign with a number to indicate lines before the current line. Use a plus sign with a number to indicate lines after the current line.

   Example

   -10,+10L

   This command line lists 10 lines before the current line, the current line, and 10 lines after the current line.

3. Multiple commands may be issued on one command line. When you issue a command to edit a single line using a line number (n), a semicolon must separate commands on the line. Otherwise, one command may follow another without any special separators.

   Example:

   15;-5,+5L<CR>

   NOTE: In the case of a Search or Replace command, the string may be ended by an <F6> or <Ctrl-Z> instead of a <CR>.

   The command line in the next example searches for "This string" and then displays 5 lines before and 5 lines after the line containing the matched string. If the search fails, then the displayed lines are those line numbers relative to the current line.

   This string <F6>-5,+5L<CR>

4. You can type EDLIN commands with or without a space between the line number and command. For example, to delete line 6, the command 6D is the same as 6 D.

5. It is possible to insert a control character (such as Ctrl-Z) into text by using the quote character <Ctrl-V> before it while in insert mode. <Ctrl-V> tells NCR-DOS to recognize the next capital letter typed as a control character. It is also possible to use a control character in any of the string arguments of Search or Replace by using the special quote character. For example:

S<Ctrl-V>Z<CR>
finds the first occurrence
of Ctrl-Z in a file

R<Ctrl-V>Z<Ctrl-Z> and <CR>
replaces all occurrences
of Ctrl-Z in a file with 'and'

It is possible to insert Ctrl-V into the text by typing <Ctrl-V)-V.

6. The Ctrl-Z (<F6> key) character ordinarily means, "This is the end of the file." If you have Ctrl- Z characters elsewhere in your file, you must tell EDLIN that these other control characters do not mean "End of File." Use the /B option when starting to tell EDLIN to ignore any Ctrl-Z characters in the file and to show you the entire file.

7. <line> when used in this text refer to the line number. Do not type "line", just the number of the desired line.

The EDLIN commands are summarized in Table 6.2. They are also described in further detail following the description of command options.

Command	**Purpose**
<line>	Edits line no.
A	Appends lines
C	Copies lines
D	Deletes lines
E	Ends editing
I	Inserts lines
L	Lists text
M	Moves lines
P	Pages text
Q	Quits editing
R	Replaces lines
S	Searches text
T	Transfers text
W	Writes lines to disk

Table 6.2 EDLIN commands

COMMAND OPTIONS

Several EDLIN commands accept one or more options. The effect of a command option varies, depending on the command. The following list describes each option.

<line>

<line> indicates a line number that you type. Line numbers must be separated by a comma or a space from other line numbers, other options, and from the command.

<line> may be specified in one of three ways:

- Number (n) — Any number less than 65534. If a number larger than the largest existing line number is specified, then <line> means the line after the last line number.
- Period (.) — If a period is specified for <n>, then <n> means the current line is marked on your screen by an asterisk (*) between the line number and the first character.
- Pound (#) — The pound sign indicates the line after the last line number. If you specify # for <line>, this has the same effect as specifying a number beyond the last line number.

<CR>

A carriage return entered without any of the <line> specifiers directs EDLIN to use a default value appropriate to the command.

?

The question mark is used only with the Replace and Search commands. The question mark option directs EDLIN to ask you if the correct string has been found. Before continuing, EDLIN waits for either a Y or <CR> for a yes response, or for any other key for a no response.

<string>

The <string> option is used only with the Search and Replace commands. <string> represents text to be found, and to be replace other text. The <string> option is used only with the Search and Replace commands. Each <string> must end with an <F6> or a <CR> (see the Replace command for details). Do not leave spaces between strings or between a string and its command letter, unless you want those spaces to be part of the string.

line-number

PURPOSE

Enters Edit mode; specifies line of text to be edited

SYNTAX

[line-number]

COMMENTS

When a line number is typed, EDLIN displays the line number and text; then, on the line below, EDLIN reprints the line number. The line is now ready for editing. You may use any of the EDLIN special editing keys to edit the line. The existing text of the line serves as the line storage area until the <CR> key is pressed.

If no line number is typed (that is, if only the <CR> key is pressed), the line after the current line (marked with an asterisk) is edited. If no changes to the current line are needed and the cursor is at the beginning or end of the line, press the <CR> key to accept the line as is.

CAUTION

> If the <CR> key is pressed while the cursor is in the middle of the line, the remainder of the line is deleted.

EXAMPLE

Assume that the following file exists and is ready to edit (You may type this in to better follow the example):

```
1:  This is a sample file.
2:  used to show
3:  the editing of line
4:  four.
```

To edit line 4, type after the *:

```
4
```

and press <CR>

The contents of the line are displayed

```
4:*four.
4:*
```

Now, using the editing keys, type:

<Ins>number<F3><CR>

You'll See:

```
4: number four.
5:*_
```

A(PPEND)

PURPOSE

Adds the specified number of lines from disk to the file being edited in memory. The lines are added at the end of lines that are currently in memory.

SYNTAX

[line-number]A

COMMENTS

This command is meaningful only if the file being edited is too large to fit into memory. As many lines as possible are read into memory for editing when you start EDLIN.

To edit the remainder of the file that will not fit into memory, you must write the lines already edited to disk. Then you can load unedited lines from disk into memory with the Append command. (Refer to the Write command in this chapter for information on how to write edited lines to disk.)

If you do not specify the number of lines to append, lines are appended to memory until available memory is 3/4 full. No action is taken if available memory is already 3/4 full.

The message "End of Input file" is displayed when the Append command has read the last line of the file into memory.

C(OPY)

PURPOSE

Inserts a range of lines ahead of a specified line number. The lines can be copied as many times as you want by using the count option.

SYNTAX

[line1],[line2],line3,[count]C

where:

line1 is the start of the text to be copied.
line2 is the end of the text to be copied.
line3 is the line after which the text will be copied
count the number of lines being copied.

COMMENTS

If you do not specify a number in "count", EDLIN copies the lines one time. If the first or the second line-number is omitted and a comma used in its place, the default is the current line. The file is renumbered automatically after the copy.

The line numbers must not overlap or you will get an "Entry error" message. For example, 3,20,15C would result in an error message.

EXAMPLES

Assume that the following file exists and is ready to edit:

```
1: This is a sample file
2: used to show copying lines.
3: See what happens when you use
4: the Copy command
5: (the C command)
6: to copy text in your file.
```

You can copy this entire block of text by issuing the following command:

```
1,6,7C<CR>
```

The result is:

```
1: This is a sample file
2: used to show copying lines.
3: See what happens when you use
```

```
 4: the Copy command
 5: (the C command)
 6: to copy text in your file.
 7: This is a sample file
 8: used to show copying lines.
 9: See what happens when you use
10: the Copy command
11: (the C command)
12: to copy text in your file.
```

If you want to place the text within other text, the third line-number should specify the line before which you want the copied text to appear. For example, assume that you want to copy lines and insert them within the following file:

```
 1: This is a sample file
 2: used to show copying lines.
 3: See what happens when you use
 4: the Copy command
 5: (the C command)
 6: to copy text in your file.
 7: You can also use COPY
 8: to copy lines of text
 9: to the middle of your file.
10: End of sample file.
```

The command **3,6,10C** results in the following file:

```
 1: This is a sample file
 2: used to show copying lines.
 3: See what happens when you use
 4: the Copy command
 5: (the C command)
 6: to copy text in your file.
 7: You can also use COPY
 8: to copy lines of text
 9: to the middle of your file.
10: See what happens when you use
11: the Copy command
12: (the C command)
13: to copy text in your
14: End of sample file.
```

D(ELETE)

PURPOSE

Deletes a specified range of lines in a file.

SYNTAX

[line-number] [,line-number]D<CR>

COMMENTS

If the first line-number is omitted, that option will default to the current line (the line with the asterisk next to the line number). If the second line-number is omitted, then just the first line-number will be deleted. When lines have been deleted, the line immediately after the deleted section becomes the current line and has the same line number that the first deleted line had before the deletion occurred.

EXAMPLES

Assume that the following file exists and is ready to edit:

```
  1: This is a sample file
  2: used to show dynamic line numbers.
  3: See what happens when you use
  4: Delete and Insert
   .
   .
   .
 25: (the D and I commands)
 26: to edit the text
 27: in your file.
```

To delete multiple lines (lines 5 through 24), type:

```
5,24D<CR>
```

The result is:

```
  1: This is a sample file
  2: used to show dynamic line numbers
  3: See what happens when you
  4: Delete and Insert
  5:*(the D and I commands)
  6: to edit text
  7: in your file.
```

To delete a single line, type only the line number desired. For example,

```
6D<CR>
```

Then, when you list the sample file (L<CR>), the result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6:*in your file.
```

Next, delete a range of lines from the following file:

```
1: This is a sample file
2: used to show dynamic line numbers.
3:*See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
```

To delete a range of lines beginning with the current line, and then to list the text again type:

```
,6D<CR>
L<CR>
```

The result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3:*in your file.
```

Notice that the lines are automatically renumbered.

(E)ND

PURPOSE

Ends the editing session.

SYNTAX

E<CR>

COMMENTS

This command saves the edited file on disk, renames the original input file <filename>.BAK, and then exits EDLIN. If the file is created during the editing session, no .BAK file is created.

The E command takes no options. Therefore, you cannot tell EDLIN on which drive to save the file. The drive you want to save the file on must be selected when the editing session is started. If the drive is not selected when EDLIN is started, the file is saved on the disk in the default drive. You can, however, copy the file to a different drive using the COPY command.

You must be sure that the disk contains enough free space for the entire file. If the disk does not contain enough free space, the write is canceled and the edited file is lost, although part of the file might be written out to the disk.

EXAMPLE

```
  E
```

After execution of the E command, the default drive prompt (for example, A>) is displayed.

I(NSERT)

PURPOSE

Inserts text immediately before the specified line-number.

SYNTAX

```
[line-number]I<CR>
```

COMMENTS

If you are creating a new file, the I command must be given before text can be typed (inserted). Text begins with line number 1. Successive line numbers appear automatically each time <CR> is pressed.

EDLIN remains in insert mode until <Ctrl-BREAK> is typed. When the insert is completed and insert mode has been exited, the line immediately following the inserted lines becomes the current line. All line numbers following the inserted section are incremented by the number of lines inserted.

If the line-number is not specified, the default is the current line number and the lines are inserted immediately before the current line. If the line-number is any number larger than the last number, or if a pound sign (#) is specified in place of 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.

EXAMPLES

Assume that the following file exists and is ready to edit:

```
    1: This is a sample file
    2: used to show dynamic line numbers.
    3: See what happens when you use
    4: Delete and Insert
    5: (the D and I commands)
    6: to edit text
    7: in your file.
```

To insert text before a specific line that is not the current line, type the line number. For example, type

```
    7I
```

The result is:

```
7: _
```

Now, type the new text for line 7:

```
7: and renumber lines<CR>
```

Then to end the insertion:

```
8: <F6><CR>
```

Now type L to list the file. The result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: and renumber lines
8:*in your file.
```

To insert lines immediately before the current line, type:

```
I<CR>
```

The result is:

```
8: _
```

Now, type the following text:

```
8: so they are consecutive<CR>
9: <F6><CR>
```

Now, when you list the file you'll see that the result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: and renumber lines
8: so they are consecutive
9:*in your file.
```

To append new lines to the end of the file, type:

```
    #I<CR>
```

This produces the following:

```
    10: _
```

Now, type the following new lines:

```
    10: The insert command can place new lines<CR>
    11: in the file; there's no problem<CR>
    12: because the line numbers are dynamic;<CR>
    13: they'll go all the way to 65533.<CR>
```

End the insertion by pressing <F6> <CR> on line 14.

```
    14: <F6><CR>
```

The new lines appear at the end of all previous lines in the file. Now type the list command, L:

The result is:

```
     1: This is a sample file
     2: used to show dynamic line numbers
     3: See what happens when you use
     4: Delete and Insert
     5: (the D and I commands)
     6: to edit text
     7: and renumber lines
     8: so they are consecutive
     9: in your file.
    10: The insert command can place new lines
    11: in the file; there's no problem
    12: because the line numbers are dynamic;
    13: they'll go all the way to 65533.
```

L(IST)

PURPOSE

Lists a range of lines, including the two lines specified.

SYNTAX

[line1] [,line2]L<CR>

COMMENTS

Default values are provided if either one or both of the options are omitted. If you omit the first option, as in:

```
,line2L
```

the display starts 11 lines before the current line and ends with the specified line. The beginning comma is required to indicate the omitted first option.

NOTE: If the specified line2 is more than 11 lines before the current line, the display is the same as if you omitted both options.

If you omit the second option, as in

```
line1L
```

23 lines are displayed, starting with the specified line-number.

If you omit both parameters, as in

```
L
```

23 lines are displayed: the 11 lines before the current line, the current line, and the 11 lines after the current line. If there are less than 11 lines before the current line, more than 11 lines after the current line are displayed to make a total of 23 lines.

EXAMPLES

Assume that the following file exists and is ready to edit:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
 .
 .
```

```
15:*The current line contains an asterisk.
 .
 .
 .
26:  to edit text
27:  in your file.
```

To list a range of lines without reference to the current line, type:

2,5L

The result is:

```
 2:  used to show dynamic line numbers.
 3:  See what happens when you use
 4:  Delete and Insert
 5:  (the D and I commands)
```

To list a range of lines beginning with the current line, type:

,26L

The result is:

```
15:*The current line contains an asterisk.
 .
 .
 .
26:  to edit text
```

To list a range of 23 lines centered around the current line, type:

L<CR>

The result is:

```
  4: Delete and Insert
  5: (the D and I command)
   .
   .
   .
 13: The current line is listed in the middle of the range.
 14: The current line remains unchanged by the L command.
 15:*The current line contains an asterisk.
   .
   .
   .
 26: to edit text
```

M(OVE)

PURPOSE

Moves a range of text to the line specified.

SYNTAX

```
[line1],[line2],line3M<CR>
```

COMMENTS

Use the Move command to move a block of text (from the first line-number (line1) to the second (line2)) to another location in the file (line3). The lines are renumbered according to the direction of the move. In other words, the lines following the section to be moved are moved up accordingly. For example,

```
,+25,100M
```

Inserts the block of text from the current line plus 25 lines to be put in front of line 100. If the line numbers overlap, EDLIN displays an "Entry error" message.

To move lines 20-30 to line 100, type:

```
20,30,100M
```

P(AGE)

PURPOSE

Pages through a file 23 lines at a time.

SYNTAX

[line1] [,line2]P<CR>

COMMENTS

If the first line (line1) is omitted, that number defaults to the current line plus one. If the second line (line2) is omitted, 23 lines are listed. The new current line becomes the last line displayed and is marked with an asterisk.

Q(UIT)

PURPOSE

Quits the editing session, does not save any editing changes, and exits to the NCR-DOS operating system.

SYNTAX

Q<CR>

COMMENTS

EDLIN prompts you to make sure you don't want to save the changes.

Type Y if you want to quit the editing session. No editing changes are saved and no .BAK file is created. Refer to the End command in this chapter for information about the .BAK file.

Type N or any other character if you want to continue the editing session.

NOTE: When started, EDLIN erases any previous copy of the file with an extension of .BAK to make room to save the new copy. If you reply Y to the "(Y/N)?" message, your previous backup copy no longer exists.

EXAMPLE

```
              Q
              Abort edit (Y/N)? Y
    A> _
```

R(EPLACE)

PURPOSE

Replaces all occurrences of a string of text in the specified range with a different string of text.

SYNTAX

[line1] [,line2] [?]   R[<string1>] [<F6>string2]<CR>

COMMENTS

As each occurrence of string1 is found, it is replaced by string2. Each line in which a replacement occurs is displayed. If a line contains two or more replacements of string1 with string2, then the line is displayed once for each occurrence. When all occurrences of string1 in the specified range are replaced by string2, the R command terminates and the asterisk prompt returns.

If a second string is to be given as a replacement, then String1 must be separated from string2 with a "space". String2 may end with an <F6> <CR> combination or with a simple <CR>.

If string1 is omitted, then Replace takes the old string1 as its value. If there is no old string1 (that is, this is the first Search or Replace done), then the replacement process is terminated immediately. If string2 is omitted, then string1 may end with a <CR>. If the first line-number is omitted in the range argument by the use of a comma (as in ",line2") then line1 defaults to the line after the current line. If the second line-number (line2) is omitted (as in "line1" or "line1,"), the second line defaults to #. Remember that # indicates the line after the last line of the file.

If string1 ends with <F6> and there is no string2, string2 is taken as an empty string and becomes the new replace string. For example,

Rstring1<F6><CR>

deletes occurrences of string1, but

Rstring1<CR> and
R<CR>

replaces the new string1 with the old string2.

Note that "old" here refers to a previous string specified either in a Search or a Replace command.

If the question mark (?) option is given, the Replace command stops at each line with a string that matches string1, displays the line with string2 in place, and then displays the prompt "O.K.?." If you press Y or the <CR> key, then string2 replaces string1, and the next occurrence of string1 is found. Again, the "O.K.?" prompt is displayed. This process continues until the end of the range or until the end of the file. After the last occurrence of string1 is found, EDLIN displays the asterisk prompt.

If you press any key except Y or <CR> after the "O.K.?" prompt, the string1 is left as it was in the line, and Replace goes to the next occurrence of string1. In this way, only the desired string1 is replaced, and you can prevent unwanted substitutions.

EXAMPLES

Assume that the following file exists and is ready for editing:

```
 1: This is a sample file
 2: used to show dynamic line numbers.
 3: See what happens when you use
 4: Delete and Insert
 5: (the D and I commands)
 6: to edit text
 7: in your file.
 8: The insert command can place new lines
 9: in the file; there's no problem
10: because the line numbers are dynamic;
11: they'll go all the way to 65533.
```

To replace all occurrences of string1 (in this example, "and") with string2 ("or") in a specified range (lines 2 thru 12), type:

```
2,12 Rand<F6> or<CR>
```

The result is:

```
4: Delete or Insert
5: (the D or I commors)
8: The insert commor can place new lines
```

Note that in the replacements, some unwanted substitutions have occurred. To avoid these and to confirm each replacement, the same original file can be used with a slightly different command. In the next example, to replace only certain occurrences of the first string with the second string, type:

2,12?R and<Ctrl-Z> or <CR>

The result is:

```
4:  Delete or Insert
O.K.? Y
5:  (the D or I commands)
O.K.? Y
5:  (the D or I commors)
O.K.? N
8:  The insert commor can place new lines
O.K.? N
* _
```

Now, type the List command (L) to see the result of all these changes:

```
    .
4:  Delete or Insert
5:  (the D or I commands)
    .
    .
8:  The insert command can place new lines
    .
```

S(EARCH)

PURPOSE

Searches the specified range of lines for a specified string of text.

SYNTAX

[line1] [,line2] [?]Sstring<CR>

COMMENTS

The string must end with a <CR>. The first line that matches string is displayed and becomes the current line. If the question mark option is not specified, the Search command terminates when a match is found. If no line contains a match for string, the message "Not found" is displayed.

If the question mark option (?) is included in the command, EDLIN displays the first line with a matching string; it then prompts you with the message "O.K.?". If you press either the Y or <CR> key, the line becomes the current line and the search terminates. If you press any other key, the search continues until another match is found, or until all lines are searched (and the "Not found" message is displayed).

If the first line (line1) is omitted (as in, line2 S string), line1 defaults to the line after the current line. If line2 is omitted (as in "line1 S string" or "line, S string"), line2 defaults to the pound sign "#" (the line after the last line of the file), which is the same as line1, # S string. If string is omitted, Search takes the old string if there is one. (Note that "old" here refers to a string specified in a previous Search or Replace command.) If there is not an old string (that is, no previous search or replace has been done), the command terminates immediately.

EXAMPLES

Assume that the following file exists and is ready for editing:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
8: The insert command can place new lines
```

```
 9:  in the file: there's no problem
10:  because the line numbers are dynamic;
11:*they'll go all the way to 65533.
```

To search for the first occurrence of the string "and," type

```
2,12 Sand<CR>
```

The following line is displayed:

```
4:  Delete and Insert
```

to get the "and" in line 5, modify the search command by typing:

```
<Del><F3><CR>
```

The search then continues from the line after the current line (line 4), since no first line was given. The result is:

```
5:  (the D and I commands)
```

To search through several occurrences of a string until the correct string is found, type:

```
1, ?Sand<CR>
```

The result is:

```
4:  Delete and Insert
O.K.? _
```

If you press any key (except Y or <CR>), the search continues, so type N here:

```
O.K.? N
```

Continue:

```
5:  (the D and I commands)
O.K.? _
```

Now press Y to terminate the search:

```
O.K? Y
* _
```

To search for string XYZ without the verification (O.K.?), type:

SXYZ<CR>

EDLIN finds a match and continues to search for the same string when you issue the S command:

S<CR>

EDLIN will continue to search when you issue the S command until there are no more occurrences, which will bring about the message.

```
Not found
```

Note that <string> defaults to any string specified by a previous Replace or Search command.

T(RANSFER)

PURPOSE

Inserts (merges) the contents of <filename> into the file currently being edited at <line>. If <line> is omitted, then the current line is used.

SYNTAX

[line-number]Tfilename<CR>

COMMENTS

This command is useful if you want to put the contents of a file into another file or into the text you are typing. The transferred text is inserted ahead of the line number specified by "line-number" and the lines are renumbered after the inserted numbers.

W(RITE)

PURPOSE

Writes a specified number of lines to disk from the lines that are being edited in memory. Lines are written to disk beginning with line number 1.

SYNTAX

```
[line-number]W
```

COMMENTS

This command is meaningful only if the file you are editing is too large to fit into memory. When you start EDLIN, EDLIN reads lines into memory until memory is 3/4 full.

To edit the remainder of your file, you must write edited lines in memory to disk. Then you can load additional lines from disk into memory by using the Append command.

NOTE: If you do not specify the number of lines, lines are written until memory is 1/4 full. No action is taken if available memory is already less than 1/4 full. All lines are renumbered, so that the first remaining line becomes line number 1.

ERROR MESSAGES

When EDLIN finds an error, one of the following error messages is displayed:

Cannot edit .BAK file- -rename file

Explanation

You attempted to edit a file with a filename extension of .BAK! .BAK files cannot be edited because this extension is reserved for backup copies.

Action

If you need the .BAK file for editing purposes, you must either RENAME the file with a different extension, or COPY the .BAK file and give it a different filename extension.

No room in directory for file

Explanation

When you attempted to create a new file, either the file directory was full or you specified an illegal disk drive or an illegal filename.

Action

Check the command line that started EDLIN for illegal filename and illegal disk drive entries. If the command is no longer on the screen and if you have not yet typed a new command, the EDLIN start command can be recovered by pressing the <F3> key.

If this command line contains no illegal entries, run the CHKDSK program for the specified disk drive. If the status report shows that the disk directory is full, remove the disk. Insert and format a new disk.

Entry Error

Explanation

The last command typed contained a syntax error.

Action

Retype the command with the correct syntax and press <CR>.

Line too long

Explanation

During a Replace command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. EDLIN aborted the Replace command.

Action

Divide the long line into two lines; then try the Replace command twice.

Disk Full- -file write not completed

Explanation

You gave the End command, but the disk did not contain enough free space for the whole file. EDLIN aborted the E command and returned you to the operating system. Some of the file may have been written to the disk.

Action

Only a portion (if any) of the file has been saved. You should probably delete that portion of the file and restart the editing session. The file is not available after this error. Always be sure that the disk has sufficient free space for the file to be written to disk before you begin your editing session.

Incorrect DOS version

Explanation

You attempted to run EDLIN under a version of DOS that was not 2.1 or higher.

Action

You must make sure that the version of DOS that you are using is 2.1 or higher.

Invalid drive name or file

Explanation

You have not specified a valid drive or filename when starting EDLIN.

Action

Specify the correct drive or filename.

Filename must be specified

Explanation

You did not specify a filename when you started EDLIN.

Action

Specify a filename.

Invalid parameter

Explanation

You specified a switch other than /B when starting EDLIN.

Action

Specify the /B switch when you start EDLIN.

Insufficient memory

Explanation

There is not enough memory to run EDLIN.

Action

You must free some memory by writing files to disk or by deleting files before restarting EDLIN.

File not found

Explanation

The filename specified during a Transfer command was not found.

Action

Specify a valid filename when issuing a Transfer command.

Must specify destination number

Explanation

A destination line number was not specified for a Copy or Move command.

Action

Reissue the command with a destination line number.

Not enough room to merge the entire file

Explanation

There was not enough room in memory to hold the file during a Transfer command.

Action

You must free some memory by writing some lines to disk.

Link Program (NCR-LINK)

GENERAL INFORMATION

In this chapter you learn about NCR-LINK. Read the entire chapter before you use LINK.

NOTE: If you are not going to compile and link programs, you do not need to read this chapter.

LINK is a program that performs the following functions:

- Combines separately produced object modules into one relocatable load module — a program you can run.
- Searches library files for definitions of unresolved external references.
- Resolves external cross-references.
- Produces a listing that shows both the resolution of external references and error messages.

PROGRAM OVERVIEW

When you write a program, you write it in source code. This source code is passed through a compiler or an assembler, which produces object modules. The object modules must be passed through the link process to produce an executable module that the computer can understand directly.

You may wish to link (combine) several programs and run them together. Any of your programs may refer to one or more symbols that are defined in other modules. These references are called external references.

LINK combines several object modules into one relocatable load module, or Run file (called an .EXE or Executable file). As it combines modules, LINK makes sure that all external references between object modules are defined. LINK can search several library files for definitions of any external references that are not defined in the object modules.

LINK also produces a List file that shows external references resolved, and it displays all error messages.

LINK uses available memory as much as possible. When available memory is exhausted, LINK creates a temporary disk file named VM.TMP.

Figure 7.1 illustrates the various parts of the LINK operation.

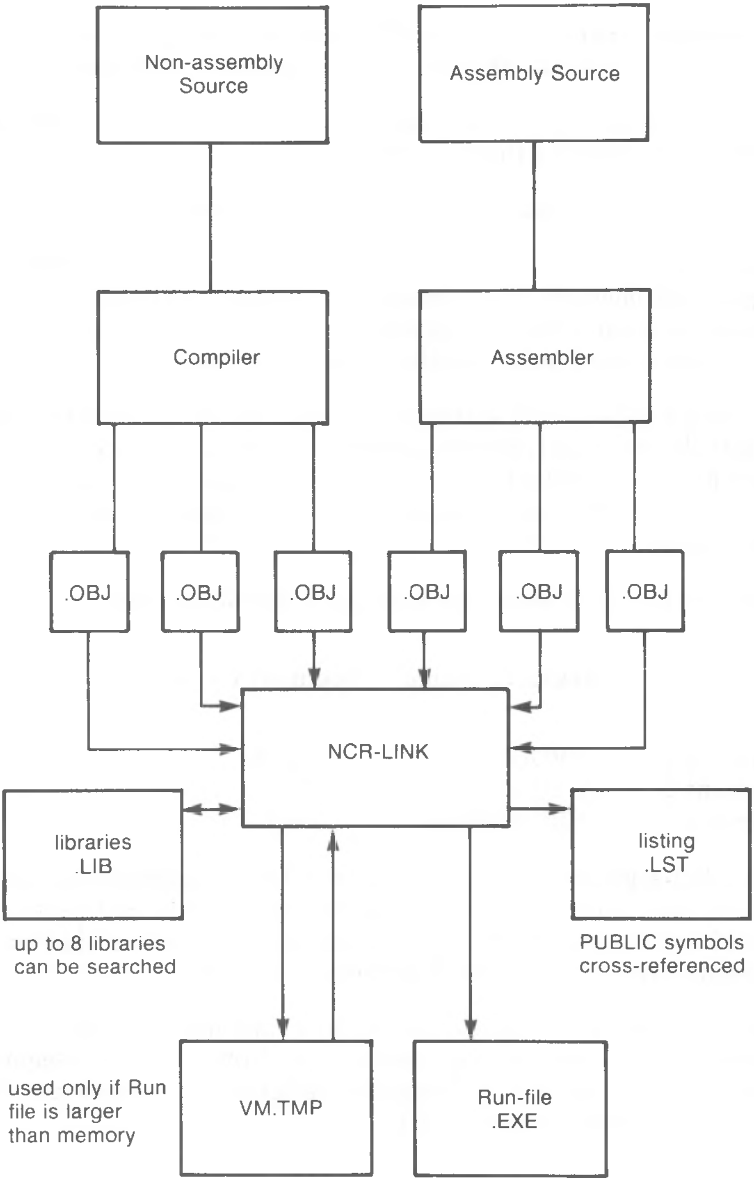


Figure 7.1 The LINK Operation

DEFINITIONS YOU'LL NEED TO KNOW

Some of the terms used in this chapter are explained below to help you understand how LINK works. Generally, if you are linking object modules compiled from BASIC, Pascal, or another high-level language, you do not need to know these terms. If you are writing and assembling programs in assembly language, however, you need to understand LINK and the definitions described in this section.

In DOS, memory can be divided into segments, classes, and groups. Figure 7.2 illustrates these concepts.

A *segment* is a contiguous area of memory up to 64K bytes in length.

A *group* is a collection of segments that fit together within a 64K-byte segment of memory. The segments are named to the group by the assembler or compiler. A program may consist of one or more groups. The group is used for addressing segments in memory.

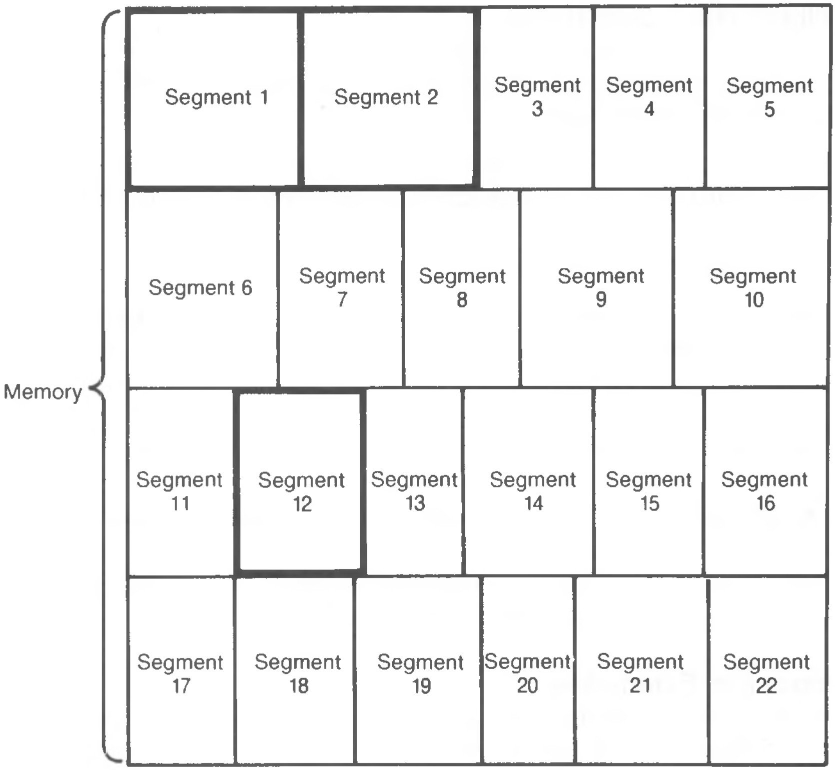
A *class* is a collection of segments. The naming of segments to a class affects the order and relative placement of segments in memory. The class name is specified by the assembler or compiler. All portions assigned to the same class name are loaded into memory contiguously.

Assume that three segments have the following names:

	Segment Name	**Segment Class**
Segment 1	PROG1.OBJ	CODE
Segment 2	PROG2.OBJ	CODE
Segment 12	PROG12.OBJ	DATA

Note that segments 1, 2, and 12 have different segment names but may or may not have the same segment class name. Segments 1, 2, and 12 form a group with a group address of the lowest address of segment 1 (that is, the lowest address in memory).

Each segment has a segment name and a class name. LINK loads all segments into memory by class name from the first segment encountered to the last. All segments assigned to the same class are loaded into memory contiguously.



Highlighted area = a group (64 K bytes addressable)

Figure 7.2 How memory is divided

During processing, LINK references segments by their addresses in memory (where they are located). LINK does this by finding groups of segments.

The segments do not need to be contiguous to form a group (see Figure 7.2). The address of any group is the lowest address of the segments in that group. At link time, LINK analyzes the groups and then references the segments by the address in memory of that group. A program may consist of one or more groups.

If you are writing in assembly language, you may assign the group and class names in your program. In high-level languages (BASIC, COBOL, FORTRAN, Pascal), the naming is done automatically by the compiler.

FILES THAT LINK USES

LINK works with one or more input files, produces one or two output files, may create a temporary disk file, and may be directed to search up to eight library files.

For each type of file, you may give a 4-part file specification. The format for LINK file specifications is the same as that of a disk file:

```
[d:][pathname] filename [.ext]
```

- d: is the drive designation. Permissible drive designations for LINK are A: through O: The colon is always required as part of the drive designation.
- pathname is the directory path in which the file is located.
- filename is any legal filename of one to eight characters.
- .ext is a 1- to 3-character extension to the filename. The period is always required as part of the extension.

Input File Extensions

If no filename extensions are given in the input (object) file specifications, LINK appends the following extensions by default:

.OBJ	Object
.LIB	Library

Output File Extensions

LINK appends the following default extensions to the output (Run and List) files:

.EXE	Run (may not be overridden)
.MAP	List (may be overridden)

VM.TMP (Temporary) File

LINK uses available memory for the link session. If the files to be linked create an output file that exceeds available memory, LINK creates a temporary file, names it VM.TMP, and puts it on the disk in the default drive. If LINK creates VM.TMP, it displays the message:

```
VM.TMP has been created.
Do not change disk in drive, <d:>
```

Once this message has been displayed, you must not remove the disk from the default drive until the link session ends. If the disk is removed, the operation of LINK will be unpredictable, and LINK might display the error message:

```
Unexpected end of file on VM.TMP
```

The contents of VM.TMP are written to the file named following the Run File: prompt. VM.TMP is a working file only and is deleted at the end of the linking session.

CAUTION

> Do not use VM.TMP as a filename for any file. If you have a file named VM.TMP on the default drive and LINK requires the VM.TMP file, LINK deletes the VM.TMP already on the disk and creates a new VM.TMP. Thus, the contents of the previous VM.TMP file will be lost.

USING LINK

STARTING LINK

LINK requires two types of input: a command to start LINK and responses to command prompts. In addition, six options control LINK features. Usually, you type all the commands to LINK on the terminal keyboard. As an option, answers to the command prompts and any options may be contained in a response file. Three special command characters [plus sign(+), semicolon(;), and at sign (@)] can be used to modify the sequence of the commands you give to LINK.

You may start LINK in any of three ways. The first method is to type the commands in response to individual prompts. In the second method, you type all commands on the line used to start LINK. To start LINK by the third method, you must create a response file that contains all the necessary commands and tell LINK where that file is when you start LINK.

Method 1	**LINK**
Method 2	**LINK filenames [/options]**
Method 3	**LINK @ filespec**

Method 1: Prompts

To start LINK with method 1, type:

LINK

when the DOS diskette is loaded

LINK is loaded into memory, and then LINK displays four text prompts that appear one at a time. You answer the prompts to tell LINK to perform specific tasks.

At the end of each line, you may type one or more options, preceded by the parameter character (in this case, a forward slash).

The command prompts are summarized in the following table and are described in more detail in the "Command Prompts" section.

PROMPT	RESPONSES
Object Modules [.OBJ]:	List .OBJ files to be linked. They must be separated by spaces or plus signs (+). If a plus sign is the last character typed, the prompt reappears. There is no default; a response is required.
Run File [Object-file.EXE]:	Give filename for executable object code. The default is first-object-filename.EXE. (You cannot change the output extension.)
List File [Run-file.MAP]:	Give filename for listing. The default is no listing. (You cannot change the output syntax.)
Libraries [ ]:	List filenames to be searched, separated by spaces or plus signs (+). If a plus sign is the last character typed, the prompt reappears. The default is an automatic library search using the .LIB extension.

Method 2: Command Line

To start LINK using method 2, type all commands on one line. The entries following LINK are responses to the command prompts. The entry fields for the different prompts must be separated by commas. Use the following syntax:

LINK object-list, runfile, listfile, lib-list [/option...]

- object-list is a list of object modules, separated by plus signs.
- runfile is the name of the file to receive the executable output.
- listfile is the name of the file to receive the listing.
- lib-list is a list of library modules to be searched.
- /option refers to optional options, which may be placed following any of the response entries (just before any of the commas or after the 'lib-list', as shown).

To select the default for a field, simply type a second comma with no spaces between the two commas.

LINK PROCESS+TEXT+TABLE+CALC/P/M,,
PROCLIST, COBLIB.LIB

This command causes LINK to be loaded, followed by the object modules PROCESS.OBJ, TEXT.OBJ, TABLE.OBJ, and CALC.OBJ. LINK then pauses (as a result of using /P option). LINK links the object modules when you press any key, and produces a universal symbol map (the /M option); defaults to PROCESS.EXE run file; creates a list file named PROCLIST.MAP; and searches the library file COBLIB.LIB.

Method 3: Response File

To start LINK with method 3, type:

LINK @ filespec

where filespec is the name of a response file. A response file contains answers to the LINK prompts (shown in method 1) and may also contain any of the options. When naming a response file, the use of filename extensions is optional. Method 3 permits the command that starts LINK to be entered from the keyboard or within a batch file without requiring you to take any further action.

To use this option, you must create a response file containing several lines of text, each of which is the response to a LINK prompt. The responses must be in the same order as the LINK prompts discussed in method 1. If desired, a long response to the "Object Modules:" or "Libraries:" prompt may be typed on several lines by using a plus sign (+) to continue the same response onto the next line.

Use options and command characters in the response file the same way as they are used for responses typed on the terminal keyboard.

When the LINK session begins, each prompt is displayed in order with the responses from the response file. If the response file does not contain answers for all the prompts (in the form of filenames, the semicolon command character, or <CR>), LINK displays the prompt that does not have a response and then waits for you to type a legal response. When a legal response is typed, LINK continues the link session. Consider the following example:

```
PROCESS TEXT TABLE CALC
/PAUSE/MAP
PROCLIST
COBLIB.LIB
```

This response file tells LINK to load the four object modules named PROCESS, TEXT, TABLE, and CALC. LINK pauses before producing a public symbol map to permit you to swap disks (see discussion under /PAUSE in the options section before using this feature). When you press any key, the output files are named PROCESS.EXE and PROCLIST.MAP. LINK searches the library file COBLIB.LIB and uses the default setting for the options.

COMMAND CHARACTERS

LINK provides three command characters.

Plus sign

Use the plus sign (+) to separate entries and to extend the current line in response to the “Object Modules:” and “Libraries:” prompts. (A space may be used to separate object modules.) To type a large number of responses (each of which may be very long), type a plus sign and a <CR> at the end of the line to extend it. If the plus sign and <CR> is the last entry following either of these two prompts, LINK prompts you for more module or library names. When the “Object Modules:” or “Libraries:” prompt appears again, continue to type responses. When all the modules to be linked and libraries to be searched are listed be sure the response line ends with a module name and a <CR> and not a plus sign and <CR>.

Example:

```
Object Modules [.OBJ]: PROCESS TEXT TABLE
CALC+<CR>
```

Object Modules [.OBJ]: **READDATA+VALDATE+<CR>**
Object Modules [.OBJ]: **BILDPRNT+<CR>**
Object Modules [.OBJ]: **PIECHART**

Semicolon

To select default responses to the remaining prompts, use a single semicolon (;) followed immediately by a <CR> at any time after the first prompt (Run File:). This feature saves time and overrides the need to press a series of <CR> keys.

NOTE: Once the semicolon has been typed and entered (by pressing the <CR> key), you can no longer respond to any of the prompts for that link session. Therefore, do not use the semicolon to skip some prompts. To skip prompts, use the <CR> key.

Example:

Object Modules [.OBJ]: **PROCESS TEXT TABLE CALC<CR>**
Run Module [PROCESS.EXT]: **;<CR>**

No other prompts appear, and LINK uses the default values (including PROCESS.MAP for the list file).

Ctrl-Break

Use the <Ctrl-Break> keys to cancel the link session at any time. If you type an erroneous response, such as the wrong filename or an incorrectly spelled filename, you must press <Ctrl-Break> to exit LINK, and then restart LINK. However, if you typed the error but did not press the <CR> key, you may delete the erroneous characters with the backspace key, but for that line only.

COMMAND PROMPTS

LINK asks you for responses to four text prompts. When you type a response to a prompt and press <CR>, the next prompt appears. When the last prompt is answered, LINK begins linking automatically without further command. When the link session is finished, LINK exits to the operating system. When the operating system prompt appears, LINK has finished successfully. If the link session is unsuccessful, LINK displays the appropriate error message.

LINK prompts the user for the names of Object, Run, and List files, and for Libraries. The prompts are listed in order of appearance. The default response is shown in square brackets ([]) following the prompt, for prompts which can default to preset responses. The "Object Modules:" prompt, however, has no preset filename response and requires you to type a filename.

Object Modules [.OBJ]:

Type a list of the object modules to be linked. LINK assumes by default that the filename extension is .OBJ. If an object module has any other filename extension, the extension must be given. Otherwise, the extension may be omitted.

Modules must be separated by plus signs (+) or spaces.

Remember that LINK loads segments into classes in the order encountered. You can use this information to set the order in which the object modules are read by LINK.

Run File [First-Object-filename.EXE]:

Typing a filename creates a file for storing the Run (executable) file that results from the link session. All Run files receive the filename extension .EXE, even if you specify an extension other than .EXE.

If no response is typed to the "Run File:" prompt, LINK uses the first filename typed in response to the "Object Modules:" prompt as the RUN filename.

Example:

Run File [PROCESS.EXE]: **B:PAYROLL/P**

This response directs LINK to create the Run file PAYROLL.EXE on drive B:. The /P causes link to pause, which allows you to insert a new disk to receive the Run file.

List File [Run-Filename.MAP]:

The list file contains an entry for each segment in the input (object) modules. Each entry shows the addressing in the Run file.

The default response is the Run filename with the default filename extension .MAP.

Libraries [ ]:

The valid responses are up to eight library filenames or simply a <CR>. (A <CR> means no library search.) Library files must have been created by a library utility.

LINK assumes by default that the filename extension is .LIB for library files.

Library filenames must be separated by spaces or plus signs (+).

LINK searches library files in the order listed to resolve external references. When it finds the module that defines the external symbol, LINK processes that module as another object module.

If LINK cannot find a library file on the disk(s) currently in the disk drive(s), it displays the message:

```
Cannot find library <d:library-name>
Enter new drive letter:
```

Press the letter for the drive designation (for example, B).

LINK OPTIONS

The six LINK options control various LINK functions. Options must be typed at the end of a prompt response, regardless of the method used to start LINK. Options may be grouped at the end of any response, or may be scattered at the end of several. If more than one option is typed at the end of one response, each option must be preceded by a forward slash (/).

All options may be abbreviated. The only restriction is that an abbreviation must be sequential from the first letter through the last typed; no gaps or transpositions are allowed. For example, examine the following lists of valid and invalid abbreviations.

Legal	**Illegal**
/D	/DSL
/DS	/DAL
/DSA	/DLC
/DSALLOCA	/DSALLOCT
/DSALLOCATE	/DSALLOCATE

DISALLOCATE

Using the /DSALLOCATE option tells LINK to load all data at the high end of the Data Segment. Otherwise, LINK loads all data at the low end of the Data Segment. At runtime, the DS pointer is set to the lowest possible address to allow the entire DS segment to be used. Use of the /DSALLOCATE option in combination with the /HIGH option permits the user application to dynamically allocate any available memory below the area specifically allocated within DGroup, yet to remain addressable by the same DS pointer. This dynamic allocation is needed for Pascal and FORTRAN programs.

NOTE: Your application program may dynamically allocate up to 64K bytes (or the actual amount of memory available) less the amount allocated within DGroup.

/HIGH

Use of the /HIGH option causes LINK to place the Run file as high as possible in memory. Otherwise, LINK places the Run file as low as possible.

CAUTION

Do not use the /HIGH option with Pascal or FORTRAN programs.

/LINE

The /LINE option tells LINK to include in the List file the line numbers and addresses of the source statements in the input modules. Otherwise, line numbers are not included in the List file.

NOTE: Not all compilers produce object modules that contain line number information. In these cases, of course, LINK cannot include line numbers.

/MAP

/MAP directs LINK to list all public (universal) symbols defined in the input modules. If /MAP is not given, LINK lists only errors (including undefined universals).

The symbols are listed alphabetically. For each symbol, LINK lists its value and its segment offset location in the Run file. The symbols are listed at the end of the List file.

/PAUSE

The /PAUSE option causes LINK to pause in the link session when the option is encountered. Normally, LINK performs the linking session from beginning to end without stopping. This option allows the user to swap the disks before Link outputs the Run (.EXE) file.

When LINK encounters the /PAUSE option, it displays the message:

```
About to generate .EXE file
Press any key
```

LINK resumes processing when the user presses the <CR> key.

CAUTION

Do not remove the disk that receives the List file, or the disk used for the VM.TMP file, if one has been created.

/STACK:<number>

The number entry represents any positive numeric value (in hexadecimal radix) up to 65536 bytes. If a value from 1 to 511 is typed, LINK uses 512. If the /STACK option is not used for a link session, LINK calculates the necessary stack size automatically.

All compilers and assemblers should provide information in the object modules that allow the linker to compute the required stack size.

At least one object (input) module must contain a stack allocation statement. If not, LINK displays the following error message:

```
WARNING: NO STACK STATEMENT
```

SAMPLE LINK SESSION

This sample shows you the type of information that is displayed during an LINK session.

In response to the DOS prompt, type:

LINK

The system displays a copyright message followed by the prompts (your answers are in bold type):

Object Modules [.OBJ]: **NCRIO SYSINIT**
Run File [NCRIO.EXE]: **/MAP**
List File [NUL.MAP]: **PRN/LINE**
Libraries [.LIB]: **;<CR>**

Consider how your answers direct LINK and how others affect the output:

- By specifying /MAP, you get both an alphabetic listing and a chronological listing of public symbols.

- By responding PRN to the "List File:" prompt, you can redirect your output to the printer.

- By specifying the /LINE option, LINK gives you a listing of all line numbers for all modules. (Note that the /LINE option can generate a large volume of output.)

- By pressing <CR> in response to the "Libraries:" prompt, and automatic library search is performed.

Once LINK locates all libraries, the linker map displays a list of segments in the order of their appearance within the load module. The list might look like this:

```
Start     Stop      Length    Name      Class

00000H    009ECH    09EDH     PROCESS   CODE
009F0H    01166H    0777H     SYSINIT   CODE
```

The information in the Start and Stop columns shows the 20-bit hex address of each segment relative to location zero. Location zero is the beginning of the load module.

The addresses displayed are not the absolute addresses where these segments are loaded. Consult the *NCR-DOS PROGRAMMER'S MANUAL* for information on how to determine where relative zero is actually located, and also on how to determine the absolute address of a segment.

Because the /MAP option was used, LINK displays the public symbols by name and value. For example:

```
ADDRESS       PUBLICS BY NAME

009F:0012     BUFFERS
009F:0005     CURRLOC (Current DOS Location)
009F:0011     DEFDRV (Default Drive)
009F:000B     DEVLIST (Device List)
009F:0013     FILES
009F:0009     FINALLOC (Final DOS Location)
009F:000F     MEMSIZE (Memory Size)
009F:0000     SYSINIT

ADDRESS       PUBLICS BY VALUE

009F:0000     SYSINIT
009F:0005     CURRLOC
```

```
009F:0009     FINALLOC
009F:000B     DEVLIST
009F:000F     MEMSIZE
009F:0011     DEVDRV
009F:0012     BUFFERS
009f:0013     FILES
```

ERROR MESSAGES

All errors cause the link session to abort. After you find the cause of the error and correct it, rerun LINK. The following error messages are displayed by LINK; they are mostly self- explanatory.

ATTEMPT TO ACCESS DATA OUTSIDE OF SEGMENT BOUNDS

There is probably a bad object file.

BAD NUMERIC PARAMETER

Numeric value entered after /STACK is not in digits.

CANNOT OPEN TEMPORARY FILE

LINK is unable to create the file VM.TMP because the disk directory is full. Insert a new disk. Do not remove the disk that will receive the List.MAP file.

DUP RECORD TOO COMPLEX

DUP record in assembly language module is too complex. Eliminate some structures or DUP records in assembly language program, reassemble, and rerun LINK.

FIXUP OFFSET EXCEEDS FIELD WIDTH

An assembly language instruction refers to an address with a short instruction instead of a long instruction. Edit assemble language source and reassemble.

INPUT FILE READ ERROR

There is probably a bad object file.

INVALID OBJECT MODULE

An object module(s) is incorrectly formed or incomplete (as when assembly is stopped in the middle).

SYMBOL DEFINED MORE THAN ONCE

LINK found two or more modules that define a single symbol name.

PROGRAM SIZE EXCEEDS
CAPACITY OF LINK

The total size may not exceed 384K bytes and the number of segments may not exceed 255.

REQUESTED STACK SIZE EXCEEDS 64K

Specify a size less than or equal to 64K bytes with the /STACK option.

SEGMENT SIZE EXCEEDS 64K

64K bytes is the addressing system limit.

SYMBOL TABLE CAPACITY EXCEEDED

Too many and/or very long names were typed, exceeding the limit of approximately 50K bytes.

TOO MANY EXTERNAL SYMBOLS IN ONE MODULE

The limit is 256 external symbols per module.

TOO MANY GROUPS

The limit is 10 groups.

TOO MANY LIBRARIES SPECIFIED

The limit is 8 libraries.

TOO MANY PUBLIC SYMBOLS

The limit is 1024 public symbols.

TOO MANY SEGMENTS OR CLASSES

The limit is 247 (segments and classes taken together).

UNRESOLVED EXTERNALS: <symbol(s)>

The external symbols listed have no defining module among the modules or library files specified.

VM READ ERROR

This is a disk error; it is not caused by LINK.

WARNING: NO STACK SEGMENT

None of the object modules specified contains a statement allocating stack space, but the user typed the /STACK option.

WARNING: SEGMENT OF ABSOLUTE OR UNKNOWN TYPE
There is a bad object module or an attempt has been made to line modules that LINK cannot handle (for example, an absolute module).

OUT OF SPACE ON VM.TMP
No more disk space remains to expand VM.TMP file.

OUT OF SPACE ON RUN FILE
There is not enough disk space for the Run file.

Advanced Configuration Feature

In many cases, there are installation-specific settings for NCR-DOS that need to be configured at system startup. An example of this is a standard device driver, such as an online printer.

The NCR-DOS configuration file (CONFIG.SYS) allows you to configure your system with a minimum of effort. With this file, you can add device drivers to your system at startup. The configuration file is simply an ASCII file that has certain commands for NCR-DOS startup (boot). The boot process is as follows:

1. The disk boot sector is read. This contains enough code to read NCR-DOS code and the installation's BIOS machine-dependent code .

2. The NCR-DOS code and BIOS are read.

3. A variety of BIOS initializations are done.

4. A system initialization routine reads the configuration file (CONFIG.SYS), if it exists, to perform device installation and other user options. Its final task is to execute the command interpreter, which finishes the NCR-DOS boot process.

CHANGING THE CONFIG.SYS FILE

If there is not a CONFIG.SYS file on the NCR-DOS disk, you can use the NCR-DOS editor, EDLIN, to create a file; then save it on the NCR-DOS disk in your root directory.

The following is a list of commands for the configuration file CONFIG.SYS:

BUFFERS=<number>

Where the number is between 1 and 99. This is the number of sector buffers that NCR-DOS should allocate in memory when it starts up. It is installation-dependent. If not set, 10 is a reasonable number. The default value is 2.

BUFFERS=<number>

Where the number is between 1 and 99. This is the number of sector buffers that NCR-DOS should allocate in memory when it starts up. It is installation-dependent. If not set, 10 is a reasonable number. The default value is 2.

FILES=<number>

Where the number is between 1 and 99. This is the number of open files that the XENIX system calls can access. It is installation-dependent. If not set, 10 is a reasonable number. The default value is 8.

DEVICE=<filename>

This installs the device driver in <filename> into the system list. (See below.)

BREAK=<ON or OFF>

If ON is specified (the default is OFF), a check for <Cntrl-C> as input will be made every time the system is called. ON improves the ability to abort programs over previous versions of the NCR-DOS. BREAK ON/OFF may be changed by issuing a BREAK command (see Chapter 4).

SHELL=<filename>

This begins execution of the shell (top-level command processor) from <filename>.

A typical configuration file might look like this:

```
Buffers= 10
Files = 10
Device = \BIN\NETWORK.SYS
Break = ON
Shell = A:\BIN\COMMAND.COM A:\BIN/P
```

Note here that the Buffers and Files parameters are set to 10. The system initialization routine will search for the filename \BIN\NETWORK.SYS to find the device that is being added to the system. This file is usually supplied on disk with your device. Make sure that you save the device file in the pathname that you specify in the Device parameter.

This configuration file also sets the NCR-DOS command EXEC to the COMMAND.COM file located on disk A: in the \BIN directory. The A:\BIN tells COMMAND.COM where to look for itself when it needs to be re-read from disk. The /P tells COMMAND.COM that it is the first program running on the system so that it can process the NCR-DOS EXIT command.

How to Obtain and Install Software

MAKING THE RIGHT PURCHASE

Your NCR PERSONAL COMPUTER MODEL with NCR-DOS is highly flexible. It can accommodate all NCR-DOS compatible application and language software, as well as virtually all IBM-PC-DOS compatible software. Think of the software that you can use as being in one of three categories:

- Software distributed by NCR for the NCR Personal Computer
- Third-Party Software (available from your supplier) that has been tested by NCR for use on the NCR Personal Computer Model
- Third-Party Software that has not been tested for use on the NCR Personal Computer

Packages in the first category can be obtained directly from NCR through your NCR representative or licensed dealer. Software in the other categories can be purchased from your dealer or any reputable software supplier. You will want to be sure, however, that the software will run on your computer. Knowing what questions to ask and working with a knowledgeable dealer will help you to make the right decision.

1. Ask if the software is on an NCR-DOS formatted disk.

2. If the software is not available on an NCR-DOS formatted disk, ask if it is available on an IBM PC-DOS compatible disk. (Your system can use a 5 1/4-inch disk with either 160/180/320/360KB capacity.)

3. Most application packages and some language software require an installation program to interface with the specific hardware. Ask if the software is readily installable for use on an NCR PC or

if the package includes an installation (or install) program so that you can tailor the software yourself. If the software is intended for use on an NCR PC or an IBM PC, it can be easily installed on your computer.

INSTALLING THE SOFTWARE

To fully use every feature on NCR Personal Computer, each application must be tailored to, or installed on, your computer. The installation procedure is different for each application, so every "off the shelf" package contains its own program for accomplishing this task. The documentation that accompanies the application disk you purchase describes the way this program is run. We recommend that before you customize your application, you make a copy of the purchased disk and use the copy as your work disk. (See the FORMAT and DISKCOPY commands in Chapter 4.) Save the original disk in a safe place and use it only to make copies.

To begin, most installation programs display a list of computer terminals. If this list includes the NCR Personal Computer, select it; otherwise, look for a compatibility with other vendors, such as Chamelion, Columbia VP, Compaq, Hyperion, etc.

Your keyboard has distinctive functional capabilities to help you operate third-party software and application programs. Table 1 gives you a complete list of these keyboard functions. Other tables include information you may need, depending on the particular application. Table 2 contains keyboard characteristics, and Table 3 is an ASCII code chart included for convenient reference.

KEYBOARD FUNCTIONS

	KEYS	DESCRIPTION
	F1	Copies one designated character.
	F2	Copy up to designated character.
	F3	Copies (remainder of) a line.
	F4	Skips up to designated character.
	F5	Moves cursor down one line to column 1.
	F6	End-of-file Marker (Ctrl-Z).
	F7-10	Programmable Function keys.
Shift	F1-10	Programmable function keys F11-F20.
Ctrl	F1-10	Programmable function keys F21-F30.

Alt	F1-10	Programmable function keys F31-F40.
	Esc	Cancel current line. Also moves to main menu in certain applications.
Shift	¦<-	Tab to left, (handled by application).
	->¦	Tab to right.
Ctrl	C	Aborts current command.
Ctrl	H	Removes last character from command line.
Ctrl	J	Inserts physical end-of-line.
Ctrl	P	Prints/Ends print of terminal output.
Ctrl	S	Suspends display of output to console.
Ctrl	BREAK	Interrupts operation.
Ctrl	End	Deletes remainder of line.
Ctrl	Home	Clears screen; returns cursor to home.
Ctrl	NumLock	Suspends operations.
Ctrl	PgDn	Deletes remainder of screen.
Ctrl	PgUp	Goes to top of file in text-editors.
Ctrl	PrtSc	Prints/Ends print of screen.
Ctrl	>	Scrolls right.
Ctrl	<	Scrolls left.
	<-	Back cursor up one space and delete character occupying space.
	↵	Move cursor to start of next line. (<CR>-key)
	PrtSc	Print screen contents.
	Ins	Enter/exit insert character mode.
	Del	Deletes a character within text editor.
	PgUp	Scroll up 25 lines.
	PgDn	Scroll down 25 lines.
	↑	Move the cursor up one line.
	>	Move the cursor to the right one character.
	V	Move the cursor down one line.
	<	Move the cursor to the left one character.
	Home	Returns cursor to line 1,column 1. Also, returns to main menu in some applications
	End	Go to end of line/text in text editors.
Scroll-Lock ↑		Scroll up one screen.
Scroll-Lock >		Scroll right one screen.
Scroll-Lock V		Scroll down one screen.
Scroll-Lock <		Scroll left one screen.

The following table shows the keyboard characteristics of the NCR Personal Computer.

KEYBOARD INFORMATION — Table 2

Number of rows	24 (1-24)
Number of columns	80 (1-80)
Cursor origin	1/1
Input/Output technique	NCR-DOS calls and commands (e.g. TYPE)
Cursor on/off	not active
Keyboard click on/off	not active

ASCII CODE CHART — Table 3

NCR ASCII CODE CHART

B_4-B_1 →		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
B_8 ↓ B_5		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0000	0	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
0001	1	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	US
0010	2	⍁	!	"	#	$	%	&	'	(	)	•	+	,	–	.	/
0011	3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
0100	4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
0101	5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
0110	6	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
0111	7	p	q	r	s	t	u	v	w	x	y	z	{	¦	}	~	DEL

ANSI ESCAPE SEQUENCES

ANSI escape sequences are processed by the ANSI driver which is provided on your NCR-DOS disk. The filename of this driver is ANSI.SYS. Use the line editor EDLIN or the COPY CON: command to create CONFIG.SYS and add the line

```
DEVICE = ANSI.SYS
```

Now you can send the ANSI escape sequences listed below to the CRT driver.

The syntax of ANSI sequences is as follows:

```
ESC [{[=]|[?]}{<#>|“<string>”}[;...]<end-of-sequence sign>
```

For further information refer to Chapter 4, the ESC command, and Chapter 5, Programming of Function Keys.

The ESC sequences listed must be preceded by: ESC [.

Function	ESC sequence

Unless specified in <rows#> or <columns#>, the number of rows and columns defaults to 1.

Cursor Positioning	<rows#>;<col.#> {H¦f}
Cursor Up	<rows#> A
Cursor Down	<rows#> B
Cursor Right	<col.#> C
Cursor Left	<col.#> D
Cursor Status Report	6n (result: [<rows#>;<col.#>R
Save Cursor Position	s
Restore Cursor Position	u
Erase Screen	2J
Erase to End of Line	K

Function	ESC sequence
Switch Graphic Attributes OFF	0m (or m)
BOLD ON (high intensity; black and white)	1m
UNDERSCORE ON (Monochrome and Alpha adapters only	4m
BLINKING ON	5m
INVERSE VIDEO ON	7m
CANCELLED ON (invisible)	8m
FOREGROUND	
Black	30m
Red	31m
Green	32m
Yellow	33m
Blue	34m
Magenta	35m
Cyan	36m
White	37m
BACKGROUND	
Black	40m
Red	41m
Green	42m
Yellow	43m
Blue	44m
Magenta	45m
Cyan	46m
White	47m
Set Screen Mode	{[=]¦[?]}<Functions# >{h¦l}

Mode:	Functions#
40x25 black and white	0 (or blank)
40x25 color	1
80x25 black and white	2
80x25 color	3

Function	ESC sequence
320x200 color	4
320x200 black and white	5
640x200 black and white	6
Wrap at end-of-line ON (typing past-end-of-line results in new line)	ESC [?7h
Wrap at end-of-line OFF (characters past end of line are lost	ESC [?7l

Messages

This appendix provides detailed information about NCR-DOS messages. It is organized into two sections, **device errors** and **general messages**. The first section describes the errors that may occur while reading or writing to devices on your system, the second section deals with NCR-DOS messages in general.

The messages are listed in alphabetical order. They are followed by an explanation, which begins with the program or command that generated the message.

DEVICE ERROR MESSAGES

If a device error occurs at any time during a command or program, DOS retries the operation three times. If the operation cannot be completed successfully, DOS returns an error message in the following format:

```
<yyy> ERROR WHILE <I/O-action> <TYPE>
Abort, Ignore, Retry:_
```

In this message, <yyy> may be one of the following:

- Bad call format
- Bad command
- Bad unit
- Data
- Disk
- Non-DOS disk
- No paper
- Not ready
- Read fault
- Sector not found
- Seek

- Write fault
- Write protect

The <I/O-action> may be either of the following:

- Reading
- Writing

<TYPE> is the name of the device in error, such as PRN, or else, represents the drive and the drive designation in which the error occurred, such as DRIVE B:.

DOS waits for you to enter one of the following responses:

A (Abort)
Terminate the program requesting the disk read or write.
I (Ignore)
Ignore the bad sector and pretend the error did not occur.
R (Retry)
Repeat the operation. Use this response when the error is corrected (such as with NOT READY or WRITE PROTECT errors.)

Usually, you will want to attempt recovery by entering responses in this order:

R (to try again)
A (to terminate the program and try a new disk)

EXPLANATIONS

Bad call format

A device driver has been passed an incorrect length request header. If this occurs, contact the dealer you purchased the device driver from.

Bad command

You have issued an invalid command to <device>.

Bad unit

A device driver has been passed an invalid sub-unit number. If this occurs, contact the dealer you purchased the device driver from.

Data

Owing to a defective spot on the disk, the data could not be read or written correctly.

Disk

An error other than the ones listed above has occurred.

Non-DOS disk

The File Allocation Table contains invalid information.

Attention:
If R (for retry) were pressed, DOS would consider the Non-DOS disk to be double-sided, with 8 sectors per track. On the one hand, this would probably cause the disk to be destroyed by a WRITE command. On the other hand, pressing R enables you to access at least the first nine sectors of a Non-DOS disk with the DEBUG utility.

No paper

The printer is out of paper.

Not ready

The indicated device is not ready to accept or transmit data.

Read fault

An error occurred while DOS was reading the data.

Sector not found

Owing to a defective spot on the disk, the sector containing the data could not be found.

Seek

The disk drive could not locate the appropriate track.

Write fault

An error occurred while DOS was reading data to/from the device.

Write protect

The disk in the specified drive is write protected. You cannot write on this disk.

GENERAL MESSAGES

A

About to generate .EXE file
Change disks <hit ENTER>

LINK. You have specified the /PAUSE option. Insert your Runfile diskette into the appropriate drive and press the <CR> key.

Abort edit (Y/N)?

EDLIN. You pressed Q to end the editing session. If Y is entered, the editing changes will not be saved, and, in addition, your previous backup copy will no longer exist.

Acess denied

DEBUG. An attempt was made to write to an R/O file.

All files cancelled by operator

PRINT. This message is returned if you specified the /T option which empties the print queue.

Allocate error on VM.TMP

MS-LIB. The disk is out of space.

Allocation error, size adjusted

CHKDSK. This message is preceded by a filename. An invalid sector number was found in the file allocation table. If you specified the /F switch, the file is truncated at the end of the last valid sector.

All partitions are currently in use.

FDISK. You must delete one of the current partitions before defining a new one.

All specified file(s) are contiguous

CHKDSK. The file or files you named are all written sequentially on the disk.

Ambiguous switch:z

LINK. The characters specified by z do not uniquely identify a linker parameter. Use more characters from the parameter name.

Amount read less than size in header

EXE2BIN. The program portion of the file was smaller than indicated in the file's header. You should re-compile or re-assemble, and re-LINK the program.

An internal failure has occurred

LINK. An error has occurred in the linker program. Contact your NCR PC dealer.

Are you sure (Y/N)?

Commands. DOS asks for verification.

Attempt to access data outside of segment bounds

LINK. There is probably an invalid object file.

B

Backing up files to diskette xx

BACKUP. This message will be followed by a list of files that were backed up on the indicated disk.

Backup diskette is not properly formatted

BACKUP. Use the FORMAT command to re-format the backup diskette.

Backup file sequence error

RESTORE. The files must be restored in the same sequence as they were backed up.

Bad command or file name

DOS. Check for spelling mistakes. The command line you entered is invalid.

Bad numeric parameter

LINK. The value you specified with the /STACK option is not in digits.

Bad or missing Command Interpreter

DOS. The disk that DOS is being started from does not contain a copy of COMMAND.COM, or an error occurred while the disk was being loaded. If System Reset fails to solve the problem, copy COMMAND-.COM from a backup disk to the root directory of the disk that failed.

Bad or missing <filename>

DOS. The message is returned only at startup. DOS was unable to find a device driver named in a DEVICE= <filename> parameter in the CONFIG.SYS file. It could also mean that a break address that was out of bounds for the machine size was set, or an error occurred while the driver was being loaded. That driver was not installed by DOS.

Batch file missing

DOS. The diskette was removed before the batch job could be terminated or the file was erased or renamed by one of the steps within it.

BF

DEBUG. Bad flag. You attempted to alter a flag, but the characters typed were not one of the acceptable pairs of flag values. See the Register command for the list of acceptable flag entries.

BP

DEBUG. Too many breakpoints. You specified more than ten breakpoints as parameters to the G command. Retype the Go command with ten or fewer breakpoints.

BR

DEBUG. Bad register. You typed the R command with an invalid register name. See the Register command for the list of valid register names.

C

<filespec> Canceled by operator

PRINT. This message appears on the printer after the current file being printed has been cancelled by a /C.

Cannot do binary reads from a device.

Commands. You have specified the /B switch with a device name while attempting to copy from the device. The copy cannot be performed in binary mode because COPY must be able to detect end-of-file from the device. You should omit the /B switch or specify the /A switch after the device name.

Cannot CHDIR to xxx
tree past this point not processed

CHKDSK. CHKDSK cannot continue processing the directory path currently being examined because track 0 is bad.

Cannot CHDIR to root
Processing cannot continue

CHKDSK. The disk you are checking is bad. Try restarting DOS and

RECOVER the disk.

Cannot create <file name>

MS-LIB. The specified file cannot be created because there is not enough room in the directory.

Cannot edit .BAK file - rename file

EDLIN. You attempted to edit a file with a filename extension of BAK! BAK files cannot be edited because this extension is reserved for backup copies. If you need the BAK file for editing purposes, you must either RENAME the file with a different extension or COPY the BAK file and give it a different filename extension.

Cannot find file <file name>
Change diskette <hit ENTER>

LINK. The linker could not locate the specified object module on the drive. Insert the diskette with the specified module on it and press the <CR> key.

Cannot find library drive x: filename
Enter new drive letter:

LINK. The linker was not able to find the library on the drive specified. Enter the correct drive designation.

Cannot format an ASSIGNed drive

FORMAT. The drive to be formatted has been assigned another drive designation. The reassignment must be removed before using the FORMAT command.

Cannot format while drive is reassigned

FORMAT. An attempt was made to format a drive that had been reassigned.

Cannot load COMMAND, system halted

DOS. While attempting to reload the command processor, DOS determined that the area in which it keeps track of available memory has been destroyed, or the command proces sor could not be found in the path specified by the COMSPEC environment parameter. You should restart DOS.

Cannot nest response file

MS-LIB. LINK. You used filespec within an automatic response file. Automatic response files cannot be nested.

Cannot open <filespec>

PRINT. The file to be printed is being written on or Crtl-C was pressed. Close the file and enter the PRINT command again.

Cannot open list file

LINK. The directory or disk is full.

Cannot open overlay

LINK. The directory or disk is full.

Cannot open temporary file

LINK. The directory or disk is full. MS-LINK is unable to create the file VM.TMP. Do not remove the disk that will receive the List.MAP file.

Cannot open VM.TMP

MS-LIB.There is no room for the file in the disk directory.

Cannot recover . (or ..) entry, processing continued.

CHKDSK. The working directory (or the working directory's parent directory) could not be recovered.

Cannot start COMMAND, exiting

DOS. While attempting to load a second copy of the command processor, either the FILES= parameter in the configuration file was found to contain too small a value, or there is insufficient available memory to contain the new copy of COMMAND.COM.

Cannot write library file

MS-LIB. The disk is out of space.

Close error on extract file

MS-LIB. The disk is out of space.

Command error

Commands. An invalid command was entered.

Compare another disk (Y/N?)

DISKCOMP. Enter Y if you wish to compare another disk and N if you do not want to continue.

Compare error at offset nnnn
File1= xx
File2= yy

COMP. There is an inequality with the files. The offset number refers to the number of bytes in the file where the error occurred (using base 16). The next two lines show the hexadecimal code of the offending characters.

Comparing x sectors per track, n side(s)

DISKCOMP. n represents the number of sides that DISKCOMP will compare. x indicates the number of sectors per track.

<filename> contains x non-contiguous blocks

CHKDSK. This message is for your information. The disk is fragmented, and the file is not written sequentially on the disk. As it takes longer to read files containing non-contiguous blocks, you should make a copy to increase performance.

Content of destination lost before copy

COPY. You entered a concatenation COPY command where one of the source filenames had the same extension as the destination. That one input file was skipped.

Copy another disk (Y/N)

DISKCOPY. Press Y if you want to copy another entire disk. DISKCOPY will prompt you to insert the required diskette. Enter N if you do not want to make another copy.

Copy complete

DISKCOPY. The copying process is completed. The contents of the source diskette have been copied to the target diskette.

Copying x sectors per track, n side(s)

DISKCOPY. n indicates the number of sides that DISKCOPY has read from the source diskette. x represents the number of sectors per track.

Couldn't open list device PRN
Enter the name of list device?

DEBUG. An incorrect list device was entered.

D

DF

DEBUG. Double flag. Conflicting codes were specified for a single flag. A flag can be changed only once per Register (RF) command.

Directory error in TREE

TREE. TREE has found a bad directory.

Directory is totally empty, no . or ..

CHKDSK. The specified directory does not contain a working directory or a parent directory.

Disk already has an MS-DOS partition

FDISK. A fixed disk may have only one MS-DOS partition.

Disk error reading FAT

CHKDSK. A disk error was encountered when CHKDSK was attempting to update the file allocation table (FAT) on the specified drive. Copy or BACKUP all files on disk and reformat.

Disk error writing FAT

CHKDSK. A disk error was encountered when CHKDSK was attempting to update the file allocation table (FAT) on the specified drive. Copy or BACKUP all files on disk and reformat.

Disk full - file write not completed

EDLIN. You gave the End command, but the disk did not contain enough free space for the whole file. EDLIN aborted the E command and returned you to the operating system. Some of the file may have been written to the disk. Only a portion (if any) of the file has been saved. You should probably delete that portion of the file and restart the editing session. The file is not available after this error. Always be sure that the disk has sufficient free space for the file to be written to disk before you begin your editing session.

Diskettes compare OK

DISKCOMP. The information on the two diskettes is identical.

Diskette is not a backup diskette

BACKUP and RESTORE. The diskette was not created by BACKUP. The first file on a backup disk is always BACKUPID.. Rerun with the correct disk.

Diskette is not last backup diskette.
"Can't add in middle of a series".

BACKUP. Data can be added only to the last backup diskette. Make sure that you have the right diskette ready for the backup.

Disk unsuitable for system disk

FORMAT. A defective track was detected where the DOS files were to reside. The disk is suitable only for data.

Divide overflow

DOS. A program attempted to divide a number by zero, or the program had a logic error that caused an internal malfunction. The system simulates CONTROL-C processing.

Does not exist

CHKDSK. CHKDSK could not find the specified path.

Do you see the leftmost 0 (Y/N)?

MODE. If you enter Y, the display is okay. If you enter N, MODE again shifts the display by one character and repeats the process.

Do you see the rightmost 9 (Y/N)?

MODE. Entering Y, means the display is okay. If you enter N, MODE shifts the display by one character and repeats the process.

Dup record too complex

LINK. Problem resides in object module created from an assembler source program. A single DUP requires 1024 bytes before expansion. Debug the source program; then rerun LINK.

Duplicate filename or File not found

RENAME. Your disk already contains a file with the specified filename, or DOS could not find the file to be renamed on the specified drive.

E

End of input file

EDLIN. The entire file was loaded and can be edited.

Enter primary file specification
Enter secondary file specification

COMP. Enter the filespecs of the two files to be compared.

Entry error

EDLIN. The last command typed contained a syntax error. Retype the command with the correct syntax and press the <CR> key.

Entry has a bad attribute

CHKDSK. CHKDSK could not read the entry.

Entry has a bad link

CHKDSK. The disk contains clusters that are cross-linked. Run CHKDSK again using the /F: option.

Entry has a bad size

CHKDSK. The´directory contains more data than is space on the cluster.

EOF mark not found

COMP. This message may be displayed when comparing non-text files. For further information see the COMP command, Chapter 4.

ˆError

DEBUG. A syntax error occurred. DEBUG responds with ˆError and redisplays the current assembly address.

Error: An internal error has occurred.

MS-LIB. Contact your dealer.

Error in EXE file

DOS. An error was detected in the relocation information placed in the file by the LINK program. This may be due to a modification to the file.

Error in EXE/HEX file

DEBUG. The file contained invalid records or characters.

Error loading from fixed disk

The system could not be loaded from fixed disk. Insert a system disk in drive A: and load from flex disk.

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

CHKDSK. You did not specify the /F switch. Consequently, CHKDSK will not correct any errors detected.

Errors on list device indicate that
it may be off-line. Please check it.

PRINT. An off-line device is being used for background printing, and a new PRINT command was issued.

EXE and HEX files cannot be written

DEBUG. The data would require a backwards conversion that DEBUG doesn't support.

EXEC failure

Commands. An error was found while reading a command from disk, or the FILES= command in the configuration file (CONFIG.SYS) does not specify a large enough value. You should increase that value and restart DOS.

F

Fatal error: Cannot open input file

MS-LIB. You have mistyped the filename.

Fatal error: Module is not in the library.

MS-LIB. You tried to delete a module that is not in the library.

File allocation table bad drive x

This message means that the copy in memory of one of the allocation tables has pointers to non-existent blocks. Possibly the disk was incorrectly formatted or not formatted before use. If this error persists, the disk is currently unusable and must be formatted prior to use.

File cannot be converted

EXE2BIN. CS.IP does not meet the COM.file criteria, or it has bad segment fixups.

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

COPY. The first filespec you specified is on the default drive. You did not specify the second filespec. The COPY is aborted (copying files onto themselves is not allowed.)

File creation error

DOS and commands. An unsuccessful attempt was made to add a new filename to the directory or to replace a file that was already there. If the file was already there, it was marked read-only and therefore could not be replaced. Run CHKDSK to determine if the directory is full, or if some other condition caused the error.

Filename must be specified

EDLIN. You are prompted to specify the file you wish to edit.

Files are different sizes

COMP. Files of different sizes cannot be compared.

Files compare ok
Compare more files (Y/N)?

COMP. Your files are identical.

File not found or <filespec>, File not found

DOS and commands. You have specified a file that is not present on the disk.

Files were backed up

RESTORE. The files on the backup diskette were backed up on the indicated date.

FIND: file not found <filename>

FIND. The filename you have specified does not exist or FIND cannot find it.

FIND: Invalid number of parameters

FIND. You did not specify a string when issuing the FIND command.

FIND: Invalid parameter <option-name>

FIND. You specified an option that does not exist.

FIND: Read error in <filename>

FIND. An error occurred when FIND tried to read the file specified in the command.

FIND: Syntax error

FIND. You typed an illegal string when issuing the FIND command.

First cluster number is invalid, entry truncated

CHKDSK. If you specified the /F switch, the file is truncated to zero-length.

Fixups needed - base segment (hex):

EXE2BIN. The source (.EXE) file contained information indicating that a load segment is required for the file. Specify the absolute segment address at which the finished module is to be located.

Fixup offset exceeds field width

LINK. An assembly language instruction refers to an address with a short instruction instead of a long instruction. Edit assembly language source and reassemble.

FOR cannot be nested

Batch. More than one FOR subcommand was found on one command line in the batch file. Only one FOR subcommand is allowed per command line.

Format Aborted

FORMAT. The drive lever was opened.

Format another (Y/N)?

FORMAT. Enter N for "no" if you do not wish to format any other disks now.

Format Complete

FORMAT. The diskette has been successfully formatted.

Format failure

FORMAT. A disk error was encountered while creating the target diskette. The diskette is unusable.

Formatting target while copying

DISKCOPY. When using the DISKCOPY command, the destination diskette need not be formatted.

G

Graphics Characters Loaded

GRAFTABL. This message is displayed after selecting either 1 (for Low or Medium Resolution) or 2 (for High Resolution). GRAFTABL is loaded and graphics display of extended characters ensured.

H

<filename>Has invalid cluster, file truncated

The file whose name precedes this message contains an invalid pointer to the data area. The file is truncated if the /F switch was used.

I

Illegal baud rate

MODE. Legal values are: 11, 15, 30, 60, 12, 24, 48, 96, 13, 18, 20, 36, 72, 19.

Illegal characters per line

MODE. A line may consist of 80 or 132 characters.

Illegal CRT adjustment parameter

MODE. The CRT adjustment parameters are R, L, and T.

Illegal data size

MODE. An illegal data size has been entered. Legal values are 7 and 8.

Illegal device

MODE. An illegal device was specified. Legal devices are:
LPT1, LPT2, LPT3, COM1, and COM2.

Illegal lines per inch

MODE. Legal values are 6 and 8.

Illegal parameter

MODE. An illegal parameter was specified.

Illegal parity specification

MODE. The parity may be N (none), O (odd) or E (even).

Illegal redirection

MODE. An illegal device has been specified.

Illegal stop bit count

MODE. Stop bits may be 1 or 2.

Incorrect DOS version

Commands. The command you entered can be used only with DOS version 2.11.

Input file read error

MS-LIB and LINK. There is probably a bad object file/module or faulty disk.

Insert backup diskette xx in drive x:

RESTORE. Insert the new backup diskette in sequence. RESTORE will continue when you press a key.

Insert backup diskette xx in drive x:
Warning! Diskette files will be erased
Strike any key when ready

BACKUP. Insert the next diskette to be used for the backup. Use DOS formatted disks only. BACKUP will continue when you press a key.

Insert COMMAND.COM disk in drive X:
and strike any key when ready

DOS. DOS is attempting to reload the command processor, but COMMAND. COM is not present on the drive DOS was started from. Insert a system diskette into the specified drive and press any key.

Insert disk with batch file
and press any key when ready

DOS. You have removed the disk containing the batch file. The batch processor is trying to find the next command in the file. Processing can continue when the disk is inserted into the proper drive. Press any key.

Insert diskette for drive x and strike any key when ready

DOS. You are prompted to insert a diskette in the specified drive to continue processing.

Insert first diskette in drive x
Insert second diskette in drive x

DISKCOPY. Insert diskettes as instructed.

Insert new diskette
and strike any key when ready

FORMAT. Insert a new disk and start the formatting process.

Insert DOS disk in X:
and strike any key when ready

FORMAT. FORMAT is trying to load the DOS files, but there is no system disk present in drive x.

Insert source diskette in drive x
Insert destination diskette in drive x

DISKCOPY. Insert diskettes as instructed, and press any key when ready. The copying process will continue.

Insufficient disk space

DOS and commands. There is not enough free space on the disk to contain the file being written. To verify, run CHKDSK to determine the status of the disk.

Insufficient memory

Commands.There is not enough memory in your machine to process these commands. Try to obtain more memory. You could, for example, change the BUFFERS=parameter in the CONFIG.SYS file to a smaller value (this is true only if you have specified BUFFERS=) and restart the system.

Insufficient memory
Processing cannot continue

CHKDSK. There is not enough memory in your machine to process CHKDSK for this disk. You must obtain more memory to run CHKDSK.

Insufficient room in root directory
Erase files in root and repeat CHKDSK

CHKDSK. The "lost" data blocks found by CHKDSK shall be stored in a file. However, the directory is full. You can either delete files from the directory or copy the data blocks onto another disk.

Insufficient space on disk

DEBUG. A Write command was issued to a disk that doesn't have enough free space to hold the data being written to disk, you may insert a disk that does have enough free space and re-issue the Write command. Otherwise, you should erase files from the disk and run DEBUG again.

Intermediate file error during pipe

There is probably not enough space for piping or no directory entry free. Erase a file or use another disk.

Invalid characters in volume label

FORMAT. One or more of the characters you entered in the volume label is not a valid filename character, or the name contained a period (volume labels contain 1 to 11 valid characters without a period).

Invalid COMMAND.COM
Insert COMMAND.COM disk in default drive
and strike any key when ready

DOS. While trying to reload the command processor, the copy of COMMAND.COM on the disk was found to be an incorrect version. You are prompted to insert the correct DOS disk and press any key to continue.

Invalid country code

You have made an invalid entry in the CONFIG.SYS file.

Invalid current directory

CHKDSK. You cannot continue processing. Restart the system and rerun CHKDSK.

Invalid date

DATE. You entered an invalid date or an incorrect delimiter. Remember that only hyphens and slashes are allowed.

Invalid device

Commands. DOS does not accept the device name you specified.

Invalid directory

DOS and commands. One of the directories in the specified path does not exist.

Invalid drive in search path

DOS. An invalid drive specification was found in one of the paths specified in the PATH command. This message appears when DOS attempts to locate a command or batch file, not at the time you issued the erroneous PATH command.

Invalid drive specification

DOS and commands. You have entered an invalid drive specification with the command.

Invalid drive or filename

EDLIN. You entered an invalid drive name or file.

Invalid format file

LINK. A library is in error.

Invalid number of parameters

Commands. You have specified the wrong number of options on the command line.

Invalid object module

MS-LIB and LINK. An object module(s) is incorrectly formed or incomplete (as when assembly is stopped in the middle).

Invalid parameter

DOS and commands. You have entered one or more invalid parameter(s). If a drive designation is required, make sure that the letter is followed by a colon.

Invalid operation: R/O file

EDLIN. The file is a read-only file and cannot be written to.

Invalid partition table on fixed disk.

Start-up. The fixed disk partition contains invalid information. Use the FDISK command for correction.

Invalid path, not directory or directory not empty

RMDIR. The specified directory was not removed because an invalid directory name was given, or because it still contains entries for files or other sub-directories. A directory can be removed only if it is empty (with the exception of the . and .. entries).

Invalid path or file name

Commands. The specified directory or file name does not exist.

Invalid sub-directory entry

CHKDSK. The sub-directory specified contains invalid information. CHKDSK will try to correct the error. Reenter the CHKDSK command and specify the /V switch for additional information on the error.

Invalid switch specification

FORMAT. An invalid switch was specified. See chapter 4, the FORMAT command for the correct syntax.

Invalid time

TIME. an invalid time or delimiter was entered. The delimiters allowed are the colon to separate hours from minutes, and minutes from seconds; and the period to separate seconds from hundredths of seconds.

<filename> is cross-linked on cluster

CHKDSK. This message will appear twice for each cross-linked cluster number, naming the two files in error. Make a copy of the file you want to keep, and then delete both files that are cross-linked.

<filespec> is currently being printed
<filespec> is in queue

PRINT. These two messages are displayed when a PRINT command is given without any parameters. They may also be returned individually when you queue the first of a subsequent file for printing. This message is for your information.

L

Label not found

Batch. A GOTO command named a label which does not exist in the batch file.

Library disk full

MS-LIB. There is no more room on the disk.

Library file does not exist. Create?

MS-LIB. The filename given for the library fields does not exist. Enter Yes (or any response beginning with Y) to create a new library file. Enter No (or any response not beginning with Y) to abort the library session.

Line too long

EDLIN. During a Replace command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. EDLIN aborted the Replace command. Divide the long line into two lines; then try the Replace command twice.

Listing file write error

MS-LIB. The file could not be written because the disk is out of space.

List output is not assigned to a device

This message is displayed if the "Name of list device" specified to the preceding prompt is invalid. Subsequent attempts return the same message until a valid device is specified.

x lost clusters found in chains
Convert lost chains to file (Y/N)?

CHKDSK. If you respond Y to this prompt, CHKDSK creates a directory entry and a file for you to resolve this problem (files created by CHKDSK are named FILEnnnn.

M

Memory allocation error
Cannot load COMMAND, system halted

DOS. A program has destroyed the area in which DOS keeps track of available memory. You should restart DOS.

10 Mismatches - ending compare

COMP. COMP does not continue because the files are too different.

- MORE -

MORE. Press any key to see the remainder of the data to be displayed.

MS-DOS incorrect on default drive

MS-DOS. The file could not be loaded. Check if the diskette in the default drive is a system disk.

MS-DOS Installed OK

MS-DOS. MS-DOS has been successfully loaded.

MS-DOS not found on default drive

MS-DOS. MS-DOS could not be loaded. Check whether the diskette in the default drive is a system disk.

Must specify destination line number

EDLIN. A destination line number was not specified for a Copy or Move command. Reissue the command with a destination line number.

Must specify ON or OFF

Commands. You are prompted to enter either ON or OFF, e.g., VERIFY is ON.

N

Name of list device [PRN:]

PRINT. This prompt appeares when PRINT is run the first time. Any current device may be specified and that device then becomes the PRINT output device. As indicated in the [ ], simply pressing <NEW LINE> results in the device PRN being used.

No fixed disk present

FDISK. Your system does not have any fixed disks. It is not possible to run FDISK.

No free file handles

DOS. 8 files may be open at one time. If you want to open more files, use the CONFIG.SYS file to change the default value.

No library file specified

MS-LIB. The message is displayed because there was no response to the Library File prompt.

No MS-DOS partition to delete.

FDISK. There was no MS-DOS partition defined to FDISK.

No MS-DOS partition . Use FDISK to correct.

FORMAT. The partition that you wanted to format is not an MS-DOS partition. It must be made active before formatting.

No path

There is currently no path specified.

No object modules specified

LINK. You did not name any object modules in the command line or in response to the prompt. The linker needs some files to link.

No operating system on fixed disk

Start-up. The fixed disk does not contain the operating system. Use the FDISK and FORMAT commands described in Chapter 4 to transfer the system to fixed disk, or boot from flexible disk drive A.

No room in directory for file

When you attempted to create a new file, either the file directory was full or you specified an illegal disk drive or an illegal filename. Check the command line that started EDLIN for illegal filename and illegal disk drive entries. If the command is no longer on the screen and if you have not yet typed a new command, the EDLIN start command can be recovered by pressing the <F3> key.
If this command line contains no illegal entries, run the CHKDSK program for the specified disk drive. If the status report shows that the disk directory is full, remove the disk. Insert and format a new disk.

Non-system disk or disk error.

Start-up. Use another system disk for booting.

No space for a xxx cylinder partition at cylinder yyy.

FDISK. There is not enough space for a partition of this size at the specified cylinder.

Not enough room for MS-DOS on this disk

There is not enough disk space on the diskette. Erase all non-essential files and try again or use another diskette.

Not enough room to merge the entire file

EDLIN. There was not enough room in memory to hold the file during a Transfer command. You must free some memory by writing some lines to disk.

Not found

EDLIN. Either the specified range of lines does not contain the string being searched for by the Replace Text or Search Text commands; or if a search is resumed by replying N to the OK? prompt, no further occurrences of the string were found.

O

Out of environment space

The environment passed to the child process exceeds 32 KB.

Out of memory

Commands. Processing cannot continue. Try to obtain more memory.

Out of space on <filename>

LINK. This error usually occurs when there is not enough disk space for <filename>.

P

Path name too long

TREE. The path name specified is too long. Enter the TREE command again, starting from a subdirectory.

Parity error or nonexistant memory error detected.

DEBUG. Bad parity or attempt to read address outside of memory location.

Press any key to begin formatting x:

FORMAT. The fixed disk (drive x) is about to be formatted. Formatting will lose track of all previously existing data on the disk. If you do not want the disk formatted, press CONTROL-C. If you do want the disk formatted, press a character key.

Press any key to begin recovery of the file(s) on drive x

RECOVER. The specified file(s) are about to be recovered. Press any key to start the process. You must understand "any key" as "almost any key". See Chapter 1 for the keys that cannot be used in this manner.

PRINT queue is empty

PRINT. There are no files in the print queue.

PRINT queue is full

PRINT. There is room for 10 files in the queue. If you attempt to put more than 10 files in the queue, this message appears on the console.

Printer fault

MODE. The printer hangs up. Check for syntax errors.

Probale non-DOS disk
Continue (Y/N)?

CHKDSK. The disk you are using is a non-DOS disk. You must indicate whether or not you want CHKDSK to continue processing.

Program size exceeds capacity of LINK

LINK. Load module is too large for processing. The total size may not exceed 384 K bytes, and the number of segments may not exceed 255.

Program terminated normally

DEBUG. The program is terminated. Do not restart a program after this message is displayed. You must reload the program with the N and L commands for it to run properly.

Program too big to fit in memory

DOS. The file containing the external command cannot be loaded because it is larger than the available memory. You should reduce the number in the BUFFERS= parameter in your CONFIG.SYS file (if you have specified BUFFERS=), restart your system and re-issue the command. If the message reappears, your system does not have enough memory to execute the command.

R

Read error on VM.TMP

MS-LIB. The disk was not ready for read.

Re-insert diskette for drive x: and strike any key when ready

FORMAT. You are prompted to insert your diskette in theindicated drive.

Resident part of PRINT installed.

PRINT. The external command is in memory and is ready for use.

Resident portion of MODE installed.

MODE. The file has been loaded into memory.

Requested drive is not available

FORMAT. An invalid drive name has been specified.

Requested stack size exceeds 64K

LINK. Specify a size greater than or equal to 64K with the /STACK option.

S

Sector size too large in file <filename>

Startup. The device driver named in <filename> specifies a device sector size larger than the devices previously defined to DOS.

Segment size exceeds 64K

LINK. Attempt to combine identically named segments which resulted in a segment requirement of greater than 64K bytes. The addressing limit is 64K bytes.

Specified COMMAND search directory bad

An incorrect pathname was given in the SHELL command.

0 size is not allowed. Enter a number

FIDKS. The minimum practical size is 10 (roughly the size of a flexible diskette). If you want you may leave this blank and DOS will default to the largest amount of cylinders allowable.

Stack size exceeds 65535 bytes

LINK. The size specified for the stack must be less than or equal to 65535.

Strike any key when ready

Commands. You must understand "any key" as "almost any key". See chapter 1 of the NCR-DOS User's Guide for further information.

Sub-directories: None

TREE. There is no directory path to display because the specified drive contains only a root directory.

Symbol defined more than once

LINK. A single symbol name is defined by two or more modules.

Symbol table capacity exceeded

LINK. Very many, very long names were entered, exceeding the limit of approximately 50K bytes. Use shorter and/or fewer names.
MS-LIB. (about 30K chars in symbol(s))

Syntax error

DOS and Commands. You have entered an incorrectly formatted command. Check for spelling mistakes.

SYS cannot install MS-DOS on this disk

SYS. MS-DOS cannot be transferred to the specified disk. Try again using another disk.

System transferred

FORMAT. The target disk has been successfully formatted and the operating system has been copied to it.

T

Terminate batch job (Y/N)?

DOS. This message is displayed if you press CONTROL-C while executing a batch file. Pressing Y will stop proces sing while pressing N only ends the command that was being executed when CONTROL-C was typed. DOS will continue executing the batch file with the next command.

The last file was not restored

RESTORE. RESTORE was terminated by the user before it was completely executed or there is not enough space on the fixed disk. The last file is deleted.

There was/were number errors detected

LINK. This message is displayed for your information at the end of the link session.

Too many external symbols in one module

LINK. The limit is 256 external symbols per module.

Too many groups

LINK. The limit is 10 groups.

Too many libraries specified

LINK. The limit is 8 libraries.

Too many object modules

MS-LIB. The limit is 500 object modules.

Too many overlays

LINK. The limit is 64.

Too many public symbols

LINK. The limit is 1024 public symbols.

Too many segments or classes

LINK. The limit is 247 (segments and classes taken together).

Track 0 bad-disk unusable

FORMAT. Track 0 is where the boot record, file allocation table, and directory must reside. The disk is unusable.

U

Unable to create directory

MKDIR. The directory you wish to create already exists, or one of the directory path names you specified could not be found, or you attempted to add a directory to the root directory and it is full.

Unexpected end-of-file on library

LINK. This is probably caused by an error in the library file.

Unexpected end of file on VM.TMP

LINK. The disk was removed from the drive during the link session.

Unknown drive

SYS. An illegal drive name was entered.

Unknown target media - must be formatted.

SYS. A diskette must be formatted for use with DOS (FORMAT command).

Unrecognized command in CONFIG.SYS

Startup. An invalid command was detected in the configuration file CONFIG.SYS. You should edit the file, correct the invalid command and restart DOS.

Unrecoverable error in directory
Convert directory to file (Y/N)?

CHKDSK. The directory whose name precedes this message contains enough invalid information that it is no longer usable as a directory. If you respond Y to this prompt, CHKDSK converts the bad directory into a file. You can then fix the directory yourself using the DEBUG utility, or you can delete it.

Unrecoverable read error

DISKCOPY. A sector could not be read. Try again or use another diskette.

Unresolved externals: <symbol(s)>

LINK. The external symbols listed have no defining module among the modules or library files specified.

V

VM read error

LINK. This is a disk error; it is not caused by LINK.

VM.TMP has been created
Do not change disk in drive <d:>

LINK.DOS has created VM.TMP. Once this message has been displayed, you must not remove the disk from the default drive until the link session ends.

VM.TMP is an illegal filename and has been ignored

LINK. VM.TMP cannot be used for an object filename. This is only a warning.

Volume label (11 characters, ENTER for none)?

FORMAT. You are requested to enter a 1 to 11 character volume label which will be written on the disk being formatted. If you do not want a volume label on the disk, press only the <CR> key.

W

Warning! All MS-DOS partition data will be destroyed.

FDISK. This message is displayed when the "Delete MS-DOS Partition" Function is selected. Make sure to make a backup copy of important data before executing this command.

Warning-directory full

RECOVER. There is insufficient directory space to recover more files. You should copy some of the files to another disk, erase them from this disk and run RECOVER again.

Warning! Diskette is out of sequence
Replace the diskette or continue
Strike any key when ready

RESTORE. The backup diskette is not the next one in sequence. Replace the diskette unless you are sure no files on the diskette(s) you skipped would be restored. RESTORE will continue when you press a key. This message will be repeated if you try to skip a diskette which contains part of a file being restored.

Warning - Diskette may be unusable

DISKCOMP. Use another diskette for processing. Make a backup copy of this disk.

Warning! No files were found to back up.

Backup. No fixed disk files were found that matched the backup file specification.

Warning! No files were found to restore

RESTORE. No backup diskettes files were found that matched the restore file specification.

Warning: no stack segment

LINK. None of the object modules specified contains a statement allocation stack space.

Warning - Read error on EXE file

EXE2BIN. An error occurred while reading the input file. EXE2BIN will attempt to continue, but the result file may be unusable.

Warning: Read error on .EXE file
Amount read less than size in header

EXE2BIN. This is a warning message only.

Warning: Segment of absolute or unknown type

LINK. There is a bad object module or an attempt has been made to line modules that LINK cannot handle (for example, an absolute module).

Warning! The file above is marked read-only
Replace the file (Y/N)?

RESTORE. The indicated is a read-only file. If Y is entered, it is replaced. RESTORE will continue when the <CR> key is pressed. This message is returned only if the /P option was specified.

Warning! The file above
was changed after it was backed up
Replace the file (Y/N)?

RESTORE. The date and time values on the fixed disk are not equivalent to the ones on the backup disk. Enter Y if you want to replace it with the backed up version or N if you do not. RESTORE will continue after you press the <CR> key. This message is displayed only if the /P switch was specified.

Write error on <filename>

MS-LIB. The error occurred because the disk is out of space.

RAM-DISK Demonstration

Your NCR-DOS diskette includes a program to demonstrate the power of the RAM-Disk in an I/O intensive environment. To run this program follow the steps given below.

1. Create a CONFIG.SYS file on your copy of the DOS diskette. The file must contain the following statement:

   DEVICE=RAMDISK.SYS /2/9/S4/D5

See "Configuration File Commands" for details of creating a CONFIG.SYS file.

2. Now boot the system.

3. A message will appear on the screen and inform you that the RAM-Disk has been installed.

4. Copy the two data files RAMDATA.1 and RAMDATA.2 using the 'copy' command when the system prompt appears.

   COPY RAMDATA.1 <D:>
   COPY RAMDATA.2 <D:>

   The <D:> refers to the drive letter of the RAM-Disk.

5. Now run the RAMDEMO.EXE program by typing in "RAMDEMO" after the system prompt.

6. The demonstration program will ask for the drive letters for the flexible or fixed disk and the RAM-Disk. By answering these questions correctly the program will execute a multiplication table reading in the multiplicand from these data files alternately. For each input of the data, the file is opened, the data is read and the file is closed. The operation will be repeated on the RAM-Disk files also. The start and end times in each case are also displayed.

The speed improvements will be 13-15 times compared to flexible disk drives and 4-5 times compared to fixed drives.

KEYBOARD CONSIDERATIONS

This appendix comprises information on the keyboard support files — how they can be used with your own programs and with application programs —, and technical data on their operation and use.

General Information

The NCR-DOS master diskette, the one that came with this binder, contains an AUTOEXEC.BAT file and a number of keyboard support files. As you will remember from copying your NCR-DOS master diskette,, the AUTOEXEC.BAT file is executed when the diskette is inserted and DOS is loaded. During execution, a new disk (target diskette) is formatted and the NCR-DOS master diskette — all keyboard support files and all other files included — is copied to this new DOS disk. Then the new DOS disk is set up to start using the keyboard format for your language. (If you are not yet familiar with this procedure, you should refer to Chapter 0, "Getting Started".)

The keyboard support files include keyboard programs which can be loaded to set up the keyboard for any one of five languages. This enables you to change from the keyboard format created on the new DOS disk to another keyboard format. The keyboard programs are loaded into memory with the KEYB command, which is described in more detail in Chapter 4, NCR-DOS commands.

Any disk formatted for NCR-DOS may be set up to start using a specific keybaord format. This can be done by means of the GWBASIC KBPGM program, which copies the selected keyboard program into the AUTOEXEC.BAT file residing on the program disk.

Setting Up a Disk for a Specific Keyboard Format

Any disk can be set up for a specific keyboard format provided that it contains an AUTOEXEC.BAT file and has enough disk space available for the keyboard program file. So we will first make sure that the disk meets these requirements.

To find out whether the disk contains the AUTOEXEC.BAT file, insert the disk into drive A, type DIR and press the <CR> key. As a result, the disk directory will be displayed on the screen. If scrolling is too fast, hold down the Ctrl key and press the Num Lock key. You may release both keys, search for the AUTOEXEC. BAT file and continue scrolling by pressing the space bar.

Now we will determine whether there is enough disk space on the disk to copy the keyboard program to it. This requires different procedures for one and two-drive flexible disk systems. If you have a one-drive system, turn to "For 1 flexible disk drive", with a two-drive system, to "For 2 flexible disk drives".

For 1 flexible disk drive

Place the DOS diskette in drive A and load DOS. When the system prompt A> is displayed, type:

```
CHKDSK B: <CR>
```

The following message will be returned:

```
Insert diskette for drive B: and press
any key when ready
```

Remove the DOS diskette and insert the program diskette. Continue by pressing any key.

Next DOS will return the disk status. Among other things, you will be informed about the number of bytes available on disk. If the number is greater than 2048, there is enough space available for the keyboard program.

For 2 flexible disk drives

Place the DOS diskette in drive A, the program diskette in drive B:, and load DOS. When the system prompt A> is displayed, type:

```
CHKDSK  B: <CR>
```

DOS will return the disk status. Among other things, you will be informed about the number of bytes available on disk. If the number is greater than 2048, there is enough space available for the keyboard program.

Setting Up the Disk

NOTE: It is possible to set up the disk for a keyboard only if it has an AUTOEXEC.BAT file and if there is enough disk space available to add the keyboard program file.

Insert the DOS diskette into drive A and load DOS. When the system prompt A> is displayed, enter:

```
GWBASIC KBPGM <CR>
```

The program displays:

```
1 = U.S. English
2 = French
3 = German
4 = Italian
5 = Spanish
6 = U.K. English
0 = exit

Enter your selection
```

Enter the number for the keyboard format.

The following message will be displayed:

```
Remove the DOS diskette from drive A
and insert the program diskette
Press any key to continue
```

If necessary remove the write protect tab covering the write protect cutout on your program diskette. Then replace the DOS diskette in drive A with the program diskette. Press any key.

Wait until the following message is returned:

```
Program diskette contains keyboard program

A>
```

Your program diskette is now set up to start using the selected keyboard format when it is loaded. If the write protect tab was removed, it should be replaced.

The Keyboard Support Files

The NCR-DOS master diskette contains a number of keyboard support files which are automatically copied to any new DOS master disk. The list below shows these files and gives a brief explanation of their functions. You may decide that you do not need all of them on your DOS disk in which case this list may help you to determine which files to delete.

KEYBUK.COM	United Kingdom keyboard program
KEYBFR.COM	French keyboard program
KEYBGR.COM	German keyboard program
KEYBIT.COM	Italian keyboard program
KEYBSP.COM	Spanish keyboard program
WTDATIM.COM	An assembler program which causes the date and time prompts to be displayed in the language selected. It searches the AUTOEXEC.BAT file of the specified disk for the keyboard program invoked by the AUTOEXEC.BAT and then prompts for date and time in the corresponding language. If the disk is not set up for a certain keyboard format, the prompts are displayed in English.
KBPGM.BAS	A BASIC program designed to set up a program disk to automatically start a keyboard program when a program is loaded.
AUTOEXEC.BAT	The AUTOEXEC.BAT file on the NCR-DOS master diskette that is used to create your new master disk.

When invoked, the keyboard program is loaded into the lowest portion of available user memory. The file remains in memory until a system reset is performed or the system is turned off. Exit to DOS is via DOS call 31H (Keep Process). Interrupt vector 9H, which initially points to the ROM BIOS keyboard interrupt service routine, is redirected to point to the interrupt service routine of the keyboard program. This is provided with its own lookup table to convert the scan codes received from the keyboard to United States ASCII codes and to hex values 80 through FF for non-United States characters. The scan codes are returned in register AH, the ASCII codes in AL.

When a keyboard program is loaded in read/write memory, interrupt vector 9H is always serviced by the interrupt service routine in read/write memory. Access to the U.S. English Keyboard format is through holding down the Ctrl and Alt keys and pressing the F1 key at the same time. The interrupt vector is still serviced by the program in read/write memory, but tables identical with those in ROM BIOS are used for lookup.

Pressing the Ctrl and Alt keys together with the F2 key reactivates the table in read/write memory.

Programming-oriented Characters

The U.S.English keyboard includes some programming-oriented characters, such as the "at" sign, the backslash and left and right square brackets. On the French, German, Italian and Spanish keyboards, different, mostly language-oriented characters take their positions on the keyboard. The language-oriented character is usually printed on top of the keytips concerned, the programming-oriented character on the front face. To access a programming-oriented character, hold down the Ctrl and Alt keys and press the appropriate key. As a result, the scan and ASCII codes will be returned in the AH and AL registers, respectively. If your keyboard does not show the programming-oriented characters on the front face, turn to Appendix A of the Owner's Manual and compare the U.S.English keyboard layout chart with your keyboard.

The "Dead Key" Feature

The NCR Personal Computer employs "dead" keys to provide diacritical marks on alphabetic characters. They operate in the same way as do European typewriters. To type the letter "e" with an acute accent, the following keyboard actions are taken:

Keyboard Action	Display Screen
	Attach_ ↑ cursor
Press □	Attach_ (the diacritic does not appear)
Press e	Attaché (the combined character appears)

After the diacritical sign has been pressed, its character code is stored in a buffer until another key is pressed. If this second character code is found to be suited for a dead key combination, the two character codes are combined. The resulting code is returned in register AL, the scan code (0) in register AH.

Should the operator add an alphanumeric character to a diacritical mark with which it may not be combined (for possible dead key combinations see the table below), the diacritical mark will be returned first, followed by the character. In the case of any other invalid character, the diacritic does not have any effect. If an attempt is made to combine the diaresis (¨) with an invalid second character, a box (■))254 decimal) will appear on the screen and an error tone will be briefly sounded. Most diacritics can also be a character by themselves; a space should follow the diacritic in this case. The diaresis followed by a space results in the box being printed.

Possible Dead Key Combinations

	Diacritical marks available			
	´	`	¨	^
Germany	x	x		
France	x		x	x
Spain	x	x	x	x

X-diacritical mark available for that country

	Valid Dead Key Characters
Germany	á é É í ó ú à è ì ò ù
France	ä Ä ë i ö Ö ü Ü ÿ â é î ô û
Spain	ä Ä ë ï ö Ö ü Ü ÿ á é É í ó ú à è ì ò ù â ê î ô û

The Paragraph Sign

When DOS is loaded, the system searches the disk for the COMMAND.COM file and loads it into memory. The COMMAND. COM program, that is, the Command Processor, processes the commands you enter and runs the appropriate programs. When the Command Processor is in control of a program, the paragraph sign, which is provided on the German, French and Italian keyboards, cannot be displayed. ↑ U is returned instead.

The Caps Lock Key

When this key is in the active mode, pressing an alphabetic key results in an uppercase (capital) letter being entered. On a French keyboard it has the additional effect of reversing the characters on the top row of the alphanumeric key pad.

Graphics Support

The NCR Personal Computer operates in either of two modes, text mode and graphics mode. In text mode, the entire character set can be displayed on the screen, whereas in graphics mode, the GRAFTABL. COM file must be loaded to ensure full character support.

This may need some further explanation. In graphics mode you may choose between 3 degrees of resolution of graphic display, namely low, medium, and high resolution e.g. by use of the SCREEN command in GW-BASIC. In low and medium resolution, a character consists of 8x8 bits, in high resolution of 8x16 bits.

The pixel-bit maps for all characters which are not provided on the United States keyboard (hex values 128 through 255) are contained in tables defined by the GRAFTABL.COM file.

When loaded, the GRAFTABL.COM file initializes an 8x8 or an 8x8 and an 8x16 character generator and causes interrupt vector 1FH to point to an 8 by 128 byte or an 8 by 128 byte and a 16 by 128 byte table, respectively.

The file is loaded with the GRAFTABL command described in Chapter 4, NCR-DOS commands.

GSX-86 CRT Driver

This driver provides high resolution color graphics for your NCR Personal Computer. It is the hardware dependent module for GSX(TM) (Graphics System Extension) which gives your operating system high resolution (640x400 pixel) color graphics capability. Among other graphics applications you can use DR Graph(TM), for instance.

Device Specifications

FILENAME — NCRPC4.SYS

DEVICE INDEX — The actual device index for this device is determined in the ASSIGN.SYS file, which associates a device index with a GIOS module (device driver). We recommend that this number be in the range 1-10 for CRTs.

GRAPHIC INPUT (GIN) — If GIN is involved on the NCR Personal Computer a large plus sign (+) appears on the screen (cursor). The cursor can be moved by pressing one of the four arrow keys (up, down, left, right) on the keyboard.
The special cursor home key enables or disables fast cursor movement. When the cursor is at the desired location the point can be selected by pressing any alphanumeric key on the keyboard. This causes the coordinates of the point to be transmitted back to the user program.
The +sign key and the —sign key are the valuator keys.

TEXT — NCR Personal Computer supports one character size.
Text can be rotated in 90-degree increments.

MARKERS — NCR Personal Computer support one marker size and 8 different markers.

1 .   2 +   3 *   4 ○
5 X   6 □   7 ◇   8 △

HATCHSTYLE

The NCR Personal Computer has eight hatchstyles.

Hatchstyle 1 vertical crosshatch
2 horizontal crosshatch
3 diagonal crosshatch +45°
4 diagonal crosshatch —45°
5 square crosshatch
6 x-crosshatch
7 vertical —45° crosshatch
8 cross diagonals

LINESTYLE

The NCR Personal Computer has eight linestyles.

Linestyle 1 is solid
2 is dashed
3 is dotted
4 is dashdot
5-8 are combinations of dashed and dotted lines.

PATTERN

The NCR Personal Computer has eight fill patterns.

COLOR

By using escape sequence 60 two different color palettes are selectable.

1. Background Green Red Yellow
2. Background Cyan Magenta White

Other default values that should be set by the driver during initialization are:

GENERALIZED DRAWING PRIMITIVES (GDPs)

The available GDP and its identifier is:

1 — Bars

ESCAPES

The escape functions available on the NCR Personal Computer are:

1 Inquire addressable character cells
2 Enter graphics mode
3 Exit graphics mode
4 Cursor up
5 Cursor down
6 Cursor right
7 Cursor left
8 Home cursor
9 Erase to end of screen
10 Erase to end of line
11 Direct cursor address
12 Output cursor addressable text
13 Reverse video on
14 Reverse video off
15 Inquire current cursor address
16 Inquire tablet status
18 Place cursor at location
19 Remove cursor
60 Background color

SUMMARY

The functions available through GIOS in the NCR Personal Computer are:

Opcode	Definition
1	Open workstation
2	Close workstation
3	Clear workstation
4	Update workstation
5	Escape
6	Polyline
7	Polymarker
8	Text
9	Filled area
11	GDP
12	Set character height
13	Set character up vector
15	Set polyline linetype
17	Set polyline color index
18	Set polymarker type
20	Set polymarker color index

Opcode	Definition
22	Set text color index
23	Set fill interior style
24	Set fill style index
25	Set fill color index
28	Input locator
29	Input valuator
30	Input choice (function keys)
31	Input string
32	Set writing mode
33	Set input mode

ERROR MESSAGE Not enough memory on your graphics card GSX aborted

means: there exists no high resolution card

ADDTIONAL INFORMATION

"Text"-Function (Opcode 8):
If you type past the end-of-line a line feed is performed. It may happen that the characters printed in the new line will overwrite those in the line above.

"Input Locator"-Function (Opcode 28)
The values may be incorrect due to rounding errors that may occur during conversion of coordinates.

"Input-String"-Function (Opcode 31)
In request mode characters are stored but not printed on the screen.

Glossary

A

ABORT

Premature termination of a program during execution.

ACCESS

The ability to locate information in a file and change it, if necessary.

ADDRESS

Data is contained in the electronic devices that make up your computer's memory and the registers in its CPU. Each device is assigned a number, which is known as an "address". Your computer is able to locate data using these addresses.

ALPHA

Alphabetic characters.

ALPHANUMERIC

Alphabetic, numeric, plus other characters such as parenthesis, dollar sign, etc.

ANSI

American National Standards Institute. An organization that establishes standards for protocols, computer languages, etc.

APPLICATION PROGRAM

A program that you use on your computer to perform a specific job, such as balancing your checkbook.

ASCII

American Standard Code for Information Interchange: A binary code widely accepted for use in sending and receiving character information in digital form.

ASSEMBLE

To prepare an object language program from a symbolic language program by substituting machine operation codes for symbolic operation codes and absolute or relocatable addresses for symbolic addresses.

ASSEMBLER

A computer program that accepts routines and subroutines that have been prepared in a mnemonic language and translates them into one main program in numeric machine language. Ratio of machine language to numeric language is usually 1 to 1.

ASSEMBLY LANGUAGE

The operational language used in a specific machine-oriented computer program.

B

BACKUP

1. V: The process of duplicating data from one disk to another disk to prevent total information loss in case the original diskette is lost or damaged. 2. n: A copy of a file or diskette.

BASIC

1. Acronym for the programming language Beginner's All-purpose Symbolic Instruction Code. BASIC uses common English words and is relatively easy to use. 2. The computer program that translates statements made in BASIC programming language into instructions to the computer.

BINARY CODE

A system of numbering that uses only two numbers, zero and one. For example, the ASCII letter D is represented as 01000100.

BIT

Acronym for Binary Unit. A switch in the CPU that is controlled by electronic impulses. A group of eight bits is called a "byte."

BUFFER

An intermediary storage area between two data processing storage or data handling systems with different access times or formats. An interim system to facilitate interface between two other systems. Information is held in the buffer until the computer or device is ready to process it.

BYTE

In your CPU, one byte represents one character and each byte is made up of 8 bits.

C

CLEAR

Placement of a memory device into a prescribed state. usually a state denoting zero or blank.

COBOL

An ancronym for Common Business Oriented Language. A high-level programming language that makes use of English language commands.

COMMAND

An instruction that tells the operating system what you want to do.

COMMAND LANGUAGE

A set of commands that control the overall operation of a computer system.

COMPILE

To translate a high-level language program into a machine language or intermediate language program.

COMPILER

A computer program used to translate high-level, human understandable languages into machine-oriented computer language.

COMPONENT

One item of a total system. For example, your keyboard is a component of your NCR Personal Computer.

CONTROL CHARACTER

A character whose occurrence in a particular context initiates, modifies, or stops an action that affects the recording, processing, transmission, or interpretation of data.

COPY

To reproduce a program or data while leaving the original information unchanged.

CPU

Acronym for Central Processing Unit.

CR

Carriage Return or Return. A format effector that moves the active position to the first character position on the next line.

CURSOR

The small blinking indicator on your display screen that shows you where the next data that you type will appear.

D

DATA

Any information — letters, numbers, symbols — that is input to, or output from, the computer for storage or processing.

DEFAULT

A value or direction automatically selected by the computer unless you specify a different one.

DEVICE

A computer peripheral or an electronic component.

DIRECTORY

A list of file names kept on each disk.

DISK (HARD)

A thin, flat, circular piece of rigid plastic on which data and information is stored. Hard disks (also called "Fixed" disks) are able to store much more data than a flexible disk.

DISKETTE

A thin, flat piece of flexible plastic that is coated with magnetic material and encased in a protective paper jacket. Data is recorded and stored on a diskette. Flexible Disk, Floppy Disk, Disk, Flexible Diskette, Floppy Diskette, Floppy, are all common names used to refer to a removable diskette.

DISK DRIVE

The mechanical device, contained within the display unit, that spins the diskettes and either reads information from them, or transfers information to them.

E

EDIT
To modify programs, data files, text files or database information by adding, deleting or changing characters, lines, records, words, etc.

EDITOR
A utility program that performs editing operations on programs and data.

ERASE
In the context of a computer, to remove data from internal memory or from a disk.

ENCODE
To put into code.

EXECUTE
To perform a computer instruction or run a program.

F

FILE
An identified collection of data and information stored on a disk or tape. A file can also refer to a body of information stored in the memory of a computer.

FORMAT (DISK/DISKETTE)
To prepare a disk for use with a particular computer system.

FORTRAN
FORmula TRANslator. A type of high-level language commonly used for the solution of engineering and scientific problems.

FUNCTION KEY
A push button on a computer keyboard that transmits a unique code that the computer interprets in a specific manner.

H

HARDWARE
All the physical components that make up your computer system. Examples include the keyboard, the display monitor, the system unit, or the printer.

HEXADECIMAL NUMBER

A number written in the base 16 number system. The first 10 numbers in the system are the integers 0 through 9; the last 6 "numbers" are the letters A, B, C, D, E, F. Also called hex number.

HIGH-LEVEL LANGUAGE

A computer programming language that is less dependent on the limitations of a specific computer; for instance, pseudolanguages; problem oriented languages; languages common to most computer systems, such as PASCAL, FORTRAN, and COBOL.

I

INITIALIZE

To originate or establish the startup state of a computer system.

INPUT

Information or data supplied to a computer or program for processing.

INTERFACE

The bringing together (in an organized manner) of entities, such as hardware, software, and a human.

K

KB (K)

Abbreviation for kilobyte. A common measure of computer storage equalling 1024 bytes.

KEYBOARD

A device for the inputting data by pushing keys.

M

MACHINE LANGUAGE

Instructions in binary notation that the hardware can understand and carry out. Machine language is always produced as a result of passing high-order programming language instructions through translation routines.

MAGNETIC DISK

Storage device(s) consisting of magnetically coated disks, on the surface of which information is stored in the form of magnetic spots arranged in a manner to represent binary data.

MAIN MEMORY

Internal memory; the temporary memory that is maintained electrically in the computer.

MEMORY

The part of the personal computer that stores information. Personal computers always have some amount of internal memory capacity made up of memory chips; this amount can vary greatly.

MESSAGE

1) A finite sequence of letters, digits, symbols, etc. 2) A transmitted series of words or symbols that are designed and intended to convey information. 3) A group of words, variable in length, transported as a unit; a transported item of information.

O

OUTPUT

In electronics, the power, energy, or signal delivered by a device or system. In computer terminology, the processed information being delivered by a computer.

P

PASCAL

A high-level programming language named after Blaise Pascal and developed by Niklaus Wirth. It emphasizes structured programming and provides convenient data structuring facilities.

PERIPHERAL

A device that is separate from the computer but that works in conjuction with it. A printer, keyboard, and disk drive are all examples.

PROCEDURE

The course of action taken to solve a problem.

PROGRAM

1. (n.) A list of instructions for the computer to follow in order to perform a desired operation. 2. (v.) To prepare a list of instructions for the computer to follow.

R

RAM

Acronym for Random Access Memory. The type of internal memory of a computer in which data can be written to, read from, erased, or stored in any order. RAM is maintained by electrical current and makes up much of the internal memory.

RANDOM ACCESS MEMORY

A storage technique in which the time required to obtain information is independent of the location of the information most recently obtained. Contrast with Sequential Access Storage.

ROM

Acronym for Read Only Memory. instructions that control essential operations of the computer are stored permanently in ROM. You can only read from ROM; you cannot change its contents.

S

SECTOR

A portion of each of the tracks on a disk or diskette. There are usually nine sectors on a track.

SOFTWARE

The non-physical portion of your system. Any kind of program, language, or data within the electrical circuits of a computer is software.

STARTUP

The process of loading and initializing an operating system.

SYNTAX

The rules of a computer language that must be followed when using the language.

SYSTEM

A collection of parts or devices making an organized whole, through some form of regular interaction or interdependence.

T

TERMINAL

A device for communication with a computer. A typical terminal consists of a keyboard and a printer or video display.

TRACKS

A series of concentric rings on a disk. Data can be written to or read from the tracks by the read-write head. There are forty tracks on each side of a 5 1/4-inch disk.

V

VARIABLE

A factor or condition that can be measured or altered by a control system, such as temperature, pressure, flow, liquid level, humidity, weight, or chemical composition.

W

WORD

In computer terminology, a unit of information that occupies one storage location. In most digital computers, a word is composed of a group of bits (binary digits).

WRITE

To copy from one form of storage to another, particularly from an internal storage to an external or secondary storage.

WRITE PROTECT NOTCH

A small notch located on an edge of most diskettes. it protects the data that is recorded on that diskette from being recorded over and thus lost.

NCR-DOS INDEX

.BAT, 3-6
.EXE (LINK), 7-6, 7-12
.EXE file (LINK), 7-1
.LIB (LINK), 7-6
.MAP (LINK), 7-6, 7-12
.OBJ (LINK), 7-6, 7-12
; (command character) (LINK)
* character, 2-3, 2-4
. character, 2-12
.. characters, 2-12
+(command character), 7-10

> or >> Output to file, 3-12
< Input from file, 3-13
<CR> key, 1-5
<Num Lock> key, 2-2

/DSALLOCATE option, 7-13
/HIGH option, 7-14
/LINE option, 7-14
/MAP option, 7-14
/PAUSE option, 7-14
/STACK option, 7-15
| pipe, separator 3-3, 3-13, 4-2
\ character, 2-11

? character, 2-3, 2-4

Address (LINK), 7-4
ANSI escape sequences, B-6
ANSI.SYS file, 4-26
appending data, 3-12
ASCII (chart), B-5
ASSIGN command, 4-5
assigning
  disk drive, 4-5
  RAM as disk, 4-29
AUTOEXEC.BAT file, 3-8, 4-9
automatic program execution, 1-11
AUX, device name, 2-5

Backing-up disks, 1-3, 1-10, 4-7
BACKUP command, 1-10, 4-7
batch processing, 4-9
  suspending, 4-15
BREAK command, 4-27
BUFFERS command, 4-28

Caps lock, E-7
changing
  drives, 1-10
  partitions, 4-51
characters, LINK command, 7-10
CHDIR command, 2-13, 2-15, 4-18
CHKDSK command, 4-19
Command Prompts
  Libraries (LINK), 7-8, 7-12
  List File (LINK), 7-8, 7-12
  Object Modules (LINK), 7-8, 7-12
  Run File (LINK), 7-8, 7-12
class, 7-4
clearing, screen, 4-22
CLS command, 4-22
command characters (LINK), 7-10
command options, 3-4
COMMAND.COM file, 1-6, 4-23
commands,
  batch, 4-9
  CONFIG.SYS, 4-26
  editing, 6-3
  editing of, 5-3
  external, 2-12, 3-2
  filters, 3-13
  internal, 2-13, 3-1
  piping, 3-13
  summary, 4-3
COMP command, 4-24
comparing files, 4-24
CON, device name, 2-5
concatenation, 4-35
CONFIG.SYS file, 4-26, A-1
configuring
  (system), 4-26
  system (advanced), A-1

converting files to binary form, 4-47
COPY command, 2-7, 4-34
copying files, 2-7, 4-34
copying your master software disk, 0-1
creating
- a batch file, 3-6
- a directory, 2-14
- command processor, 4-33

CITTY command, 4-37
current date, 1-7
current directory, 2-11
current time, 1-7

Data error, C-1
data input/output, 3-12
DATE command, 4-38
dead key, E-5
default drive designation, 1-10
defining
- communications adapters, 4-64
- disk configuration, 1-12
- printer, 4-62
- video displays, 4-65

DELETE command, 4-40
delimiters, 3-5
deleting partitions, 4-52
DEVICE command, 4-29
device names, 2-5
DIR command, 1-8, 4-41
directory, 2-1, 2-9
- creating, 2-14
- path, 2-11
- removing, 2-15

disk backup, 1-10
disk copy, 1-10
Disk Errors
- Abort, C-1
- Data, C-1
- Disk, C-1
- Ignore, C-1
- Not Ready, C-1
- Retry, C-1
- Sector Not Found, C-1

Seek, C-1
Write Protect, C-2
disk, master, 0-1
DISKCOMP command, 4-43
DISKCOPY command, 1-10, 4-44
displaying
disk directory, 1-8
files, 4-87
partition, 4-53
remarks, 4-16
the directory, 2-2
volume identifier, 4-90
working directory, 2-13
drive designations, 1-2
default, 1-10
dummy parameters, 3-10

ECHO command, 4-11
edit mode, 6-7
editing
commands, 6-3
files, 6-1
keys, 5-1, 6-2
of commands, 5-3
options, 6-6
programs, 6-1
appending, 6-9
copying, 6-10
deleting, 6-12
exiting from, 6-14
inserting, 6-15
listing (lines), 6-18
moving text, 6-21
paging, 6-22
quit, 6-23
replacing, 6-24
searching, 6-27
transferring, 6-30
writing (to disk), 6-31
EDLIN Error Messages, 6-32
EDLIN Format Conventions, 6-4
EDLIN Command Options, 6-6
entering date and time, 1-6

ERASE command, 4-46
ESC commamd, 4-46a
EXE2BIN command, 4-47
EXIT command, 4-48
extension,
    common, 2-6
    filename, 2-3
external commands, 3-2
external reference, 7-1

FDISK command, 1-13, 4-49
file
    allocation table, 2-1
    protection, 2-8
filenames, 2-2, 2-11
    naming conventions, 2-5
files, 2-1
Files that LINK uses, 7-6
FILES command, 4-32
filters, 3-13
FIND command, 4-55
fixed disk, 1-2
flexible disk, 1-2
FOR command, 4-12
FORMAT command, 1-9, 4-57
formatting
    a disk, 1-9
    a fixed disk, 1-12
function keys, 1-11, 5-1, 5-9

GOTO command, 4-13
GRAFTABL command, 4-58a
Graphic attributes, B-7
GRAPHICS command, 4-59
graphics support, E-7
group, 7-4
GSX-86 CRT Driver, F-1

HELP, 1-15, 4-60
hidden files, 2-2
hierarchical directory structure, 2-9

IF command, 4-14
Ignore, C-1
input,
  data, 3-12
  filtering, 3-13
  redirection, 3-12
installing software, B-1
internal commands, 3-1

Keeping track of files, 2-1
KEYB command, 4-60a
keyboard,
  format, 0-1, 0-6, E-1
  support files, E-1, E-4
  program, 4-6, E-1
  functions, B-2
Keyboard Key Reassignment, 5-10

LINK options, 7-13
linking programs, 7-1
loading DOS, 0-1,
LST, device name, 2-5

Messages, C-1
MKDIR command, 2-14, 4-61
MODE command, 4-62
MORE command, 4-67
multiple operating systems, 1-11, 1-12

Naming files, 2-2
IO.SYS file, 2-2
MSDOS.SYS file, 2-2, 4-82
Not Ready Error, C-1
NUL, device name, 2-6

Object modules, 7-1
operating system, 0-1
operating systems, multiple, 1-11
options, LINK, 7-13
options, command, 3-4
output, redirecting, 3-12

Paragraph, E-7
parameters, 3-10
partition, 1-11, 1-12
PATH command, 4-68
pathing, 2-12, 2-13
pathnames, 2-11
PAUSE command, 4-15
piping, 3-13
PRINT command, 4-69
printing
  files, 4-69
  graphics, 4-59
PRN, device name, 2-5
programming-oriented characters, E-5
prompt, 1-6, 4-72
PROMPT command, 4-72
prompts, LINK, 7-11
protecting files, 2-8

RAMDISK.SYS file, 4-29
reassignment, 5-10
Redirecting, 3-12
RECOVER command, 4-74
relocatable module, 7-1
REM command, 4-16
RENAME command, 4-75
replaceable parameters, 3-10, 4-17
RESTORE command, 4-76
Ready, C-1
RMDIR command, 2-15, 4-78
root directory, 2-9

Sector Not Found, C-1
Seek Error, C-1
segment, 7-4
SET command, 4-79
Set screen mode, B-7
SHELL command, 4-33
SHIFT command, 4-17
SORT command, 4-80

starting
  DOS, 1-5
  LINK, command line, 7-8
    prompts, 7-7
    response file, 7-9
syntax,
  command, 3-2, 4-1
  EDLIN, 6-4
  pathname, 2-11
SYS command, 4-82

TIME command, 4-84
TREE command, 4-86
TYPE command, 4-87

Universal characters, 2-3
user notes, 8-1

VER command, 4-88
VERIFY command, 4-89
VM.TMP file, 7-2
VOL command, 4-90
volume label, 1-9

Write Fault Error, C-4
Write Protect Error, C-4
working directory, 2-11

NCR

NCR PERSONAL COMPUTER

OWNER'S MANUAL

- How Do You Rate This Documentation?

	Clarity	Accuracy	Organization	Completeness	Usability
Excellent (Needs No Improvement)					
Good (Could Use Minor Imptovements)					
Fair (Should Have Major Improvements)					
Poor (Should Be Completely Changed)					

- How Often Do You Use This Document?

Often ____ Occasionally ____ Rarely ____

Your Name: ____

Position: ____

Company: ____

Address: ____

Date: ____ Tele. No. ____

Comments:

NCR GmbH
Technical Publications Department
Ulmer Strasse 160 A
8900 Augsburg
Federal Republic of Germany

NCR
Personal Computer Division
Augsburg, Germany

017—0043182