

NCR

**Personal
Computer**

NCR-DOS

Microsoft, MS and GW (MS™-DOS, MS™-MACRO Assembler and GW™-BASIC) are registered trademarks of Microsoft Corporation; TUTOR is a product of Comprehensive Software Support, Inc. IBM is a registered trademark of International Business Machines Corporation. Dr. Graph and GSX are trademarks of Digital Research Inc.

Copyright © 1982, 1983, 1984, 1985 by Microsoft Corporation

Copyright © 1984 by NCR Corporation

Dayton, Ohio

All Rights Reserved

Printed in the Federal Republic of Germany

Second Edition, March 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 NCR representative or NCR office for information that is applicable and current.



CUSTOMER PROGRAM LICENSE AGREEMENT

YOU SHOULD CAREFULLY READ THE FOLLOWING TERMS AND CONDITIONS BEFORE OPENING THIS PACKAGE. OPENING THIS 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.

This License Agreement applies to the Program contained in the accompanying package. Unauthorized copying is prohibited by the Copyright Law. YOU MAY NOT USE, COPY, MODIFY OR TRANSFER THE PROGRAM, IN WHOLE OR IN PART, EXCEPT AS EXPRESSLY PROVIDED FOR IN THIS LICENSE AGREEMENT. The Program is or may be considered by the copyright owner or licensor of NCR as confidential, proprietary and a trade secret and should be safeguarded by you as such. Title to the Program and copies of it remains with the copyright owner.

LICENSE

You may:

- a. use the Program only on a single machine at a single location, unless the Program has been specifically designated by NCR, in writing or on this package, for your use on a group of machines.
- b. copy the Program into any machine readable or printed form for backup or modification purposes only, to support your use of it on the single machine or designated group of machines. (Certain programs may include mechanisms to limit or inhibit copying. They may be designated by a "Copy Protected" notice);
- c. modify the Program for your use on the single machine or designated group of machines. (Any portion of the Program merged into another program will be considered to be a modification and will be subject to the terms and conditions of this License Agreement); and
- d. transfer the Program and license to another party only if the other party agrees to accept the terms and conditions of this License Agreement. You must advise NCR of the name and address of the other party and the other party must accept the terms of this License Agreement by signing a copy of it and providing NCR with the signed copy. If you transfer the Program, you must at the same time destroy all copies whether in printed or machine-readable form which you have not transferred to the other party and this includes all modifications of the Program (including portions of it 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.

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

TERM

Your license will be effective until terminated. You may terminate it at any time by destroying the Program, including all copies, modifications and merged portions in any form. It will also terminate upon conditions set forth elsewhere in this License Agreement or if you fail to comply with any term or condition in this License Agreement. You agree that upon any such termination you will destroy the Program, including all copies, modifications and merged portions in any form.

EXCLUSION OF WARRANTY

EXCEPT AS STATED IN THE "LIMITED WARRANTY" BELOW, THE PROGRAM IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU (AND NOT NCR OR ITS DEALER OR DISTRIBUTOR OR ANY LICENSOR OF NCR OR OWNER OF THE PROGRAM) ASSUME THE ENTIRE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION. There is no warranty by NCR or any other party or person that the functions contained in the Program will meet your requirements or that the operation of the Program will be uninterrupted or error free. You assume all responsibility for the selection of the Program to achieve your intended results, and for the installation, use and results obtained from it.

LIMITED WARRANTY

NCR warrants both the media on which the Program is reproduced and the reproduction of the Program on the media 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 repair or replacement of any media not meeting NCR's "Limited Warranty" and which is returned to NCR or an authorized NCR dealer or distributor within the 90-day period, with a copy of your receipt, or
2. If NCR or its authorized dealer or distributor is unable to deliver replacement media and repair is not practicable or cannot be timely made, you may terminate this License Agreement by returning the Program and your money will be refunded.

IN NO EVENT WILL NCR OR ANY OTHER PARTY OR PERSON BE LIABLE TO YOU OR ANYONE ELSE FOR ANY DAMAGES, INCLUDING LOST PROFITS, LOST SAVINGS OR OTHER INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM OR MEDIA EVEN IF NCR OR THE OTHER PARTY OR PERSON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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 limitations 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 other rights which vary from state to state.

MISCELLANEOUS

You may not sublicense, assign or transfer the license or the Program except as expressly provided in this License Agreement. Any attempt otherwise to sublicense, assign or transfer any of the rights, duties or obligations hereunder is prohibited, and will automatically terminate your license and right to use the Program.

This License Agreement will be governed by the laws of the State of Ohio where NCR Corporation has its principal office.

YOU ACKNOWLEDGE THAT YOU HAVE READ THIS AGREEMENT, UNDERSTAND IT AND AGREE TO BE BOUND BY ITS TERMS AND CONDITIONS. YOU FURTHER AGREE THAT IT IS THE COMPLETE AND EXCLUSIVE STATEMENT OF THE AGREEMENT BETWEEN US AND THAT IT SUPERSEDES ANY PROPOSAL OR PRIOR AGREEMENT, ORAL OR WRITTEN, AND ANY OTHER COMMUNICATIONS BETWEEN US OR BETWEEN YOU AND ANY NCR DEALER OR DISTRIBUTOR RELATING TO THE SUBJECT MATTER OF THIS LICENSE AGREEMENT.

Should you have questions concerning this License Agreement or should you desire to obtain information on warranty performance, you may contact NCR by writing to:

NCR CORPORATION
1150 Anderson Drive
Liberty, SC 29657
USA

HOW TO USE THIS MANUAL

Before you use this manual you should be familiar with the information contained in your NCR *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.

You also may have inserted the NCR PAL familiarization diskette into your disk drive and explored each of its major topics. Now you are ready to learn about NCR-DOS. (NCR-DOS is NCR's version of Microsoft® Corporation's MS™-DOS.) To make a backup copy of your NCR-DOS master diskettes (included in the back of this manual), you need 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 to look for is the indication that the disks are double-density, preferably double-sided (though single-sided is acceptable for everything but 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 term "diskette" is used in these manuals when referring to flexible disks. 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 1 tells you how to begin using NCR-DOS and how to format and make backup copies of your disks. It discusses disk drive configurations and identification, and leads you through some frequently performed operations. It also describes NCR User Interface and 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. Use of the RAM-DISK (memory-resident files and directories) feature is also discussed. Chapter 3 teaches you about commands — types, options, entry conventions (standard formats and procedures that apply to all commands), batch files, and the configuration change feature. Chapter 4 describes (in reference format) each NCR-DOS system, batch and configuration command

individually, giving you its purpose, its format (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.

After you have read the first four chapters, it would be valuable for you to load and run the NCR TUTOR diskette, which is in the back of this manual along with the NCR-DOS diskettes. You will find the NCR TUTOR diskette both a helpful review and a new insight into NCR-DOS. To load TUTOR, insert the diskette into the "A" drive and turn on your NCR PERSONAL COMPUTER. Major Topics 5 and 6 of NCR TUTOR are the subject of Chapters 5 and 6 of this manual, so you should skip them until you have read these corresponding chapters.

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 can 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. Chapter 8 describes the NCR DEBUG Utility that provides a controlled testing environment for binary and executable object program files.

Appendices to this manual include a set of command systems summary charts, guidelines for buying, installing and using applications software, information on disk errors and messages and how you can respond to them, and instructions for running a RAM.DISK demonstration program. Information for installing international keyboard drivers and the GSX-86 CRT drivers are also included.

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 four 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

- The NCR User Interface which offers assistance in entering commands
- The NCR TUTOR diskette, any portion of which can be studied independently

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 MS-MACRO ASSEMBLER package (which includes the *NCR-DOS PROGRAMMER'S MANUAL* from NCR or from your NCR Personal Computer dealer.

NCR-DOS

Contents

CHAPTER 1 WORKING WITH DOS

UNDERSTANDING THE SOFTWARE'S ROLE.	1-1
What Is NCR-DOS?	1-1
What Is An Operating System?	1-1
What Is NCR User Interface	1-2
IDENTIFYING YOUR DISK DRIVES	1-2
How DOS Assigns Disk Drive Designations	1-2
Using Flexible/Fixed Disk Systems	1-3
GETTING STARTED	1-3
PERFORMING FREQUENT OPERATIONS	1-4
Making Keyboard Entries	1-4
Loading DOS/NCR User Interface	1-5
Entering Date And Time.	1-6
Displaying The Directory	1-6
Formatting A Blank Diskette	1-6
Changing The Default Drive	1-6
Backing-Up Your Disks	1-7
Using The Programmable Function Keys	1-7
Running Programs Automatically	1-8
Running Multiple Operating Systems	1-8
Turning The System Off	1-8
IF YOU HAVE A FIXED DISK.	1-8
Partitioning The Fixed Disk	1-9
DEFINING A SERIAL PRINTER.	1-9
NCR USER INTERFACE	1-9
CHAPTER 2 FILES AND DIRECTORIES	
WHAT ARE FILES	2-1

How DOS Keeps Track Of Your Files	2-1
The DIR (Show Directory) Command	2-2
Naming Your Files	2-3
Using Shorthand File Names	2-4
Naming Files According To DOS Conventions	2-5
Copying Files	2-8
Protecting Files	2-9
ORGANIZING FILES INTO MULTIPLE DIRECTORIES.	2-9
Using Pathing With Filenames	2-12
Pathnames	2-12
Using Pathing With External Commands	2-14
Using Pathing With Internal Commands	2-14
Displaying Your Working Directory	2-15
Creating A Directory	2-16
Changing Your Working Directory	2-16
Removing A Directory	2-16
RAM-DISK (MEMORY-RESIDENT) FILES AND	
DIRECTORIES	2-17
How RAM-DISK Simulates A Flexible Disk Drive	2-17
Defining RAM-DISK(S) Using The Configuration	
Change Feature	2-18
RAM-DISK Drive Designation(s)	2-19
CHAPTER 3 LEARNING ABOUT COMMANDS	
TYPES OF DOS COMMANDS.	3-1
Internal Commands	3-1
External Commands	3-2
THE CONCEPT OF FORMAT (SYNTAX)	3-2
DOS Format Notation	3-3
Command Options	3-4
Common Entry Conventions	3-5
INPUT AND OUTPUT	3-6
Redirecting Your Output	3-6
Filters	3-7
Command Piping	3-7
BATCH PROCESSING	3-9
Definition Of A Batch (.BAT) File.	3-9
The AUTOEXEC.BAT File	3-11

Creating An AUTOEXEC.BAT File	3-13
Multiple-Use .BAT Files	3-14
Creating .BAT Files with Replaceable Parameters	3-14
Executing .BAT Files with Replaceable Parameters	3-16
BATCH COMMANDS	3-18
Table Of Commands	3-18
CONFIGURATION CHANGE FEATURE	3-19
The Configuration (CONFIG.SYS) File	3-20
Creating The CONFIG.SYS File	3-21
Changing The CONFIG.SYS File	3-22
Device Driver Routines	3-22
ANSI.SYS Extended Screen And Keyboard Functions Driver	3-22
RAMDISK.SYS Driver For Memory-Resident	
CONFIG.SYS COMMANDS	3-23
Table Of Commands	3-24
CHAPTER 4 DOS COMMANDS	
Format Notation Review	4-1
Table Of Commands	4-3
ASSIGN	4-7
AUTOEXEC.BAT	4-9
BACKUP	4-10
BREAK	4-13
BUFFERS	4-14
CHDIR	4-15
CHKDSK	4-16
CLS	4-20
COMMAND	4-21
COMP	4-22
CONFIG.SYS	4-24
COPY	4-25
CTTY	4-28
DATE	4-29
DEL	4-31
DEVICE	4-32
DIR	4-36
DISKCOMP	4-38
DISKCOPY	4-39
ECHO	4-41
ERASE	4-42
ESC	4-42a

EXE2BIN	4-43
EXIT	4-44
FDISK	4-45
FILES	4-51
FIND	4-52
FOR..IN..DO	4-54
FORMAT	4-55
GOTO	4-57
GRAFTABL	4-57a
GRAPHICS	4-58
HELP	4-59
IF	4-60
KEYB	4-60a
MKDIR	4-61
MODE	4-62
MORE	4-69
PATH	4-70
PAUSE	4-71
PRINT	4-72
PROMPT	4-74
RECOVER	4-76
REM	4-77
RENAME	4-78
RESTORE	4-79
RMDIR	4-82
SET	4-83
SHELL	4-84
SHIFT	4-85
SORT	4-86
SYS	4-88
TIME	4-89
TREE	4-91
TYPE	4-92
VER	4-93
VERIFY	4-94
VOL	4-95

CHAPTER 5 EDITING, FUNCTION KEYS, THE CURSOR AND DISPLAY GRAPHICS

SPECIAL EDITING KEYS	5-1
Programming Of Editing Keys	5-1
Editing Of DOS Commands	5-4
Examples	5-4
Programming The Keyboard	5-10

Programming Of Function Keys	5-10
Programming Alternate Mode Of Alphabetic Keys . . .	5-11
Restoring The System Prompt	5-11
Examples	5-11
Programming The Cursor and Display Graphics . . .	5-13
Programming The Cursor	5-14
Programming The Display Graphics	5-16

CHAPTER 6 LINE EDITOR (EDLIN)

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	6-5
line-number	6-8
A(PPEND)	6-10
C(COPY)	6-11
D(ELETE)	6-13
E(ND)	6-15
I(NSERT)	6-16
L(IST)	6-19
M(OVE)	6-22
P(AGE)	6-23
Q(UIT)	6-24
R(EPLACE)	6-25
S(EARCH)	6-28
T(RANSFER)	6-31
W(RITE)	6-32
ERROR MESSAGES	6-33

CHAPTER 7 LINK PROGRAM (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 Prompts	7-11
Link Options	7-13
 SAMPLE LINK SESSION	 7-15
 ERROR MESSAGES	 7-17
 CHAPTER 8 DEBUG UTILITY	
INTRODUCTION	8-1
Overview Of DEBUG	8-1
 COMMANDS	 8-2
Command Conventions	8-2
Command Parameters	8-2
Table Of Commands	8-6
Command Descriptions	8-8
A(SSEMBLE)	8-9
C(OMPARE)	8-11
D(UMP)	8-12
E(NTER)	8-14
F(ILL)	8-16
G(O)	8-17
H(EX)	8-19
I(NPUT)	8-20
L(OAD)	8-21
M(OVE)	8-23
N(AME)	8-24
O(UTPUT)	8-26
Q(UIT)	8-27
R(EGISTER)	8-28
S(EARCH)	8-31
T(RACE)	8-32
U(NASSEMBLE)	8-33
W(RITE)	8-35
 ERROR MESSAGES	 8-37
 SYSTEM REQUIREMENTS	 8-38

APPENDIX A COMMAND SUMMARY REFERENCE	
CHARTS	A-1
NCR-DOS COMMANDS	A-2
BATCH COMMANDS.	A-6
CONFIGURATION COMMANDS	A-6
EDLIN COMMANDS	A-7
DEBUG COMMANDS.	A-8
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
GLOSSARY	G-1
INDEX	

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* (DOS) 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 original DOS master diskettes, the two you received with this book, contain 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 diskettes, and format other new disks. Later, in the Learning about Commands chapter, you learn the various conventions used in DOS commands. In the DOS Commands chapter, each DOS system command is fully described in a reference format.

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 computer 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.

WHAT IS NCR USER INTERFACE?

NCR DOS makes the task of learning and using DOS easy through the use of a User Interface (UI) which intercepts the user's commands and provides assistance in entering a command. You can, at the same time, leave the current command, seek help from the "HELP" utility and return to the command without losing earlier entries. There are three levels of assistance, the higher the level, the greater the assistance. The default level is three.

All three levels are under the control of a command preprocessor (UI) which controls interpretation of the input to and output from DOS. Commands are still executed by DOS. This UI resides on the NCR-DOS diskette (version 2.11.30 or higher) and is loaded as a part of DOS.

A detailed description is not given in this manual because the UI is self-explanatory through its menus and prompts. See the NCR User Interface section at the end of this chapter.

IDENTIFYING YOUR DISK DRIVES

Before actually loading your software, you may need to know a little more about the NCR PERSONAL COMPUTER 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.

HOW DOS ASSIGNS DISK DRIVE DESIGNATIONS

When you copy your master diskette, or load almost any software or data, you are directed to insert or change a diskette. 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, DOS assigns each drive its own designation. Consider the following disk configuration examples:

- You have two flexible disk drives. One drive is designated A; the other drive is designated B.
- You have one flexible disk drive. The drive designated as both A and B by DOS.
- You have one flexible disk drive and one fixed disk unit. The flexible disk drive is designated as both A and B by DOS. The fixed disk unit is designated C.
- 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.

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 assigns both letters to the one flexible disk. If you have a fixed disk drive, it is designated C. Any additional freestanding disk units you may add are designated D, E, and so on.

USING FLEXIBLE/FIXED DISK SYSTEMS

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 probably use the flexible disk drive only to format and make copies of flexible disks. In these situations, DOS always "prompts" you to change diskettes and waits for you to insert the new diskette. You then continue processing by pressing any key.

GETTING STARTED

The NCR-DOS master disks (or diskettes), the two you received with this book, contain all operating system software files and all commands and programming tools. For your protection, master diskette 1 requires you to make a copy of NCR-DOS. You should also make a copy of diskette 2. After DOS is loaded, screen instructions lead you through the procedure.

Insert the DOS disk (1) 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. The time it takes for DOS to load depends on the system configuration (see *LOADING DOS/User Interface* later in this chapter).

After the system prompt *A>* appears, follow the instructions that appear next on the screen. Label your new copy as the new master NCR-DOS diskette (1). Always use this diskette for processing.

NCR-DOS diskette 2 contains the Basic interpreter as well as some programming utilities. The Basic interpreter resides on the diskette under four names: *BASIC.EXE*, *BASICA.EXE*, *GWBASIC.EXE*, and *GW-BASIC.EXE*. Through a special procedure all these files share the same storage area. To transfer this interpreter to another diskette, use the 'COPY' command under NCR-DOS (see the *DOS Commands* chapter). It is only necessary to copy one of these files over to another diskette; as all four files are the same — only the names are different. You may choose the file name you are most comfortable with. You can also make a backup copy of this diskette by using the *DISKCOPY* command (see the *DOS Commands* chapter).

If you have a fixed disk, complete setup procedures as described in "If you have a Fixed Disk" at the end of that chapter.

PERFORMING FREQUENT OPERATIONS

NCR User Interface easily guides you through many of the most frequently performed operations. There are, however, some procedures that you still need to understand.

MAKING KEYBOARD ENTRIES

To do this, you must know the system command format notation used. The concept of command format, and a complete list of the format notations used in 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 terminate an entry you just made. These keys, which are equivalent in function, have a short vertical line on the right connected to a left-pointing arrow <←> with the word <ENTER>. To relate to a commonly-used term, these keys are referred to as the <ENTER> keys. The notation <ENTER> is used when either one of these keys is pressed. Do not confuse the <ENTER> <←> keys with the backspace keys, all of which have a left-pointing arrow without the vertical line (←).

Most entries you make while using DOS system commands, creating batch files or CONFIG.SYS commands, or using the Line Editor (EDLIN) or NCR-LINK, must be terminated with the <ENTER> key. The most common exception is when the software calls for a Y/N (yes/no) response. In this case, and occasionally in other situations, the software tests the entered character(s) and proceeds according to the value entered without waiting for entry of a termination key.

In this manual, termination of each entry, except a Y/N, by pressing the <ENTER> key is assumed. The <ENTER> key is shown in the format or example only when critical to your understanding.

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.

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.

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.

LOADING DOS/NCR USER INTERFACE

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

Insert the DOS diskette (1) in drive A, turn the drive access lever a quarter turn to the right, and turn on your computer. This is always the standard startup procedure. The time it takes for DOS to load depends on the system configuration.

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 DOS *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

You should adopt the practice of entering date and time 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.

Control Break Error Handling

NCR User Interface helps you enter information in an easy manner. However, when you are in normal DOS systems, if you discover you have made a mistake after pressing the <ENTER> key, 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” followed by >. You can then re-answer the prompt or type another command. To correct a line before you press <ENTER>, use the backspace (←) key to erase one letter at a time, or press <Escape> to delete the current entry and reenter the line.

DISPLAYING THE DISKETTE DIRECTORY

To find out what you have stored on a disk or diskette, use the DIR command under NCR-DOS (see the *DOS Commands* chapter).

FORMATTING A BLANK DISKETTE

Every new diskette must be formatted to allow the diskette to accept data for storage. Use the FORMAT command under NCR-DOS (see the *DOS Commands* chapter).

CHANGING THE DEFAULT DRIVE

The “A>” is the DOS prompt from the command processor when in normal DOS system. It tells you that DOS is ready to accept commands. The A in the prompt is the default disk drive. This means that DOS searches only the diskette in drive A for any filenames you enter and writes files to that diskette 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 or follow the procedure in User Interface.

```
A>
```

```
A>B: (you type B: and press <ENTER> in response to the prompt)
```

```
B>
```


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

```
B>A:  
A>
```

BACKING-UP YOUR DISKS

You've made a backup copy of your master software diskette; it is a good policy to make backup copies of all your disks. If a disk becomes damaged or if files are accidentally erased, you still have all of the information on your backup diskette(s).

You can make backup copies of flexible disks with the DISKCOPY or COPY command; you make backup copies of fixed disks with the BACKUP command. These commands are discussed in detail in the *DOS Commands* chapter.)

USING THE PROGRAMMABLE FUNCTION KEYS

Your standard NCR PC keyboard 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, Function Keys, the Cursor and Display Graphics* chapter.

NCR User Interface provides an easy way of defining these keys.

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 bring the system up, the command processor always searches for a file named AUTOEXEC.BAT on the DOS disk. This file, if present, contains a set of commands and/or program names that DOS runs each time DOS is started. The *Learning About Commands* chapter tells you how to create an AUTOEXEC.BAT file.

RUNNING MULTIPLE OPERATING SYSTEMS

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 reserve the desired portion of the available space for the exclusive use of DOS with the FDISK command as described in the "Partitioning The Fixed Disk" section of this chapter. FDISK is more fully described in the *DOS Commands* chapter.

TURNING THE SYSTEM OFF

There is no specific command in DOS to turn the system off. To end your session, wait until any current processing function is completed, and the default drive prompt ("A>" or other) appears. Place the diskette(s) in their protective jacket and store appropriately.

IF YOU HAVE A FIXED DISK

If you have one or more fixed disk drives (also called hard disks), you must perform extra setup procedures by defining your disk configuration. (If you have only flexible disks, you can skip this section.) FDSET.UP, a self-guiding batch file, is provided on your NCR-DOS diskette 1 for this purpose. To invoke FDSET.UP, for initially setting up drive C, select the appropriate option in the main menu of UI.

When you move your computer, you should use the PARK utility. This prevents possible damage to data on your hard disk.

PARTITIONING THE FIXED DISK

If you want DOS to share a fixed disk drive with one or more other operating systems, you can allocate portions of the disk's storage space to each operating system. These allocated areas are called partitions. A maximum of four partitions is possible on the fixed disk. You can make the partitions any size (the total of partition sizes can

not exceed the storage capacity of the disk). After partitioning the disk, the partitions must be formatted with the operating system you want.

FDISK creates, changes or deletes a partition only for DOS. Other operating systems may each create, change or delete their own partitions.

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

- Keep compatible operating system software, application software, and data on 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 A SERIAL PRINTER

If you are 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.

NCR USER INTERFACE

NCR DOS operates with NCR User Interface (UI); UI resides on the NCR DOS diskette (version 2.11.30 or higher). UI operates at three levels of assistance; the higher the level, the greater the assistance. Initially, the system comes up in the Main Menu mode (level 3). It will always come up in level 3 unless you change the assistance level.

The Main Menu (level 3) can be entered from any level and provides you with various helpful features including: a way to execute an application with a few keystrokes and a way to define up to 40 function keys. You can set the assistance level to one that is comfortable for you.

DOS commands can be executed in all three levels. Level 3 is completely menu driven. Level 2 provides a command line, a command syntax, and examples of the command upon request. Assistance is provided in both level 2 and 3 upon request or when entry errors are discovered in the command parameters. Level 1 provides no prompts or guidance at all.

In addition to User Interface, your NCR DOS diskette contains a HELP utility which contains information on various DOS commands. The HELP utility can be invoked in all three levels and may be entered at any time, even in the middle of entering a command. If HELP and the command are entered, an appropriate description for the command is displayed. If just HELP is entered the command list is displayed. Upon exiting HELP, you are returned to the place you were before invoking HELP, with no loss of entries.

HELP can also be entered from the normal DOS system. Enter HELP and the command for a description of that command. For the command list to be displayed, enter HELP.

SHELUTIL Utility

This utility is menu driven and allows you to enable, disable, install or delete UI and turn the clock display on or off. After installing or removing UI, the system may be rebooted. SHELUTIL, however does not transfer files used in the HELP utility.

The function keys you may have defined while UI was installed will no longer be defined if UI is removed.

If you remove UI, you can still turn on the clock display. TIME.SYS is the user installable clock used by UI to display time. Create a CONFIG.SYS file (see the Command section) by entering "DEVICE=TIME.SYS". The operating system will install the clock software, addressed internally as INT1C, at boot time.

You can turn the display OFF/ON with the SHELUTIL utility or by entering:

```
COPY CON INT1C <Enter>  
0 <Ctrl-Z> <Enter>
```

To turn the display ON, enter a "1" instead of a "0" in line two.

If you experience difficulties in running applications, turn off the display. TIME.SYS takes over user timer interrupt hexadecimal 1C. Some applications may not run correctly with the display on or with TIME.SYS installed.

NCR User Interface Error Messages

Re-install UI with the SHELUTIL utility, if any of the following messages appear:

Bad or missing SHELLDRV.SYS

Bad or missing TIME.SYS

NCR User Interface Error xx — If xx=53, the disk is full and some unneeded files should be deleted to make room. If xx is other than 53, insert the UI diskette in the drive or install UI.

Reserved File Names

The following names are reserved for use with UI and HELP. They cannot be used as filenames in the root directory with UI installed:

HELP.EXE	SHELL2.OVR
INT1C	SHELL3.OVR
NCR.DIR	SHELL4.OVR
NCRDIR.DCD	SHELLDRV.SYS
NCRSCRN	SHELLHLP.DTA
SHELL IF	SHELLRES.EXE
SHELL.TMP	TIME.SYS
SHELL1.OVR	



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.

WHAT ARE FILES?

A file is a collection of related information. A file on your disk can be compared to a set of file folders in a file cabinet drawer. For example, one file drawer might contain a folder for each employee in the office with a name and address card, employment application, work and salary history record, copies of annual performance reviews, and any other appropriate data. You might refer to this drawer as the Employee Master File. Similarly, a file on your disk could contain a "record" for each employee (corresponding to a folder). This record could contain data elements, or "fields", for name, address, phone number, date of employment, salary, position title, performance rating, etc. You could also call this file your 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 on your computer. 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 one or more directories on each disk. A disk directory can be thought of as a machine-readable table of contents. These directories also contain information on the sizes of the files, their locations 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, and is located on the disk drive indicated by the system prompt (A>).

An additional system area is called the File Allocation Table. It keeps track of the locations 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.

USING 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 named files in the current directory on the disk. For example, if your DOS diskette is in drive A and you want to see the listing for the current directory on that diskette, type:

DIR[A:]

NOTE: This example includes another type of format notation with which you need to become familiar. The square brackets [] enclose an entry which is optional. Assuming that A is the current default drive, this directory listing request can be entered as either:

DIR
or
DIR A:

DOS responds with a directory listing of all the files in the current (A:) directory on your DOS diskette. 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" system 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

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 the *DOS Commands* chapter.

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 consists of a period and 1 to 3 characters. You can type any filename or extension 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 diskette in drive B (when drive B is NOT the default drive), type the following command:

A> 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 diskette in drive A for the filename **NEWFILE.EXE** and so the drive designation is not necessary if the file is on drive A. A drive designation is needed only 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. Valid characters for filename extensions are the same as those for filenames. Here is a complete list of the characters you can use in filenames 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 occasionally in this manual to indicate the complete filename format:

[drive designation:][pathname]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.EXE
A:NEWFILE
TEXT

Pathnames are discussed later in this chapter.

USING SHORTHAND FILE NAMES

Two special characters (called universal or 'wildcard' characters) can be used in filenames and extensions: the question mark (?) and the asterisk (*). These special characters give you a shorthand method of selecting certain filenames when used with DOS commands.

Using The Question Mark ? Character — A question mark (?) specified in any character position 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????.???

displays:

TEST1RUN.EXE
TEST2RUN.EXE
TEST3RUN.EXE
TESTSALL.EXE

However, the command:

DIR TEST1RUN.??

does not display TEST1RUN.EXE because only two characters of filename extension are indicated by "??".

Using The Asterisk * Character — An asterisk (*) specified in any character position in a filename or filename extension indicates that any character can occupy that position or any of the remaining 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 diskette 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 Files According to DOS Conventions

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

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 either input from the keyboard or output to the computer console (screen).

LST

or

PRN

Used when referring to the printer device. These may be used as filename extensions but never as filenames. PRN can be used in place of a filespec to define the desired destination for data as discussed in the "Redirecting Your Output" section of the *Learning About Commands* chapter; an example is also shown in the description of the DIR command in the *DOS Commands* chapter.

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 you create.

The filename, and especially the filename extension, is 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 input (source coding) for MS™-MACRO Assembler programming language
- BAK — Line Editor or other backup copy
- BAS — BASIC language program (source coding)

- BAT — An executable batch file
- BIN — Binary image version of an executable object code file (EXE2BIN default)
- COB — Compiler input (source coding) for COBOL programming language
- COM — Command file on system disk
- C — Compiler input (source coding) for "C" programming language
- DAT — Common usage for data files
- DOC — Documentation for software
- EXE — Any executable object code file
- FOR — Compiler input (source coding) for FORTRAN programming language
- LIB — Linker library file input
- MAC — Macro filename
- MAP — Linker listing default
- OBJ — Compiler output for most programming languages (object code)
- OVR — File overlay
- PAS — Compiler input (source coding) for PASCAL programming language
- 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 get the error message "FILE CANNOT BE COPIED ONTO ITSELF". You can not make a copy of a file on the same disk unless you specify a different filename for the new copy.

The simplified format of the COPY command is:

COPY filespec1 [filespec2]

Where = filespec1 is the file from which the copy is to be made.
filespec2 defines the location and/or the new name for the new copy.

For example:

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

copies the file MYFILE.TXT on diskette A to a file that is named MYFILE.TXT on diskette B. A duplicate copy of MYFILE.TXT now exists on drive B.

Figure 2.1 illustrates what a COPY command does.

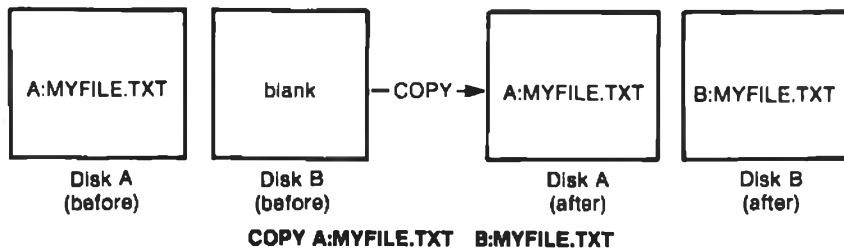


Figure 2.1 Copying files to another diskette

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 diskette A, one named MYFILE.TXT and the other named NEWNAME.TXT. Figure 2.2 illustrates this example.

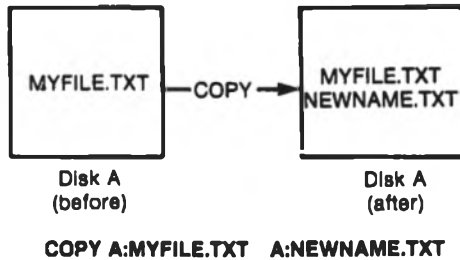


Figure 2.2 Copying files on the same diskette

You can also copy all the 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 *DOS 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 can not 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 diskettes 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.

ORGANIZING FILES INTO MULTIPLE 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 sizes 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 listed 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 multiple directories. These directories are a way of dividing your files into convenient groups of files. For example, you may want all of your accounting files listed in one directory and text files in another. Any one directory can contain the names of any reasonable number of files, and it may also contain the names of 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 on 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 root 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 contained in a different directory.

Figure 2.3 illustrates a typical hierarchical directory structure

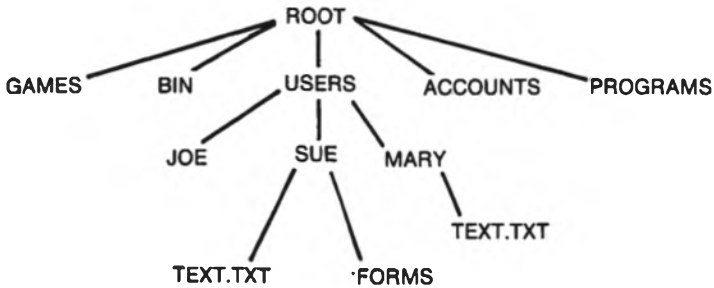


Figure 2.3 A sample hierarchical directory structure

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 *DOS 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 “Using Pathing With External Commands” section of this chapter for a description of the possible use of such a 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 all have their own directories which are sub-directories of the USERS directory. Sue has a sub-directory named FORMS under the SUE directory. Sue and Mary both have files named TEXT.TXT listed in their directories. 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. However, 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 backslash mark (\) is used to separate directories from other directories and files.

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

```
DIR \USERS\MARY
```

USING PATHING WITH FILENAMES

When you use hierarchical directories, you must tell DOS where your files are located in the directory structure. Both Mary and Sue, for example, have files named TEXT.TXT. Each must tell DOS in which directory her file resides if she wants to access it. One way to do this is by preceding the filename[.ext] with a pathname.

Pathnames

A simple filename is a sequence of characters that can optionally be preceded by a pathname. A pathname is a sequence of one or more directory names, each separated from the previous one by a backslash (\). The pathname is followed in many cases by a simple filename (plus extension if applicable), separated from the last directory name by a backslash.

The format of pathnames is:

```
[ \ [directory] [ \ directory ] ... ] \
```

where:

... (ellipses) indicate that the [\ directory] entry is repeated as many times as needed

and:

the final backslash [\] is present only if filename[.ext] follows

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 plus the filename[.ext] of Sue's TEXT.TXT file is \USERS\SUE\TEXT.TXT.

When you are in your working directory, a filename[.ext] alone or the pathname plus the filename[.ext] may be used interchangeably. The following list shows some sample names:

\
Indicates the root directory.

\ PROGRAMS

Sub-directory under the root directory; containing program files.

\ USERS \ SUE \ FORMS \ 1A

A typical full pathname plus filename[.ext]. This one designates a file named 1A in the directory named FORMS which is a sub-directory of the directory named SUE. SUE is a sub-directory of the directory named USERS, which is a sub-directory under the root directory.

MARY

A relative pathname; it names the sub-directory MARY when the sub-directory USERS is the working directory. If the working directory is the root (\), the entry to access MARY is **\ USERS \ MARY**.

TEXT.TXT

Name of a file when the working directory is either SUE or MARY.

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 any directory (except the root) is made.

.. (double period)

The shorthand name of the parent working directory.

If you type:

DIR..

then the file names listed in the parent directory of your working directory are displayed. If you type:

DIR..\..

Then the file names listed in the directory of the parent directory are displayed.

Using Pathing With 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 in 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. If the external commands are not in your working directory, you must tell DOS in which directory they 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 to DOS that the \BIN path be used to find the FORMAT command. The command

```
PATH \BIN
```

tells DOS to search in your working directory, the root directory and the \BIN directory for all commands. You only have to specify this PATH once during your computer session. DOS then searches in \BIN for each external command if it does not find it in the working directory or the root directory. If you want to know what the current PATH is, type the word PATH and the current value of PATH is displayed.

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

Using Pathing With 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 pathname1 pathname2
```

If both pathnames specified are existing directories, all files are copied from the first directory into the second 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 contents of the directory for a specific path.

TYPE pathname

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

DISPLAYING YOUR WORKING DIRECTORY

As mentioned earlier, DOS looks for all commands that you issue in your working directory. If you are confused about which directory you are in, you can find out the name of the directory by issuing the **CHDIR** (Change Directory) command with no options. For example if your current directory is `\USERS\JOE`, when you type:

CHDIR

you see:

A: \USERS\JOE

This is your current drive designation and your 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 ID
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 ID for this diskette was not assigned when the diskette was formatted. Note that both files and directories are listed in this output. The '.' indicates the working directory \USERS\JOE, and '..' is the shorthand notation for the parent directory \USERS. As you can see, Joe has another directory in this "tree" structure named TEXT. FILE1.COM is a file in the \USERS\JOE directory. All of these directories and files reside on the diskette 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 can not 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 the *Line Editor (EDLIN)* chapter, 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 or CD \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.. puts 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 or RD NEWDIR

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

RMDIR \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 \USERS\SUE directory, type:

DEL \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 will learn more about DOS commands.

RAM-DISK (MEMORY RESIDENT) FILES AND DIRECTORIES

NCR-DOS includes a feature called RAM-DISK (Random Access Memory Disk) which allows you to store data files and directories in the computer memory rather than on a disk. You can read and write these files and directories using standard DOS commands such as COPY, DIR or TYPE.

HOW A RAM-DISK SIMULATES A FLEXIBLE DISK DRIVE

You can use DOS commands to access RAM-DISK because each area set aside for storing RAM-DISK data files and directories is considered by DOS to be logically equivalent to an additional flexible disk drive.

However, instead of routing each request for access to the "diskette" to the standard memory resident DOS flexible disk driver software routine, which in turn interfaces with a "real" diskette through the flexible disk drive controller circuit board, DOS routes requests for access to a RAM-DISK to the RAMDISK.SYS software driver routine, which interfaces directly with the area(s) set aside in memory.

Retrieving or writing data stored in a RAM-DISK is much faster than access to a "real" diskette because the access takes place at memory speed and CPU data transfer rates. Data transfer rates for an actual diskette are restricted by the rotational speed of the device. Data transfer can not begin until the read/write head mechanism has been moved to the correct circular track (this is called "seek time" on a disk) and sector within the track.

A RAM-DISK demonstration program is included on your NCR-DOS diskettes. This program shows you graphically the increase in processing speed that can be attained using the RAM-DISK feature. Instructions for running the demonstrations are in Appendix D of this manual.

Note that a RAM-DISK area is "volatile". That means that each time the computer is turned off or reset (<Ctrl-Alt-Del>), the data in the RAM-DISK is lost or destroyed. Therefore, before turning off or resetting your computer, make certain that any RAM-DISK files you wish to keep are copied to a real disk. The COPY command can be used to copy specific files to and from the RAM-DISK. The DISKCOPY command can also be used to copy an entire diskette to or from an identically sized RAM-DISK.

DEFINING RAM-DISK(S) USING THE CONFIGURATION CHANGE FEATURE

To use the RAM-DISK feature, you tell DOS to set aside one or more areas of computer memory as RAM-DISK(S). You do this by including one or more DEVICE=RAMDISK.SYS... entries in a CONFIG.SYS file on your DOS disk. See the "Configuration Change Feature" section in the *Learning About Commands* chapter of this manual for instructions on creating or changing a CONFIG.SYS file. See also the sections in that chapter on "Device Driver Routines" and "CONFIG.SYS Commands" for further description of the RAMDISK.SYS driver routine and the DEVICE=RAMDISK.SYS format.

Each time DOS is started up (each time the computer is turned on, or reset using <Ctrl-Alt-Del>), DOS tests for the presence of a CONFIG.SYS file. If the file is present, DOS reads each command in the CONFIG.SYS file and establishes its operating conditions accordingly. Any DEVICE=RAMDISK.SYS... entry in a CONFIG.SYS file causes DOS to set aside the amount of area specified in the DEVICE command for a RAM-DISK. One copy of the RAMDISK.SYS driver routine (whether one or more RAM-DISKS

are defined) is also loaded into memory to function as part of the memory resident DOS.

RAM-DISK DRIVE DESIGNATION(S)

DOS always assigns disk drive designations each time it is started up. RAM-DISK(S) are assigned the next available designation letter(s) after the standard assignments for real disk drives.

Two examples:

- You have a dual flexible disk drive machine, and you create a CONFIG.SYS file with two DEVICE=RAMDISK.SYS... entries. The real drives are, as always, designated A and B. The two RAM-DISKS are designated C and D.
- Your computer has one flexible disk drive and one integrated fixed disk drive. You add a DEVICE=RAMDISK.SYS... command to an existing CONFIG.SYS file which previously contained no RAM-DISK entries. DOS designates the flexible disk drive as both A and B, designates the fixed disk drive as C, and designates the RAM-DISK as D.

These designations are made each time DOS is started up until the CONFIG.SYS file is changed or deleted.



Learning About Commands

Commands are a way of communicating with the computer. By entering NCR-DOS commands on your keyboard you can direct 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

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 can not see these commands when you do a directory listing of your DOS disk; they are part of the memory resident command processor (COMMAND.COM). When you type these commands, they execute immediately. The following internal commands are described in the next chapter, DOS Commands.

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

NOTE: Where two alternate spellings of a command are equally acceptable to the command processor, the second spelling is in parentheses ().

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). .EXE files that you convert to memory image using `EXE2BIN` will be .BIN files unless you specify a different extension.

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

ASSIGN	GRAFTABL
BACKUP	GRAPHICS
CHKDSK	KEYB
COMMAND	HELP
COMP	MODE
DISKCOMP	MORE
DISKCOPY	PRINT
ESC	RECOVER
EXE2BIN	RESTORE
FDISK	SORT
FIND	SYS
FORMAT	TREE

THE CONCEPT OF FORMAT (SYNTAX)

Before you begin to use the DOS commands, you must understand the format notation. Format refers to the precise presentation sequence and structure that DOS recognizes for legitimate commands. Errors in format will cause DOS to respond with an appropriate error message. Other errors, more serious, can cause DOS to do something other than what you meant to tell it to do.

DOS FORMAT NOTATION

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

[] Square Brackets

Indicate that the enclosed entry is optional.

< > Angle Brackets

Indicate 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 , press the specific key whose keytop has that value printed on it.

When the angle brackets enclose two keytop names connected with a hyphen (-), 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 <ENTER> is used when the (ENTER) key is to be pressed. Second, since there is no universally-accepted symbol for a space (blank) character, the notation <space> is used in some instances where the entry of a space character is critical in the format.

{ } 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 braces are also enclosed in square brackets.

(. . .) ellipsis

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

| A Broken Bar

When used with a DOS filter, the broken bar indicates a pipe. (This feature is fully explained later in this chapter under "Filters".)

CAPS Capital Letters

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

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 1 to 8 character name for a disk file. The filename option does not include a disk drive designation or a filename extension.

.ext

Refers to an optional filename extension consisting of a period and 1 to 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]

Some commands permit the use of pathname as part of the filespec between the drive designation and filename.

pathname

Refers to a pathname in the following format:

[\]directory[[\]directory]...[\]

Where the final backslash [\] is present only if filename [.ext] follows.

option parameter

Refers to a selection of processing options (for example, /B or /P or an ON/OFF choice).

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 or 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:

```
COPY 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 the three parts of a file specification with delimiters, since the colon and the period already serve as delimiters.
5. When instructions say "Strike any key," you can press any alpha (A-Z) or numeric (0-9) key, the space bar, <ENTER>, or certain special character keys (refer to "Copying Your Master Software Diskette" in the *Working With DOS* chapter.)
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 <ENTER> 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 <Ctrl-Num Lock> to suspend the scrolling. Press any key to resume scrolling.
11. Editing and function keys can be used when entering commands. Refer to the *Editing, Functions Keys, the Cursor and Display Graphics* 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> (unless you have changed the prompt by issuing a PROMPT command).
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.

INPUT AND OUTPUT

DOS always assumes that input comes from the keyboard and output goes to the CRT display. However, the flow of command input and output can be redirected. Input can come from a file rather than the keyboard, and output can go to a file or to a line printer instead of to the CRT display. 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 CRT display. You can send this information to a file by using a greater-than sign (>) in your command. For example, the command:

DIR

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

DIR >MYFILES

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

appends your directory listing to a currently existing file named MYFILES. If MYFILES does not exist, it is created. To print the directory of the disk in the default drive on your printer, enter

DIR >PRN

It is often useful to have input for a command come from a file rather than from the keyboard. This is possible by using a less-than sign (<) in your command. For example, the command:

SORT <NAMES>LIST1

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 CRT display 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 CRT display 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 broken bar symbol (|). For example, the command:

DIR | SORT

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

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 and redirects (>) the output from SORT to the DIREC.FIL file rather than the console. DIREC.FIL contains a sorted listing of the directory on the default drive, since no other drive was specified in the command. To specify a destination drive other than the default drive, type:

DIR | SORT>B:DIREC.FIL

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

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, *DOS Commands* .

BATCH PROCESSING

DEFINITION OF A BATCH (.BAT) FILE

Often you may find yourself typing the same sequence of commands and/or program names 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 entered one at a time from the keyboard. Each batch file must be named with the .BAT extension, and can be executed by typing the filename with or without its extension.

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

Two of the DOS commands which are available for use expressly in batch files are 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 *DOS Commands* chapter.

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:

```

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

```

To execute each command in 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 as an individual command.

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

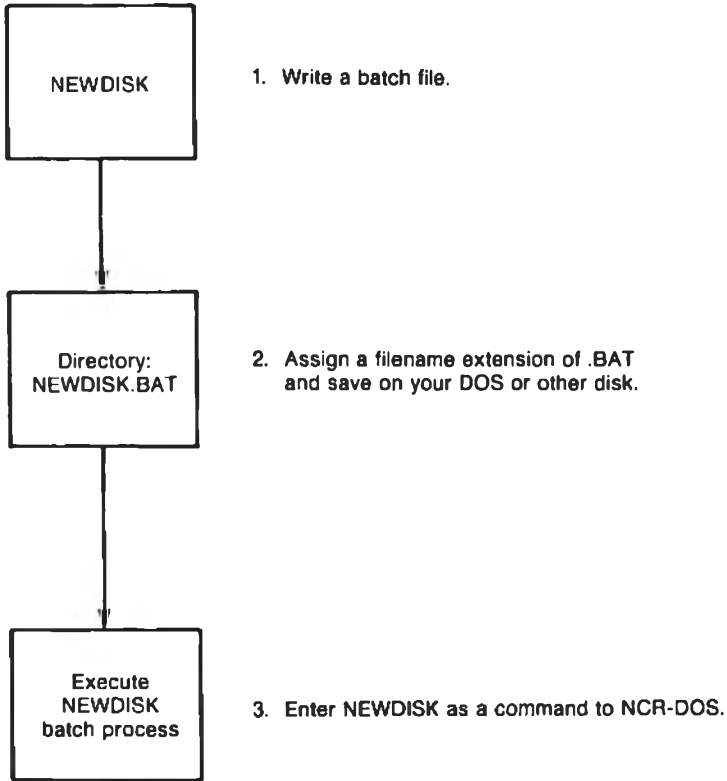


Figure 3.1

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 the filename to execute the batch file with or without 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 message 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 so that 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

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 DOS system is booted. This is convenient if you wish to execute a specific series of commands and/or programs 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.

Automatic program execution is useful when you want to run a specific application package or execute a standard set of utility functions (for example: DIR, CHKDSK) automatically each time you start the system.

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.

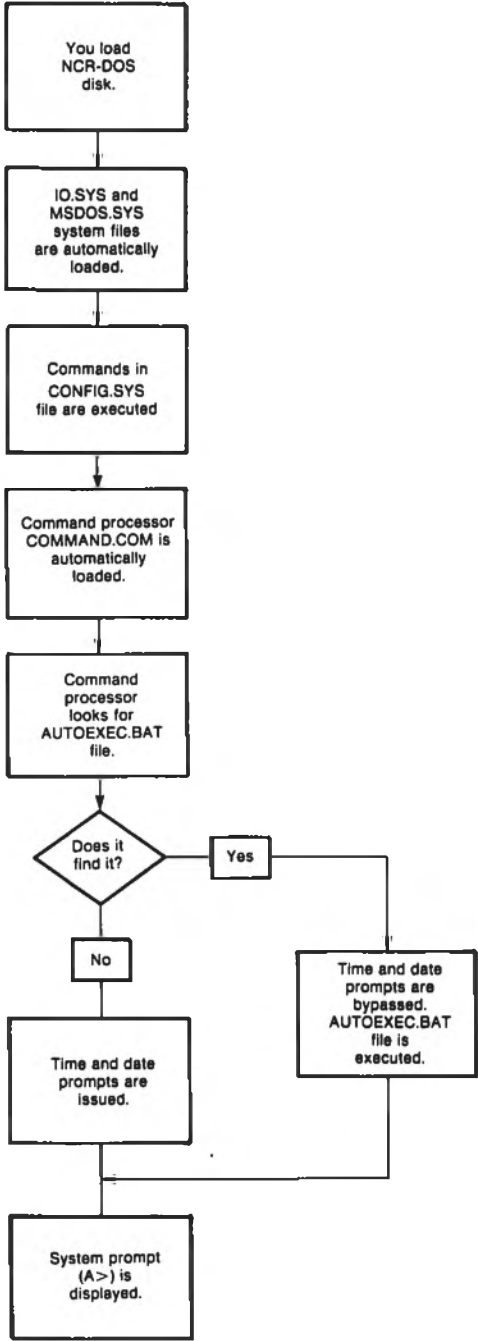


Figure 3.2

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. You could create an AUTOEXEC.BAT file as follows:

1. Type:

COPY CON: AUTOEXEC.BAT

This statement tells DOS to copy the information from the console (keyboard) into the AUTOEXEC.BAT file. (CON: replaces [d1:[pathname]filename[.ext1]). Note that the AUTOEXEC.BAT file must be located in the root directory of your DOS disk.

2. Now type:

CALENDAR

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

3. Press the <F6> key or <Ctrl-Z> to terminate the entry of the AUTOEXEC.BAT file; then press the <ENTER> key to put the command into the AUTOEXEC.BAT file.
4. The CALENDAR program will now run automatically whenever you start DOS.

You can enter any batch COMMAND, DOS system command or program name, or a series of commands and/or program names, into the AUTOEXEC.BAT file, one command or program name per line. End each line with <ENTER>; press <F6> or <ENTER> only after the last 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.

An AUTOEXEC.BAT file can also be created using the methods described in the Line Editor (EDLIN) chapter.

MULTIPLE-USE .BAT FILES

There may be times when you want to create a standard procedure and run it with different sets of data. The data may be located in a number of files.

When used in a normal DOS command, an option is a specific value which you define. Options in batch commands can be entered the same way. Or, you can create a batch (.BAT) file with dummy (replaceable) parameters. These parameters, expressed as %0—%9, can be replaced with values supplied when you run the .BAT file.

Creating .BAT Files With Replacement Parameters

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 the first such variable. ("% 0" is reserved for the filename of 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 does 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 you should remember that the batch program substitutes the parameter for %n throughout the file. Therefore, if within the same batch file you have %1.TXT and %1.LST, the batch program substitutes the same filename for both file extensions. A universal character is a valid parameter 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 file.

As another example, if you have a number of BASIC source program 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.LST
A:
TYPE %2.LST
TYPE %0.BAT
```

Now, press <F6> and then press <ENTER>. 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), unless the SHIFT command has been executed.

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 .BAT Files With Replaceable Parameters

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
```

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 its parameters, as follows:

COPY A:PROG1.BAS B:PROG2.BAS

B:

A:GWBCOM B:PROG2.BAS,B:PROG2.OBJ,B:PROG2.LST

A:

TYPE B:PROG2.LST

TYPE BASICOMP.BAT

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

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

Remember that the dummy parameter %0 is always replaced by the drive designator (if specified) and the filename of the batch file, unless a SHIFT command has been executed.

BATCH COMMANDS

To execute a batch file, you type in the filename if it is on the default drive, or the drive, path, and filename if it is on a non-default drive. Since all batch files have the file name extension .BAT, DOS locates it and begins executing it automatically. Do not use a .BAT filename that is identical to a .COM or a .EXE filename since the system executes the .COM or .EXE file and does not look for the .BAT file.

You can create a batch file using EDLIN, the DOS COPY command, or any other text editor. There are several commands that are primarily designed for use in batch processing. These commands are called "batch commands" and are similar to DOS system commands in that they are "interactive" commands. This means that they can be issued from the DOS command line also (although they would be of little use if issued this way). The batch commands are ECHO, FOR, GOTO, IF, PAUSE, REM, and SHIFT, and all are thoroughly explained in the *DOS Commands* chapter.

TABLE OF COMMANDS

Figure 3.3, Table of batch commands, summarizes the available batch commands. This table shows the purpose and format of each command, and the page number in the *DOS Commands* chapter, on which the detailed description of each command is found.

COMMAND	PURPOSE	FORMAT
ECHO	To turn echo on and off during batch file processing	ECHO ON OFF message
FOR..IN..DO..	To selectively process files by a DOS command	FOR %%<variable> IN (list)DO <command> %%<variable>
GOTO	To change sequence of execution of batch file statements	GOTO <label>
IF	To conditionally execute a DOS or batch command during batch file processing	IF [NOT] <condition> <command>
PAUSE	To suspend execution of the batch file	PAUSE [message]
REM	To display a message during batch file processing	REM [message]
SHIFT	To allow access to more than 10 replaceable parameters	SHIFT

Figure 3.3 Table of batch commands

CONFIGURATION CHANGE FEATURE

The term **configure** means to establish a specific arrangement of the parts. Accordingly, to **configure** your operating system is to set up your computer to operate in a particular way.

You may want to improve, or “tune”, the performance and flexibility of your computer system. The **BREAK**, **BUFFERS** and **FILES** commands, described in the *DOS Commands* chapter, can provide this improvement.

You may want to have the ability to use extended CRT display and keyboard control functions, or to use one or more memory resident **RAM-DISKS** to simulate diskettes for temporary storage of files and directories. The **DEVICE=ANSI.SYS** and **DEVICE=RAMDISK.SYS** options of the **DEVICE** command make these features available.

You may want to be able to interface with added special devices, such as security systems, heating and air-conditioning controls, or a

computer-assisted design (CAD) capability for engineers. These interfaces are made by using non-standard "device driver" software routines.

The "Device Driver Routines" discussion later in this section should help clarify the meaning of this term. All devices, standard or optional, that interface to the computer's main processor board, have device driver routines which DOS uses for servicing the interfaces. (transfer data and control signals to and/or from the devices). These devices include the keyboard, the flexible disk drive(s), the fixed disk drive, the monochrome or color/graphics CRT display, a parallel printer, or any device connected to an asynchronous communications adapter port (such as a communications modem or a serial printer).

All of these device drivers are provided as part of your NCR-DOS operating system software. Those which DOS determines are needed by your specific hardware, based on configuration switch settings or presence of a response at the port, are automatically used by DOS when access to those devices is called for by your programs or DOS system commands. These standard drivers do not need to be identified to DOS by DEVICE=commands.

Eventually, you may even want to replace the standard DOS command processor, COMMAND.COM, temporarily or permanently with a different command processor that you have developed or acquired. The SHELL command makes this possible.

THE CONFIGURATION (CONFIG.SYS) FILE

The configuration change feature of NCR-DOS uses the configuration file (CONFIG.SYS) to allow you to tailor your system with a minimum of effort. This special file is accessed through the root directory of your boot disk. It may contain any commands needed to accomplish the functions described in the previous section. The following commands can be placed in the configuration file: BREAK, BUFFERS, DEVICE, FILES and SHELL.

The CONFIG.SYS file is loaded at the same time as DOS. If you make changes to the CONFIG.SYS file, they are not effective until the next time DOS is booted (started up). The boot process is as follows:

1. The Read-Only Memory Basic Input-Output System (ROM BIOS) chip in the computer is executed. This chip contains the power-on diagnostics routines, initialization routines, and the standard device handling routines (the device drivers).
2. The power-on diagnostics are performed.

3. The initialization routines are performed. These include the zeroing of Dynamic Random Access Memory (DRAM), and the setting up in DRAM memory of the tables and control information needed by the operating system.
4. Part of the initialization process includes loading the DOS system routines, and executing the command in the CONFIG.SYS file, if one exists. These commands modify the control information to reflect your configuration.
5. The final initialization routine loads IO.SYS, MSDOS.SYS the command processor, COMMAND.COM (or an alternate command processor, if the SHELL command was used). The BIOS then transfers control to the command processor.

CREATING THE CONFIG.SYS FILE

You can use the text editing capability described in the Line Editor (EDLIN) chapter or the COPY CON: command option as shown in the example below to create a CONFIG.SYS file. Your current (working) directory must be the root directory of the diskette or fixed disk from which you plan to boot DOS.

As an example, you may want to initially use the BUFFERS and DEVICE commands. You can create a new (or replacement) CONFIG.SYS file with the following entries:

```
COPY CON: CONFIG.SYS
BUFFERS=10
DEVICE=ANSI.SYS
```

Terminate the first two lines with <ENTER>; terminate the final line with <F6> or <ENTER>.

This places the CONFIG.SYS file in the root directory of your DOS boot diskette or fixed disk, where the initialization routines can find it. If you have included the line DEVICE=ANSI.SYS, you must also copy the ANSI.SYS file to your boot disk.

After you have stored the file, reboot the system. DOS reads the CONFIG.SYS file and alters its control information to accommodate the commands in the file.

Remember that CONFIG.SYS is read and used only when DOS starts up (when the computer is turned on, or when it is reset using <Ctrl-Alt-Del>). Thus, when you change the CONFIG.SYS file, the changes are not implemented until the next time you cause the system to be booted from the disk.

CHANGING THE CONFIG.SYS FILE

An existing CONFIG.SYS file can also be changed either by text editing or by use of the COPY CON: option. Lines can be added, modified or deleted using the techniques described in the Line Editor (EDLIN) chapter. Use of the COPY CON: option, however, involves complete replacement of the existing CONFIG.SYS file with a new version of the file. The entire file is entered as it should be after the change, and it written over the previous file.

DEVICE DRIVER ROUTINES

A device driver is a very detailed software routine. It is designed and programmed to provide a common means for all programs and commands to communicate with the hardware interface (built-in port, expander board, etc.). It is designed for a specific hardware device, such as a keyboard, a flexible disk drive, a fixed disk drive, a CRT display, a parallel printer, a serial modem or printer, or a special added device.

The driver routine performs such tasks as:

- testing whether the device is ready to accept or send data
- testing whether any type of error return has been transmitted by the device
- initiating the actual read or write of data when the device is ready, etc.

Thus, this level of detail testing and control is not required in each application program or DOS system command that uses the device.

As mentioned earlier in this section, standard device drivers are included in the basic ROM BIOS chip installed in your computer. Device driver routines that you write or acquire for non-standard devices must be present (listed) in the root directory of your boot disk as xxx.SYS files (ANSI.SYS and RAMDISK.SYS have already been mentioned). The presence of such a driver file, and your intent to have DOS use it, are defined to DOS by the use of the DEVICE=xxx.SYS command. As many DEVICE commands as needed can be entered into your CONFIG.SYS file.

The ANSI.SYS and RAMDISK.SYS files are the two non-standard device drivers supplied by NCR, and their use is so widespread that they are discussed in more detail in the next two sections.

ANSI.SYS Extended Screen And Keyboard Functions Driver

The standard CRT driver provides a complete functional capability for a monochrome CRT display and adequate color CRT capability for

many users. However, to be able to make full use of the color and optional higher resolution of a color/graphics CRT display, you need to commit more of your computer memory to this driver with its extended screen functions.

This driver is also needed if you want to be able to reprogram certain keys on your keyboard by using the programming of function keys, or programming of alternate mode of alphabetic keys, option of the PROMPT command. This option is detailed in the description of the PROMPT command in the *DOS Commands* chapter.

RAMDISK.SYS Driver For Memory Resident Simulated Diskette(s)

This driver allows DOS to interface with one or more areas of computer memory set aside as RAM-DISK(S) for temporary storage of files and diskettes. When any DEVICE=RAMDISK.SYS entries exist in the CONFIG.SYS file, any input or output function required by a program or a DOS system command that accesses the files and/or directories within a RAM-DISK is routed to the RAMDISK.SYS driver instead of to the flexible (or fixed) disk driver.

RAMDISK.SYS, like ANSI.SYS, increases the amount of memory used by DOS.

CONFIG.SYS COMMANDS

CONFIG.SYS "commands" differ from DOS system commands or batch commands in that they are not "interactive" commands. They can not be executed from the DOS command line like DOS commands, or as they are encountered during processing of a batch file, like batch commands (or DOS commands in a batch file). Instead, commands entered into a CONFIG.SYS file are used by the DOS initialization routines to modify the control information in memory. In that way, they can affect all subsequent processing without ever being executed in the normal command sense.

TABLE OF COMMANDS

The table in Figure 3.4 summarizes the commands which can be entered into a CONFIG.SYS file and the purpose and format of each command.

COMMAND	PURPOSE	FORMAT
BREAK	To allow more-frequent test for <Ctrl-Break>	BREAK ON OFF
BUFFERS	To change number of disk data storage areas in memory	BUFFERS=n
DEVICE	To define non-standard device and its driver routine	DEVICE=[d:][path-name]filename[.ext]
FILES	To set the number of files that can be open at one time	FILES=n
SHELL	To cause alternate command processor to be loaded	SHELL=[d:][path-name]filename[.ext]

Figure 3.4 Table of CONFIG.SYS commands

DOS 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 this chapter. Similarly, the configuration commands BREAK, BUFFERS, DEVICE (that includes the RAM-Disk facility), FILES, and SHELL are also located in this chapter. The individual command descriptions are preceded by a table summarizing the complete set.

Format Notation Review

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 Capital Letters

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.

< > Angle Brackets

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, pathname, filename, and file extension; such as d:pathname filename.ext)

[] Square Brackets

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

{ } Braces

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

| Broken Bar

A broken bar when used with a filter indicates a pipe.

... Ellipsis

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

<ENTER> The ENTER Key

To execute any command you must press the <ENTER> (↵) 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.

Table of Commands

The following table provides an alphabetical listing of each command, the purpose of the command and the format for entering the command.

Name	Purpose	Format
ASSIGN	To assign a disk drive	ASSIGN [d1=d2...]
AUTOEXEC.BAT	A file containing a series of commands for batch processing: FOR, GOTO, IF, PAUSE, REM, SHIFT	
BACKUP	To copy fixed disk to flexible disks	BACKUP d1:[pathname] [filename[.ext]] d2:[/S] [/M] [/A] [/D:mm-dd-yy]
BREAK	To set Ctrl-C check (CONFIG.SYS)	BREAK ON OFF
BUFFERS	To tell DOS how many disk buffers to use (CONFIG.SYS)	BUFFER=nn
CHDIR (CD)	To change directories; display working directories	CHDIR [[d:]pathname]
CHKDSK	To scan the directory of the default or designated drive and check for consistency	CHKDSK [d:] [filename[.ext]] [/F] [/V]
CLS	To clear the screen	CLS
COMMAND	To invoke secondary command processor	COMMAND [d:][pathname][/P] [/C string]
COMP	To compare two files	COMP [filespec1] [filespec2]
CONFIG.SYS	A file containing commands that configures the operating system (BREAK, BUFFER, DEVICE, FILES, SHELL commands)	
COPY	To copy specified file(s) to disk	COPY [filespec1] [/A] [/B] [filespec2] [/V] [/A] [/B]
CTTY	To change console TTY	CTTY DEV
DATE	To display and set date	DATE [mm-dd-yy]
DEL	To delete file(s) specified	DEL [d:] [pathname][filespec]

Name	Purpose	Format
DEVICE	To allow peripheral connection (CONFIG.SYS)	DEVICE=[d:][pathname] filename[.ext]
DIR	To list requested directory entries	DIR [d:][pathname] [filename[.ext]] [/P] [/W]
DISKCOMP	To compare flexible disks	DISKCOMP [d1:] [d2:] [/1] [/8]
DISKCOPY	To copy entire disks	DISKCOPY [d1:] [d2:] [/1]
ECHO	To turn batch file echo feature on/off (BATCH)	ECHO ON ECHO OFF ECHO <message>
ERASE	To delete specified files	ERASE [d:][pathname] filename[.ext]
ESC	Enables input of escape sequences through the keyboard	ESC [#;#.<delimiter> ESC ["string";...]
EXE2BIN	To convert a file to .COM format	EXE2BIN [d1:][pathname1] filename1[.ext1] [d2:] [pathname2] filename2 [.ext2]
EXIT	To exit a command and return to lower level	EXIT
FDISK	To set up fixed disk	FDISK
FILES	To designate number of files (CONFIG.SYS)	File = nn
FIND	To search for a constant string of text	FIND [N] [/C] [/N] "string" [[d:][pathname] filename[.ext]...]
FOR..IN..DO..	To selectively process files by a DOS command. (BATCH) ● for batch processing ● for interactive processing	FOR %% <variable> IN <list> DO <command> %% <variable> FOR % <variable> IN <list> DO <command> % <variable>
FORMAT	To format disk for NCR-DOS files ● flexible diskettes ● fixed disks	FORMAT [d:] [/8] [/1] [/V] [/S] FORMAT [d:] [/V] [/S]
GOTO	To change execution sequence. (BATCH)	GOTO <label>
GRAFTABL	Supports character display in graphics mode	GRAFTABL
GRAPHICS	To print display on matrix printer	GRAPHICS

Name	Purpose	Format
HELP	To display on-line data about commands	HELP [command]
IF	Batch command extension (BATCH)	IF [NOT] <condition> <command>
KEYB	Changes the keyboard format	See description
MKDIR (MD)	To make a directory	MKDIR [d:] [pathname]
MODE	To define peripheral interfaces	See description
MORE	To display output screen by screen	MORE
PATH	To set a command search path	PATH [[d1:]pathname1] [;[d2:]pathname2] [...]
PAUSE	To pause for input (BATCH)	PAUSE [<message>]
PRINT	Background print feature; to print a text file on a line printer	PRINT [[d:] filename1[.ext1]] [/T] [/C] [/P]...
PROMPT	To designate command prompt	PROMPT [\$][prompt-text]
RAMDISK	To allocate Ram-Disk memory (CONFIG.SYS/DEVICE)	DEVICE=RAMDISK.SYS
RECOVER	To recover a bad disk	RECOVER [d:] [pathname] filename[.ext] RECOVER d:
REM	To display a comment in a batch file (BATCH)	REM [comment]
REN (RENAME)	To rename a file	REN [d:] [pathname]filename1[.ext1] filename2[.ext2]
RESTORE	To restore backed-up files from a flexible disk to the hard disk	RESTORE [d1:] [d2:] [pathname][filename.ext] [/S] [/P]
RMDIR (RD)	To remove a directory	RMDIR [d:] pathname
SET	To set one string value equal to another	SET [string1]=[string2]
SHELL	To move command processor out of root directory (CONFIG.SYS)	SHELL = [d:][pathname]filename[.ext] [/P][/C string]

Name	Purpose	Format
SHIFT	To increase number of replaceable parameters in batch file (BATCH)	SHIFT
SORT	To sort data forward or backwards	SORT [/R] [/+n] [<filespec1][>filespec2]
SYS	To transfer NCR-DOS hidden system files from drive A to specified drive	SYS d:
TIME	To display and set time	TIME [hh[.mm[.ss[.cc]]]]
TREE	To display all directory paths	TREE [d:] [/F] [>PRN]
TYPE	To display contents of ASCII file on the screen	TYPE [d:] [pathname]filename[.ext]
VER	To display DOS version number	VER
VERIFY	To verify data when written to disk	VERIFY ON OFF
VOL	To display disk volume identification number	VOL [d:]

TYPE

External

PURPOSE

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

FORMAT

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.
- 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.
- If you try to reassign a valid disk drive to an invalid drive, no error message appears until the command or program attempts to access the invalid drive. If this occurs, you should repeat the ASSIGN command with a valid drive.
- 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.

ASSIGN

CAUTION

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

AUTOEXEC.BAT

TYPE

Batch

PURPOSE

A file containing a series of commands to be executed in a system start up.

FORMAT

AUTOEXEC.BAT

COMMENTS

This AUTOEXEC.BAT is a batch file that may contain any of the seven batch commands shown below. If the file exists at system start up, DOS executes each of the batch commands in sequence. The AUTOEXEC.BAT file may also contain any DOS command or any command you have created.

A typical AUTOEXEC.BAT file contains the DATE and TIME commands as well as a PROMPT command to display the directory name in the prompt:

```
PROMPT $p
```

You can also establish the directory path you are in by using the Change Directory command CHDIR (or CD).

The following batch commands may be executed in the AUTOEXEC.BAT file:

```
ECHO  
FOR..IN..D..  
GOTO  
IF  
PAUSE  
REM  
SHIFT
```

BACKUP

TYPE

External

PURPOSE

To copy one or more files from the fixed disk drive (or the DOS partition of the fixed disk, if you are using multiple operating systems) to the flexible disk drive.

FORMAT

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

where:

d1 = fixed disk drive to be backed up

d2 = destination disk drive of backup diskette

pathname = directory path for files on the fixed disk

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. You must use the last flexible diskette from the previous backup, otherwise your previous backup will be written over.

/D: = Back up all files created or modified since the given date mm/dd/yy.

Some things to remember when using BACKUP:

- The source of a BACKUP can be the entire DOS partition, one or several specified files, a directory path, or any combination of the above.
- The destination can be any flexible disk drive.
- The default source drive is the current fixed disk drive in use. Also if no parameters are given with the source drive, the default source to be backed up is the current directory only. Use the pathname to specify other directories for the backup and the /S option if you want to backup the files within the subdirectories of the current directory.

BACKUP

- The destination flexible diskettes must have been formatted under DOS prior to executing BACKUP.
- The BACKUP utility deletes all files currently on the destination diskette. (The only exception to this is when you use the /A option you may append the new files to the end of your last diskette of your previous backup).

NOTE: The /A option only works in conjunction with the last backup diskette where multiple diskettes are required to perform the backup.

- The BACKUP utility prompts you to insert the next diskette in the destination disk drive when the total size of all the backup files is larger than the capacity of one flexible diskette. BACKUP numbers these diskettes so that when you are RESTORING the files it can check if you have inserted the diskettes in the correct order.

EXAMPLES USING BACKUP:

A>BACKUP C: A:

Backs up the current directory of C to flexible disk drive A. If subdirectories exist in the current directory of C, they are ignored by the backup.

A>BACKUP C: A: /S

Backs up the current directory files plus any subdirectories and the files within them. If the current directory is the root, then the entire DOS partition is backed up.

A>BACKUP C:\ A: /A /M

Backs up all files in the root directory that have changed since the last backup was performed. These new backup files are appended to the end of the last backup if the last diskette from the previous backup is put in drive A. If any other diskette is put in drive A, the files are overwritten.

A>BACKUP C:\LEVEL2*.COM A: /S /D:01/01/84

Backs up all files with the .COM extension and a file date of Jan 1, 1984 or later in the LEVEL2 directory and all of its subdirectories.

BACKUP

BACKUP and RESTORE can also be used to copy from one fixed disk to another. NOTE: Since you cannot backup directly from one fixed disk to another, you must backup the first fixed disk to flexible diskettes then restore the diskettes to the DOS partition of the second fixed disk.

DOS sets an exit code after completing the BACKUP utility. Exit code "0" denotes successful completion, a "1" means there are no eligible files to backup on the designated source drive, a "3" denotes that the backup was terminated by the user (using a <Ctrl-Brk>), and a "4" means the backup terminated because of an unrecoverable error. Any other number also means the backup was not successful. In this case you should check the BACKUP command format, make sure the destination drive is closed and a formatted, non-write-protected flexible diskette is in the drive. Then retry the utility. The exit code can be tested by using the ERRORLEVEL *number* condition parameter of the batch processing IF command.

To retrieve a file that has been backed up, the RESTORE command must be used, files on a BACKUP diskette cannot be used directly.

To check for accuracy while writing the disks, you may want to set VERIFY to "ON" before doing a BACKUP or RESTORE (see VERIFY later in this chapter).

BREAK

TYPE

Configuration

PURPOSE

To allow you to specify a more-frequent test for a <Ctrl-Break> entry.

FORMAT

```
BREAK {ON }  
      {OFF}
```

COMMENTS

With BREAK OFF, which is the default condition, DOS checks for entry of a <Ctrl-Break> sequence during the execution of a program only when accessing either a standard input, output or printer device or an asynchronous communications adapter. A compiler or other program which performs few such accesses is difficult to interrupt.

When BREAK is ON, DOS tests for <Ctrl-Break> any time DOS is entered for any reason by the executing program.

If you are running an application program that uses Ctrl-Break function keys, you should 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 unless BREAK ON is in CONFIG.SYS, there is no need to set the BREAK OFF unless it has been previously set to BREAK=ON. BREAK is an interactive command that can be used in a batch or configuration file or issued from the command line.

BUFFERS

TYPE

Configuration

PURPOSE

To increase or decrease the number of buffer storage areas.

FORMAT

BUFFERS=nn

where:

nn = 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. When the **BUFFER** command is not used, DOS defaults to two buffers with the capacity of 512 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 usually speeds 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 probably falls between 12 and 24.

Preferably 14 to 30 buffers should be assigned if you are using a fixed disk drive.

CHDIR

TYPE

Internal

PURPOSE

To change the current (working) directory to a different path or display current (working) directory.

FORMAT

CHDIR [[d:]pathname]

CD [[d:]pathname]

where:

d: = disk drive

COMMENTS

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

CHDIR \USER\JOE\FORMS

or

CHDIR \FORMS

and DOS puts you in the new directory.

A shorthand notation is also available with this command:

CHDIR..

or

CHDIR

Either of the above commands 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 enter **CHDIR**, DOS displays:

B:\BIN\USER\JOE

CHKDSK

TYPE

External

PURPOSE

To scan the directory of the specified disk drive, check it for consistency, report disk and memory capacities, and correct any disk errors.

FORMAT

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 file
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.

If you enter a filename in addition to the disk drive designator, DOS checks the entire disk and produces the status report. Additionally, DOS reports to you any non-contiguous blocks that exist for the specified filename. You can use the COPY command to collect non-contiguous blocks into one block.

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

Error reading FAT

Copy or BACKUP all files on disk and reformat

Error writing FAT

Copy or BACKUP all files on disk and reformat

*<filename> contains
non-contiguous blocks*

Disk is fragmented; use copy to better utilize space.

All specified file(s) are contiguous

Good disk

Invalid drive specification

Re-enter command correctly

"entry" Invalid parameter

Re-enter command correctly

Invalid sub-directory entry

Re-enter command correctly

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.

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 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, where X = nnn is a sequential number beginning with 0000).

CHKDSK then displays:

X bytes disk space freed

If you respond N to this prompt, CHKDSK erases all lost clusters.

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.

CHKDSK

*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 DEBUG utility, or you can delete it (refer to the DEBUG Utility chapter).

CLS

TYPE

Internal

PURPOSE

To clear the terminal screen.

FORMAT

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.

COMMAND

Type

Internal

PURPOSE

To load a secondary version of the command processor

FORMAT

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

where:

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

/C = causes the following command line (string) to be executed by the secondary command processor.

string = command to be added to the command processor.

COMMENTS

COMMAND allows you to utilize a secondary version of the COMMAND.COM file from the disk and the directory of your choice. For example, the secondary version of COMMAND.COM could be a test shell that you have developed; or it could be a copy of the standard COMMAND.COM in which you have changed the programming environment using the SET command. The COMMAND.COM file within DOS handles your housekeeping activities, your internal DOS commands, your system prompt, and your error handling routines. If you have the MS-MACRO ASSEMBLER package, you can find more information on the COMMAND command in the *NCR-DOS PROGRAMMER'S MANUAL*.

COMP

TYPE

External

PURPOSE

To compare two files and report their differences.

FORMAT

```
COMP [[d1:] [pathname1] [filename1[.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 for the comparison

COMMENTS:

The compare utility is useful to check a recently copied file, or to compare a text file to a backup (.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 command. Some things to remember about the COMP command:

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

If your files are identical, the message:

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

appears. However, if there is an inequality between the files the message:

```
Compare error at offset nnnn
```


COMP

File1 = xx

File2 = yy

is displayed. The offset number refers to the number of bytes from the beginning of the file that 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

TYPE

Configuration

PURPOSE

A file containing commands to configure the system, that are executed at system startup before the DOS COMMAND.COM file is given control.

FORMAT

CONFIG.SYS

COMMENTS

The CONFIG.SYS file contains commands that essentially configure or setup system parameters. These parameters must be established before the DOS system begins to interact with batch and interactive system commands. This is why the configuration commands are executed prior to giving control to DOS COMMAND.COM.

Only the following configuration system commands can be contained in the CONFIG.SYS file:

BREAK
BUFFERS
DEVICE
FILES
SHELL

TYPE

Internal

PURPOSE

To copy 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.

FORMAT

```
COPY [d1:] [pathname1] filename1[.ext1][/A] [/B]
[d2:] [pathname2] filename2[.ext2] [/V] [/A] [/B]
```

where:

d1: source drive or CON: = console

d2: destination drive

filename1 = source file

filename2 = destination file

pathname1 = source directory path

pathname2 = destination directory path

/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 the destination is not given, the copy is to the default drive and has the same name as the source file. If the source is on the default drive and the destination 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

The destination may have three forms:

COPY

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 is overwritten, with the new copy of the file replacing the original.

The /V option 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
```

This command concatenates files named FILE1.XYZ, FILE2.COM, and FILE3.TXT and places them in sequence 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 is combined with MON.REF to form MON.WK, then TUR.WKD and TUR.REF forms 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.)

During concatenation the COPY command 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.

The COPY command does not copy hidden files. For this reason, refer to the SYS command to copy the hidden files to a diskette.

CTTY

TYPE

Internal

PURPOSE

To change the input device from which commands are issued (TTY represents the console).

FORMAT

CTTY DEV

COMMENTS

DEV stands for "device", through which you communicate with 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

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

DATE

TYPE

Internal

PURPOSE

To inquire, enter or change the NCR-DOS internal system date. This date is recorded in the directory for any files you create or alter.

You can change the date from your keyboard 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.)

FORMAT

DATE [mm-dd-yy]

COMMENTS

If you type DATE, DOS responds with the message:

```
Current date is MM-DD-YY
Enter new date:
```

Press <ENTER> 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, DOS displays the message:

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

DATE

DATE then waits for you to enter a valid date.



TYPE

Internal

PURPOSE

To delete all files with the designated filespec.

FORMAT

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

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 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.

DEVICE

TYPE

Configuration

PURPOSE

To define to DOS a device driver file; and cause loading of the driver file each time DOS is started.

FORMAT

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

COMMENTS

The device driver is the software used to control a peripheral. Your NCR-DOS includes the drivers needed for all standard 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 need to identify the appropriate software through your CONFIG.SYS file so that DOS can talk with the new hardware devices. This is done through the DEVICE command.

ANSI.SYS

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 display by controlling the color through programs or to reprogram the keyboard. Standard CRT and keyboard driver functions are replaced by a set of extended screen and keyboard functions. To place the definition of the ANSI.SYS file within CONFIG.SYS, you may use EDLIN, another text editor, or the COPY CON: command. Place the command:

DEVICE = ANSI.SYS

in the CONFIG.SYS file. If you are using a fixed disk, you should add the directory path as well:

DEVICE = C:\SYS\ANSI.SYS

In this example, \SYS\ indicates that the ANSI.SYS file is listed in a directory named SYS which is a subdirectory of the root directory. Typically, flexible disk users place the device drivers in the root directory.

RAMDISK.SYS

Your DOS flexible disk contains a RAMDISK.SYS file that you can use to maintain a "RAM-DISK". A RAM-DISK is an area of computer

DEVICE

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 that 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

DEVICE

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 (nnn) of directory entries which are to be allowed on the RAM-DISK. nnn should be a multiple of 16. DOS accommodates up to 752 directory 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/L64
```

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 is labeled as the D disk.

DEVICE

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.

DIR

TYPE

Internal

PURPOSE

To list the filenames in a directory.

FORMAT

DIR [d:][pathname]filename[.ext]] [/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.

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 in the root directory 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

Two optional parameters 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 directory names and filenames are displayed, without other information. Files are displayed five filenames per line. When using the wide option without a filename parameter, you may have difficulty in distinguishing between files and directories.

DIR

You may also redirect the directory listing as follows:

DIR >PRN to print the directory listing

DIR >filespec to place a copy of the directory listing in a file

DISKCOMP

TYPE

External

PURPOSE

To compare two (identical) flexible disks.

FORMAT

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

where:

d1,d2 = two drives containing diskettes or RAM-DISKS 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 you have created a duplicate diskette or RAM-DISK 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 number and side of the unequal track.

Since the COPY utility only copies files instead of duplicating sectors or tracks (that may be bad or empty), it is unlikely that a copy of a diskette is physically identical to the original. Similarly, due to the differences in size, a fixed disk can not be compared to a flexible disk.

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

Before you DISKCOMP a diskette to a RAMDISK, be certain the RAMDISK is exactly the same physical size as the diskette (has the same configuration of sectors and tracks).

DISKCOPY

TYPE

External

PURPOSE

To make an identical copy of the diskette or RAM-DISK in the source drive to the diskette or RAM-DISK in the destination drive (including any hidden files). Copies only diskettes or RAM-DISKS.

FORMAT

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

where:

- d1 = source diskette drive or RAM-DISK drive
- d2 = destination diskette drive or RAM-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 is reformatted if it is not already formatted the same as the source diskette. You may 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 diskettes at the appropriate times. DISKCOPY waits for you to press any key before continuing.

After copying, DISKCOPY prompts:

Copy another (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 drive designations, a single-drive copy operation is performed on the default drive.
- If you omit the destination drive, the default drive is used as the destination drive.
- Both diskettes must have the same number of physical sectors and those sectors must be the same size. The destination diskettes is automatically reformatted to match the source diskette.

DISKCOPY

- If the RAM-DISKS and/or the diskettes to be copied are not identical in size, the copy is not completed and an error message is issued.
- Diskettes that have had extensive 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). This condition causes a loss of performance when accessing files. To correct this condition, you should use the COPY*.* command to eliminate these bad or unused and inbedded sectors.
- If there is any doubt about the success of the DISKCOPY, you can use the DISKCOMP to verify the accuracy of the copy.

EXAMPLE

DISKCOPY A: B:

copies the contents of the diskette in drive A to the diskette in drive B.

- DISKCOPY automatically determines the number of sides to copy, based on the source diskette in the drive, unless you use the /1 option.
- If disk errors are encountered during a DISKCOPY, DOS displays an error message.

Refer to Appendix C, on Disk Errors, for information on this error message.

ADVANCED INFORMATION

A fragmented disk can cause poor performance due to delays involved in finding, reading, or writing a file. Since DISKCOPY is a track-for-track, sector-for-sector utility, you may want to use the COPY*.* command, instead of DISKCOPY, to copy your diskette file for file and eliminate the fragmentation.

ECHO

TYPE

Batch

PURPOSE

To turn the batch echo feature on and off.

FORMAT

```
ECHO    ON
        OFF
        <message>
```

COMMENTS

Normally, commands in a batch file are displayed (echoed) on the screen when they are seen by the command processor. ECHO OFF turns off this feature. ECHO ON turns the screen echo back on. The ECHO <message> is displayed even if ECHO is currently off.

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

ERASE

TYPE

Internal

PURPOSE

To erase all files with the designated filespec

FORMAT

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

where:

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

COMMENTS

ERASE can be used to delete one or more files. If the filespec is *.* (meaning erase all files), the prompt "Are you sure? Y/N" appears. If a Y or "y" is typed as a response, then all files are erased as requested. You can also type DELETE for the ERASE command. If you enter only pathname, you erase all files within the directory. To erase 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 erased without reformatting the disk.

TYPE

External

PURPOSE

Enables input of ESC sequences through the keyboard

FORMAT

ESC <delimiter> <ESC sequence>

The keyword ESC must always be separated 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 lower-case p.

COMMENTS

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

To change an .EXE file into a memory image (.COM format) file.

FORMAT

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

where:

- d1 = source disk drive containing the .EXE format file
- pathname1 = source directory containing the .EXE format file
- filename1 = filename to be changed
- .ext1 = extension of the file to be changed (defaults to .EXE)
- d2 = destination disk drive for newly created .COM file
- pathname2 = destination directory for newly created .BIN or .COM file
- filename2 = filename to be created (if different than filename1)
- .ext2 = extension of file the file to be created (defaults to .BIN)

COMMENTS

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

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

EXIT

TYPE

Internal

PURPOSE

To exit a secondary COMMAND.COM (the command processor) and return to a previous level unless the secondary command processor was made permanent by the use of the /P option of COMMAND.

FORMAT

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 such as in GW-BASIC, you enter the SHELL command to load the COMMAND.COM. You now are able to issue DOS commands.

You can now type the DIR command and DOS displays the directory for the default drive. When you enter EXIT, you return to the previous level GW-BASIC application program.

FDISK

TYPE

External

PURPOSE

To set up your fixed disk to handle DOS and optionally to set up partitions for other operating systems.

FORMAT

FDISK

COMMENTS

This command displays and alters your fixed disk partitioning or adds a new fixed disk partition for another operating system. This command is completely menu driven.

When you type FDISK and press <ENTER>, the following screen appears:

FIXED DISK SETUP PROGRAM VERSION N.NNN

FDISK Options

CURRENT FIXED DISK DRIVE: 1

CHOOSE ONE OF THE FOLLOWING:

1. Create DOS Partition
2. Change Active Partition
3. Delete DOS Partition
4. Display Partition Data
5. Select Next Fixed Disk Drive — if your system has more than one fixed disk drive

INITIAL FDISK USE

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

CREATE DOS PARTITION

FDISK

Current Fixed Disk Drive: 1

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

If you intend to use DOS as your only operating system, type "Y" and press <ENTER> or just press <ENTER>. Make sure your DOS diskette is loaded in your default disk drive. Your entire fixed disk is dedicated to DOS and requires formatting. (Refer to the FORMAT command in this chapter.) After formatting, you can copy all the DOS commands and hidden system files by typing COPY *.* C: and SYS 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 <ENTER>. The following screen appears:

Total fixed disk space is nnn cylinders

Maximum available space is nnn

Cylinders at cylinder is nnn

These show the total number of cylinders available on your fixed disk. Press <ENTER> and the following prompt appears:

ENTER PARTITION SIZE....

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 defaults to the largest amount of cylinders allowable.

The next prompt is:

Enter starting cylinder number nnn

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

When you press <ENTER>, the screen changes to show your new, fixed disk partition statistics, and the prompt:

Press Esc to return to FDISK options.

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

FDISK

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:

```
NCR Personal Computer
Fixed Disk Setup Program Version N.NN
(C) Copyright NCR Corp. 19YY
```

Change Active Partition

Current Fixed Disk Drive: 1

Press Esc to return to Utility Options

Partition	Status	Type	Start	End	Size
1	N	non-DOS	000	049	50
2	N	non-DOS	060	149	100
3	A	DOS	150	304	155

Total disk space is xxx cylinders

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

Enter the number of the partition whose operating system you want to execute when the system is started from the fixed disk. The following message appears:

Press Esc to return to FDISK Options []

NOTE: The lines on the screen change to show the new active partition.

Press the Esc key to return to the FDISK options menu and press it again to return to DOS.

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 then starts.

FDISK

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

NOTE: This option results in loss of the data in the DOS partition, so make sure you have backed up all of the files that you want to keep before you proceed.

You 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 then see a screen similar to the following:

```
NCR Personal Computer
Fixed Disk Setup Program Version N.NN
(C) Copyright NCR Corp. 19YY
```

Delete DOS Partition

Current Fixed Disk Drive: 1

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 fixed disk space is xxx cylinders

Warning! All data in the DOS partition will be destroyed. Do you wish to continue.....? [N]

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

If you type Y and press <ENTER>, the partition information displayed on the screen is updated, and the following message appears:

Press Esc to return to FDISK Options []

The DOS partition has now been deleted. You need to start another system from the fixed disk or restart DOS from a diskette to proceed.

FDISK

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

```
NCR Personal Computer
Fixed Disk Setup Program Version N.NN
(C) Copyright NCR Corp. 19YY
```

Display Partition Information

Current Fixed Disk Drive: 1

Partition	Status	Type	Start	End	Size
1	A	DOS	000	199	900
2	N	non-DOS	900	904	108

Total fixed disk space is xxx cylinders

Press Esc to return to FDISK Options []

One line is shown for each 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 another fixed disk drive.

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

FDISK

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



FILES

TYPE

Configuration

PURPOSE

To designate the number of files that can be opened concurrently.

FORMAT

FILES = nn

where:

nn 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 allows you to open 5 anyway. If you write assembler language programs using the MS™-MACRO ASSEMBLER package, see the section on "Handles" in the *NCR-DOS PROGRAMMER'S MANUAL*.

FIND

TYPE

External

PURPOSE

To search for a specific string of text in one or more files and display each occurrence of the string.

FORMAT

FIND [/V] [/C] [/N] "string" [[d:[pathname] filename[.ext]...]

where:

/V = Displays all lines that do not contain the string

/C = Counts the number of occurrences of the string in the file without displaying each occurrence

/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 *not* acceptable to the FIND command

COMMENTS

FIND is a filter that searches the specified file(s) for the specified string. It then optionally displays all file lines that contain the specified string, that do not contain the specified string or displays only the file line numbers that contain the string.

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.

FIND

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 can not 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.

FOR..IN..DO

TYPE

Batch

PURPOSE

To permit selective file processing by a DOS command.

FORMAT

- For batch processing:

```
FOR %% <variable> IN (list) DO <command>
%% <variable>
```

- For interactive processing:

```
FOR % <variable> IN (list) DO <command> % <variable>
```

where:

<variable> = any non-capitalized single letter

(list) = one or more filespecs; the entire list is enclosed in parentheses

<command> = any DOS system command or batch sub command

COMMENTS

This command provides you with the capability to:

- Select certain files (IN <list>)
- Define this selection as one variable (FOR <variable>)
- Perform a DOS command (DO <command>) on each of your selections (<variable>) in sequence

<variable> can be any character except 0,1,2,3,...,9 (to avoid confusion with the %0-%9 batch processing replacement parameters.) Use the %% in front of the <variable> for the FOR..IN..DO command that is included in a batch processing file and use the % in front of the <variable> for issuing the command from the DOS command line (interactive processing).

Consider the following example:

```
FOR %%F IN (FILE1.EXT FILE2.EXT FILE3.EXT) DO
COPY %%F C:
```

This command could appear in a batch file to restore the FILE1.EXT, FILE2.EXT and FILE3.EXT files from the default diskette drive to the C disk each time the system is brought up.

FORMAT

TYPE

External

PURPOSE

To format the disk in the specified drive to accept NCR-DOS files.

FORMAT

FORMAT [d:] [/8] [/1] [/V] [/B] [/S] (flexible diskettes)

FORMAT [d:] [/V] [/S] (fixed disks)

where:

/1 = 8 sectors/track, single-sided, double density (160 KB)

/8 = 8 sectors/track, double-sided, double density (320 KB)

/V = Prompt for a volume label

/B = Leaves room for DOS operating system without copying it

/S = Copies operating system on newly formatted disk (must be last parameter in command)

COMMENTS

This command formats the disk and initializes the Directory and File Allocation Tables (FAT). If no drive is specified, the disk in the default drive is formatted. The default values for all flexible disks are double density, double sided with 9 sectors per track. A fixed disk is formatted with 17 sectors per track.

When formatting a fixed disk, FORMAT displays a message asking for the number of certifications (read-after-write checks on each track). The default value is 5; increasing the number significantly increases the time for formatting.

Four options can be requested when formatting a flexible disk; only the /S and /V options are valid when formatting a fixed disk.

/V

Causes FORMAT to pause in the formatting process and display a message asking for a volume label (useful in disk identification).

/B

Causes FORMAT to leave room for the DOS operating system without copying it, in case you wish to put another version of NCR-DOS on the disk.

FORMAT

/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.)

NOTE: To copy an entire system diskette, use **DISKCOPY** to produce a "mirror image" and write over any label, if one exists.

The next group of options allows some other format to be created on a flexible disk. (These formats may be needed if you want to copy and use information from a non-NCR format diskette.) If no option is specified, the default format is assumed: 9 sectors per track; double sided, double density (360 KB disk capacity).

/8

Formats at 8 sectors per track; double sided, double density (320KB disk capacity).

/1

Formats at 8 sectors per track; single sided, double density (160KB disk capacity).

If you are formatting your fixed disk, the **/V** option is recommended. Identifying the disk with a 1- to 11-character volume label is good computer operational practice, and may 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.

GOTO

TYPE

Batch

PURPOSE

To change the sequence of execution of batch file statements.

FORMAT

GOTO<label>

COMMENTS

GOTO causes batch commands to be executed beginning with the line following the <label> line. <label> is recognized by the system by a colon followed by a label name. If a label has not been defined, the current batch file terminates. The label line can either precede or follow the GOTO line in the batch file.

For example:

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

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

Note that GOTO abc and GOTO:abc are equivalent.

ADVANCED INFORMATION

Note that the line containing the label (:abc) is not displayed during execution of the batch file. You can take advantage of this fact by entering comment lines which begin with labels not referenced by any GOTO command. These comments are shown when you display, edit or print the batch file, but not when you execute the batch file.

GRAFTABL

TYPE

External

PURPOSE

Enables full character support in graphics mode.

FORMAT

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

TYPE

External

PURPOSE

To load a graphics screen/printer driver.

FORMAT

GRAPHICS

COMMENTS

To load the graphics screen/printer driver, simply invoke the command by entering 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 every time it is 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
```


HELP

TYPE

External

PURPOSE

To access on-line information concerning internal and external commands.

FORMAT

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

ANSI	DEL	GOTO	RENAME
ASSIGN	DEVICE	GRAFTABL	RESTORE
BACKUP	DIR	GRAPHICS	RMDIR
BATCH	DISKCOMP	IF	SET
BREAK	DISKCOPY	KEYS	SHELL
BUFPERS	ECHO	MKDIR	SHIFT
CHDIR	ERASE	MODE	SORT
CHKDSK	ESC	MORE	SYS
CLS	EXE2BIN	PATH	TIME
COMMAND	EXIT	PAUSE	TREE
COMP	FDISK	PRINT	TYPE
CONFIG	FILES	PROMPT	VER
COPY	FIND	RAMDISK	VERIFY
CTTY	FOR IN DO	RECOVER	VOL
DATE	FORMAT	REM	

IF

TYPE

Batch

PURPOSE

To conditionally execute a DOS or batch command during batch file processing.

FORMAT

IF [NOT]<condition> <command>

COMMENTS

The IF command first tests <condition>. If <condition> is true or when modified by the NOT parameter, is false, then the <command> is executed. If <condition> is found to be false or when modified by the NOT parameter, is found to be true, the <command> is not executed.

The <condition> parameter to be tested is one of the following:

ERRORLEVEL n

True if and only if the previously executed program or 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 batch replacement parameter (%%, if used) substitution. Strings may not have embedded separators. A lowercase character in one string is not considered to be identical to an uppercase character in the corresponding position of the other string.

EXIST filespec

True if and only if filespec exists as specified ((d:)[pathname] filename[.ext])

NOT <condition>

True if and only if the condition is false.

Consider the following examples:

IF NOT EXIST TMP.BID ECHO Can't find file

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

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.

FORMAT

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 earlier 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 keyboard program last loaded into memory.

For additional information on keyboard formats, turn to Appendix E "Keyboard Considerations".

MKDIR

TYPE

Internal

PURPOSE

To create a subdirectory on a disk.

FORMAT

MKDIR [d:] [pathname]

or

MD [d:] [pathname]

COMMENTS

This command is used to create a hierarchical directory structure on a hard disk or diskette. 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, enter:

MKDIR \USER\JOE

MODE

TYPE

External

PURPOSE

To define which printer(s), asynchronous communications adapter(s), or CRT display adapter DOS is to use and to establish the mode of use through specification or selection of operational characteristics.

FORMAT

See corresponding details under "Comments" for each format.

Format 1: To select printer characteristics:

```
MODE LPT#[cpl[,lpi[,P]
```

Format 2: To direct printer output to a serial printer via an asynchronous communications adapter:

```
MODE LPT#:=COMx
```

Format 3: To specify the operational characteristics of an asynchronous communications adapter:

```
MODE COMx: baud rate[,parity[,data bits[,stop bits[,P]]]]
```

Format 4: To specify the display mode of the Color/Graphics CRT display adapter, to shift the display, or to switch active status between a Color/Graphics CRT display adapter and a monochrome CRT display adapter:

```
MODE display type  
or  
MODE [display type],shift[,T]
```

COMMENTS

Once you have added a parallel printer to your system, or a color/graphics CRT display, or a serial printer, plotter, modem or other device(s) which connect through the asynchronous communications adapter(s), you may find that you are entering the same MODE command(s) every time you turn on the machine and boot in DOS. When you reach the point where one or more "standard" MODE commands are required for DOS to properly use the components of your system, the repetitive MODE command(s) should be included in an AUTOEXEC.BAT file.

MODE

An existing AUTOEXEC.BAT file can be changed or added to by use of the Line Editor (EDLIN) program as described in the Line Editor (EDLIN) chapter of this manual. Or, it can be recreated and replaced with needed changes and/or additions using the COPY CON: technique described in the "Creating an AUTOEXEC.BAT File" section of the learning about Commands chapter. If you have not currently been using an AUTOEXEC.BAT file, you can create one using either the EDLIN program or the COPY CON: technique.

Format 1: To select printer characteristics:

For some printer models, MODE Format 1 sets the printer number, characters per line, line per inch and time-out retry parameters. (If a serial printer is installed via an asynchronous communications adapter, MODE Format 2 must also be entered to assign the printer output to that device.)

You can have from one to three printers, addressed as LPT1, LPT2, and LPT3, on the NCR PC. The first, or primary, printer, must be identified as LPT1. The reserved device names, LST and PRN, discussed in the "Naming Convention" section of the Files and Directives chapter, always direct DOS to use LPT1 when entered in lieu of a filename for redirecting output (as in DIR | SORT > PRN).

The format for MODE Format 1 is:

```
MODE LPT#[cpl][,lpi][,P]
```

where:

= the printer number (1,2 or 3).

cpl = the number of characters per (8-inch wide) line. The number can be either 80 (10 characters per inch) or 132 (16.5 characters per inch). The default value is 80.

lpi = the number of lines per inch, either 6 or 8. The default value is 6.

P = continuous retry on printer not ready ("time-out") condition.

DOS starts with the printer at 80 characters per line and 6 lines per inch. To see how your printer is set up, type:

MODE LPT#:

substituting the number of the printer (usually 1) for the #.

This command works best with NCR or Epson printers (or printers that "act" like Epson printers). If you don't have this type of printer,

MODE

MODE is not able to set the characters per line or the lines per inch. In fact, MODE gives you a printer error message if you try to use some other type of printer.

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 can not 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 does not retry continuously but gives an error message when the printer "times out."

If you give the ,P and your printer hangs up, your computer locks up. You can get the computer out of this condition 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 does not retry on a time-out because the ,P has not been included.

When power is turned on and DOS is booted in, the default printer mode is 80 characters per line and 6 lines per inch. The cpl and lpi parameters, once set to either of their respective values, do not

MODE

change during a session unless specifically changed by another MODE command. Either parameter, if omitted or invalid in a subsequent MODE command, remains unchanged from its last assigned value. The P option, by contrast, is turned off if a MODE command is entered without the ,P parameter.

If you enter:

MODE LPT1:132,8,P

the command sets the characters per line (cpl) to 132 and lines per inch (lpi) to 8, and specifies continuous retry on time-out.

If you later enter:

MODE LPT1;6

this command leaves cpl unchanged (at 132), changes the lpi to 6, and stops continuous retry on time-out.

Format 2

The second format of MODE also relates to the printer. The format is:

MODE LPT#:=COMx

where:

- # = the number of the printer you are reassigning
- x = the communications adapter for your serial printer.

This command redirects 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 the MODE LPT1 command.

To use this version of MODE, you should also set up the serial asynchronous communications adapter, using format 3 of the MODE command covered in the next section before you start printing.

MODE

Format 3

This format of **MODE** sets the asynchronous communications adapter characteristics: baud rate, parity, data bits, stop bits, and retries. The format is:

MODE COMx: baud rate[, [parity], [data bits], [stop bits] [,P]]

where:

x = the number of the communications adapter, either 1 or 2.
baud rate = 110, 150, 300, 600, 1200, 2400, 4800, or 9600. (You may abbreviate and give just the first two numbers of the rate).

parity = O, E, or N for Odd, Even, or None. 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 stop 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 format line. Use a comma to indicate an omitted parameter.

If you don't set P or forget to, then it is off (It should be set for a serial printer). 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

This 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

MODE

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 you have connected to your Color/Graphics Monitor Adapter Board. If you have only the Monochrome Display, this command does nothing.

The format of this command is:

```
MODE display type  
or  
MODE [display type],[shift] [,T]
```

where:

display type=40,80

shift=L(left) or R(right)

T=Test Pattern

The display type can be any one of the following. (The C/G display refers to the RGB monitor, television monitor or composite video monitor attached to the Color/Graphics Monitor Adapter Board.)

- | | |
|------|---|
| 40 | 40-column lines for the display |
| 80 | 80-column lines for the display |
| BW40 | Makes the C/G display the active display, uses 40-character lines, and turns off color for the composite monitor. |
| BW80 | Makes the C/G display the active display, uses 80-character lines, and turn off color for the composite monitor. |
| 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 |

The active display is the current monitor that is being used by DOS and your programs to display information. If you have two displays,

MODE

such as a Monochrome display and another monitor or TV attached to the C/G Monitor Board, either one can be used with this command.

Color is optional. Some programs use color; others do not. Specifying CO40, or CO80 does not guarantee that your programs will display in color; the selection merely lets DOS display CRT output in color. It's up to you and your programs to make color appear.

MODE can only partially be overridden by 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) and the active display selected by the MODE command remains in effect until changed by another MODE command.

The shift option (R or L), shifts the display left or right one character position. Shift can be used with either a television set or a composite monitor to adjust the display. With some composite monitors, characters "fall off" the edge of the screen. This command lets you shift the line back into position.

The T option produces a test pattern line. When you request it, MODE displays 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.

Both the S and T options should be entered when the prompt in on the active display that you want to adjust.

For example

```
MODE 40,L,T
```

sets the mode to 40 characters per line and shifts the entire screen one character position to the left. A test pattern is displayed that essentially counts the character positions on the line. You can then shift the screen further to the left without re-entering the MODE command.

MORE

TYPE

External

PURPOSE

To send output to the console one screen at a time, allowing you to see the displayed data before continuing.

FORMAT

<COMMAND> | MORE

COMMENTS

MORE is a filter that reads from standard input (such as a command from your terminal) and displays one screen of data 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 data. 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 enter:

TYPE MYFILES.COM | MORE

NCR-DOS displays the file MYFILES.COM (on the default drive) one screen at a time. Notice that there is a space both before and after the pipe(|). These spaces are optional.

The MORE command writes a temporary file on the default disk drive. It uses this file as a buffer during the display operation. Therefore, if the disk in the default drive is either "write protected" or is full, the MORE command can not function.

PATH

TYPE

Internal

PURPOSE

To establish the directory path that DOS is to follow when looking for an external DOS command, user command or application command. PATH only works for external commands ending in .EXE, .COM or .BAT.

FORMAT

```
PATH [[d1:]pathname1] [:[d2:]pathname2]
[:[d3:]pathname3];[...]
```

COMMENTS

This command allows you to define to DOS which directories to search for any external command after DOS has searched your working directory and not found the command. The default value is the root directory (\), where all DOS external commands typically reside. However, if the root directory is not on the default drive or the commands are in another directory, DOS can not find them without a PATH command directory path such as:

```
PATH C:\
```

that tells DOS to look on the C drive for the root directory. This command is used only for external commands that have an extension of .EXE, .COM or .BAT.

To tell DOS to search your USER directory then your JOE directory after it has searched for the external commands, enter:

```
PATH \USER\JOE
```

DOS now also searches the USER and JOE directories on the default drive for external commands until you set another path.

You can tell DOS to search more than one path by specifying several pathnames separated by semicolons. For example,

```
PATH B:\USER\JOE; B:\USER\SUE; C:\DEV
```

tells DOS to search for external command on the drives and in the directories specified by the above pathnames. DOS searches the pathnames in the order specified in the PATH command.

The PATH command entered with no options displays the current path. If you specify PATH;, DOS sets the NUL path, meaning that only the working directory is searched for external commands.

PAUSE

TYPE

Batch

PURPOSE

To suspend the execution of the batch file and display a message.

FORMAT

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>).

When the command processor encounters PAUSE, it suspend execution and prints the specified message (if any):

[message]

Strike any key when ready...

When you press any key the batch process resumes execution. If you press <Ctrl-Break> the following prompt is displayed:

Terminate batch job (Y/N)?

If you type **Y** in response to this prompt, execution of the remainder of the batch file is aborted and control is returned to the operating system command level. Therefore, PAUSE may be used to break a batch file into segments, 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.

PRINT

TYPE

External

PURPOSE

To queue a text file for printing on a line printer. After placing the text file in the print queue you may continue processing other NCR-DOS commands (usually called "background printing").

FORMAT

PRINT [[d:] [filename[.ext]] [/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) and stops printing after the print buffer is empty. A message to this effect is printed.

/C — CANCEL

This option turns on the cancel mode. The preceding file and all following files defined on the command line are removed from the print queue until you type a **/P** option.

/P — PRINT

This option turns on the print mode. The preceding files and all following files defined on the command line 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

PRINT A:TEMP1.TST/C A:TEMP1.TST A:TEMP2.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.

The PRINT command displays the following 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 <ENTER> 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 in 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.

File not found d:XXXXXXXXX.XXX

A filename was given for files to add to the queue, but no files match a specification. (If there are no files in the queue to match a cancelled filename, no error message appears.)

Error reading drive N

If this message occurs when PRINT attempts a disk access in drive N, PRINT keeps trying until the drive is ready. Any other error causes the current file to be cancelled. In such a case, an error message is output to your printer.

PROMPT

TYPE

Internal

PURPOSE

To change the DOS command prompt.

FORMAT

PROMPT [[**\$**]prompt-text]

COMMENTS

This command allows you to change the text of the DOS system prompt. If no text is typed, the prompt is set to the default prompt-drive designation followed by the greater than (>) character.

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 text (but may be intermixed with non-\$ characters to build a prompt line):

Use This Character	To Get This Prompt:
\$	- The '\$' character
t	- The current time in the form "hh:mm:ss"
d	- The current date in the form "day-of-week mm-dd-yy"
. \p	- the current drive and directory in the form "d:\current path"
v	- The version number in the form "MS-DOS Version X.XX"
n	- The default drive letter
g	- The ">" character
l	- The "<" character
b	- The " " character

PROMPT

- - A carriage return/line feed sequence (<ENTER>)
- s - A space
- h - A backspace, erasing the previous character
- e - ASCII code X'1B' (escape) (see ADVANCED INFORMATION below)
- q - The equal sign (=)

Any keyboard character(s) except the =, \$, ", > and < characters, *not* preceded by the \$ character sets the prompt text to the entered character(s). Combinations of prompt text may be entered. For example, the command:

PROMPT \$P \$g

sets the prompt to the current drive letter and pathname followed by >.

Consider the following example:

PROMPT \$n:FML

Sets the prompt to the default drive letter followed by a colon and the letters FML (which could be your initials).

ADVANCED INFORMATION

The PROMPT command has a second usage within NCR-DOS using the PROMPT \$e escape functions. It is used to program your function keys, reprogram your keyboard, your cursor and display graphics. See the section in the Chapter entitled "Editing, Function keys, The Cursor and Display Graphics".

RECOVER

TYPE

External

PURPOSE

To recover a file or an entire disk containing bad sectors or a bad directory.

FORMAT

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

or

RECOVER d:

COMMENTS

If sectors on a disk are bad, you can recover either the file that contains the bad sectors (without the bad data in the sector but saving the rest of the file) or the entire disk (if the bad sector was not in the directory).

To recover a particular file, type:

RECOVER filename

DOS responds with:

Press any key to begin recovery of the file(s) on drive=h

NCR-DOS reads the file sector-by-sector and skips 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 the bad sector(s).

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 is rebuilt, and all files renamed FILEnnnn.REC, where nnnn is a four digit number beginning with 0000. Sub-directories are also renamed, but not rebuilt, in the FILEnnnn.REC format and put into the root directory.

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.

TYPE

Batch

PURPOSE

To display a message that is on the same line as the REM command during execution of the batch file.

FORMAT

REM [<message>]

COMMENTS

The only separators allowed in the <message> are the space, tab, and comma. The maximum length of the message is 123 bytes. The following is an example of a batch file using REM commands:

```
REM This file checks new disks
```

```
REM It is named NEWDISK.BAT
```

```
PAUSE Insert new disk in drive B:
```

```
FORMAT B:/S
```

```
DIR B:
```

```
CHKDSK B:
```

REM commands with no messages can be included in your batch files to create spacing between other commands for better readability when the batch files are displayed or printed.

Note that the REM command and its associated message is not displayed during batch file processing if ECHO is OFF.

RENAME

TYPE

Internal

PURPOSE

To change the name of a file to a different name without changing the drive designation or pathname.

FORMAT

```
RENAME [d:] [pathname] filename1[.ext1] filename2[.ext2]  
or  
REN [d:] [pathname] filename1 [.ext1] filename2 [.ext2]
```

COMMENTS

The first 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 filename is ignored. The file remains in the same path and directory and on the same 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

TYPE

External

PURPOSE

To restore files, created by the BACKUP command, from the flexible diskette(s) to a fixed disk.

FORMAT

RESTORE d1: [d2:] [pathname] [filename.ext.] [/S] [/P]

where:

d1: = the disk drive containing the files to be restored

d2: = the fixed disk drive to be restored

pathname = the directory path to the desired location of the files

filename.ext = the name of the file to be restored (you may use the universal characters * and ?)

/S = this option restores the current directory plus its subdirectories, if any exist. With this option the directory tree structure which existed when the files were backed up is recreated and added to your current tree structure before the files are RESTORED. If this option is not used, any subdirectory files are ignored.

/P = [option to supply prompts on the screen for you to determine which recently altered files are to be restored. This prompt only shows when it sees that the file size of the file to be restored is not the same as the file size of the file it is replacing on the fixed disk. If the file sizes are the same, RESTORE the fixed disk. If the file sizes are the same, RESTORE assumes that the file has not changed and no prompt is shown.

COMMENTS

The RESTORE command replaces previously backed-up files from a flexible diskette to the fixed disk. This command only restores files that you have saved through the DOS BACKUP command (see the BACKUP command in this chapter).

Things to remember when using RESTORE:

- The source of the RESTORE utility must be one or more flexible diskettes containing files created through the BACKUP utility.
- The destination of the RESTORE utility must be a fixed disk drive.

RESTORE

- If the backup files to be restored are on several diskettes, the RESTORE utility prompts you to insert the next diskette until all the files have been restored.
- If the /S option is not used, only the current directory is restored.
- When restoring files to a directory other than the current directory, the destination path must originate from the root directory.
- Back up files can be restored only to a directory that has the same pathname as the pathname the file was backed up under.

EXAMPLES USING RESTORE

A>RESTORE A: C:OLDFILE.DAT

Restores the file called OLDFILE.DAT from the diskette in drive A to the current directory of drive C. This command fails if OLDFILE.DAT did not originate from a directory with the same name as your current directory. If OLDFILE.DAT is not found on that diskette, then RESTORE prompts you to insert the next sequential BACKUP diskette (if the diskette in drive A was not the last in the backup series of diskettes). If the diskette is the last of the BACKUP series, then the message "file not found" is displayed.

A>RESTORE A: C: /S

Restores all files on the backup diskettes, including all subdirectory files.

A>RESTORE A: C:*.DAT

Restores all files ending in .DAT, to the current directory. All subdirectory files are ignored.

A>RESTORE A: C:\LEVEL2 /P

Restores all files from diskette A in the LEVEL2 directory and displays a prompt if a backup file and a file presently on the fixed disk, that have the same file name, but different file sizes. This example requires that the current directory be the root directory.

A>RESTORE A: C:\LEVEL2\LEVEL3\OLDFILE.DAT

If the current directory is not LEVEL3, this restores OLDFILE.DAT from drive A to the LEVEL3 directory on drive C. If the current

RESTORE

directory is LEVEL3, then the following command can be used —
A>RESTORE A: C:OLDFILE.DAT.

Exit Codes

The exit code for this command is one of the following:

- 0 NORMAL COMPETITION
- 1 NO FILES FOUND TO RESTORE
- 3 TERMINATED BY THE OPERATOR (through <Ctrl-Brk> or <Esc>)
- 4 TERMINATED DUE TO ERROR

You can test the returned exit codes with the ERRORLEVEL number condition parameter of the IF batch command.

You may also want to set VERIFY to "ON" to check if there are any errors while writing to the disk (see the VERIFY command in this chapter).

RMDIR

TYPE

Internal

PURPOSE

To remove a directory or subdirectory from a hierarchical directory structure.

FORMAT

RMDIR [d:]pathname

or

RD [d:]pathname

COMMENTS

This command removes a (sub) directory that is empty of files (except for the "." and ".." shorthand symbols).

To remove the \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 files which can be deleted do exist, use DELETE or ERASE to delete all except the . and .. entries in the (sub)directory. Then type:

```
RMDIR \USER\JOE
```

The JOE (sub) directory is deleted from the directory structure.

TYPE

Internal

PURPOSE

To set one string equal to another string for access by programs and commands.

FORMAT

SET[string1=[string2]]

where:

string1 = the actual string you are defining.

string2 = the parameter you want string1 to equal.

COMMENTS

This command is meaningful only if you want to define (set) values that are used by programs you have written or by DOS commands.

Entering the Set command without an accompanying string parameters, displays the current set values. These values initially consist of:

- PATH=pathnames
- COMSPEC=A:\COMMAND.COM (the location of your command processor)
- PROMPT= for any prompt variations. Using SET with string1 but not string2 removes string1 = string2 from the environment.

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 SET TTY= command

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 uses 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.

SHELL

TYPE

Configuration

PURPOSE

To cause DOS initialization routines to load a specified command processor file instead of COMMAND.COM.

FORMAT

SHELL = [d:[pathname] filename[.ext] [/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 replaces the standard command processor, COMMAND.COM, with one you have developed. You should be totally conversant with the assembler language before attempting to create a new command processor.

"

5071C

SHIFT

TYPE

Batch

PURPOSE

To allow access to more than 10 replaceable parameters in batch file processing by shifting all parameter down one parameter position.

FORMAT

SHIFT

COMMENTS

Usually, command files are limited to handling 10 replaceable parameters, %0 through %9. To allow access to more than ten parameters, use SHIFT to move the command line parameters down one position. For example, if you enter:

```
BASICOMP A:PROG1 B:PROG2
%0 = BASICOMP
%1 = A:PROG1
%2 = B:PROG2
%3...%9 are empty
```

then a SHIFT command results in the following:

```
%0 = A:PROG1
%1 = B:PROG2
%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 (and %9 is shifted to %8 and so on, with 0% being replaced by %1).

SORT

TYPE

External

PURPOSE

To read input from a file or your terminal, sort the data according to the ASCII collating sequence, then write it to your output device.

FORMAT

SORT [/R] [/+n] [<filespec1] [>filespec2]

where:

<filespec1 = source file to be sorted; default is the console keyboard unless SORT is preceded by another command and a piping symbol(|)

>filespec2 = output file to contain the sorted data; default is the console display.

/R = Reverse order of sort

/+n = Starting column of sort key

COMMENTS

The SORT command is used to arrange the sequence of records in the file according to ASCII sequence (normally ascending). The sort begins in the specified column (either column 1 or /+n) and includes *all* characters in the record from the specified column. There are two options that you are allowed to select:

/R

Reverses the sort sequence from ascending to descending order; that is, sorts from Z to A.

/+n

Sorts starting from column n. If you do not specify this option, 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:

SORT

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. The maximum file size that can be sorted is 63K characters.

ADVANCED INFORMATION

Although the SORT command can be a very useful tool, it has some limitations. Since it sorts according to the ASCII collecting sequence, 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 number 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 0001, 0002, 0010, 0186, and so on, the sort performs 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 corrects that situation.

SYS

TYPE

External

PURPOSE

To transfer the NCR-DOS hidden system files from the disk in the default drive to the disk in the drive specified by d:

FORMAT

SYS d:

COMMENTS

SYS is normally used to update the system files or to place the system files on a formatted disk that contains no files. An entry for d: is required. There must be enough space available at the beginning of the destination disk to accommodate the system files, preferably through the use of the /S option with the FORMAT command.

If the hidden system files IO.SYS and MSDOS.SYS are already on the destination disk, they must occupy the same amount of space on the disk as the new system files require. If the destination disk is completely blank, DOS copies the hidden system files into the blank disk.

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 is displayed:

No room for system on destination disk

There is not enough room on the destination disk for the IO.SYS and MSDOS.SYS files. This means that the file space presently allocated for the hidden system files is too small for the files to be copied.

Incompatible system size

The system files IO.SYS and MSDOS.SYS do not take up the same amount of space on the destination disk as was allocated for them.

TYPE

Internal

PURPOSE

To display or set the time.

FORMAT

TIME [hh[.mm[.ss[.cc]]]]

where:

hh = hour; 00-24

mm = minute; 00-59

ss = second; 00-59

cc = hundredths of a second; 00-99

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 <ENTER> 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 hour, minute and second entries must be separated by periods or colons. The hundredths of a second entry must be separated by a period only. You do not have to enter 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.

TIME

ADVANCED INFORMATION

If you are using the AUTOEXEC.BAT file, the command TIME may be included if you wish to enter the correct time at the start of your session.

TREE

TYPE

External

PURPOSE

To display all directory paths and subdirectory names on a disk and, if specified, all files contained within each subdirectory.

FORMAT

```
TREE [d:] [/F] [>PRN]
```

where:

d: = the disk drive to be displayed

/F = displays all the files in each directory

>PRN = redirection of the display to the printer

COMMENTS

If you don't specify a disk drive the default drive is used. The TREE command displays all sub-directories. With the /F option it displays all file names in each subdirectory. If there are many files and sub-directories on the disk, you may want to route the TREE display to a printer through the use of >PRN. You may also use the MORE filter to look at a page at a time.

TREE | MORE

TYPE

TYPE

Internal

PURPOSE

To display the contents of an ASCII file on the display screen.

FORMAT

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

COMMENTS

Use this command to examine an ASCII file without modifying it. (Use DIR to find the name of a file and EDLIN to alter the contents of a file.) You can not display 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. You should not try to display binary files because some non-ASCII control characters (such as Ctrl-Z) are sent to your computer, causing bells, form feeds, and escape sequences to be executed. The display can be redirected to another output device such as a printer (>PRN) or it can be printed after it is displayed on the screen by pressing <Ctrl-Ptr Sc>.

VER

TYPE

Internal

PURPOSE

To display the current MS-DOS version number.

FORMAT

VER

COMMENTS

This command displays the current version of the MS-DOS system that is loaded in your computer.

VERIFY

TYPE

Internal

PURPOSE

To turn the verify switch on or off (when writing to disk).

FORMAT

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). After this, DOS performs a VERIFY each time you write data to a disk. You receive an error message only if DOS was unable to successfully reread the data written 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. The default for VERIFY is OFF.

TYPE

Internal

PURPOSE

To display the disk volume identifier, if it exists.

FORMAT

VOL [d:]

COMMENTS

This command displays the volume ID (label name) of the disk in drive d:. If no drive is specified, DOS displays the volume ID of the disk in the default drive.



Editing, Function Keys, the Cursor and Display Graphics

SPECIAL EDITING KEYS

The special editing keys allow you to save some keyboard entry effort by repeating all or part of 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 current line can be selectively copied from the last entered stored line into the current 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 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. This is 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 *DOS 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.

Warning: If any editing function is programmed to a different key, or eliminated by reprogramming any of the F1-F6 keys, the descriptions of the use of that function key as given in this chapter and in the Line Editor (EDLIN) chapter are no longer accurate.

Function	Key	Description
Copy one character	F1 or →	Copies one character from the stored line to the command line. (Copies the last command entered).
Copy up to character	F2x	Copies characters up to the character (x) 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. (Copies the last command entered).
Skip one character	Del	Skips over (does not copy) one character in the stored line.
Skip up to character	F4x	Skips over (does not copy) the characters in the stored line up to the character (x) specified.
Quit input	Esc	Voids the current input line; leaves the stored line unchanged.
Insert and Replace mode	Ins	Enters insert mode. Press again to enter replace mode.
Store new line	F5	Makes the new line the stored line.

End of Keyboard entry F6

**Signals the end of command
line for certain keyboard entry
operations.**

Figure 5.1 Special editing functions

EDITING OF DOS 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 <ENTER>.
- 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 amount of typing.

The relationship between the command line and the line storage area is shown in Figure 5-2.

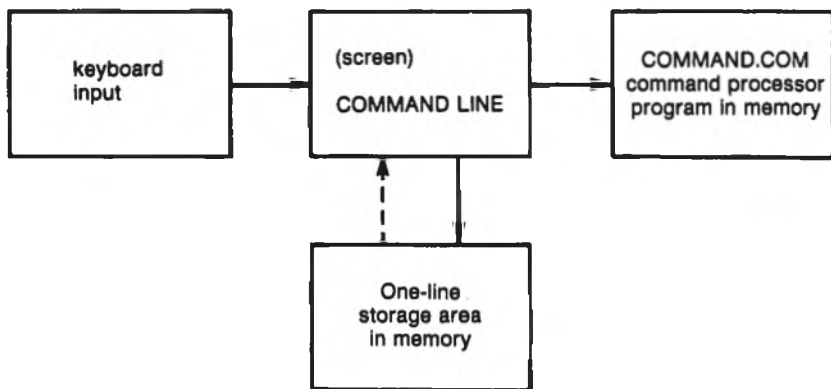


Figure 5.2 Command line usage

You type a command to NCR-DOS on the command line. When you press the <ENTER> 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

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><ENTER>

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. Pressing <ENTER> 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>l

Insert mode is entered; a "l" is inserted. The command line now reads:

DIR PROG1

Now type:

<F3><ENTER>

Any function key 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

<ENTER> causes the command line to be executed and to be copied into the special storage area.

Note that the 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 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:

This skips over the "1" in the stored line. The displayed command line does not change.

Next type:

<F1> (or →)

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 **<ENTER>** 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 can be used to accomplish the same results as in this example. This sequence, which 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, press **<ENTER>**.

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><ENTER>

(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 <Ins> and the <space> character are needed now to insert a space after "TYPE" before copying "PROG.ASM" into the command line. <F3> copies the remainder of the stored line (PROG.ASM) into the command line. <ENTER> causes the command line to be executed and to be copied into the special storage area. (<F3> turns off insert mode and returns DOS to replace mode.)

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\<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><ENTER>
```


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 **** 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
```

Pressing **<ENTER>** 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<F3>
```

The display now reads:

```
TYPE \PROGSRCE\PAYROLL\PROGW2.ASM
```

8. If, before you press **<ENTER>**, 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 skips the letter "P"; the letters "LEDGER" replace the letters "AYROLL"; and the <F3> function copies the remainder (" \PROG*.ASM") of the stored line. The display now reads:

```
TYPE \PROGSRCE\PAYROLL\PROGW2.ASM\  
DIR \PROGSRCE\LEDGER\PROG*.ASM
```

Pressing <ENTER> now causes the new command (bottom) line to be executed and to be copied into the special storage area.

PROGRAMMING THE KEYBOARD

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.

First, you have to install in the CONFIG.SYS file, the ANSI driver that is provided on your NCR-DOS diskette. The filename of this driver is ANSYSYS. Using EDLIN or another editor, add the following line to the file CONFIG.SYS (if CONFIG.SYS does not yet exist, create a new file with that name. See the Learning About Commands chapter.):

```
DEVICE=ANSYSYS
```

PROGRAMMING OF FUNCTION KEYS

Now all you have to do is type an NCR-DOS PROMPT command. You normally think of this command to alter the system prompt, but you can also use it to alter the character belonging to a particular key. The general format of the command is

```
PROMPT $e[0;#;"string";0;#....p
```

the '\$', lowercase 'e', and '[' characters are to be typed as true characters. # represents a number, this being the code for the key you wish to program. Following a semicolon is the contents you wish to assign to a particular key. This may be a string enclosed in double quotation marks, for example "DIR", or a direct (ASCII) value, for example 13 for <ENTER>, or any combination of string and direct value items. Items are separated from one another by semicolons. The last item, however, is not ended by a semicolon but by a lowercase p.

Normally, you use this form of the PROMPT command in order to program a Function Key. In this case, you should bear in mind that the code for a Function Key consists of not one but two numbers. The first of these two numbers is always 0. The second number is the one that selects a particular Function Key, including shifted (uppercase), Ctrl and Alt modes:

0; 59...68 Function Keys F1...F10 in base mode
 0; 84...93 Function Keys F1...F10 in shifted (uppercase) mode
 0; 94...103 Function Keys F1...F10 in Ctrl mode
 0;104...113 Function Keys F1...F10 in Alt mode

PROGRAMMING ALTERNATE MODE OF ALPHABETIC KEYS

If you wish to program an alphabetical key in the Alt mode, you have to select an appropriate second number from the following:

0;16...25 Alt Q, W, E, R, T, Y, U, I, O, P
 0;30...38 Alt A, S, D, F, G, H, J, K, L
 0;44...50 Alt Z, X, C, V, B, N, M

Second numbers for other keys and key combinations are given in the "Extended Codes" section of the *NCR-DOS PROGRAMMER'S MANUAL*, and in the diagram of the keyboard scan codes in the appendix of your *GW-BASIC* manual.

RESTORING THE SYSTEM PROMPT

After you have defined one or more keys using the PROMPT command, the system prompt disappears from the screen. To restore it, simply issue a second prompt command, this time without any characters or values, or define the prompt at the end of the PROMPT command line that defines the Key (\$...).

The new key assignment remains until you reload the operating system. If you frequently need a specific set of key assignments, it is a good idea to store a set of corresponding PROMPT commands in a batch file for execution immediately after loading the operating system.

EXAMPLES

The following PROMPT command assigns the DOS command "DIR" with a subsequent <ENTER>, to Function Key <F10>:

PROMPT \$e[0;68;"DIR";13p

To give the <ALT A> key combination the DOS command "TYPE"
followed by <ENTER>, issue the following command:

PROMPT \$e[0;30;"TYPE";13p

PROGRAMMING THE CURSOR AND DISPLAY GRAPHICS

You can change the location and appearance of the cursor and the appearance of the display screen by using the PROMPT $\$e[...$ format of the PROMPT command. The $\$e$ used with the PROMPT command defines the ESC(hex 1B) sequence. Include the DEVICE=ANSI.SYS command in the CONFIG.SYS file for the cursor and display change functions to work. The ANSI.SYS acts as an extended keyboard and screen *device driver* in this case, and supports the use of the $\$e$ (ESC) sequence with the PROMPT command.

Exercise extreme caution when using these functions. Your errors can cause unpredictable results and bring down the system.

Several cursor and display functions can be combined on a single line to perform a sequential series of functions using only one PROMPT $\$e[...$ command. This series of functions must end with a function to set the prompt, or, in some cases, the very next command must be a PROMPT command to set the prompt. If you do not set the prompt, no prompt is displayed after the function is executed. Do not combine functions to set the Function keys with functions of any other type function within one PROMPT $\$e[...$ command. Several functions can be combined to set multiple Function keys on one line, or several cursor and display functions can be combined on one line; but do not combine the two types of functions.

Use exactly the same case of alphabetic character, as is defined for each function. If an upper case or lower case character is defined, do not use the opposite case character; such as, h for H.

Default values are defined for each variable associated with a function. The default values for a function variable are used if no other value is entered or if you enter a value of zero for the variable.

The PROMPT $\$e[...$ command can be entered from the DOS command line. The more typical usage for this command is to have several PROMPT commands that set Function keys, place the cursor, set color attributes and define the prompt, in the AUTOEXEC.BAT file. Entering an individual PROMPT command for a cursor movement function from the command line, has little practical use.

The cursor and display graphics functions can be used with the PROMPT command, or these functions can also be used as control sequences when issued by DOS programming function calls that can write to an output device (the keyboard and display screen). (See your NCR MS-Macro Assembler manual for the commands that write to an output device.)

PROGRAMMING THE CURSOR

The cursor functions operate as indicated when you use them as control sequences for DOS programming function calls that write to a standard output device. The function notation is ESC[(hex 1B5B) followed by the function parameters (...).

When using these functions with the PROMPT command, you enter PROMPT \$e[...] instead of the ESC[...] notation. These functions perform differently when used with the PROMPT command.

- For cursor movement or status functions — the function is performed after the PROMPT command is entered. Therefore, the position of the cursor is at the beginning of the next line when the cursor movement or status function is executed.
- If a cursor movement function is executed and not immediately followed by a function to define the prompt on the same line, then each time the <ENTER> key is pressed, the PROMPT command you entered is executed again. This condition continues in effect only if or until the entire screen pages to a clear screen, then the PROMPT command is no longer in effect. If you do not define the prompt, no prompt text is displayed after the PROMPT function is executed.
- You should follow the PROMPT function with a function to define the prompt on the same line. This makes both the cursor movement or status function and prompt definition function remain in effect until another PROMPT command is entered. After the <ENTER> key is pressed, DOS executes the command entered on the command line first, then the PROMPT \$e[...] command that you entered.

FUNCTION DEFINITION

ESC[#;#H Cursor Position — moves the cursor to the position specified by the two parameters. The first parameter specifies the line number and the second parameter specifies the column number. The default value is one, which moves the cursor to the home position.

ESC[#A Cursor Up — moves the cursor up the number of lines specified. The default value is one. This function is ignored if the cursor is already on the top line.

- ESC[#B** **Cursor Down** — moves the cursor down the number of lines specified. The default value is one. This function is ignored if the cursor is already on the bottom line.
- ESC[#C** **Cursor Forward** — moves the cursor forward the number of columns specified. The default value is one. This function is ignored if the cursor is already in the rightmost column.
- ESC[#D** **Cursor Backward** — moves the cursor back the number of columns specified. The default value is one. This function is ignored if the cursor is already in the leftmost column.
- ESC[##f** **Horizontal and Vertical Position** — moves the cursor to the position specified by the two parameters. The first parameter specifies the line number and the second parameter specifies the column number. The default value is one, which moves the cursor to the home position (same as the **Cursor Position** function).
- ESC[6n** **Device Status Report** — reports the current cursor position. The format of this report is / [#;#R, where the first parameter specifies the current line and the second parameter specifies the current column. This function should not be used with the **PROMPT** command.
- ESC[s** **Save Cursor Position** — saves the current cursor position. This cursor position can be restored with the **Restore Cursor Position** function.
- ESC[u** **Restore Cursor Position** — restores the cursor to the value it had when the console driver received the **Save Cursor Position** function.
- ESC[2J** **Erase Display** — erases all of the screen and the cursor goes to the home position.
- ESC[k** **Erase Line** — erases from the cursor to the end of the line, including the cursor position.

PROGRAMMING THE DISPLAY GRAPHICS

The display graphics functions operate as indicated when used as control sequences for DOS programming function calls that write to a standard output device. The function notation is ESC[(hex 1B5B) followed by the function parameters (...).

When using these functions with the PROMPT command, you enter PROMPT \$e[...] instead of the ESC[...] notation.

You should follow the PROMPT function with a function to define the prompt on the same line. This makes both the display graphics function and prompt definition function remain in effect until another PROMPT command is entered. After the <ENTER> key is pressed, DOS executes the command entered on the command line first, then the PROMPT \$e[...] command that you entered. If you do not define the prompt, no prompt text is displayed after the PROMPT function is executed.

FUNCTION DEFINITION

ESC[#;...;#m Set Graphic Rendition — sets the attribute specified by the # parameter(s). All screen and display characters have the attributes defined by the # parameter(s), until this function is entered again. Parameters 30 through 47 apply to color graphic displays only.

Parameter # Definition

0	All attributes Off (white on black — DOS default)
1	Bold On (high intensity)
4	Underscore On (Monochrome Display only)
5	Blink On
7	Reverse video On (Black and White)
8	Cancelled On (invisible)
30	Black foreground
31	Red foreground

32	Green foreground
33	Yellow foreground
34	Blue foreground
35	Magenta foreground
36	Cyan foreground
37	White foreground
40	Black background
41	Red background
42	Green background
43	Yellow background
44	Blue background
45	Magenta background
46	Cyan background
47	White background

FUNCTION DEFINITION

ESC[=#h Set Mode — defines the screen width or type specified by the # parameter. You must follow this function with a second PROMPT command to set the prompt, or the prompt returns to the same starting position each time the ENTER key is pressed. Functions 0 through 6 apply to color graphic displays only.

Parameter # Definition

0	40 cols x 25 lines black and white
1	40 cols x 25 lines color

- 2 80 cols x 25 lines black and white
- 3 80 cols x 25 lines color
- 4 320 x 200 pixels color (no cursor)
- 5 320 x 200 pixels black and white (no cursor)
- 6 640 x 200 pixels black and white (no cursor)
- 7 Sets wrap at end of line (typing past end-of-line wraps around to the beginning of the next line. You can enter characters until the keyboard input buffer is full). This is the DOS default mode.

FUNCTION DEFINITION

ESC[=#1 Remove Mode — removes the screen width of type specified by the # parameter. Parameters are the same as the Set Mode function except that parameter 7 turns off wrap at end-of line (you can not enter characters past the end-of-line). You must follow this function with a second PROMPT command to set the prompt, or the prompt returns to the same starting position each time the ENTER key is pressed.

Examples

The following example displays the date and time in the upper-right corner of the screen in reverse video and sets the prompt to the current pathname preceded by "<" and followed by ">". This series of functions is executed by DOS each time the ENTER key is pressed (following the execution of the command entered on the command line).

```
PROMPT $e[s$e[1;50f$e[7m$d/$t$e[0m$e[u$1$p$g
```

- The \$e[s function saves the current position of the cursor.
- The \$e[1;50f positions the cursor on line 1 column 50.
- The \$e[7m sets the reverse video.
- The \$d/\$t gets the date and time.

- The `$e[0m` sets all attributes off (to turn off the reverse video).
- The `$e[u` restores the cursor to the position saved earlier.
- The `lp$g` sets the cursor to the `<pathname>`.

If you have a color graphic display, you could modify the above PROMPT command to set your display color attributes also. The following example sets the background to blue (44) and the foreground to yellow (33) (after resetting all attributes (`$e[0m`) following the reverse video (`$e[7m`) of the time/date display).

PROMPT \$e[s\$e[1;50f\$e[7m\$d/\$t\$e[0;33;44m\$e[u\$1\$p\$g

These PROMPT commands, or others of your own design, can easily be included in an AUTOEXEC.BAT file and executed each time the system is started.



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] filename[.ext][/B]
```

where:

/B means to ignore embedded
Ctrl-Z characters (end-of-file markers)

If you are creating a new file, the filename[.ext] 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 over (does not copy) one character in the stored line.
Esc	voids the current input line; leaves the stored line unchanged.
Ins	Enters insert mode. Press again to exit mode when done. (Second Ins is not needed if F1-F5 pressed.)
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 all remaining characters from the stored line to the new line.
F4x	Skips over (does not copy) 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, the filename[.ext] you enter 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 in order 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.

NOTE: 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 *DOS Commands* chapter); then start EDLIN and specify the new filename.ext.

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 \USERS\SUE\TEXT.TXT` allows you to edit the `TEXT.TXT` file in the subdirectory `SUE`.
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**

NOTE: In the case of a Search or Replace command, the string may be ended by an `<F6>` or `<Ctrl-Z>` instead of `<ENTER>`.

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.

***S This string <F6>-5,+5L**

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**
finds the first occurrence
of Ctrl-Z in a file

***R<Ctrl-V>Z<Ctrl-Z>**
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 show you the entire file including any embedded Ctrl-Z characters.
7. <line> when used in this text refers to the line number. Do not type "line", just the number of the desired line.

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.

<line> may be specified in one of three ways:

- Number (n) — Any number less than 65529. 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 <line>, then this means the current line which 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.

<ENTER>

Pressing the **<ENTER>** key 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 **<ENTER>** 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. A **<string>** (in the Search command) or **<string1>** (in the Replace command) represents text to be found, and **<string2>** (in the Replace command) is the text to replace other text. The **<string>** option is used only with the Search and Replace commands. Each **<string>** must end with an **<F6>** or **<ENTER>** (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.

n

The n option can be used to specify the number of lines to read from disk or write to disk with the Append (read) or Write command.

TABLE OF COMMANDS

The table in Figure 6.2 summarizes the EDLIN commands. The table defines the purpose and the format of each command.

COMMAND	PURPOSE	FORMAT
line-number	Enter edit mode at given (or next) line number	[<line>]
A(PPEND)	Append (read) more lines from disk	[n]A
C(OPY)	Copy a range of lines within the text	[line1],[line2],line3,[count]C
D(ELETE)	Delete a range of lines	{line1}[.line2]D
E(ND)	End the editing session	E
I(NSERT)	Enter Insert mode before given (or current) line number	[line]I
L(IST)	List a range of lines	[line1][.line2]L
M(OVE)	Moves a range of lines within the text	[line1],[line2],line3M
P(AGE)	Page through the text	[line1][.line2]P
Q(UIT)	Quit the editing session	Q
R(EPLACE)	Replace one string with another	[line1][.line2][?] R[<string1>] [<F6> <string2>]
S(EARCH)	Search for a string	[line1][.line2][?] S<string>
T(RANSFER)	Transfer contents of another file into current file	[line]T[d:] filename[.ext]
W(RITE)	Write lines to disk	[n]W

Figure 6.2 EDLIN Commands

COMMAND DESCRIPTIONS

The following pages contain the detailed reference descriptions of each of the EDLIN commands.

LINE-NUMBER

PURPOSE

Enters Edit mode; specifies line of text to be edited.

FORMAT

[<line>]

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 <ENTER> key is pressed.

If no line number is typed (that is, if only the <ENTER> 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 <ENTER> key to accept the line as is.

CAUTION

If the <ENTER> 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 <ENTER>

The contents of the line is displayed:

4:*four.
4:*

LINE-NUMBER

Now, using the editing keys, type:

<Ins>number <F3>

You will 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.

FORMAT

[n]A

COMMENTS

This command is used 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 does 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.

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.

FORMAT

[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 before which the text will be copied
- count the number of times the lines are to be copied.

COMMENTS

If you do not specify a number in "count", the EDLIN default is to copy 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 and the first of the copied lines becomes the current line number.

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

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

C(OPY)

- 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 file.
- 14: End of sample file.

D(ELETE)

PURPOSE

Deletes a specified range of lines in a file.

FORMAT

[line1[,line2]]D

COMMENTS

If the first line number is omitted, that option defaults 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 is 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

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 the text
7: in your file.
```

D(ELETE)

To delete a single line, type only the line number desired. For example,

6D

Then, when you list the sample file with the **L** command, 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 the 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
L

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.

FORMAT

E

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 can not 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 cancelled 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(INSERT)

PURPOSE

Inserts text immediately before the specified line number.

FORMAT

[line number]I

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 the <ENTER> key is pressed.

EDLIN remains in insert mode until <Ctrl-Break> or <F6> is entered. 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 the 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:

I(INSERT)

Now, type the new text for line 7:

7: **and renumber lines**

Then to end the insertion:

8: <F6>

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 the text
7: and renumber lines
8:*in your file.

To insert lines immediately before the current line, type:

I

The result is:

8: _

Now, type the following text:

8: **so they are consecutive**

9: <F6>

Now, when you list the file you 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 the 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

I(INSERT)

This produces the following:

10:

Now, type the following new lines:

- 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 65529.**

End the insertion by pressing <F6> on line 14.

14: <F6>

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 the 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 65529.**

PURPOSE

Lists a range of lines, including the two lines specified.

FORMAT

[line1] [,line2]L

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 fewer 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.

L(IST)

·
·
·

26: to edit the 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 the text

To list a range of 23 lines centered around the current line, type:

L

The result is:

L(IST)

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 the text

M(OVE)

PURPOSE

Moves a range of text to the line specified.

FORMAT

[line1],[line2],line3M

COMMENTS

Use the **Move** command to move a block of text (from the first line number (line1) through the second (line2)) to another location in the file (in front of 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

Relocates the block of text from the current line plus 25 lines 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

Displays a block of lines, and moves current line marking to last line displayed.

FORMAT

[line1] [,line2]P

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 last line displayed becomes the new current line and is marked with an asterisk.

Successive P commands, with no line numbers specified, page through the file 23 lines at a time.

Q(UIT)

PURPOSE

Quits the editing session, does not save any editing changes, and exits to the NCR-DOS operating system.

FORMAT

Q

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.

FORMAT

[line1] [,line2] [?] R[<string1>][<F6> <string2>]

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 <F6>(or <Ctrl-Z). String2 ends with <ENTER>.

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 <ENTER>. 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,

R<string1><F6>

deletes occurrences of string1, but

R<string1><ENTER>

replaces the new string1 with the old string2.

R<ENTER>

replaces the old string1 with the old string2.

Note that "old" here refers to a previous string specified in either a Search or a Replace command.

R(EPLACE)

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 <ENTER> 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 <ENTER> 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 the 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 65528.

To replace all occurrences of string1 (in this example, "and") with string2 ("or") in a specified range (lines 2 thru 12), type:

2,12 R and <F6> or <ENTER>

The result is:

- 4: Delete or Insert
- 5: (the D or I commands)
- 8: The insert command 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:

R(EPLACE)

2,12?R and <Ctrl-Z> or <ENTER>

The result is:

4: Delete or Insert
O.K.? Y
5: (the D or I commands)
O.K.? Y
5: (the D or I command)
O.K.? N
8: The insert command 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.

FORMAT

[line1] [,line2] [?]S<string>

COMMENTS

The string must end by pressing ENTER. 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 <ENTER> 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 the 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;

S(EARCH)

11:*they'll go all the way to 65529.

To search for the first occurrence of the string "and," type

2,12 Sand

The following line is displayed:

4: Delete and Insert

to get the "and" in line 5, modify the search command by typing:

<F3>

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

The result is:

4: Delete and Insert
O.K.?

If you press any key (except Y or <ENTER>), 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

EDLIN finds a match and continues to search for the same string when you enter the S command:

S(EARCH)

S

EDLIN will continue to search when you issue the S command until there are no more occurrences, which brings 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[.ext] into the file currently being edited at <line>. If <line> is omitted, then the current line is used.

FORMAT

[<line>]T[d:]filename[.ext]

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> 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.

FORMAT

[n]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 to disk 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 can not 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 format error.

Action

Retype the command with the correct format and press <ENTER>.

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- 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 or filename

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, etc.

Link Program (NCR-LINK)

GENERAL INFORMATION

In this chapter you learn about NCR-LINK. You should 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 one or more 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 run through a compiler or an assembler, that produces object modules (see MASM.EXE in your NCR MS-Macro Assembler manual). 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 gives you the option to produce 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.

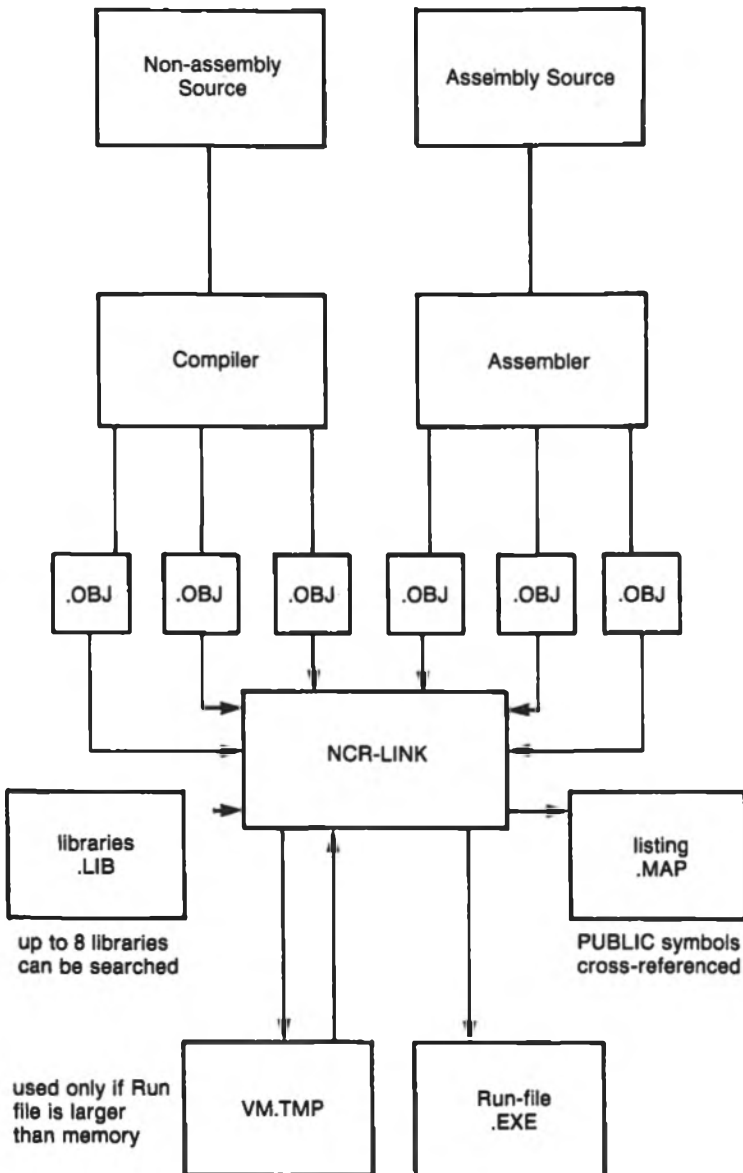


Figure 7.1 The LINK Operation

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 continuously.

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.

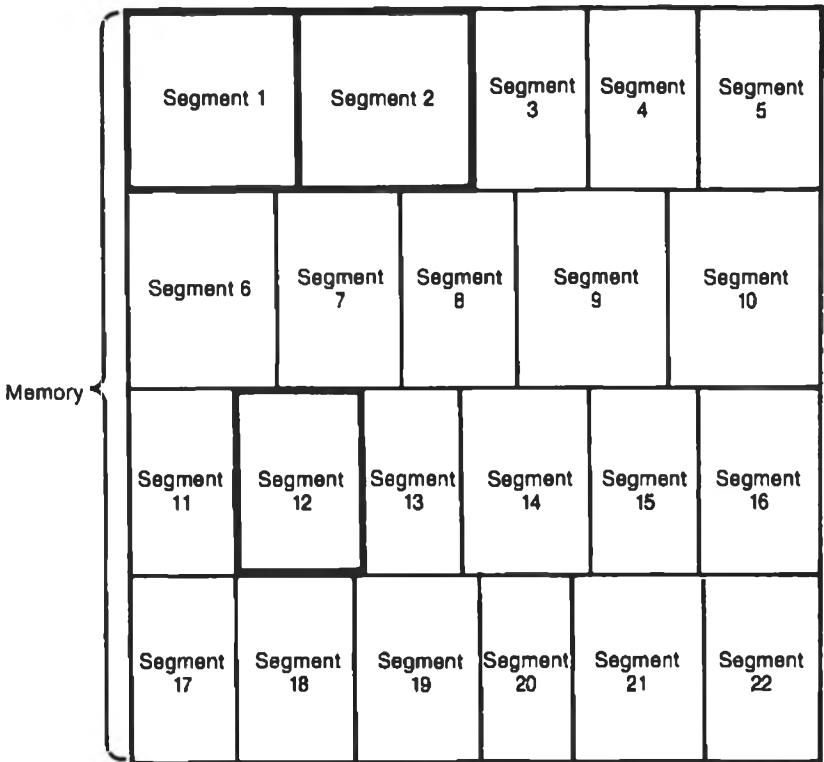


Figure 7.2 How memory is divided

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. 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 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 LINK commands and tell LINK where that file is when you start LINK. This method is best used if you have many source programs and anticipate extensive relinking.

- 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 [filename.EXE]:	Give filename for executable object code. The default is first-object-filename.EXE. (You cannot change the output filename extension.)
List File [NUL.MAP]:	Give filename if you want a listing. The default is no listing (NUL).
Libraries [.LIB]:	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 format:

LINK object-list, runfile, listfile, library-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.
- library-list is a list of library modules to be searched.
- /option refers to processing choices (see the "Link Options" section later in this chapter) which may be placed following any of the response entries (just before any of the commas or after the 'library-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 @ [d:][path] filename[.ext]

where the 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 <ENTER>), 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. Note that the second response line consists only of the /PAUSE and /MAP options. Thus the default value for Run File (first object filename + .EXE, in this case PROCESS.EXE) has effectively been chosen. 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 <ENTER>, 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 module or library names.) To type a large number of responses (each of which may be very long), type a plus sign and a <ENTER> at the end of the line to extend it. If the plus sign and <ENTER> 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 <ENTER> and not a plus sign and <ENTER>.

Example:

Object Modules [.OBJ]: **PROCESS TEXT TABLE CALC+**
 Object Modules [.OBJ]: **READDATA+VALDATE+**
 Object Modules [.OBJ]: **BILDPRNT+**
 Object Modules [.OBJ]: **PIECHART**

Semicolon

To select default responses to the remaining prompts, use a single semicolon (;) followed immediately by a <ENTER> at any time after the first prompt (Run File:). This feature saves time and overrides the need to press a series of <ENTER> keys.

NOTE: Once the semicolon has been typed and entered (by pressing the <ENTER> 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 <ENTER> key.

Example:

Object Modules [.OBJ]: **PROCESS TEXT TABLE
 CALC<ENTER>**
 Run Module [PROCESS.EXT]: **;<ENTER>**

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 <ENTER> 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 <ENTER>, 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 [NUL.MAP]:

The list file contains an entry for each segment in the input (object) modules. Each entry shows the addressing in the Run file. This file is very useful when using the DEBUG Utility.

The default response is no listing (NUL).

Libraries [.LIB]:

The valid responses are up to eight library filenames or simply <ENTER>. 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 can not 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	/DSALLOCTE

/DSALLOCATE

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 you press any 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 should contain a stack allocation statement. If not, LINK displays the following warning message:

WARNING: NO STACK STATEMENT

SAMPLE LINK SESSION

This sample shows you the type of information that is displayed during a 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]: **;<ENTER>**

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 <ENTER> in response to the "Libraries:" prompt, an 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	009E8H	09E8H	PROCESS	CODE
009F0H	01188H	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 assembly 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 link modules that LINK can not 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.



DEBUG Utility

INTRODUCTION

OVERVIEW OF DEBUG

The NCR DEBUG Utility (DEBUG) is a debugging program that provides a controlled testing environment for binary and executable object program files. DEBUG lets you enter commands to load, alter, display and execute object coding in memory, and to read or write object program files. DEBUG facilitates the development and testing of programs coded in the MS™-MACRO ASSEMBLER language. It also provides a tool for examining memory and files to aid in resolution of system problems in abnormal situations.

For example, if a program called FILE.EXE is being debugged, then the following is a typical command to start DEBUG:

DEBUG FILE.EXE

DEBUG loads FILE.EXE into memory starting at 100 hexadecimal in the lowest available segment. The BX:CX registers are loaded with the number of bytes placed into memory.

Note that DEBUG responds with the hyphen (-) prompt to notify you to enter a command.

An <arglist> may be specified if <filename> is present. The <arglist> is a list of filename parameters and switches that are to be passed to the program <filename>. Thus, when <filename> is loaded into memory, it is loaded as if it had been started with the command:

A> <filename><arglist>

ALL DEBUG commands may be aborted at any time by pressing <CONTROL-C>. <CONTROL-S> suspends the display, so that you can read it before the output scrolls away. Entering any key other than <CONTROL-C> or <CONTROL-S> restarts the display. All

of these commands are consistent with the control character functions available at the NCR-DOS command level.

Note that entering DEBUG without <filename> or arguments permit you to enter the DEBUG program with no specific file loaded. You can then use DEBUG commands to display ROM memory or operating system memory, or perform other functions as desired to resolve problems.

CAUTION

1. When DEBUG is started, it sets up a program header at offset 0 in the program work area. You can overwrite the default header if no <filespec> is given to DEBUG. If you are debugging a .COM or .EXE file, however, do not tamper with the program header below address 5CH, or DEBUG terminates.
2. Do not restart a program after the "Program terminated normally" message is displayed. You must reload the program with the N and L commands or re-enter DEBUG from DOS for it to run properly.

COMMANDS

COMMAND CONVENTIONS

Each DEBUG command consists of a single letter followed by one or more parameters. Additionally, the control characters and the special editing functions described in the *Editing and Function Keys* chapter of this manual, apply inside DEBUG.

If a format error occurs in a DEBUG command, DEBUG reprints the command line and indicates the error with an up-arrow (↑) and the word "error."

For example:

```
dcs:100 cs:110
  ↑ error
```

Any combination of uppercase and lowercase letters may be used in commands and parameters.

COMMAND PARAMETERS

All DEBUG commands accept parameters, except the Quit command.

Types of DEBUG parameters are defined as follows:

Parameter	Definition
<drive>	A one-digit hexadecimal value to indicate which drive a file is loaded from or written to. The valid values are 0-3. These values designate the drives as follows: 0=A:, 1=B:, 2=C:, 3=D:.
<bytes>	A two-digit hexadecimal value to be placed in or read from an address or register.
<record>	a 1- to 3-digit hexadecimal value used to indicate the logical record number on the disk and the number of disk sectors to be written or loaded. Logical records correspond to sectors. However, their numbering differs since they represent the entire disk space.
<value>	A hexadecimal value up to four digits used to specify a port number or the number of times a command should repeat its functions.
<address>	A two-part designation consisting of either an alphabetic segment register designation or a four-digit segment address plus an offset value. The segment designation or segment address may be omitted, in which case the default segment is used. DS is the default segment for all commands except G, L, T, U, and W, for which the default segment is CS. All numeric values are hexadecimal.

For example:

```
CS:0100
04BA:0100
```

The colon is required between a segment designation (whether numeric or alphabetic) and an offset.

<range> Two <address>es: e.g., <address> <address>; or one <address>, an L, and a <value>: e.g., <address> L <value> where <value> is the number of lines the command should operate on, and L80 is assumed. The last form cannot be used if another hex value follows the <range>, since the hex value would be interpreted as the second <address> of the <range>.

Examples:

```
CS:100 110
CS:100 L 10
CS:100
```

The following is illegal:

```
CS:100 CS:110
      ↑ error
```

The limit for <range> is 10000 hex. To specify a <value> of 10000 hex within four digits, type 0000 (or 0).

<list> A series of <byte> values or of <string>s. <list> must be the last parameter on the command line.

Example:

```
fcs:100 42 45 52 54 41
```

<string> Any number of characters enclosed in quote marks. Quote marks may be either single (') or double ("). If the delimiter quote marks must appear within a <string>, the quote marks must be doubled. For example, the following strings are legal:

```
'This is a "string" is okay.'
```

```
"This is a "string" is okay."
```


However, this string is illegal:

'This is a 'string' is not.'

Similarly, these strings are legal:

"This is a 'string' is okay."

"This is a ""string"" is okay."

However, this string is illegal:

"This is a "string" is not."

Note that the double quote marks are not necessary in the following strings:

"This is a "string" is not necessary."

"This is a ""string"" is not necessary."

The ASCII values of the characters in the string are used as a <list> of byte values.

Parameters may be separated by delimiters (spaces or commas), but a delimiter is required only between two consecutive hexadecimal values. Thus, the following commands are equivalent:

```

dcs:100 110
d cs:100 110
d,cs:100,110

```

Note that not all commands use all parameters. Individual commands and their specific parameters are discussed in the "Command Descriptions" section where commands are arranged in alphabetical order for ease of reference.

TABLE OF COMMANDS

Figure 8.1 summarizes the available Debug commands. This table shows the purpose and format of each command, and the page number on which the detailed description of each is found.

COMMAND	PURPOSE	FORMAT	PAGE
ASSEMBLE	Assembles source programs in memory	A[<address>]	8-9
COMPARE	Compares two variables	C<range> <address>	8-11
DUMP	Displays the contents of memory	D[<range>]	8-12
ENTER	Enters data values into memory	E<address> [<list>]	8-14
FILL	Fills memory with data values	F<range> <list>	8-16
GO	Executes a program in memory	[<address> [<address>...]]	8-17
HEX	Adds two values in hexadecimal	<value> <value>	8-19
INPUT	Displays input data from a port	I<value>	8-20
LOAD	Load a file into memory	L[<address> [<drive> <record> <record>]]	8-21
MOVE	Moves data from memory to memory	M<range> <address>	8-23
NAME	Names files	N<filename> [<filename>]	8-24
OUTPUT	Sends data to an output port	O<value> <byte>	8-27
QUIT	Terminates Debug	Q	8-28
REGISTER	Displays contents of registers	R[<register-name>]	8-29
SEARCH	Searches memory for data	S<range> <list>	8-32
TRACE	Displays executed commands and data	T[= <address>] [<value>]	8-33
UNASSEMBLE	Displays source of object code	U[<range>]	8-35
WRITE	Writes a file to disk	W[<address> [<drive> <record> <record>]]	8-37

Figure 8.1 Table of Debug commands

COMMAND DESCRIPTIONS

The following pages contain the detailed reference descriptions of each of the Debug commands.

A(SSEMBLE)

PURPOSE

Assembles 8086/8087/8088 mnemonics directly into memory.

FORMAT

A[<address>]

COMMENTS

If a format error is found, DEBUG responds with

↑ Error

and redisplays the current assembly address.

All numeric values are hexadecimal and must be entered as 1-4 characters. Prefix mnemonics must be specified in front of the opcode to which they refer. They may also be entered on a separate line.

The segment override mnemonics are CS:, DS:, ES:, and SS:. The mnemonic for the far return is RETF. String manipulation mnemonics must explicitly state the string size. For example, use MOVSW to move word strings and MOVSB to move byte strings.

The assembler will automatically assemble short, near or far jumps and calls, depending on byte displacement to the destination address. These may be overridden with the NEAR or FAR prefix. For example:

```
0100:0500 JMP 502           ; a 2-byte short jump
0100:0502 JMP NEAR 505      ; a 3-byte near jump
0100:505  JMP FAR 50A       ; a 5-byte far jump
```

The NEAR prefix may be abbreviated to NE, but the FAR prefix can not be abbreviated.

DEBUG can not tell whether some operands refer to a word memory location or to a byte memory location. In this case, the data type must be explicitly stated with the prefix "Word PTR" or "BYTE PTR". Acceptable abbreviations are "WO" and "BY". For example:

```
NEG  BYTE PTR [128]
DEC  WO [SI]
```

DEBUG also can not tell whether an operand refers to a memory location or to an immediate operand. DEBUG uses the common convention that operands enclosed in square brackets refer to memory. For example:

```
MOV  AX,21      Load AX with 21H
```

A(SSEMBLE)

```
MOV AX,[21]    Load AX with the
                contents
                of memory location 21H
```

Two popular pseudo-instructions are available with Assemble. The DB opcode will assemble byte values directly into memory. The DW opcode will assemble word values directly into memory. For example:

```
DB 1,2,3,4,"THIS IS AN EXAMPLE"
DB "THIS IS A QUOTE: " '
DB "THIS IS A QUOTE: ' "
```

```
DW 1000,2000,3000,"BACH"
```

Assemble supports all forms of register indirect commands. For example:

```
ADD BX,34[BP+2].[SI-1]
POP [BP+DI]
PUSH [SI]
```

All opcode synonyms are also supported. For example:

```
LOOPZ 100
LOOPE 100
```

```
JA 200
JNBE 200
```

For 8087 opcodes, the WAIT or FWAIT must be explicitly specified. For example:

```
FWAIT FADD ST,ST(3)  This line will assemble
                     an FWAIT prefix
LD TBYTE PTR [BX]   This line will not
```

C(OMPARE)

PURPOSE

Compares the portion of memory specified by <range> to a portion of the same size beginning at <address>.

FORMAT

C<range> <address>

COMMENTS

If the two areas of memory are identical, there is no display and DEBUG returns with the DOS prompt. If there *are* differences, they are displayed in this format:

<address1> <byte1> <byte2> <address2>

EXAMPLE

The following commands have the same effect:

C100,1FF 300

or

C100L100 300

Each command compares the block memory from 100 to 1FFH with the block of memory from 300 to 3FFH.

D(UMP)

PURPOSE

Displays the contents of the specified region of memory.

FORMAT

D[<range>]

COMMENTs

If a range of addresses is specified, the contents of the range are displayed. If the D command is typed without parameters, 128 bytes are displayed at the first address (DS:100) after the address displayed by the previous Dump command.

The dump is displayed in two portions: a hexadecimal dump (each byte is shown in hexadecimal value) and an ASCII dump (the bytes are shown in ASCII characters). Nonprinting characters are denoted by a period (.) in the ASCII portion of the display. Each display line shows 16 bytes with a hyphen between the eighth and ninth bytes. At times, displays are split in this manual to fit them on the page. Each displayed line begins on a 16-byte boundary.

If you type the command:

```
dcS:100 110
```

DEBUG displays the dump in the following format:

```
04BA:0100 42 45 52 54 41 ... 4E 44 TOM SAWYER
```

If you type the following command:

```
D
```

the display is formatted as described above. Each line of the display begins with an address, incremented by 16 from the address on the previous line. Each subsequent D (typed without parameters) displays the bytes immediately following those last displayed.

If you type the command:

```
DCS:100 L 20
```

the display is formatted as described above, but 20H bytes are displayed.

If then you type the command:

```
DCS:100 115
```


D(UMP)

the display is formatted as described above, but all the bytes in the range of lines from 100H to 115H in the CS segment are displayed.

E(ENTER)

PURPOSE

Enters byte values into memory at the specified <address>.

FORMAT

E<address>[<list>]

COMMENTS

If the optional <list> of values is typed, the replacement of byte values occurs automatically. (If an error occurs, no byte values are changed.)

If the <address> is typed without the optional <list>, DEBUG displays the address and its contents, then repeats the address on the next line and waits for your input. At this point, the Enter command waits for you to perform one of the following actions:

1. Replace a byte value with a value. Simply type the value after the current value. If the value typed in is not a legal hexadecimal value or if more than two digits are typed, the illegal or extra character is not echoed.
2. Press the <SPACE> bar to advance to the next byte. To change the value, simply type the new value as described in (1.) above. If you space beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning.
3. Type a hyphen (-) to return to the preceding byte. If you decide to change a byte behind the current position, typing the hyphen returns the current position to the previous byte. When the hyphen is typed, a new line is started with the address and its byte value displayed.
4. Press the <ENTER> key to terminate the Enter command. The <ENTER> key may be pressed at any byte position.

EXAMPLE

Assume that the following command is typed:

```
ECS:100
```

DEBUG displays:

```
04BA:0100 EB.
```

To change this value to 41, type 41 as shown:

E(NTER)

04BA:0100 EB.41

To step through the subsequent bytes, press the <SPACE> bar to see:

04BA:0100 EB.41 10. 00. BC.

To change BC to 42:

04BA:0100 EB.41 10. 00. BC.42

Now, realizing that 10 should be 6F, type the hyphen as many times as needed to return to byte 0101 (value 10), then replace 10 with 6F:

04BA:0100 EB.41 10. 00. BC.42-
04BA:0102 00.-
04BA:0101 10.6F

Pressing the <ENTER> key ends the Enter command and returns to the DEBUG command level.

F(ILL)

PURPOSE

Fills the addresses in the <range> with the values in the <list>.

FORMAT

F<range> <list>

COMMENTS

If the <range> contains more bytes than the number of values in the <list>, the <list> will be used repeated until all bytes in the <range> are filled. If the <list> contains more values than the number of bytes in the <range>, the extra values in the <list> will be ignored. If any of the memory in the <range> is not valid (bad or nonexistent), the error will occur in all succeeding locations.

EXAMPLE

Assume that the following command is typed:

```
F04BA:100 L 100 42 45 52 54 41
```

DEBUG fills memory locations 04BA:100 through 04BA:1FF with the bytes specified. The five values are repeated until all 100H bytes are filled.

PURPOSE

Executes the program currently in memory.

FORMAT

G[=`<address>` [`<address>`...]]

COMMENTS

If only the Go command is typed, the program executes as if the program had run outside DEBUG.

If =`<address>` is set, execution begins at the address specified. The equal sign (=) is required, so that DEBUG can distinguish the start =`<address>` from the breakpoint `<address>`s. If this option is chosen, be certain that there is an executable instruction at the =`<address>`. Otherwise unpredictable results can occur.

With the other optional addresses set, execution stops at the first `<address>` encountered, regardless of that address position in the list of addresses to halt execution or program branching. When program execution reaches a breakpoint, the registers, flags, and decoded instruction are displayed for the last instruction executed. (The result is the same as if you had typed the Register command for the breakpoint address.)

Up to ten breakpoints may be set. Breakpoints may be set only at addresses containing the first byte of an 8086 opcode. If more than ten breakpoints are set, DEBUG returns the BP Error message.

The user stack pointer must be valid and have 6 bytes available for this command. The Go command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, Code Segment register, and Instruction Pointer are pushed on the user stack. (Thus, if the user stack is not valid or is too small, the operating system may crash.) An interrupt code (0CCH) is placed at the specified breakpoint address(es).

When an instruction with the breakpoint code is encountered, all breakpoint addresses are restored to their original instructions. If execution is not halted at one of the breakpoints, the interrupt codes are not replaced with the original instructions.

EXAMPLE

Assume that the following command is typed:

```
GCS:7550
```

G(O)

The program currently in memory executes up to the address 7550 in the CS segment. DEBUG then displays registers and flags, after which the Go command is terminated.

After a breakpoint has been encountered, if you type the Go command again, then the program executes just as if you had typed the filename at the DOS command level. The only difference is that program execution begins at the instruction after the breakpoint rather than at the usual start address.

H(EX)

PURPOSE

Performs hexadecimal arithmetic on the two parameters specified.

FORMAT

H<value> <value>

COMMENTS

First, Hex adds the two parameters, then subtracts the second parameter from the first. The results of the arithmetic are displayed on one line; first the sum, then the difference.

EXAMPLE

Assume that the following command is typed:

H19F 10A

Hex performs the calculations and then displays the result:

02A9 0095

I(INPUT)

PURPOSE

Inputs and displays one byte from the port specified by <value>.

FORMAT

I<value>

COMMENTS

A 16-bit port address is allowed.

EXAMPLE

Assume that you type the following command:

I2F8

Assume also that the byte at the port is 42H. Input inputs the byte and displays the value:

42

L(OAD)

PURPOSE

Loads a file into memory.

FORMAT

L[<address> [<drive> <record> <record>]]

COMMENTS

BX:CX are set to the number of bytes read. The file must have been named either when DEBUG was started or with the N command. Both the DEBUG invocation and the N command format a filename properly in the normal format of a file control block at CS:5C.

If the Load command is typed without any parameters, DEBUG loads the file into memory beginning at address CS:100 and sets BX:CX to the number of bytes loaded. If the Load command is typed with an address parameter, loading begins at the memory <address> specified. If Load is typed with all parameters, absolute disk sectors are loaded, not a file. The <record>s are taken from the <drive> specified (the drive designation is numeric here--0=A:, 1=B:, 2=C:, etc.); DEBUG begins loading with the first <record> specified, and continues until the number of sectors specified in the second <record> have been loaded.

EXAMPLE

Assume that the following commands are typed:

```
A>DEBUG
-NFILE.COM
```

Now, to load FILE.COM, type:

```
L
```

DEBUG loads the file and then displays the DEBUG prompt. Assume that you want to load only portions of a file or certain records from a disk. To do this, type:

```
L04BA:100 2 OF 6D
```

DEBUG then loads 109 (6D hex) records from drive C: beginning with logical record number 15 into memory beginning at address 04BA:0100. When the records have been loaded, DEBUG simply returns the — prompt.

If the file has a .EXE extension, it is relocated to the load address specified in the header of the .EXE file: the <address> parameter is

L(OAD)

always ignored for .EXE files. The header itself is stripped off the .EXE file before it is loaded into memory. Thus the size of an .EXE file on disk will differ from its size in memory.

If the file named by the Name command or specified when DEBUG is started is a .HEX file, then typing the Load command with no parameters causes DEBUG to load the file beginning at the address specified in the .HEX file. If the Load command includes the option <address>, DEBUG adds the <address> specified in the Load command to the address found in the .HEX file to determine the start address for loading the file.

M(OVE)

PURPOSE

Moves the block of memory specified by <range> to the location beginning at the <address> specified.

FORMAT

M<range> <address>

COMMENTS

Overlapping moves (i.e., moves where part of the block overlaps some of the current addresses) are always performed without loss of data. Addresses that could be overwritten are moved first. The sequence for moves from higher addresses to lower addresses is to move the data beginning at the block's lowest address and then to work towards the highest. The sequence for moves from lower addresses to higher addresses is to move the data beginning at the block's highest address and to work towards the lowest.

Note that if the addresses in the block being moved will not have new data written to them, the data there before the move will remain. The Move command copies the data from one area into another, in the sequence described, and writes over the new addresses. This is why the sequence of the move is important.

EXAMPLE

Assume that you type:

```
MCS:100 110 CS:500
```

DEBUG first moves address CS:110 to address CS:510, then CS:10F to CS:50F, and so on until CS:100 is moved to CS:500. You should type the Dump command, using the <address> typed for the Move command, to review the results of the move.

N(AME)

PURPOSE

Sets filenames.

FORMAT

N<filespec>[<filespec>...]

COMMENTS

The Name command performs two functions. First, Name is used to assign a filename for a later Load or Write command. Thus, if you start DEBUG without naming any files to be debugged, then the N<filespec> command must be typed before a file can be loaded. Second, Name is used to assign filename parameters to the file being debugged. In this case, Name accepts a list of parameters that are used by the file being debugged.

These two functions overlap. Consider the following set of DEBUG commands:

```
-NFILE1.EXE  
-L  
-G
```

Because of the effects of the Name command, Name will perform the following steps:

1. (N)ame assigns the filename FILE1.EXE to the filename to be used in any later Load or Write commands.
2. (N)ame also assigns the filename FILE1.EXE to the first filename parameter used by any program that is later debugged.
3. (L)oad loads FILE1.EXE into memory.
4. (G)o causes FILE1.EXE to be executed with FILE1.EXE as the single filename parameter (that is, FILE1.EXE is executed as if FILE1.EXE had been typed at the command level).

A more useful chain of commands might look like this:

```
-NFILE1.EXE  
-L  
-NFILE2.DAT FILE3.DAT  
-G
```

Here, Name sets FILE1.EXE as the filename for the subsequent Load command. The Load command loads FILE1.EXE into memory, and

N(AME)

then the Name command is used again, this time to specify the parameters to be used by FILE1.EXE. Finally, when the Go command is executed, FILE1.EXE is executed as if FILE1.EXE FILE2.DAT FILE3.DAT had been typed at the NCR-DOS command level. Note that if a Write command were executed at this point, then FILE1.EXE (the file being debugged) would be saved with the name FILE2.DAT! To avoid such undesired results, you should always execute a Name command before either a Load or a Write command.

There are four regions of memory that can be affected by the Name command:

CS:5C	FCB for file 1
CS:6C	FCB for file 2
CS:80	Count of characters
CS:81	All characters typed

A File Control Block (FCB) for the first filename parameter given to the Name command is set up at CS:5C. If a second filename parameter is typed, then an FCB is set up for it beginning at CS:6C. The number of characters typed in the Name command (exclusive of the first character, "N") is given at location CS:80. The actual stream of characters given by the Name command (again, exclusive of the letter "N") begins at CS:81. Note that this stream of characters may contain switches and delimiters that would be legal in any command typed at the DOS command level.

EXAMPLE

A typical use of the Name command is:

```
DEBUG PROG.COM
-NPARAM1 PARAM2/C
-G
-
```

In this case, the Go command executes the file in memory as if the following command line had been typed:

```
PROG PARAM1 PARAM2/C
```

Testing and debugging therefore reflect a normal runtime environment for PROG.COM.

O(UTPUT)

PURPOSE

Sends the <byte> specified to the output port specified by <value>.

FORMAT

o<value> <byte>

COMMENTS

A 16-bit port address is allowed.

EXAMPLE

Type:

02F8 4F

DEBUG outputs the byte value 4F to output port 2F8.

Q(QUIT)

PURPOSE

Terminates the DEBUG utility.

FORMAT

Q

COMMENTS

The Q command takes no parameters and exits DEBUG without saving the current file. You are returned to the NCR-DOS command level.

EXAMPLE

To end the debugging session, type:

Q

DEBUG has been terminated, and control returns to the NCR-DOS command level.

R(EGISTER)

PURPOSE

Displays the contents of one or more CPU registers.

FORMAT

R[<register-name>]

COMMENTS

If no <register-name> is typed, the R command dumps the register save area and displays the contents of all registers and flags.

If a register name is typed, the 16-bit value of that register is displayed in hexadecimal, and then a colon appears as a prompt. You then either type a <value> to change the register, or simply press the <ENTER> key if no change is wanted.

The only valid <register-name>s are:

AX	BP	SS	
BX	SI	CS	
CX	DI	IP	(IP and PC both refer
DX	DS	PC	to the Instruction
SP	ES	F	Pointer.)

Any other entry for <register-name> results in a BR Error message.

If F is entered as the <register-name>, DEBUG displays each flag with a two-character SET/CLEAR alphabetic code (shown in the table below). To alter any flag, type the opposite two-letter SET/CLEAR code. The flags are either set or cleared.

The flags are listed below with their codes for SET and CLEAR:

FLAG NAME	SET	CLEAR
Overflow	OV	NV
Direction	DN Decrement	UP Increment
Interrupt	EI Enabled	DI Disabled
Sign	NG Negative	PL Plus
Zero	ZR	NZ
Auxiliary. Carry	AC	NA
Parity	PE Even	PO Odd
Carry	CY	NC

R(EGISTER)

Whenever you type the command **RF**, the flags are displayed in the order shown above in a row at the beginning of a line. At the end of the list of flags, **DEBUG** displays a hyphen (-). You may enter new flag values at alphabetic pairs. The new flag values can be entered in any order. You do not have to leave spaces between the flag entries. To exit the **R** command, press the **<ENTER>** key. Flags for which new values were not entered remain unchanged.

If more than one value is entered for a flag, **DEBUG** returns a **DF** Error message. If you enter a flag code other than those shown above, **DEBUG** returns a **BF** Error message. In both cases, the flags up to the error in the list are changed; flags at and after the error are not.

At startup, the segment registers are set to the bottom of free memory, the Instruction Pointer is set to 0100H, all flags are cleared, and the remaining registers are set to zero.

Type:

R

DEBUG displays all registers, flags, and the decoded instruction for the current location. If the location is **CS:11A**, then the display will look similar to this:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000
SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A NV UP DI
NG NZ AC PE NC
04BA:011A CD21 INT 21
```

If you type:

RF

DEBUG will display the flags:

```
NV UP DI NG NZ AC PE NC -
```

Now, type any valid flag designation, in any order, with or without spaces.

For example:

```
NV UP DI NG NZ AC PE NC — PLEICY<RETURN>
```

DEBUG responds only with the **DEBUG** prompt. To see the changes, type either the **R** or **RF** command:

R(EGISTER)

RF

NV UP EI PL NZ AC PE CY -

Press <ENTER> to leave the flags this way, or enter another RF command to specify different flag values.

S(EARCH)

PURPOSE

Searches the <range> specified for the <list> of bytes specified.

FORMAT

S<range> <list>

COMMENTS

The <list> may contain one or more bytes, each separated by a space or comma. If the <list> contains more than one byte, only the first address of the byte string is returned. If the <list> contains only one byte, all addresses of the byte in the <range> are displayed.

EXAMPLE

If you type:

```
SCS:100 110 41
```

DEBUG will display a response similar to this:

```
04BA:0104  
04BA:010D  
-type:
```

T(RACE)

PURPOSE

Executes one instruction and displays the contents of all registers and flags, and the decoded instruction.

FORMAT

T[=`<address>`][`<value>`]

COMMENTS

If the optional =`<address>` is typed, tracing occurs at the =`<address>` specified. The optional `<value>` causes DEBUG to execute and trace the number of steps specified by `<value>`.

The T command uses the hardware trace mode of the 8086 or 8088 microprocessor. Consequently, you may also trace instructions stored in ROM (Read Only Memory).

EXAMPLE

Type:

T

DEBUG returns a display of the registers, flags, and decoded instruction for that one instruction. Assume that the current position is 04BA:011A; DEBUG might return the display:

```
AX=0E00 BX=00FF CX=0007 DX=01FF SP=039D BP=0000
SI=005C DI=0000
DS=04BA ES=04BA SS=04BA CS=04BA IP=011A NV UP DI
NG NZ AC PE NC
04BA:011A CD21          INT          21
```

If you type

T=011A 10

DEBUG executes sixteen (10 hex) instructions beginning at 011A in the current segment, and then displays all registers and flags for each instruction as it is executed. The display scrolls away until the last instruction is executed. Then the display stops, and you can see the register and flag values for the last few instructions performed. Remember that `<CONTROL-S>` suspends the display at any point, so that you can study the registers and flags for any instruction.

U(NASSEMBLE)

PURPOSE

Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

FORMAT

U[<range>]

COMMENTS

The display of disassembled code looks like a listing for an assembled file. If you type the U command without parameters, 20 hexadecimal bytes are disassembled at the first address after that displayed by the previous Unassemble command. If you type the U command with the <range> parameter, then DEBUG disassembles all bytes in the range. If the <range> is given as an <address> only, then 20H bytes are disassembled instead of 80H.

EXAMPLE

Type:

```
U04BA:100 L10
```

DEBUG disassembles 16 bytes beginning at address 04BA:0100:

```
04BA:0100 206472 AND [SI+72],AH
04BA:0103 69 DB 69
04BA:0104 7665 JBE 016B
04BA:0106 207370 AND [BP+DI+70],DH
04BA:0109 65 DB 65
04BA:010A 63 DB 63
04BA:010B 69 DB 69
04BA:010C 66 DB 66
04BA:010D 69 DB 69
04BA:010E 63 DB 63
04BA:010F 61 DB 61
```

If you type

```
U04ba:0100 0108
```

The display will show:

```
04BA:0100 206472 AND [SI+72],AH
04BA:0103 69 DB 69
04BA:0104 7665 JBE 016B
04BA:0106 207370 AND [BP+DI+70],DH
```

U(NASSEMBLE)

If the bytes in some addresses are altered, the unassembler alters the instruction statements. The U command can be typed for the changed locations, the new instructions viewed, and the unassembled code used to edit the source file.

W(RITE)

PURPOSE

Writes the file being debugged to a disk file.

FORMAT

W[<address>[<drive> <record> <record>]]

COMMENTS

If you type Write with no parameters, BX:CX must already be set to the number of bytes to be written; the file is written beginning from CS:100. If the Write command is typed with just an address, then the file is written beginning at that address. If a Go or Trace command has been used, BX:CX must be reset before using the Write command without parameters. Note that if a file is loaded and modified, the name, length, and starting address are all set correctly to save the modified file (as long as the length has not changed).

The file must have been named either with the DEBUG invocation command or with the Name command (refer to the Name command earlier in this manual). Both the DEBUG invocation and the Name command format a filename properly in the normal format of a file control block at CS:5C.

If the Write command is typed with parameters, the write begins from the memory address specified; the file is written to the <drive> specified (the drive designation is numeric where 0=A, 1=B, 2=C, etc.); DEBUG writes the file beginning at the logical record number specified by the first <record>; DEBUG continues to write the file until the number of sectors specified in the second <record> have been written.

Files having a .HEX or .EXE extension can not be written using DEBUG Write. Therefore, if you want to make patches to these files you must first Rename the files, make the patches to the renamed file and again Rename the file back to its original name.

WARNING

Writing to absolute sectors is *EXTREMELY* dangerous because the process bypasses the file handler.

Type:

W

DEBUG will write the file to disk and then display the DEBUG prompt. Two examples are shown below.

W(RITE)

W

WCS:100 1 37 2B

DEBUG writes out the contents of memory, beginning with the address CS:100 to the disk in drive B. The data written out starts in disk logical record number 37H and consists of 2BH records. When the write is complete, DEBUG displays the prompt:

WCS:100 1 37 2B

-

ERROR MESSAGES

During the DEBUG session, you may receive any of the following error messages. Each error terminates the DEBUG command under which it occurred, but does not terminate DEBUG itself.

Error Code	Definition
BF	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	Too many breakpoints You specified more than ten breakpoints as parameters to the Go command. Retype the Go command with ten or fewer breakpoints.
BR	Bad register You typed the Register command with an invalid register name. See the Register command for the list of valid register names.
DF	Double flag You typed two values for one flag. You may specify a flag value only once per RF command.

SYSTEM REQUIREMENTS

The NCR DEBUG Utility requires:

A memory minimum that is program-dependent:

13K bytes for code

Run space is program-dependent

Disk drive(s):

1 disk drive if and only if output is sent to the same physical disk from which the input was taken. DEBUG does not allow time to swap disks during operation on a one-drive configuration. Therefore, two disk drives is a more practical configuration.

Command Format Summary Tables

The following tables summarize the five major types of commands discussed in this manual.

- Table A.1 contains the NCR-DOS commands. The contextual usage of these commands is described in Chapters 1, 2 and 3. Each command is explained in detail in Chapter 4, *DOS Commands*.
- Table A.2 contains the Batch commands. The contextual usage of these commands is described in Chapter 3, *Learning About Commands*. Each command is explained in detail in Chapter 4, *DOS Commands*.
- Table A.3 contains the Configuration commands. The contextual usage of these commands is described in Chapter 3, *Learning About Commands*. Each command is explained in detail in Chapter 4, *DOS Commands*.
- Table A.4 contains the EDLIN commands. The contextual usage and a detail explanation of each command is described in Chapter 6, *Line Editor (EDLIN)*.
- Table A.5 contains the DEBUG commands. The contextual usage and a detail explanation of each command is described in Chapter 8, *Debug Utility*.

Each of these tables contains the Name, Purpose and Format of each command.

COMMAND FORMAT SUMMARY TABLES

Name	Purpose	Format
ASSIGN	To assign a disk drive	ASSIGN [d1=d2...]
AUTOEXEC.BAT	A file containing a series of commands for batch processing: FOR, GOTO, IF, PAUSE, REM, SHIFT	
BACKUP	To copy fixed disk to flexible disks	BACKUP d1:[pathname] [filename[.ext]] d2:[/S] [/M] [/A] [/D:mm-dd-yy]
BREAK	To set Ctrl-C check (CONFIG.SYS)	BREAK ON OFF
BUFFERS	To tell DOS how many disk buffers to use (CONFIG.SYS)	BUFFER =nn
CHDIR (CD)	To change directories; display working directories	CHDIR [[d:]pathname]
CHKDSK	To scan the directory of the default or designated drive and check for consistency	CHKDSK [d:] [filename[.ext]] [/F] [/V]
CLS	To clear the screen	CLS
COMMAND	To invoke secondary command processor	COMMAND [d:][pathname][[/P] [/C string]]
COMP	To compare two files	COMP [filespec1] [filespec2]
CONFIG.SYS	A file containing commands that configures the operating system (BREAK, BUFFER, DEVICE, FILES, SHELL commands)	
COPY	To copy specified file(s) to disk	COPY [filespec1] [/A] [/B] [filespec2] [/V] [/A] [/B]
CTTY	To change console TTY	CTTY DEV
DATE	To display and set date	DATE [mm-dd-yy]
DEL	To delete file(s) specified	DEL [d:] [pathname][filespec]

Table A.1. NCR-DOS commands (1 of 4)

Name	Purpose	Format
DEVICE	To allow peripheral connection (CONFIG.SYS)	DEVICE=[d:][pathname] filename[.ext]
DIR	To list requested directory entries	DIR [d:][pathname] [filename[.ext]] [/P] [/W]
DISKCOMP	To compare flexible disks	DISKCOMP [d1:] [d2:] [/1] [/8]
DISKCOPY	To copy entire disks	DISKCOPY [d1:] [d2:] [/1]
ECHO	To turn batch file echo feature on/off (BATC H)	ECHO ON ECHO OFF ECHO <message>
ERASE	To delete specified files	ERASE [d:][pathname] filename[.ext]
ESC	Enables input of escape sequences through the keyboard	ESC [#;#..<delimiter> ESC ["string";...
EXE2BIN	To convert a file to .COM format	EXE2BIN [d1:][pathname1] filename1[.ext1] [d2:] [pathname2] filename2 [.ext2]
EXIT	To exit a command and return to lower level	EXIT
FDISK	To set up fixed disk	FDISK
FILES	To designate number of files (CONFIG.SYS)	File = nn
FIND	To search for a constant string of text	FIND [N] [/C] [/N] "string" [[d:][pathname] filename[.ext]...]
FOR..IN..DO..	To selectively process files by a DOS command. (BATC H) ● for batch processing ● for interactive processing	FOR %% <variable> IN <list> DO <command> %% <variable> FOR % <variable> IN <list> DO <command> % <variable>
FORMAT	To format disk for NCR-DOS files ● flexible diskettes ● fixed disks	FORMAT [d:] [/8] [/1] [/V] [/S] FORMAT [d:] [/V] [/S]
GOTO	To change execution sequence. (BATC H)	GOTO <label>
GRAFTABL	Supports character display in graphics mode	GRAFTABL
GRAPHICS	To print display on matrix printer	GRAPHICS

Table A.1. NCR-DOS commands (2 of 4)

Name	Purpose	Format
HELP	To display on-line data about commands	HELP [command]
IF	Batch command extension (BATCH)	IF [NOT] <condition> <command>
KEYB	Changes the keyboard format	See description
MKDIR (MD)	To make a directory	MKDIR [d:] [pathname]
MODE	To define peripheral interfaces	See description
MORE	To display output screen by screen	MORE
PATH	To set a command search path	PATH [[d1:]pathname1 [;[d2:]pathname2] [...]]
PAUSE	To pause for input (BATCH)	PAUSE [<message>]
PRINT	Background print feature; to print a text file on a line printer	PRINT [[d:] filename1[.ext1]] [/T] [/C] [/P]...
PROMPT	To designate command prompt	PROMPT [\$][prompt-text]
RAMDISK	To allocate Ram-Disk memory (CONFIG.SYS/DEVICE)	DEVICE=RAMDISK.SYS
RECOVER	To recover a bad disk	RECOVER [d:] [pathname] filename[.ext] RECOVER d:
REM	To display a comment in a batch file (BATCH)	REM [comment]
REN (RENAME)	To rename a file	REN [d:] [pathname]filename1[.ext1] filename2[.ext2]
RESTORE	To restore backed-up files from a flexible disk to the hard disk	RESTORE [d1:] [d2:] [pathname][filename.ext] [/S] [/P]
RMDIR (RD)	To remove a directory	RMDIR [d:] pathname
SET	To set one string value equal to another	SET [string1]=[string2]]
SHELL	To move command processor out of root directory (CONFIG.SYS)	SHELL = [d:][pathname]filename[.ext] [/P]/[C string]

Table A.1. NCR-DOS commands (3 of 4)

Name	Purpose	Format
SHIFT	To increase number of replaceable parameters in batch file (BATCH)	SHIFT
SORT	To sort data forward or backwards	SORT [/R] [/+n] [<filespec1][>filespec2]
SYS	To transfer NCR-DOS hidden system files from drive A to specified drive	SYS d:
TIME	To display and set time	TIME [hh[.mm[.ss[.cc]]]]
TREE	To display all directory paths	TREE [d:] [/F] [>PRN]
TYPE	To display contents of ASCII file on the screen	TYPE [d:] [pathname]filename[.ext]
VER	To display DOS version number	VER
VERIFY	To verify data when written to disk	VERIFY ON OFF
VOL	To display disk volume identification number	VOL [d:]

Table A.1. NCR-DOS commands (4 of 4)

COMMAND	PURPOSE	FORMAT
ECHO	To turn echo on and off during batch file processing	ECHO ON OFF message
FOR..IN..DO..	To selectively process files by a DOS command	FOR %%<variable> IN (list)DO <command>
GOTO	TO change sequence of execution of batch file statements	GOTO <label>
IF	TO conditional execution of a DOS or batch command during batch file processing	IF [NOT]<condition> <command>
PAUSE	TO suspend execution of the batch file	PAUSE [message]
REM	To display a message during batch file processing	REM [message]
SHIFT	To allow access to more than 10 replaceable parameters	SHIFT

Table A.2. Batch commands

COMMAND	PURPOSE	FORMAT
BREAK	allow more-frequent test for <Ctrl-Break>	BREAK ON OFF
BUFFERS	Change number of disk data storage areas in memory	BUFFERS=n
DEVICE	Define non-standard device and its driver routine	DEVICE=[d:][path-name]filename[.ext]
FILES	Set the number of files that can be open at one time	FILES=n
SHELL	Causes alternate command processor to be loaded	SHELL=[d:][path-name]filename[.ext]

Table A.3. Configuration commands

COMMAND	PURPOSE	FORMAT
line-number	Enter edit mode at given (or next) line number	[<line>]
A(PPEND)	Append (read) more lines from disk	[n]A
C(OPY)	Copy a range of lines within the text	[line1],[line2], line3,[count]C
D(ELETE)	Delete a range of lines	[line1],[line2]D
E(ND)	End the editing session	E
I(NSERT)	Enter insert mode before given (or current) line number	[line]
L(IST)	List a range of lines	[line1],[line2]L
M(QVE)	Moves a range of lines within the text	[line1],[line2], line3M
P(AGE)	Page through the text	[line1],[line2]P
Q(UIT)	Quit the editing session	Q
R(EPLACE)	Replace one string with another	[line1],[line2][?] R[<string1>] [<F6> <string2>]
S(EARCH)	Search for a string	[line1],[line2][?] S<string>
T(RANSFER)	Transfer contents of another file into current file	[line]T[d:] filename[.ext]
W(RITE)	Write lines to disk	[n]W

Table A.4. EDLIN commands

COMMAND FORMAT SUMMARY TABLES

COMMAND	PURPOSE	FORMAT	PAGE
ASSEMBLE	Assembles source programs in memory	A[<address>]	8-9
COMPARE	Compares two variables	C<range> <address>	8-11
DUMP	Displays the contents of memory	D[<range>]	8-12
ENTER	Enters data values into memory	E<address> [<list>]	8-14
FILL	Fills memory with data values	F<range> <list>	8-16
GO	Executes a program in memory	[<address> [<address>...]]	8-17
HEX	Adds two values in hexadecimal	<value> <value>	8-19
INPUT	Displays input data from a port	I<value>	8-20
LOAD	Load a file into memory	L[<address> [<drive> <record> <record>]]	8-21
MOVE	Moves data from memory to memory	M<range> <address>	8-23
NAME	Names files	N<filename> [<filename>]	8-24
OUTPUT	Sends data to an output port	O<value> <byte>	8-27
QUIT	Terminates Debug	Q	8-28
REGISTER	Displays contents of registers	R[<register-name>]	8-29
SEARCH	Searches memory for data	S<range> <list>	8-32
TRACE	Displays executed commands and data	T[= <address>] [<value>]	8-33
UNASSEMBLE	Displays source of object code	U[<range>]	8-35
WRITE	Writes a file to disk	W[<address> [<drive> <record> <record>]]	8-37

Table A.5. DEBUG commands

How to Obtain and Install Software

MAKING THE RIGHT PURCHASE

Your NCR PERSONAL COMPUTER 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
- 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 want to be sure, however, that the software runs on your computer. Knowing what questions to ask and working with a knowledgeable dealer helps 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 program so that you can tailor the software yourself. If the software is intended for use on an NCR PC or on IBM PC compatible computers, 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, select an IBM compatible computer, 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 (Refer to Chapter 5, *Editing and Function Keys*). Other tables include information you may need, depending on the particular application. Table 2 contains keyboard characteristics, Table 3 gives notes and frequencies in cycles for programming music applications, and Table 4 is an ASCII code chart included for convenient reference.

KEYBOARD FUNCTIONS

KEYS	DESCRIPTION
F1	Copies one designated character.
F2X	Copy up to designated character.
F3	Copies (remainder of) a line.
F4X FN	Skips up to designated character.
F5	Inserts an @ and 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.

←	Tab to left.
→	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 (numeric keypad).
←	Back cursor up one space and delete character occupying space (keyboard).
←	Move cursor to start of next line (ENTER 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.
↓	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 some text editors.
Scroll-Lock ↑	Scroll up one screen.
Scroll-Lock →	Scroll right one screen.
Scroll-Lock ↓	Scroll down one screen.
Scroll-Lock ←	Scroll left one screen.

The following table shows the CRT display characteristics of the NCR Personal Computer.

CRT DISPLAY 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

MUSICAL NOTES

Refer to Table 3 below for the note and corresponding frequency. The frequency can be any number between 0 and 32767. The duration (tone length) is the second parameter for musical notes, and can be any number from 0 to 65535. The duration number represents 5/91 of a second (or about 18 per second). For further information see the SOUND command in the accompanying *GW-BASIC* manual.

MUSIC CODES — Table 3

Note	Frequency in Hertz
Pause	-
A	110.
A#	116.5
B	123.5
C	131
C#	138.6
D	146.8
D#	155.8
E	164.8
Note	Frequency in hertz

F	174.6
F#	185
G	196
G#	208
A	220
A#	233
B	246.9
C (Middle C)	261.6
C#	277.4
D	293.7
D#	311
E	329.6
F	349.2
F#	370
G	392
G#	415
A (tuning note)	440
A#	465
B	493.9
C	523.3
C#	553
D	587.3
D#	622
E	659.3
F	698.5
F#	740
G	784
G#	830
A	880
A#	932
B	987.8
C	1046.5
C#	1108.7
D	1174.7
E	1238.5
F	1306.9
G	1368
A	1432
B	1495.5
# = sharp	

ASCII CODE CHART - Table 4

NCR ASCII CODE CHART																	
B_4-B_1		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
$B_8 \downarrow B_5$		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

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.

Access 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 DECISION MATE 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.

If the filename is SHELLDRV.SYS or TIME.SYS, you must re-install User Friendly SHELL.

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 processor 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 Ctrl-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.

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 flespecs 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.

^Error

DEBUG. A syntax error occurred. DEBUG responds with ^Error and redisplay 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.)

<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.

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.

<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.

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 of 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.

<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 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 <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 nonexistent 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.

**Probable 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 the indicated 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. Attempted 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 processing 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.

User Friendly SHELL error XX
Reinstall User Friendly SHELL

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.

THE UNIVERSITY OF CHICAGO

PHYSICS DEPARTMENT
5720 S. UNIVERSITY AVE.
CHICAGO, ILL. 60637

PHYSICS 351

LECTURE 10: ELECTROSTATICS

1. ELECTROSTATIC POTENTIAL

The electrostatic potential $V(\mathbf{r})$ is defined as the work done per unit charge in moving a test charge from infinity to the point \mathbf{r} .

2. ELECTROSTATIC POTENTIAL ENERGY

The electrostatic potential energy U of a system of charges is the work done in assembling the system from infinity.

3. ELECTROSTATIC POTENTIAL OF A POINT CHARGE

The electrostatic potential of a point charge q is given by

$$V(r) = \frac{1}{4\pi\epsilon_0} \frac{q}{r}$$

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
```

See "Configuration Change Feature" section of Chapter 2, Files and Directories, for details of creating a CONFIG.SYS file.

2. Now boot the system.
3. A message appears on the screen and informs you that the RAM-DISK has been installed.
4. Remove your DOS diskette and insert your DOS PROGRAMMERS TOOLS diskette into drive A.
5. Copy the two data files RAMDATA.1 and RAMDATA.2 to the RAM-DISK 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.

6. Now run the RAMDEMO.EXE program by entering:

```
"RAMDEMO"
```

after the system prompt.

7. The demonstration program asks you for the drive letters for the flexible or fixed disk and the RAM-DISK drive. By answering these questions correctly the program executes 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 is repeated on the RAM-DISK files also. The start and end times in each case are also displayed.

The speed improvements is 13-15 times faster than flexible disk drives and 4-5 times faster than fixed disk 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 1, “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 keyboard 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 diaeresis ("") 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 diaeresis followed by a space results in the box being printed.

Possible Dead Key Combinations

	Diacritical marks available			
	´	˘	¨	ˆ
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 Ê í ó ú à è ì ò ù
France	ä Ä ë ì ö Ö ü Ü ÿ â é í ò ú
Spain	ä Ä ë ì ö Ö ü Ü ÿ á é Ê í ó ú à è ì ò ù â ê í ò ù

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.

APPENDIX E

W

When loaded, the GRAFTABL.COM file initializes an 8x8 or an 8x8 and a 8x16 character generator and causes interrupt vector 1FH to point to an 8 by 128 byte or an 8 by 128 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 hatch-styles.

- 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 line-styles.

- 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:

<u>Opcode</u>	<u>Definition</u>
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

ADDITIONAL 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 data 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 data loss in case the original diskette is lost or damaged. 2. n: A copy of a file or diskette that contains backup data.

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 acronym 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.

CONSOLE

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 you may enter data or commands.

D

DATA

Any combination of letters, numbers, and 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 or aluminum alloy coated with magnetic material on which data 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 computer hardware, that spins the diskettes and either reads data from them, or transfers data 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 command or run a program.

F**FILE**

An identified collection of data stored on a disk or tape. A file can also refer to a body of data 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 one-character value represented in four bits (half a byte). The first 10 values in the system are the integers 0 through 9; the last 6 values are the letters A, B, C, D, E, F. Also called hex number. The four bits for value one is 0001; the four bits for F is 1111.

HIGH-LEVEL LANGUAGE

A computer programming language that is less dependent on the limitations of a specific computer; for instance, pseudo-languages; 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

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 where $K = 1024$ bytes.

KEYBOARD

A device for entering data to the computing system by pressing 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 data is encoded in the form of magnetic spots arranged in a manner to represent data.

MAIN MEMORY

Internal memory; the temporary memory that is maintained electrically in the computer.

MEMORY

The part of the personal computer that stores data. 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 data 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 conjunction 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 data is independent of the location of the data 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 can not change its contents.

S**SECTOR**

A portion of each of the tracks on a disk or diskette. There are usually nine sectors per track for a 5 1/4 inch diskette and 17 sector per track for an NCR PC fixed (hard) disk.

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 structural format rules of a computer language that must be followed when using the language. In this manual we call it Format.

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 changes independently or that can be changed.

W

WORD

In computer terminology, a unit of data that occupies a storage location. In most digital computers, a word is composed of a group of one or more bytes.

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. When covered with opaque material, it protects the data that is recorded on that diskette from being recorded over and thus lost.

INDEX

* character, 2-4

. character, 2-13

.. characters, 2-13

<ENTER> key, 1-4

<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

\ character, 2-12

? character, 2-4

< redirect input, 3-7

> redirect output, 3-6

ANSI.SYS file, 3-19, -21-23, 5-10, 5-13

appending data, 3-6

ASCII (chart), B-6

ASSIGN command, 4-7

assigning

 disk drive, 4-7

 RAM as disk, 2-17, 3-23

AUTOEXEC.BAT file, 1-8, 3-11-13, 4-9, A.2
automatic program execution, see AUTOEXEC.BAT
AUX, device name, 2-6

backing-up disks, 1-7, 4-10
BACKUP command, 1-7, 4-7
batch processing, 3-9
 suspending, 4-71
 replacement parameters 3-14-16
BREAK command, 4-13
BUFFERS command, 4-14

changing
 drives, 1-6
 partitions, 1-9
characters, LINK command, 7-10
CHDIR command, 2-14, 2-16, 4-15
CHKDSK command, 4-16
class, 7-4
clearing, screen, 4-20
CLS command, 4-20
command characters (LINK), 7-10
COMMAND.COM file, 4-21
commands,
 batch, 3-18, A.
 CONFIG.SYS, 3-24, A.3
 DEBUG 8-7, A.5
 DOS, 4-1, A.1
 editing, 6-7, A.3
 editing of, 5-4
 external, 2-14, 3-2
 filters, 3-7
 internal, 2-14, 3-1
 piping, 3-7
 summary, 4-3
COMP command, 4-22
comparing files, 4-22
CON, device name, 2-5
CONFIG.SYS file, 3-20, 4-24, A.2
configuring
 (system), 3-19, 4-24
converting files to binary form, 4-43

COPY command, 2-8, 4-25
copying files, 2-8, 4-25
creating
 a batch file, 3-9
 command processor, 4-84
CTTY command, 4-28
current directory, 2-12

data input/output, 3-6
DATE command, 4-29
default drive designation, 1-6
defining
 communications adapters, 4-66
 disk configuration, 1-8
 printer, 4-63
 video displays, 4-67
DELETE command, 4-31
deleting partitions, 4-48
DEVICE command, 4-32
device names, 2-3
DIR command, 1-6, 2-2, 4-36
directory, 2-1, 2-9
 path, 2-12, 4-70
disk backup, 1-7, 2-8, 4-10, 4-25
disk, master, 1-1
DISKCOMP command, 4-38
DISKCOPY command, 1-7, 4-39
display graphics 5-11
displaying
 directory, 1-6, 4-36
 files, 4-92
 partition, 4-49
 remarks, 4-77
 the directory, 2-2
 volume identifier, 4-95
drive designations, 1-2
 default, 1-6

ECHO command, 4-41
edit mode, 6-8

editing

- commands, 6-7
- files, 6-1
- keys, 5-1, 6-2
- of commands, 5-4
- options, 6-5
- programs, 6-1

editing,

- appending, 6-10
 - copying, 6-11
 - deleting, 6-13
 - exiting from, 6-15
 - inserting, 6-16
 - listing (lines), 6-19
 - moving text, 6-22
 - paging, 6-23
 - quit, 6-24
 - replacing, 6-25
 - searching, 6-28
 - transferring, 6-31
 - writing (to disk), 6-32
- entering date and time, 1-6
- ERASE** command, 4-42
- ESC** command, 4-42a
- EXE2BIN** command, 4-43
- EXIT** command, 4-44
- extension,
- common, 2-6
 - filename, 2-3, 7-6
- external commands, 3-2

FDISK command, 1-9

file

- allocation table, 2-2
 - protection, 2-9
- filenames, 2-3, 2-12
- filename extensions, 2-5, 7,6
- files, 2-1
- FILES** command, 4-51
- filters, 3-7
- FIND** command, 4-52
- fixed disk, 1-8
- flexible disk, 1-3

FOR command, 4-54
FORMAT command, 1-6, 4-55
format (syntax)
 command, 3-2, 4-1
 EDLIN, 6-4
 pathname, 2-12
formatting
 a disk, 1-6
 a fixed disk, 1-9
 function keys, 1-7, 5-1, 5-10
 prompt 5-10
 screen 5-16

GOTO command, 4-57
GRAFTABL command, 4-57a
GRAPHICS command, 4-58
group, 7-4

HELP, 1-10, 4-59
hidden files, 2-2, 4-88
hierarchical directory structure, 2-10

IF command, 4-60
input,
 data, 3-6
 filtering, 3-7
installing software, B-1
internal commands, 3-1
IO.SYS file, 2-2, 4-55, 4-88

keeping track of files, 2-1
KEYB command, 4-60a
keyboard functions, B-2

linking programs, 7-1
loading DOS/NCR User Interface, 1-5
LST, device name, 2-6

MKDIR command, 2-11, 4-61
MODE command, 4-62
MORE command, 4-69

MSDOS.SYS file, 2-2, 4-55, 4-88
multiple operating systems, 1-8
naming convention, 2-5

naming files, 2-3
NCR User Interface, 1-2, 1-9
notes, music, B-4
NUL, device name, 2-6

object modules, 7-1
operating system, 1-1
operating systems, multiple, 1-8
options, LINK, 7-13
output, redirecting, 3-6

partition, 1-9, 4-45
PATH command, 4-70
PAUSE command, 4-71
piping, 3-6
PRINT command, 4-72
printing
 files, 4-72
 graphics, 4-58
PRN, device name, 2-6
prompt, 1-5, 4-72, 5-10-19
PROMPT command, 4-72, 5-10-19
prompts, LINK, 7-13
protecting files, 2-9

RAMDISK.SYS file, 2-17
RECOVER command, 4-76
redirect output (>), 3-6
relocatable module, 7-1
REM command, 4-77
RENAME command, 4-78
replaceable parameters, 3-14, 4-85
RESTORE command, 4-79
RMDIR command, 2-15, 4-82
root directory, 2-10, 2-11

segment, 7-4

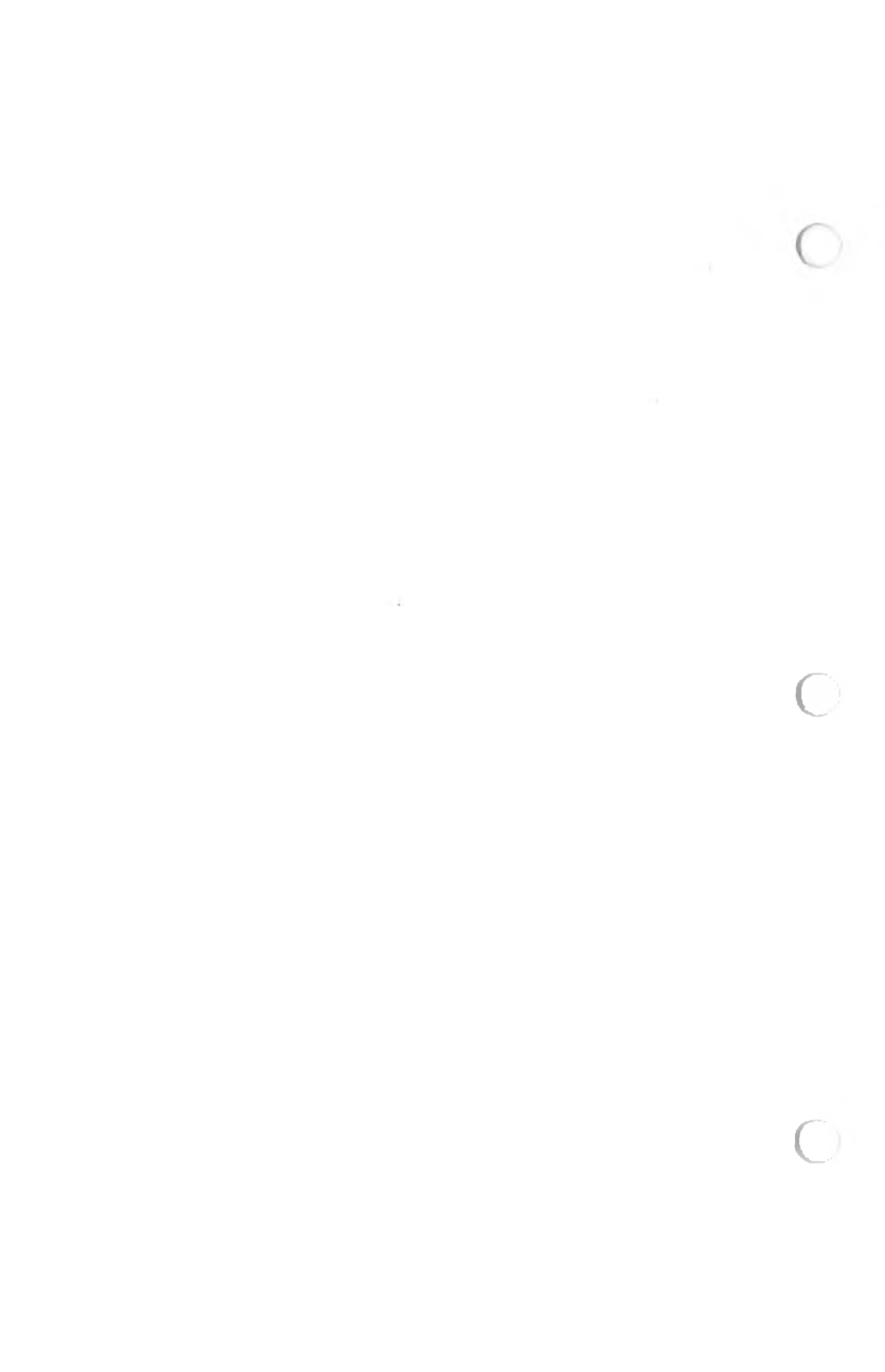
SET command, 4-83
SHELL command, 4-84
SHIFT command, 4-85
SORT command, 4-86
starting
 DOS, 1-5
 LINK, command line, 7-8
 prompts, 7-7
 response file, 7-9
SYS command, 4-88
TIME command, 4-89
TREE command, 4-91
TYPE command, 4-92

universal characters, 2-4

User Interface, 1-2, 1-9

VER command, 4-93
VERIFY command, 4-94
VM.TMP file, 7-2
VOL command, 4-95

wildcard characters, 2-4
working directory, 2-1, 2-15



NCR

NCR Corporation
Dayton, Ohio 45479

150-0000997