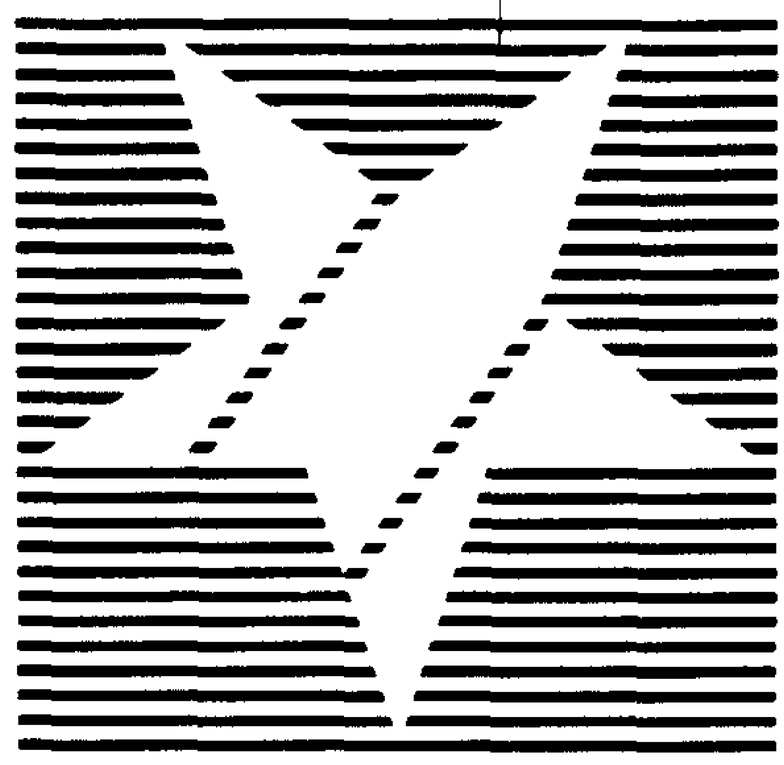
marcus bennett

NorthStar

North Star NetWare™ User's Manual

Release 1.1.0

for NetWare version 4.61
and MS-DOS version 3.10



04207B

Copyright Notice

All rights reserved No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording or otherwise, without the prior written permission of North Star Computers, Inc No patent liability is assumed with respect to the use of the information contained herein While every precaution has been taken in the preparation of this book, North Star Computers, Inc assumes no responsibility for errors or omissions Further, this publication and features described herein are subject to change without notice

North Star NetWare is an OEM version of NetWare™
Copyright © Novell Inc , 1983 All rights reserved

North Star NetWare is a trademark of North Star Computers, Inc.
NetWare is a trademark of Novell Inc.
Omninet is a trademark of Corvus Systems, Inc.
MS-DOS is a trademark of Microsoft Corporation
IBM PC and IBM PC/XT are trademarks of
International Business Machines Corporation
8086, 8087, 8088 are registered trademarks of Intel Corporation

Copyright ©1985 by North Star Computers, Inc.
All rights reserved

To reorder this manual, reorder part number 04246

PREFACE ix
Overview ix
SYMBOLS AND CONVENTIONS x
COMMAND RULES AND RESTRICTIONS xiii

CHAPTER 1 INTRODUCTION TO THE SYSTEM 1-1
THE CENTRAL MODULE 1-3
THE WORKSTATIONS 1-5
THE NORTH STAR WORKSTATIONS 1-6
THE KEYBOARD 1-7
Introduction 1-7
Typewriter and Control Keys 1-8
Function Keys and Numeric Keypad 1-10
THE SOFTWARE 1-13
STORING INFORMATION 1-14
NETWARE SECURITY FEATURES 1-22
Login/Password Security 1-23
Trustee Security 1-24
Directory Security 1-28
Effective Rights 1-29

CHAPTER 2 GETTING STARTED 2-1
Logging In 2-2
Using HELP 2-4
Identifying Yourself 2-5
Displaying the Date and Time 2-6
Determining Your Access Rights 2-7
Changing Your Password 2-8
Playing NSNIPES 2-9
Logging Out 2-10
Using Commands on the Network 2-11
Mapping and Remapping Disk Drives 2-13
More Experimenting on the Network 2-14

CHAPTER 3	COMMUNICATING WITH OTHER USERS	3-1
	SENDING MESSAGES	3-2
	ELECTRONIC MAIL	3-4
	Creating and Sending a Memo	3-5
	THE ELECTRONIC MAILBOX	3-8
	TYPES OF MAIL	3-9
	MAIL ATTRIBUTES	3-11
	SEARCHING FOR MAIL	3-12
	SENDING MAIL TO GROUPS	3-15
	SCREEN DISPLAY AND COMMAND ENTRY	3-17
	ELECTRONIC MAIL SYSTEM COMMANDS	3-20
	The CHECK Command	3-21
	The CLOSE Command	3-22
	The DIRECTORY Command	3-23
	The EDIT Command	3-24
	The HELP Command	3-27
	The LIST Command	3-29
	The LIST MY MEMOS Command	3-30
	The LIST USERS AND GROUPS Command	3-31
	The OPEN Command	3-32
	The PUT Command	3-33
	The QUIT Command	3-34
	The READ Command	3-35
	The REMOVE Command	3-37
	The REMOVE MY MEMOS Command	3-38
	The SEND Command	3-39
	The VIEW Command	3-41
CHAPTER 4	MANAGING DIRECTORIES	4-1
	WORKING WITH DIRECTORIES	4-2
	NetWare Directory Structure	4-2
	Accessing Network Files	4-3
	The CHKVOL Command	4-5
	The ERASEDIR Command	4-6
	The FLAGDIR Command	4-8
	The GRANT Command	4-11
	The LISTDIR Command	4-14

CHAPTER 4 MANAGING DIRECTORIES
WORKING WITH DIRECTORIES
The MAKEDIR Command 4-15
The NPATH Command 4-17
The RENDIR Command 4-23
The REVOKE Command 4-25
The SHOWDIR Command 4-27
The TLIST Command 4-29
The UDIR Command 4-30
The VOLINFO Command 4-32
The DOS CHDIR Command 4-34
The DOS CHKDSK Command 4-35
The DOS DIR Command 4-39
The DOS MKDIR Command 4-41
The DOS PATH Command 4-42
The DOS RMDIR Command 4-44

CHAPTER 5 MANAGING FILES 5-1
The FLAG Command 5-2
The HIDEFILE Command 5-4
The LARCHIVE Command 5-6
The LRESTORE Command 5-15
The NARCHIVE Command 5-18
The NCOPY Command 5-26
The NRESTORE Command 5-29
The PURGE Command 5-32
The SALVAGE Command 5-33
The SHOWFILE Command 5-35
The DOS ATTRIB Command 5-36
The DOS COPY Command 5-37
The DOS DEL Command 5-42
The DOS FC (File Compare) Command 5-43
The DOS RECOVER Command 5-53
The DOS REN (Rename) Command 5-55
The DOS RESTORE Command 5-56

CHAPTER 6 USING THE SHARED DISKETTE DRIVE 6-1
Introduction 6-1
DISKETTE DRIVE COMMANDS 6-2
The REQUEST Command 6-2
The RELEASE Command 6-3
The FORMAT Command 6-4
The DISKCOPY (Copy Diskette) Command 6-6
The DISKCOMP (Compare Diskettes) Command 6-9
The DISKBOOT Command 6-10
THE SIMULATED SECOND DISKETTE DRIVE 6-11

CHAPTER 7 PRINTING 7-1
Overview 7-1
HOW PRINTING WORKS 7-2
Shared Printers 7-2
Local Printers 7-4
PRINTING COMMANDS 7-5
The PRINTER Command 7-5
The LOCAL Command 7-8
The NPRINT Command 7-12
The Q Command 7-16
The SPOOL Command 7-19
The ENDSPOOL Command 7-24
The DOS PRINT Command 7-26

CHAPTER 8 OTHER DOS COMMANDS 8-1
Overview 8-1
The BREAK Command 8-2
The CHKMEM Command 8-3
The CLS Command 8-4
The COMMAND Command 8-5
The CTTY Command 8-7
The CURSOR Command 8-8
The DATE Command 8-9
The EXIT Command 8-10
The FIND Command 8-11
The MORE Command 8-13

CHAPTER 8 OTHER DOS COMMANDS
The PROMPT Command 8-14
The SET Command 8-16
The SORT Command 8-17
The TIME Command 8-19
The TYPE Command 8-20
The VER command 8-21
The VERIFY Command 8-22
The VOL Command 8-23

CHAPTER 9 USING THE SETLOGIN COMMAND 9-1
A Sample Login Script 9-2
The SETLOGIN Command 9-4
SETLOGIN's NPATH 9-5
SETLOGIN's COMSPEC 9-6
SETLOGIN's INCLUDE 9-8
SETLOGIN's DRIVE 9-9
SETLOGIN's WRITE 9-10
Using Conditionals 9-13
Login Script Parameters 9-16
#--External Program Execution 9-18
SETLOGIN's EXIT 9-19
SETLOGIN's BREAK 9-20
Displaying Files 9-21
Inserting Explanatory Text 9-22
Firing Phasers 9-23

CHAPTER 10 CREATING AND USING BATCH FILES 10-1
BATCH PROCESSING 10-2
The AUTOEXEC.BAT File 10-4
THE AUTOEXEC.BAT FILE 10-5
How To Create An AUTOEXEC.BAT File 10-5
Replaceable Parameters 10-7
Executing A .BAT File 10-9
BATCH FILE COMMANDS 10-10
Introduction 10-10
ECHO 10-11

CHAPTER 10 CREATING AND USING BATCH FILES
BATCH FILE COMANDS
FOR 10-12
GOTO 10-14
IF and THEN 10-15
PAUSE 10-16
SHIFT 10-18
REM (Remark) 10-19

CHAPTER 11 THE DOS LINE EDITOR (EDLIN) 11-1
EDLIN FILES 11-2
HOW TO START EDLIN 11-3
SPECIAL EDITING KEYS 11-5
[F1] Key 11-6
[F2] Key 11-7
[F3] Key 11-8
[Del] Key 11-9
[F4] Key 11-10
[Esc] Key 11-11
[Ins] Key 11-12
[F5] KEY 11-14
USING EDLIN COMMANDS 11-15
EDLIN COMMANDS 11-20
(A)ppend Command 11-20
(C)opy Command 11-21
(D)elete Command 11-24
<line> Edit Commands 11-27
(E)nd Command 11-29
(I)nsert Command 11-30
(L)ist Command 11-34
(M)ove Command 11-37
(P)age Command 11-38
(Q)uit Command 11-39
(R)eplace Command 11-40
(S)earch Command 11-44
(T)ransfer Command 11-48
(W)rite Command 11-49
EDLIN ERROR MESSAGES 11-50

CHAPTER 12 **PREPARING DOS PROGRAMS** 12-1
THE LINKER PROGRAM (LINK) 12-2
DEFINITIONS YOU'LL NEED TO KNOW 12-4
FILES THAT LINK USES 12-6
HOW TO START LINK 12-8
Prompts 12-9
Command Line 12-10
Response File 12-12
COMMAND CHARACTERS 12-14
COMMAND PROMPTS 12-16
LINK SWITCHES 12-19
SAMPLE LINK SESSION 12-23
LINK ERROR MESSAGES 12-26
THE EXE2BIN COMMAND 12-29
Error Messages 12-31

CHAPTER 13 **THE DOS DEBUG COMMAND** 13-1
HOW TO START DEBUG 13-2
COMMAND INFORMATION 13-4
Parameters 13-6
DEBUG COMMANDS 13-10
(A)ssemble Command 13-10
(C)ompare Command 13-13
(D)ump Command 13-14
(E)nter Command 13-16
(F)ill Command 13-18
(G)O Command 13-19
(H)ex Command 13-21
(I)nput Command 13-22
(L)oad Command 13-23
(M)ove Command 13-25
(N)ame Command 13-26
(O)utput Command 13-29
(Q)uit Command 13-30
(R)egister Command 13-31
(S)earch Command 13-35

CHAPTER 13 THE DOS DEBUG COMMAND
DEBUG COMMANDS
(T)race Command 13-36
(U)nassemble Command 13-38
(W)rite Command 13-40
Error Messages 13-43

APPENDIX A GLOSSARY A-1

APPENDIX B DOS DISK AND DEVICE ERRORS B-1

APPENDIX C ANSI ESCAPE SEQUENCES C-1

APPENDIX D HOW TO CONFIGURE YOUR SYSTEM D-1

APPENDIX E COMMAND INDEX E-1

APPENDIX F USING AN EXTERNAL WORKSTATION ON THE NETWORK F-1
Booting a Workstation F-2
Using DOS Commands on the Network F-8

INDEX Index-1

Overview

The North Star NetWare User's Manual contains information necessary to operate a workstation on the network.

For brevity and ease of use, unnecessary technical information such as hardware installation, supervisor instructions, and console commands have been omitted from this manual. These topics are covered in a separate Supervisor's Manual which is supplied with the system.

Only one User's Manual comes with the system. You may wish to purchase additional copies for use at each workstation.

This manual is organized as follows:

- The Preface contains symbols and conventions used in the text and with commands you give to the computer
- Chapter 1 provides an introduction to the computer system and presents the concepts necessary to begin understanding the NetWare multiuser operating system
- Chapters 2 through 8 are necessary reading for all users
- Chapters 9 through 13 present more complicated commands and programs intended for advanced users
- Appendices A through F cover various technical subjects and will serve as reference guides for error messages and commands

Introduction In order to avoid confusion, we have, as far as possible, used the same symbols and formatting techniques used to represent command formats in existing PC/MS-DOS manuals; however, some command format conventions are specific to NetWare, so read the following sections carefully.

CAPITALS Words in CAPITAL letters are keywords which must be entered (spelled) exactly as they are shown. Although keywords are shown in capital letters, they may be typed in either upper or lower case. Keywords are sometimes optional; optional keywords are always enclosed in square brackets [ ].

< > Words enclosed in angle brackets indicate variable data you must enter (i.e., user-supplied information which changes depending upon the circumstances). E.g., <filename>.

[ ] Square brackets indicate that the enclosed entry is optional. Do not type the brackets; type only the keyword or the information represented by the variable.

Square brackets also enclose the names of typed keys such as [Retrn] or [F1].

If you find square brackets nested in square brackets:

1) You may use the part of the command format which is enclosed only by the outside set of brackets, without using the part of the command format which is enclosed by the inside set of brackets.

2) You may not use the part of the command format enclosed by the inside set of brackets unless you also use the part of the command format which is enclosed by the outside set of brackets.

| | A bar is used to separate entry choices in NetWare commands. When you find a bar separating items in a command syntax, you may enter any one item, but not more than one item. When used with a DOS filter, the bar indicates a pipe in which the output of one DOS command is sent as the input to another command.

... Ellipses mean you may repeat an item as many times as you wish.

Boldface Boldface is used to indicate entries you type in to the system.

All punctuation and other characters (except [ ], ( ), | , and ...) must be entered exactly as shown.

Command Variables The term <drive:> represents any drive letter followed by a colon. For example if you see the term <drive:> you might enter C:, D:, H:, or any other drive letter with a colon.

The term <drive> represents any drive letter (a following colon is excluded). If you see the term <drive> you might enter C, D, H, or any other drive letter.

The term /<flag list> refers to any number of flags which may be entered in one list as part of a command. Using a flag list allows you to specify optional performance features of a command. Available flags vary from command to command. Flags entered in a flag list must be separated. The method of separation varies from command to command. Some commands require slashes between flags, some commands require spaces between flags, and some commands will accept either type of separator, so read the flag instructions for each command carefully.

Command Variables (cont.)

The term <filename> represents the name of any file, including filename and extension. A filename can be from one to eight characters long. The filename extension can be three or fewer characters, but must be preceded by a period. You can type any filename in small or capital letters and NetWare will translate these letters into uppercase characters.

The term filespec represents up to eight characters of any filename; the term filespec excludes an extension.

The term <pathspec> is used to represent any path specification. This term may be used to indicate either a full path specification or a partial path specification.

A full path specification includes the name of the volume's root directory and all directories branching down to (and including) the subdirectory you wish to identify.

A partial path specification is a shortened version used to indicate a subdirectory relative to your current pathspec. Examples of partial pathspecs are given later in this section.

The terms path/filename and path/filespec refer to the combination of the pathspec term (above) and the filename or the filespec term (above).

Examples:

SYS:INVENTORY/WIDGET.ARM (path/filename)
SYS:INVENTORY/WIDGET (path/filespec)

Introduction The following rules and restrictions apply to specific parts of NetWare commands.

Pathspecs A pathspec must not exceed 63 characters.

You should follow DOS 3.1 conventions regarding periods or slashes at the beginning of a pathspec.

In some commands, a partial pathspec may be specified. For example, if you usually work on only one volume (so that it is always your default volume), you would never really need to include volume when you specify a pathspec.

A partial pathspec may be used to identify a subdirectory above or beneath your default (current) directory.

A period (.) at the beginning of a pathspec means that the pathspec is relative to the current directory. Two periods (..) means that the pathspec is relative to the parent directory, three periods (...) the grandparent directory, etc.

A backslash (\) at the beginning of a NetWare pathspec means that the pathspec is relative to the volume's root directory.

Pathspecs (cont.)

Partial pathspec examples:

If your default directory is SYS:ACCOUNTING\PAYROLL, entering "HOURLY\JULY" as a partial pathspec in most commands will produce the same results as entering "SYS:ACCOUNT\PAYROLL\HOURLY\JULY".

Similarly, in relation to the directory SYS:ACCOUNTING\SALES, entering "..\MARKETING" would identify the network directory SYS:ACCOUNTING\MARKETING as well as if you had entered "SYS:ACCOUNTING\MARKETING".

Internal and External Commands

There are two types of commands built into the NetWare operating system.

Internal commands are the easiest to use. You cannot see these commands when you do a directory listing on your drive; they are part of the command processor, COMMAND.COM, and are always available to you no matter what drive you are on.

External commands reside on drives as program files. They are read from a drive before they run. If an external command is not present on your current drive, or you do not have a search path to the drive where the command resides, you must type the letter of the drive that contains the command before the command name in order to execute it. You could also establish a search path to the required drive, or you could make a copy of the command file on the current drive.

NetWare commands are all external commands. DOS contains both internal and external commands. The relationship between NetWare and DOS is explained in Chapter 1.

Filenames

The NetWare definition of a filename is the same as the DOS definition of a filename. All DOS 3.1 filename restrictions apply to filenames created on a NetWare network. NetWare filenames are limited to eight numbers or characters followed by a period (.) delimiter and an additional three number or character extension.

The following characters cannot be used in a NetWare filename:

- o backslash (\)
- o slash (/)
- o colon (:)
- o semicolon (;)
- o asterisk (*)
- o question mark (?)
- o tilde (~)
- o comma (,)
- o left bracket ([)
- o right bracket (])
- o period (.) (except as delimiter)

You may not use wildcard characters (*.* or ?????.???) as part of a <filename> unless the accompanying instructions specifically state that wild card characters are allowed.

<filename> examples:

MYFILE.SHX
123FIL.456
YOURFILE (filename which has no extension)

Directory Names

NetWare allows directory names used in pathspecs to be up to 14 characters long.

DOS limits directory names used with DOS commands to eight characters and a three character extension. NetWare allows you to continue using 14-character directory names, even with DOS commands. In order to use 14-character directory names with DOS commands you must know the following:

- To make any DOS command work with a directory name over eight characters long, you must provide the full directory name when entering the DOS command (you may not provide only the first eight characters).

- When using directory names over eight characters long, some DOS functions, such as DIR, will only echo the first eight characters of the directory name to the workstation screen, even though you actually enter all of the characters.

A directory name cannot include the following characters:

- backslash (\)
- slash (/)
- colon (:)
- semicolon (;)
- asterisk (*)
- question mark (?)
- comma (,)
- caret (^)
- tilde (~)
- percent sign (%)
- left bracket ([)
- right bracket (])

Sample Command

The command format for the NPRINT command is:

NPRINT [<path>\]<filename> [/<flag list>][Retrn]

According to the conventions, the only essential parts of this command format are the keyword (the command name NPRINT), and the filename.

The pathspec and /flag list are optional since they are enclosed in square brackets. Therefore,

NPRINT REPORT.A1[Retrn]

would be, in some cases, a valid command to print a file in the default directory named REPORT.A1.

You can see that if a pathspec is included, it must be followed by a backslash, and it must immediately precede the filename, with no space between. For example:

NPRINT VOL1:SALES\REPORT.A1[Retrn]

If a /flag list is included, it must be placed after the filename. The filename and /flag list must be separated by a space. Here is an example, using the flag NB (for "No Banner"):

NPRINT REPORT.A1 /NB[Retrn]

Command Entry

You must press the [Retrn] key after typing a command in order to execute the command. The [Enter] key will not execute commands on North Star workstation keyboards.

INTRODUCTION TO THE SYSTEM

Overview

Welcome to North Star NetWare, the IBM PC-DOS/MS-DOS compatible network from North Star Computers, Inc. This manual introduces the system and helps you make the best use of its features.

This manual refers to a person in your organization called the **system supervisor**, who is the person responsible for installing and maintaining the system. There may be more than one person who performs this role.

You'll read many new terms in this manual. Most terms are defined the first time they're used. A glossary is provided in Appendix A. You may want to take a few minutes to review the glossary before you continue reading. The subjects discussed in this manual are:

- The parts that make up the computer system
- The operating system software
- Using your workstation
- Sending and receiving Electronic Mail
- Organizing and managing your files
- Using printers and other peripheral devices
- Advanced operating system features: Batch files, the DOS line editor, and the LINK, EXE2BIN and DEBUG commands

Other Documents

These other documents describe related subjects:

- The **North Star NetWare Supervisor's Manual** contains the information that the supervisor needs to install and maintain the system, including complete descriptions of the NetWare and DOS commands that are restricted to the supervisor's use only.

- The **DIMENSION Technical Manual** contains detailed specifications and theories of operation for all the major hardware sub-assemblies comprising the DIMENSION computer, including schematics. This manual is intended for bench technician reference.

- The **DIMENSION Service Manual** contains information for field service technicians, including diagnostic programs and sub-assembly removal and replacement procedures.

Multi-User System

North Star NetWare is a **multi-user** operating system. Each person uses one or more workstations connected to one central computer, or **central module.** The central module stores the software and users' files and controls the components, which include hard disks, a diskette drive, printers, and options such as a tape backup unit and additional memory.

Components

A **hard disk** (or fixed disk) is a high-capacity file storage device in the central module. Every system has one hard disk. It stores the users' files and the software that manages the system. Some systems have a second hard disk which stores additional files.

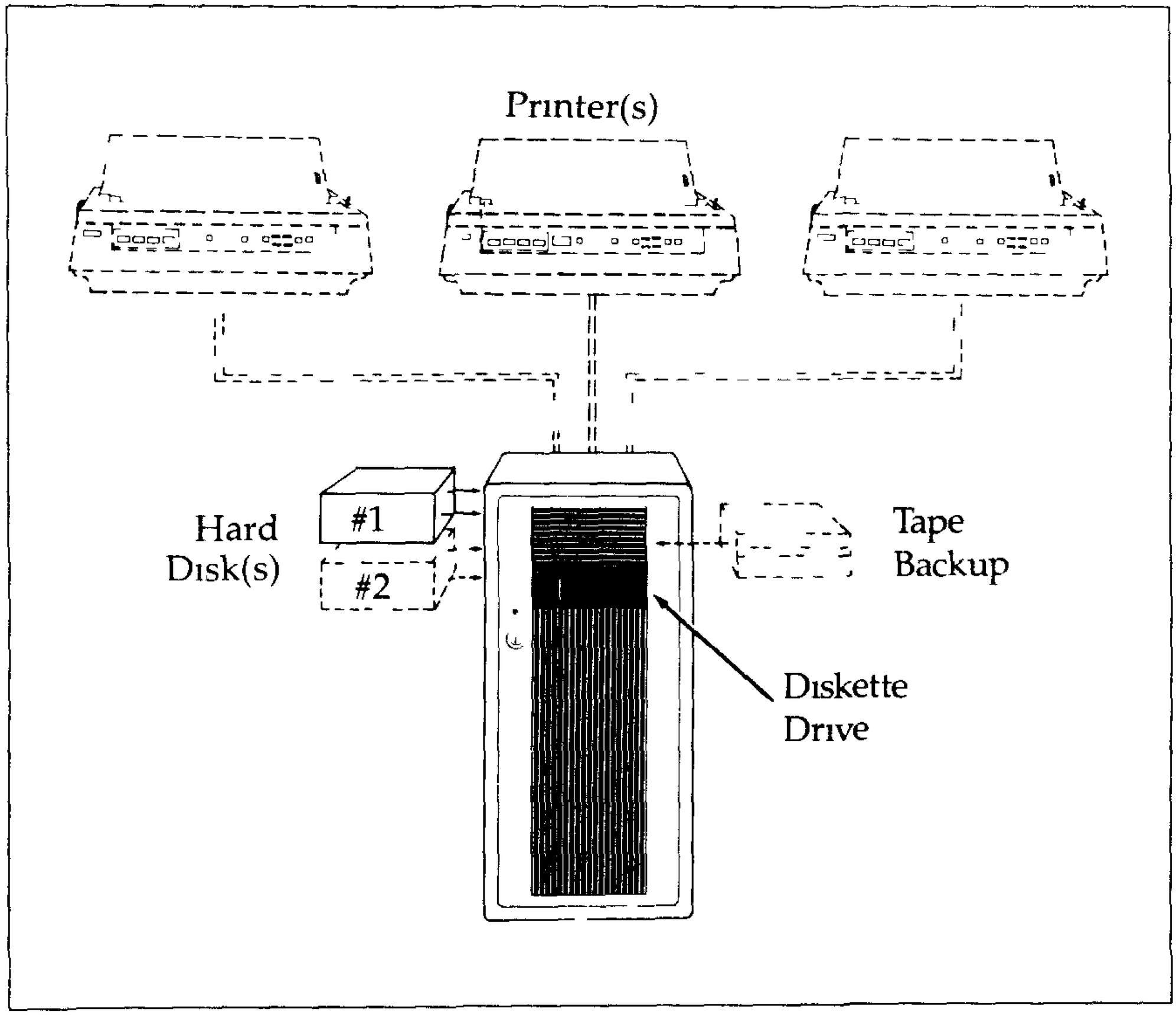
Every central module contains a **diskette drive** that is used to copy files between the hard disks and diskettes. You can use the same kind of soft sectored diskettes that you would use with an IBM PC or XT computer system.

The central module can have up to three **shared printers** connected to it.

A **network interface board** can be installed in the central module. When IBM PC (or XT) or compatible computers are attached to a network interface board they can share files stored on the central module's hard disk(s) and can use the shared printers.

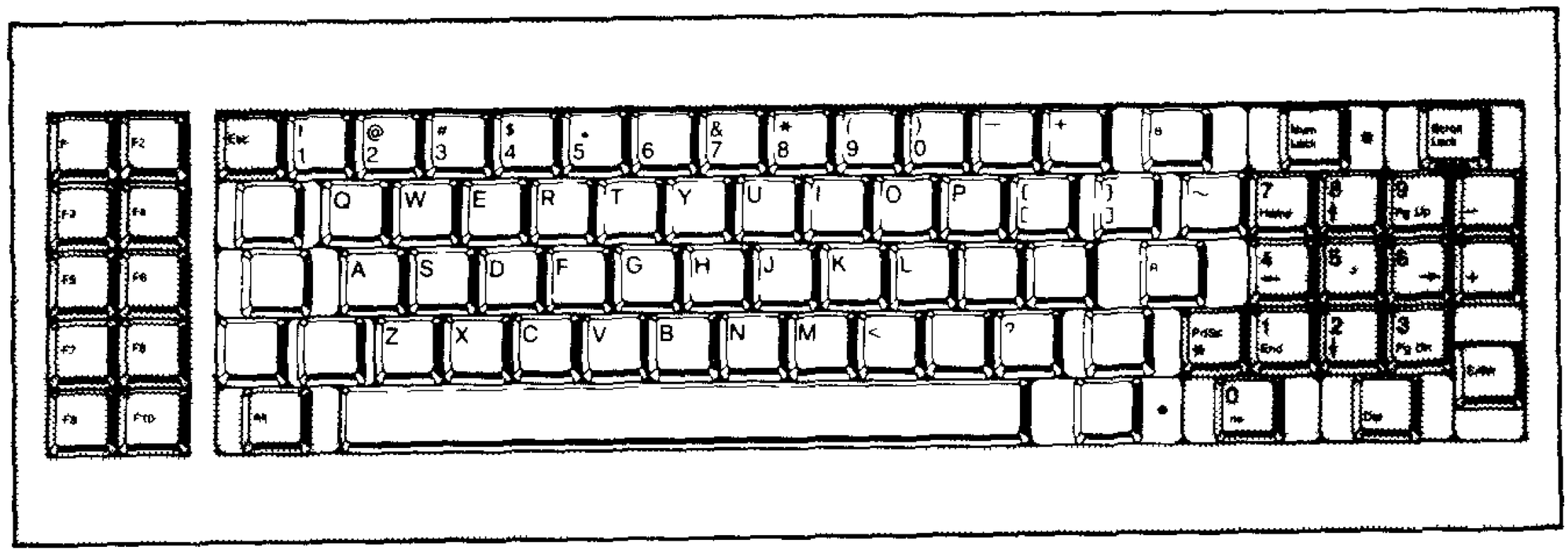
Central Module and Components

This illustration shows the central module and its components. Optional components are shown with dotted lines.

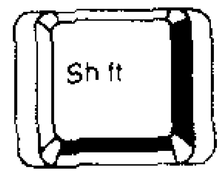


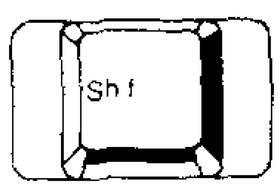
Introduction Each person uses a **workstation** connected to the central module by a cable. A system contains from two to twelve North Star workstations, and also can have external PC workstations connected through additional hardware and cabling. A North Star workstation is a series of components which are described on the following pages.

Introduction Some of the "typewriter" keys work differently than their counterparts on a standard typewriter.

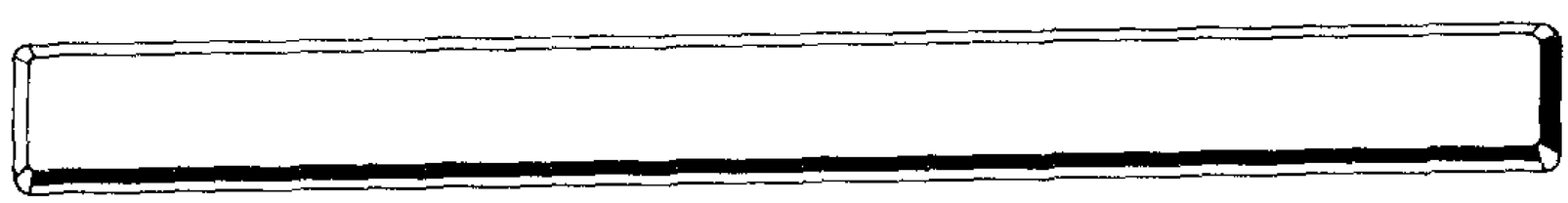


Descriptions

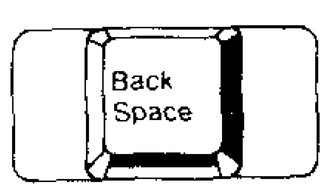




There are two shift keys. Neither key locks into place.

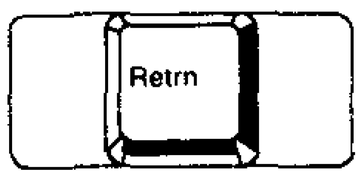


The space bar moves the cursor one character to the right, substituting a blank for any character already in that spot.

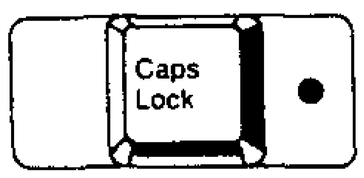


The backspace key moves the cursor one character to the left, removing any character already in that spot.

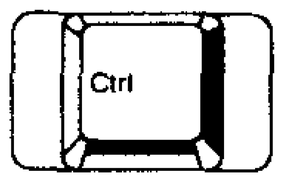
Descriptions
(cont.)



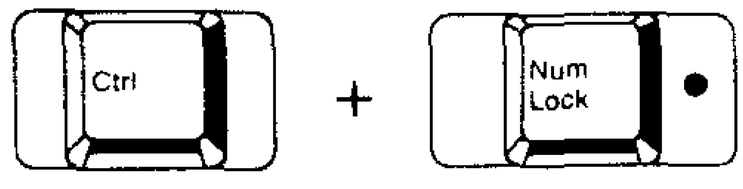
This key has three purposes: it signals the end of a typed entry and moves the cursor to the next entry; it signals the program to start executing the command you typed; or it ends a line of text and moves the cursor to the beginning of the next line.



Pressing this key so that its light comes on causes the letter keys to type in upper case. Pressing it again turns it off. When the [Caps Lock] key is on, the shift keys produce lower case letters.



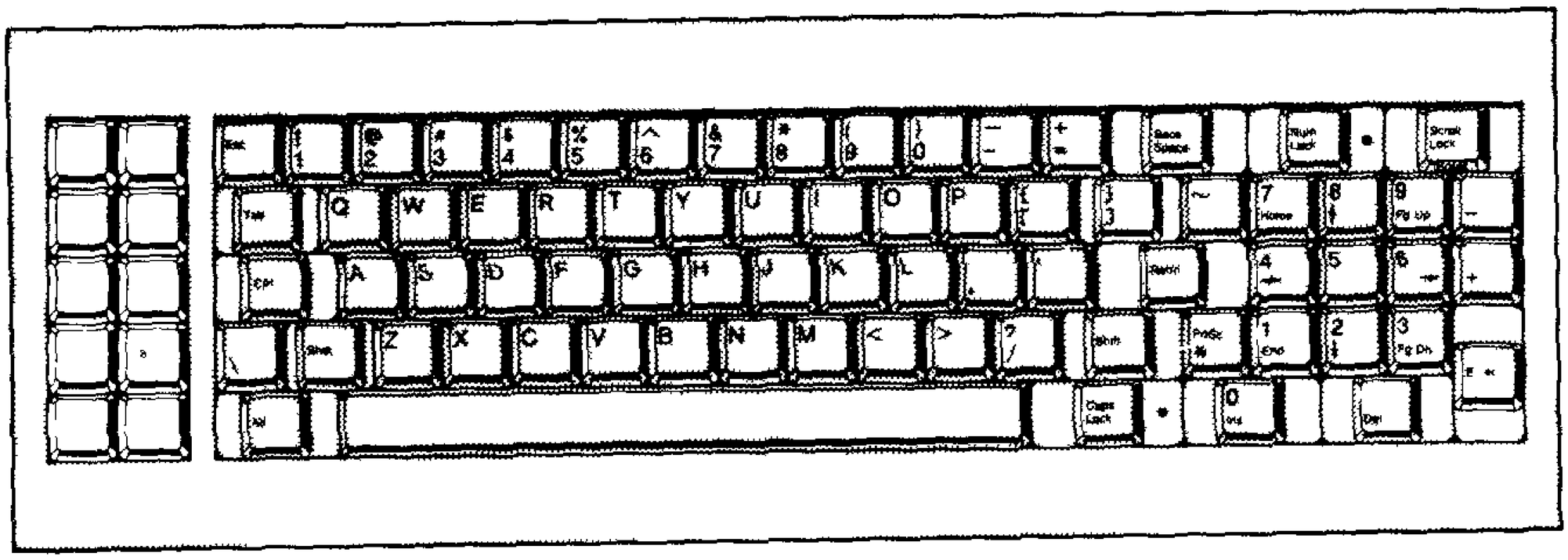
This key is used in combination with other keys to perform various functions with system programs. For example:



temporarily stops the display when a program is running. Press any key to start the display again.

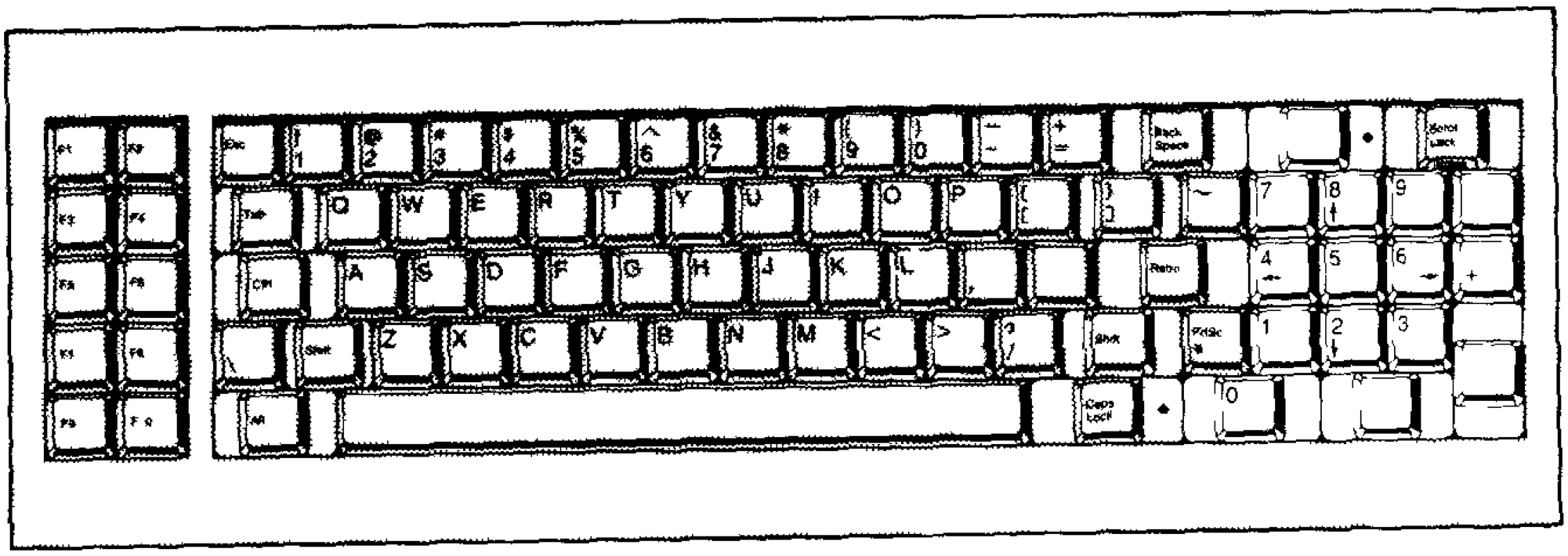
Function Keys

Generally, each program uses the function keys (not shaded in the illustration below) for different operations. Procedures in this manual explain how to use these keys with system programs. Manuals for other programs, such as applications, explain how the keys work with them.

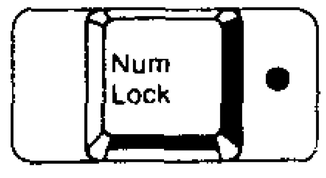


Numeric Keypad

The numeric keypad (not shaded below) can operate in two modes. The modes are explained on the following pages.

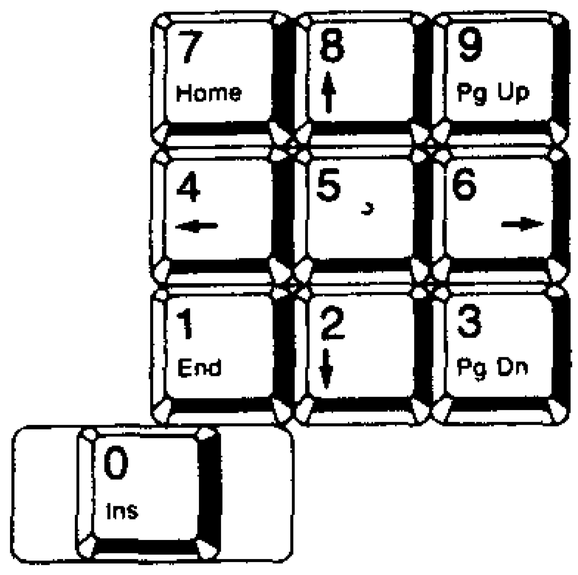


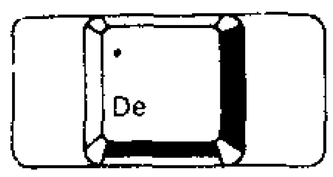
Calculator Mode



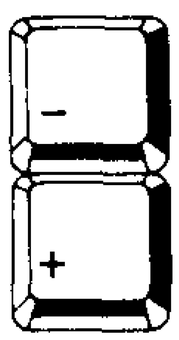
Pressing this key so that its light comes on causes the keypad to operate similarly to a calculator keypad. Then the keys work as described below.

These keys type numbers.

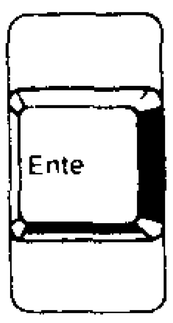




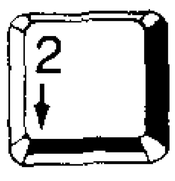
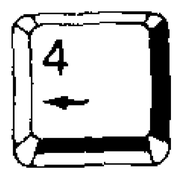
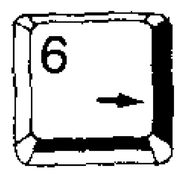
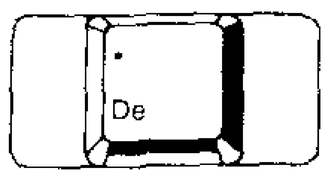
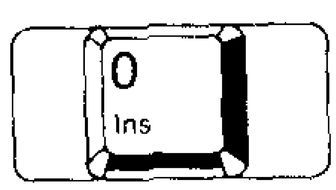
This key types the decimal point.



These keys type the minus and plus signs. In some programs, these keys also perform addition and subtraction.



This key works the same whether [Num Lock] is on or off. Like the [Retrn] key, it signals the end of a typed entry or line and moves the cursor to the next entry or line. With some programs you may use this key to execute a typed command.

Cursor Control		Pressing [Num Lock] so that the light goes off causes the keypad to control the cursor instead of working like a calculator.
		This key moves the cursor up one line.
		This key moves the cursor down one line.
		This key moves the cursor one character to the left.
		This key moves the cursor one character to the right.
		Press the [Home] key and an arrow key in combination to move the cursor in other ways. For example, press [Home] then the [ ] key to move the cursor to the last character at the right side of the line.
		This key deletes the character at the cursor.
		Pressing this key once allows you to insert characters where the cursor currently resides, moving all subsequent characters to the right. Press it again to turn insert mode off.

Introduction

The system software, which is stored on hard disk #1 of the central module, has several parts that are explained below.

Operating System

The NetWare operating system consists of two parts. The first part runs in the central module. It turns the central module into a **file server**. The second part of the NetWare operating system is a **shell** which runs in each workstation's memory, together with a North Star-modified version of Microsoft DOS release 3.10.

The NetWare shell intercepts requests that applications make to DOS. Requests that involve files and shared printers are communicated to the file server by the NetWare shell. When the file server responds to requests, the NetWare shell communicates the information to applications. Requests that don't involve files or shared printers are handled by DOS without intervention from the NetWare shell.

This manual explains the NetWare operating system, except for those commands which may be used only by the system supervisor to install and maintain the system.

Applications

An **application** is a program that performs a specific task, such as word processing or accounts payable. The applications for your system are selected by your organization. Your system supervisor will install them on the hard disk(s) in the central module and give you instruction in accessing and using them.

Electronic Mail

The system includes an **electronic mail** program that contains a simple command to send short messages, and a sophisticated mail system to send longer files to both individuals and groups of users on your network.

Introduction

Your computer contains one or two hard disks which are used to store the information created by all of the system's users. The NetWare operating system keeps track of information on the computer's hard disk(s) by organizing it into **volumes, directories, subdirectories** and **files.**

Volumes

A volume is some part of a hard disk that is used to store directories and files. A hard disk may contain as many as sixteen volumes. However, unless your organization has some special needs you will probably use one volume per hard disk. The number of volumes is controlled by the system supervisor.

If you are familiar with the MS-DOS concept of partitions, it may help you to think of a NetWare volume as a very large partition which all users can share at the same time.

Directories

A directory is used to store files and subdirectories that are related to each other. For example, you might store all of the subdirectories containing letters to your customers in a particular directory.

Sub-Directories

Subdirectories hold collections of files which are somehow related to each other. A subdirectory might be used to store all of the correspondence related to a particular customer.

Files

A file is a collection of related data. Files can be compared to file folders. One file, for instance, might contain all of your customer's names and addresses. Another file might contain a letter to a particular customer.

Organization Volumes, directories, subdirectories and files allow you to use your hard disk storage space in a very efficient manner. Your system supervisor may have organized your information by:

- o Creating a separate directory for each user in which he or she can store information that nobody else needs to access

- o Or, by creating directories to store information that groups of users will share. Shared directories might be created for the following functions:

  - Accounting
  - Customers
  - Applications

Figure 1-1 on the next page shows how files and directories might be used to organize a company's information.

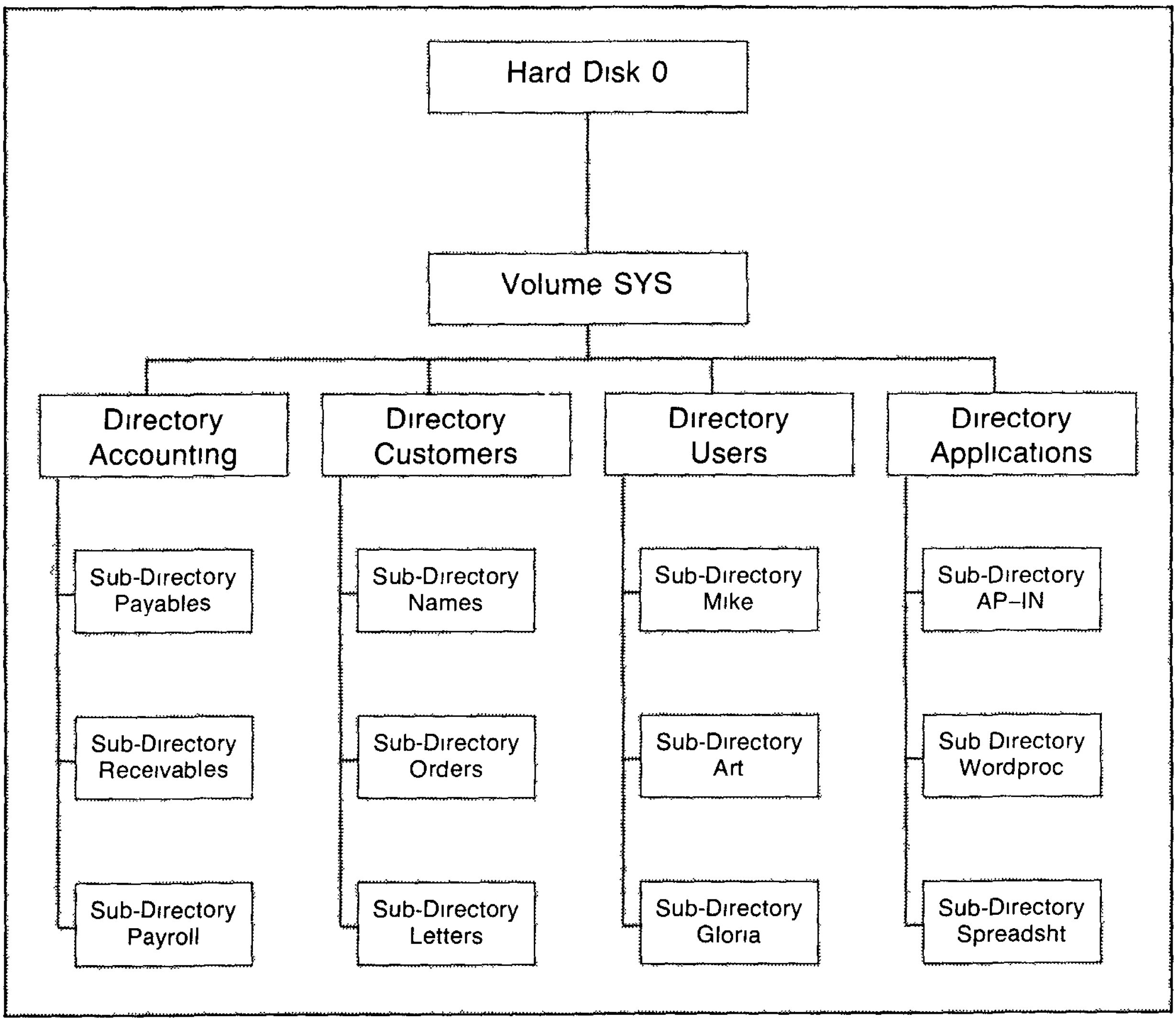


Figure 1-1
Sample Data Storage Layout

Discussion You can see from figure 1-1 that the hard disk contains only one volume. The volume is divided into four directories. Each of the four directories is further divided into three subdirectories. The twelve subdirectories would be used to store this company's files.

Names

Every volume, directory, subdirectory and file needs to be identified by a name. You must follow some rules when you select names.

Volume Names

A volume name consists of two to fifteen letters and/or numbers. The first character of a volume name may not be a number. Volume names are selected by the system supervisor.

Directory and Sub-Directory Names

A directory or subdirectory name consists of one to fourteen characters. Any printable character may be used in directory names, except for the following characters:

\ / : ; * ? , ^ ~ % [ ]

File Names

A filename has two parts:

- o A name of one to eight characters
- o An optional extension of one to three characters

Any printable character can be used for filenames and extensions, except for the following characters:

\ ? : ; * ? ~ , [ ] .

If you use an extension, the two parts of the filename must be separated by a period, as in <filename>.<ext>.

File Names (cont.)

The following examples illustrate correctly formed file names:

PAYROLL.DAT
ACCTRECV.FIL
ACCTPAY.BAK
SAMPLE

The following examples illustrate incorrectly formed file names:

PAY:ROLL.DAT	use of the colon isn't allowed
ACCTRE*.FIL	use of the asterisk isn't allowed
.DAT	the name cannot be omitted

Accessing Information

So far, you've learned that information is stored in files and that files are grouped together and stored in directories. In order to access information, you need to tell NetWare where it is located. You use a **pathspec** to tell NetWare where information is stored. A pathspec combines a volume and one or more directory names into a pointer to the place where information is stored.

The table on the following page shows how the volume, directory and subdirectory names defined in the previous diagram are combined to form pathspecs. Notice that the volume name is followed by a colon (:) and that the other parts of the pathspecs are separated by backslashes (\).

Pathspecs

```
SYS:ACCOUNTING\PAYABLES
SYS:ACCOUNTING\RECEIVABLES
SYS:ACCOUNTING\PAYROLL
SYS:CUSTOMERS\NAMES
SYS:CUSTOMERS\ORDERS
SYS:CUSTOMERS\LETTERS
SYS:USERS\MIKE
SYS:USERS\ART
SYS:USERS\GLORIA
SYS:APPLICATIONS\AP-IN
SYS:APPLICATIONS\WORDPROC
SYS:APPLICATIONS\SPREADSHEET
```

Table 1
Examples of Pathspecs

Drive Letters

Pathspecs can be quite lengthy--up to 63 characters. Drive letters provide a simpler method of accessing information.

The drive letters A through Z are used as "nicknames" for pathnames. Some examples of how drive letters are used are:

A:=SYS:ACCOUNTING\PAYROLL
B:=SYS:ACCOUNTING\RECEIVABLES

Using the above examples, a file named REPORT.DOC could be accessed in two ways:

- By preceeding the file name with the full pathspec:

  SYS:ACCOUNTING\PAYROLL\REPORT.DOC

- By using drive letter A to represent the pathspec:

  A:REPORT.DOC

It's easy to see how much simpler drive letters make file access. You will learn how to assign pathspecs to drive letters in Chapter 4.

Logged Drive

Whenever you are logged into the system, a prompt is displayed on the screen. The prompt shows which drive is being used (called the **logged drive**) by displaying its letter followed by a right angle bracket. For example, when you first log in, the prompt "C>" is displayed on the screen.

Logged Drive (cont.)

You can change from one logged drive to another by typing the desired drive letter followed by a colon and then pressing the [Retrn] key. For example, to change the logged drive from C to D you would type:

D:[Retrn]

When you type a command or file name without supplying a drive letter, NetWare looks for the file on the logged drive.

Search Drives

A search drive is a drive that NetWare uses to look for files when they aren't found in the logged drive (or in the drive that you indicated).

SYS:PUBLIC is one example of a directory that is assigned to a search drive. All of the NetWare and DOS command files are stored in SYS:PUBLIC. When you type a command, such as HELP, NetWare finds its files in drive Z even if you aren't logged to drive Z.

Search drives are set using the NPATH command as discussed in Chapter 4.

Introduction One of NetWare's most attractive features is its security system. Four levels of security are provided:

- **Login/password security** allows the supervisor to assign each user a password which must be entered with his or her username before he or she can gain access to the computer system.

- **Trustee security** allows the supervisor to specify which directories and/or subdirectories individual users and groups of users may access.

- **Directory security** allows the supervisor to control access to directories and subdirectories for all users.

- **File attributes security** provides for controlling access to individual files within directories and/or subdirectories.

These security features are described fully in the following pages.

Introduction When you want to use a workstation, you perform a **login** procedure by entering your username and optional password. Your system supervisor will provide these for you.

Usernames Every username is unique. Usernames consist of one to fifteen printable characters.

The following are examples of usernames:

GLORIA
MIKE
ART
JAMIE
PETE1
PETE2

Passwords Passwords provide a way to keep unauthorized individuals from accessing your information. Since passwords are optional, your system supervisor can assign them to all users, some users or no users. Passwords consist of one to fifteen printable characters.

The following are examples of passwords:

SECRET
MY*STARS
YOUR$CHOICE

Introduction

When your system supervisor defined the system's users, he or she assigned each user **rights** to one or more directories. He also may have created **groups** of users and assigned rights to groups. A user who is granted rights to a directory is called a **trustee** of that directory.

The combination of a user's individual rights and his or her group rights are called his or her **trustee rights.**

Types of Rights

There are eight types of rights:

- o **R** - Read
- o **W** - Write
- o **O** - Open
- o **C** - Create
- o **D** - Delete
- o **P** - Personal Ownership
- o **S** - Search Directory
- o **M**- Modify File Attribute Flags

Each of these eight trustee rights are described below.

Read

Read rights allow a trustee to read data from an open file. Without this right, a trustee may not display data on a workstation screen or send it to a printer. Thus, the trustee cannot see what is in the file.

Write

Write rights allow a trustee to write data into an open file. Without this right a trustee may not change the contents of an existing file.

Open

Open rights allow a trustee to open existing files. Without this right, a trustee may not read or write to an existing file. It is important to realize that this is not the right which allows creation of new files.

Create

Create rights allow a trustee to create a new file, opening the new file simultaneously. The trustee cannot write to the new file without write privileges.

For example, a trustee with only Create and Write privileges may create a new file, place data in it, and close the file. Once the file is closed, the trustee cannot reopen, read or edit the file since he or she does not have Open and Read rights. This is the method that is used to provide security for users' mailboxes.

The Create right also allows a trustee to create a file with the name of an existing file if he or she has delete rights. Delete rights are required so that the old file can be deleted before the new file is created.

Delete

Delete rights allow a trustee to delete a file by deleting the file's directory entry. No other rights are needed to delete files.

Personal Ownership

Personal ownership rights allow a trustee to grant or revoke trustee and directory rights in a directory and subdirectories. A trustee cannot grant or revoke rights in directories above the directory in which he or she has personal ownership rights.

Trustees with personal ownership of a directory may regrant themselves rights in subdirectories in which their rights have been revoked.

Trustees with ownership of a directory may create or delete subdirectories.

Search

Search rights allow a trustee to search through a directory for files. Without search rights a trustee cannot see the names of the files that are stored in a directory.

Modify File Attribute Flags

Modify File Attribute Flag Rights allow a trustee to change the attributes of files. The file attributes that may be changed are Read/Write, Read-Only, Non-Shareable and Shareable. A user may only rename a file if he or she has Modify rights and the file is Read/Write.

Combinations

Rights normally need to be granted in combination before a trustee can do anything useful within a directory. For example, if a user is to be allowed to use the applications stored in a directory, the minimum set of rights that he or she must be granted is Read, Open, Search. If a user is to be allowed to create files, the minimum set of rights that he or she must be granted is Create, Open, Write. Table 2 (on the next page) shows some typical rights configurations and tells what they allow users to do.

Extended Rights

A trustee's rights to a directory extend down to all subdirectories under that directory. For example, if a user has Read, Open and Search rights to the directory path SYS:ACCOUNTING, then he or she can open and read files in the paths SYS:ACCOUNTING\PAYABLES, SYS:ACCOUNTING\RECEIVABLES and SYS:ACCOUNTING\PAYROLL.

These rights...	Allow a user to...
S	List the files stored in a directory
ROS	List the files stored in a directory; open files and read their contents; run application programs
WC	Create new files and write to them
ROWS	List the files stored in a directory; open files; read and/or update their contents; run application programs
ROWDS	List the files stored in a directory; open files; read and/or update their contents; delete files; run application programs
ROWCDS	List the files in a directory; open files; read and/or update their contents; create files; delete files; run application programs
P	Change his or her rights; grant and revoke rights to other users
ROWCDSM	List the files stored in a directory; open files; read and/or update their contents; create files; delete files; flag files as shareable and non-shareable; flag files as read-only and read/write; run application programs

Table 2
Example of Combined Rights

Introduction

The system supervisor can establish the same eight rights for directories as those that apply to users and groups. The difference is that directory access by all users (rather than by selected users) is affected by directory rights.

Open Access

Directories are usually created with full access rights. Unless special circumstances exist, the system supervisor shouldn't restrict directory rights.

Note: *Directory rights do not extend down to subdirectories.* If the rights of a directory are restricted, the restriction doesn't apply to its subdirectories.

Example

The system supervisor may want to restrict the rights of a directory after your system has been in use for a while. For example, if the supervisor finds that some files are being deleted repeatedly but can't determine who is responsible, he or she might set the directory's rights to Read, Open and Search which would allow users to:

- List the files stored in the directory
- Open files
- Read from files

but would not allow users to:

- Modify files
- Delete files
- Create files
- Grant or revoke rights to the directory and its subdirectories.

Introduction

Netware combines a user's trustee rights with directory rights to form the user's **effective rights.** It is a user's effective rights that determine how he or she may use a directory. NetWare determines a user's effective rights at each level of the NetWare directory structure.

Three Concepts

In order to understand the concept of effective rights you must first understand three other concepts:

- o Trustee rights (the combination of user and group rights) are completely reset at any directory level where they are redefined. Unless trustee rights are redefined, they carry down through all succeeding subdirectories.

- o Directory rights must be set in every separate directory and subdirectory. They do not carry down through succeeding subdirectories.

- o File security takes precedence over a user's effective rights. If a file is set to Read-Only, users can't modify the file's contents or delete the file.

Determining Rights

There are three steps that your system supervisor followed to determine each user's final rights to files:

- o First, he or she determined each user's trustee rights by combining (logically ORing) his or her user and group rights

- o Second, he or she determined each user's effective rights by subtracting (logically ANDing) any right that isn't contained in **BOTH** the user's effective rights and directory's rights.

Determining Rights (cont.)

- Then, he or she determined each user's final file rights by subtracting (logically ANDing) the user's right to "write" to the file if the file's attribute is set to "read-only"

Example

Figure 1-2 on the next page shows how the NetWare security system determines user's final file rights under various circumstances.

Logical OR with

Directory or Subdirectory	User Name	Users Directory Rights	Group One Rights	Group Two Rights
ACCOUNTING	Mike	ROSWCD	ROS	ROS
ACCOUNTING/PAYABLES	Mike	ROSWCD	ROS	ROS
ACCOUNTING/RECEIVABLES	Mike	ROS	ROS	ROS
ACCOUNTING/PAYROLL	Mike	ROS	ROS	ROSD
CUSTOMERS	Gloria	ROSWD	ROSWD	ROS
CUSTOMERS/NAMES	Gloria	ROSWD	ROSWD	ROS
CUSTOMERS/ORDERS	Gloria	ROS	ROS	ROS
CUSTOMERS/LETTERS	Gloria	ROS	ROS	ROSWC

User and Directory Rights:
R = Read
O = Open
S = Search
W = Write
C = Create
D = Delete

File Attributes.
R/W = Read/Write
R-O = Read-Only

Logical AND with

Logical AND with

Group Three Rights	=	Trustee Rights
ROS		ROSWCD
ROS		ROSWCD
ROSW		ROSW
ROSWC		ROSWCD
ROS		ROSWD
ROS		ROSWD
ROSW		ROSW
ROSD		ROSWCD

Directory's Rights	=	Effective Rights	File's Attribute	=	Final File Rights
ROSWC		ROSWC	R/W		ROWC
ROSWCD		ROSWCD	R-O		RO
ROSWC		ROSW	R/W		ROW
ROSWCD		ROSWCD	R/W		ROWCI
ROSWC		ROSW	R-W		ROW
ROSWCD		ROSWD	R/W		ROWD
ROSWC		ROSW	R-O		RO
ROSWCD		ROSWCD	R/W		ROWC

GETTING STARTED

Overview

Now that you've been introduced to NetWare's basic concepts, it's time to begin using the system. In this chapter you'll learn how to:

- o Login
- o Use the HELP program
- o Identify yourself using the WHOAMI command
- o Display the date and time with the SYSTIME command
- o Determine your access rights using the RIGHTS command
- o Change your password with the SETPASS command
- o Play NSNIPES, a video game
- o Logout

You'll also learn about mapping network drives and do some exercises to help you become familiar with commonly used commands.

Before you can begin, you must obtain the copy of worksheet #3 that your system supervisor prepared for you. It shows your user name and password.

North Star NetWare User's Manual 2-1

GETTING STARTED
Logging In

Start Up

The system central module must be turned on before your workstation will operate. If the power-on indicator at your video display is on but nothing appears on the screen, check with your system supervisor to be sure your workstation is properly installed and that the central module is operating.

Login

When your video display is turned on, you should see the login screen:

```
                    Workstation #___

     North Star NetWare Version 1.x.x

     Copyright 1985 North Star Computers, Inc.
     Copyright 1983, 1984, 1985 Novell, Inc.


                 Enter your user name:--------------
```

Notice the cursor flashing beneath the first dash in the row at the bottom of the screen. Type your user name and press [Retrn]. If you have been assigned a password, the system prompts you to enter it. Your password remains confidential because the letters do not appear on the screen. After typing your password, press [Retrn].

Login (cont.)

In response to your login (entering your user name and password), the system will display your drive assignments like this:

```
DRIVE C := SYS:YOURNAME
DRIVE Z := SYS:PUBLIC
-----
SEARCH1 := Z:. [SYS:PUBLIC]
C>
```

Your specific drive assignments may be different than those shown above. Compare the assignments shown on the screen with those listed on your User Worksheet. If you find a discrepancy, use the NPATH or SETLOGIN comands (described in Chapters 4 and 9) to make corrections.

If your login was not successful, try again. You may have typed your username or password incorrectly. If your difficulty persists, see your system supervisor for assistance.

NetWare Commands

Now that you're logged into NetWare, you can begin using the NetWare commands. The rest of this chapter explains some of the commands you need to get acquainted with the NetWare operating system.

HELP

NetWare contains its own information system that can assist you in learning about NetWare concepts and commands. Type:

HELP[Retrn]

The system will display the Help Index and provide instructions for using the help program. Use the cursor control keys to highlight the topic you wish to learn about. Press [Retrn] to view information on the selected topic.

When you are in the help program you can press [Esc] and ask your own questions. For example, typing:

[Esc]What is security?[Retrn]

will bring you an explanation of NetWare's security concepts.

Exit the help program by pressing the [Ctrl] and C keys simultaneously.

The fastest way to get specific help when you are running Netware (but are not sure about a NetWare command) is to type:

HELP <command name>[Retrn]

The system will display the help screen for that command. Press the [Ctrl] and C keys simultaneously to exit the help program and return to NetWare.

Note: DOS commands are not included in the help program. This manual contains instructions for using DOS commands.

WHOAMI

You can use the WHOAMI command to display your login information.

Command Format

WHOAMI [<flag>][Retrn]

<flag> is one of the following:

G (for Group) lists any network groups you belong to
R (for Rights) lists directories you have rights to
A (for All) lists both of the above

Remarks

This command will tell you your username, your workstation number, the date and time of your last login, and, if you specified a flag, the groups you belong to, the rights you have to directories, or both.

SYSTIME This command displays the current time and date.

Command Format SYSTIME[Retrn]

Remarks When you issue this command, the system responds with a display similar to the following:

Current System Time: Friday, May 24 1985 3:30 pm

RIGHTS

This command displays the effective rights you have in a particular directory. Use the pathspec information on your worksheet #3 to specify a directory and see how this command works.

Command Format

RIGHTS [<pathspec>][Retrn]

Remarks

If you do not supply a pathspec, NetWare will display the rights you have in the directory you are currently using.

If you were to ask for your rights in the SYS:MAIL directory (type RIGHTS SYS:MAIL[Retrn]), the system would respond with:

```
SYS:MAIL
Your Directory Rights are [ W C S ]:

You may Write to Files.          (W)
You may Create new Files.        (C)
You may Search the Directory.    (S)
```

SETPASS Your system supervisor assigned your login password when he or she initially configured the system software. You can change your password at any time.

Command Format SETPASS[Retrn]

Remarks When you type SETPASS[Retrn] the system asks you to enter your old password, enter your new password, and finally to retype your new password to confirm the entry. To protect its confidentiality, your password will never appear on the screen. If you've entered all the information correctly, you'll see:

Your password has been changed.

If your entry was incorrect, you will receive one of the following error messages:

Incorrect old password
New password not correctly retyped
Error: Control characters are not allowed in passwords

If you wish to retry, you must begin again by typing SETPASS[Retrn]

NSNIPES

NSNIPES is a network game that demonstrates the multi-user aspect of the system by allowing many users to play in the same maze. The object is to destroy the snipes (the faces with arrows), the factories that make the snipes (the boxes), and other players before you lose all five of your men.

Before beginning to play the game, all users wishing to play must log into the same directory area and have at least Read, Write, Open, and Create rights in the directory.

To begin the game, one user types:

NSNIPES <skill level>

<skill level> is from 1 to 10. Try using 4 to begin. The other players then enter:

NSNIPES[Retrn]

After each player has entered, the first player presses **[Retrn]** to start the game.

To move through the maze, use the **4**, **8**, **6**, and **5** keys on the cursor control keypad. Pressing two keys at a time will result in diagonal movement.

To fire, press the **A**, **W**, **S**, and **D** keys. Pressing two at once will result in diagonal shooting.

NSNIPES was written for monochrome video displays. If you are using a color monitor, you should run NCSNIPES instead of NSNIPES. All stations on the network can be in the same game even if some players are running NSNIPES and some NCSNIPES. (NCSNIPES begins with "flying cursors.")

You can play NSNIPES by yourself for practice. To leave the game, press **[Ctrl][Scroll Lock]** simultaneously.

LOGOUT This command allows you to exit the network system.

Command Format LOGOUT[Retrn]

Remarks You should always log out before turning off your workstation so that data files will be closed properly, and so that no one else can access your files. When you log out at a North Star workstation, the login screen will reappear. You can reenter your username and password whenever you're ready.

If you are using an external PC workstation, all network drive mappings are removed when you log out, except that the default drive is mapped to SYS:LOGIN so that you may log in without re-booting (re-starting) your workstation (refer to Appendix F if you are using this type of workstation).

If you're used to using a 3-key reset ([Ctrl][Alt][Del]) to exit from the operating system, you should be aware that in NetWare this action does not log you out, but instead performs a warm boot and reinitalizes your path connections, leaving you logged in with the username and password that were in effect before the 3-key reset.

Using DOS Commands

>> CAUTION <<

If you are already familiar with DOS commands, there are several DOS comands you must not use when you are using NetWare.

Most DOS 3.1 commands will work normally on network drives. However, there are some exceptions. In order to ensure optimum system performance, follow these important command guidelines:

o BACKUP and RESTORE

Don't use the PC/MS-DOS BACKUP or RESTORE commands to back up or restore files stored on network drives.

You must use the NetWare LARCHIVE/LRESTORE, and NARCHIVE/NRESTORE commands to back up and restore files stored on network drives.

If you are using an external PC workstation, you may use the PC/MS-DOS BACKUP and RESTORE commands to back up and restore files stored on DOS partitions of a local hard drive. Another valid use of BACKUP and RESTORE is to transfer extremely large files from a stand-alone machine with a hard disk to a network at a different location.

The RESTORE command supplied with NetWare may be used to restore files that were created under NS-DOS to NetWare volumes. This procedure is described in the North Star NetWare Supervisor's Manual.

Using DOS Commands (cont.)

- PATH -- Although NetWare does support the DOS PATH command, you should not use it on network drives. Use the NetWare NPATH command for setting search drives. NPATH will allow you to specify local drives as part of the search drive list (see Chapter 4). If you use PATH, each time you change it you will lose your network search drives.

- PRINT --To print any file on a shared network printer, you must use the NPRINT command.

  You may use PRINT to send local or network files to a local printer.

- CHKDSK (on network hard disk drives) -- Use the NetWare CHKVOL command instead to get hard disk information. You may use CHKDSK on diskettes and local hard disk drives.

- DIR -- The DOS DIR command will not work properly if a volume name other than the default volume is specified. Use the UDIR command instead.

- COPY -- The DOS COPY command may also work improperly if a volume name is specified. NCOPY should be used in these cases.

- CHDIR (CD) -- If CD is invoked from a local drive and a volume name is specified, the network shell will automatically change that local drive to a network drive.

Introduction

Mapping disk drives is a further way of organizing your data. You can use drive letter designations to connect you with your directories. Once you have logged into the network, you can use the NPATH command to map or remap any of your network drives (including those initially set up for you by your system supervisor).

You can assign as many as 26 network drives, if you go so far as to reassign letters assigned to local drives as network drives. (The number of assigned drives is limited to the number of upper-case letters in the alphabet.)

However, if you have only been given access rights to three or four directories, it is impractical to reassign local drives or to map out more drives than you can use.

You can switch back and forth to different drives by typing the drive letter and a colon at the DOS prompt. If you leave a drive then return to it, the drive will still be mapped to the last assigned directory.

NPATH

To display a full listing of your drive mappings, type NPATH[Retrn]. NetWare will respond with a list of your mapped network drives, including the directories the drives are mapped to. Your first network drive should be mapped to your home directory, and your first network search drive (drive Z) should be mapped to SYS:PUBLIC.

These original mappings are permanent until you remap them. If you do change them, the drives remain mapped to the directories to which you assign them until you log out of the network.

You may use SETLOGIN to edit your LOGIN script to include the drive mappings you want. In this way, frequently-used directories can be automatically mapped to drives when you log in. These commands and procedures are discussed further in Chapters 4 and 9.

Exercises Let's do a little experimenting to help you become more familiar with some basics.

NPATH Follow the directions below to try an example NPATH command. Type:

NPATH H:=SYS:SUPER[Retrn]

You've just assigned network drive H to the supervisor's directory (SUPER). Note that this doesn't do you much practical good, unless you have access rights to that directory, but it illustrates how drives are remapped with NPATH.

To display the new drive assignment, once again type:

NPATH[Retrn]

The screen display should now show that drive H is mapped to the SUPER directory.

Now let's check to see how NetWare's security works.

If drive H is not already your default, change to drive H by typing:

H:[Retrn]

Now, see if you can gain access to the SUPER directory. Try to get a listing of the files in that directory by typing:

UDIR[Retrn]

There are plenty of files in that directory, but unless the supervisor has given you Search rights in directory SYS:SUPER, you'll see a message that no files are found.

NCOPY, UDIR, and DELETE

Here's an exercise that will introduce you to some common commands. Change to the network Search drive (Z) by typing:

Z:[Retrn]

(Remember this drive is mapped to SYS:PUBLIC.)

Now use the UDIR command to list the files in this directory. Type:

UDIR[Retrn]

A listing of the files in the PUBLIC subdirectory will appear on your monitor screen.

Next, use the NCOPY command to move a file from drive Z to your home directory on drive C. To do this, pick a filename that is listed in the Z: directory (preferably a small file), then type:

NCOPY <filename> C:[Retrn]

NetWare will look through the drive Z directory, find the file you indicated, and copy the file into the directory of drive C.

To verify that the NCOPY command worked, change to drive C by typing:

C:[Retrn] then **UDIR[Retrn]**

The file you copied should appear in the directory listing for drive C.

Now you can erase the file from your personal drive by typing:

DEL <filename>[Retrn]

USERLIST

This command allows you to find out who else is logged in to the network by typing:

USERLIST[Retrn]

Your name, and the names of any other users who are logged in, will be displayed on your workstation screen.

SEND

If another user is logged in, you can send a brief message to the user's workstation. For example, if user GLORIA is logged in, you can type:

SEND "Want to go to lunch?" TO GLORIA[Retrn]

GLORIA will immediately get your message on her screen.

Communicating with other users is covered completely in Chapter 3.

NPRINT

If you want a file to be printed, type

NPRINT <filename>[Retrn]

The file will be queued to the first network printer.

The NPRINT Command in Chapter 7 will explain various options and flags you can use with NPRINT.

Summary

Now that you've experimented with a few commands you can see how simple it is to start using NetWare. The next chapter goes further into another NetWare feature, Electronic Mail, and then Chapter 4 discusses commands related to directory management. You will need to become familiar with the material in Chapters 3 through 8 in order to use the network proficiently.

COMMUNICATING WITH OTHER USERS

Overview

NetWare provides you with two methods to communicate with other users: a simple command to send short messages, and a sophisticated mail system to send memos, letters, documents and files to both individuals and groups of users on your network. This chapter, as well as the on-line HELP screens, will guide you through communicating with other users on your NetWare system.

Introduction

With NetWare, you can send a short message to another user on the system and the user will receive the message immediately. You may also send the same message to more than one user.

USERLIST

You can send a message to any other network user as long as the user is signed onto the system. Before you attempt to send your message, first find the names and workstations of users signed onto your network by typing:

USERLIST[Retrn]

How to Send

To send a message to another user, you type:

SEND "<message>" to [<user>|<station>][Retrn]

You must enclose your message in quotes. You may send up to one line of text in the message. If you want to send the message to more than one user, type each user's name or workstation number on the same line separated by a comma.

If you have sent your message correctly, it will immediately flash across the bottom of the user's screen.

Disabling Message Receipt

If you don't want to receive any messages from other users, type the command:

CASTOFF[Retrn]

CASTOFF disables message reception. The screen will confirm with a notice saying:

Broadcast messages from other stations will now be rejected.

Disabling Message Receipt (cont.)

If you try to send a message to another user while he or she has disabled his message reception, you will get a notice on your screen that says that the user or station did not receive your message.

Enabling Message Receipt

When you are ready to receive messages from other users, type:

CASTON[Retrn]

CASTON cancels your previous CASTOFF command.

Introduction

With the Electronic Mail System (EMS) you can compose, send, and file network mail in the form of documents, letters, files, and memos. You can send mail to both network users and groups.

An EXPRESS MAIL feature enables you to send mail to a user or group so that they will be immediately notified by a prompt on the 25th line of their screen.

Mail you receive will be stored in your personal mailbox directory, which cannot be accessed by any other user. You can open your mail and read it, remove it (if you do not wish to keep it), forward it to other users or groups, or close it and keep it for later use.

You can create letters and documents using any text editor or word processing package. Memos can be created within the Electronic Mail System using the EMS built-in editor.

You may access a comprehensive HELP facility by pressing [F1] or by typing "HELP". This help facility also allows you to pose questions. HELP can be used at any time during command entry. When you have finished using the HELP facility, you will be returned to the same point where you left off.

Many EMS commands let you request and manipulate items of mail via database-like phrases. You may select mail based on its type, source, and date or time of mailing. These qualifying phrases are called **Search Descriptions.**

The available commands can be called by several different names or synonyms. They are described in the detail in this chapter.

Introduction The following procedure is designed to introduce you to the Electronic Mail System.

Procedure: **Create and Send a Memo**

1. Log into the network.

2. Type:

   MAIL[Retrn]

   The screen displays a copyright notice and version number and tells you if you have any mail.

3. Type:

   EDIT TEST[Retrn]

   The cursor moves to the top of the editing screen. You can now type a memo of up to 30,000 characters. Use the arrow keys to move the cursor around if you need to edit.

4. Press:

   [F2]

   to save the memo.

Procedure: Create and Send a Memo

5. To send the memo you just created and saved, type:

   LIST USERS[Retrn]

   to get a list of users on the network

   or

   type:

   LIST GROUPS[Retrn]

   to get a list of groups on the network

   or

   type:

   LIST USERS IN GROUP <group name>

   to get a list of users in a group.

6. After you decide who you want to send the memo to, type:

   SEND TEST TO <username|group name>[Retrn]

   The memo is sent to the user's mailbox. When the user checks his mailbox, he or she will be notified that mail is waiting.

Introduction The following procedure is designed to introduce you to the Electronic Mail System.

Procedure: **Create and Send a Memo**

1. Log into the network.

2. Type:

   MAIL[Retrn]

   The screen displays a copyright notice and version number and tells you if you have any mail.

3. Type:

   EDIT TEST[Retrn]

   The cursor moves to the top of the editing screen. You can now type a memo of up to 30,000 characters. Use the arrow keys to move the cursor around if you need to edit.

4. Press:

   [F2]

   to save the memo.

Procedure: ***Create and Send a Memo***

5. To send the memo you just created and saved, type:

   LIST USERS[Retrn]

   to get a list of users on the network

   or

   type:

   LIST GROUPS[Retrn]

   to get a list of groups on the network

   or

   type:

   LIST USERS IN GROUP <group name>

   to get a list of users in a group.

6. After you decide who you want to send the memo to, type:

   SEND TEST TO <username|group name>[Retrn]

   The memo is sent to the user's mailbox. When the user checks his mailbox, he or she will be notified that mail is waiting.

Procedure: Create and Send a Memo

7. If you want the user to know he or she has mail immediately after you've sent it, type:

   SEND EXPRESS MEMO TEST TO <*username*>[Retrn]

   The memo is sent to the user's mailbox. If he is signed on, a message appears across the bottom of his workstation screen.

8. Type:

   QUIT[Retrn]

   when you are ready to exit to the system.

You have just learned how to create, save and send a memo. The rest of the chapter will guide you through using the many features of the Electronic Mail System.

Rights

Each network user has a unique, encoded mail subdirectory which was created when the system supervisor first defined the users on the system. Mail subdirectories are located in the directory SYS:MAIL. Each user has the following rights to his mail subdirectory: Read, Write, Open, Create, Delete, Search, and Modify. No other users have all rights to your mail subdirectory. This means that another user who wanted to read your mail would have to log in with your user name and current password.

Mail sent to you from other users is deposited in temporary files until you run the Electronic Mail System. When you bring up the EMS, the mail is automatically filed in your mailbox. Network security allows other users to place new mail into your mail subdirectory but restricts them from viewing or altering any other information.

Remove Old Mail

A maximum of 127 items of mail may be archived at one time. The only limit to the size of your mailbox is the size of the volume SYS. Because it is not practical to use the entire disk for mailboxes, you should use the REMOVE command to get rid of old, useless mail. You should avoid mailing large files, especially to many users, because each user will have a duplicate copy of the file which will use up valuable disk space. Instead, send a short memo informing users which directory to look in to find a large file that many users will work with.

Introduction Any network mail will be one of four types:

- o File
- o Document
- o Letter
- o Memo

The first three types are standard MS-DOS files that can be created by any program, text editor (such as EDLIN), or word processing application. A memo is a special text file you create using the EMS memo editor.

The following descriptions will help you determine how to classify a piece of mail.

Files Use files for raw data files, program source files, binary image files, or other such non-message oriented files.

Documents Use documents for word-processed documents or text files.

Letters Use letters for message-oriented text files, usually lengthy or general purpose in nature.

Memos Use memos for relatively short messages addressing specific subjects. Memos belong exclusively to the user who created them originally, as they are special purpose files which reside in the user's mail subdirectory. Memos should be created and maintained ONLY with the EMS Editor. Copies may then be distributed to others via the Electronic Mail System.

Identification

When you identify an item of mail to be distributed, the item should be prefixed by "FILE", "DOCUMENT", or "LETTER". Here are some examples:

FILE E:../MYFILE.DAT

DOCUMENT SYSHELP.DOC

DOCUMENT SYS:ARTICLE/NETWORK.DOC

LETTER C:SALES.RPT

If you do not specify a prefix, EMS will assume your item is a file.

Special Memo Prefixes

A MEMO prefix is much the same except that a memo name is restricted to eight characters. Memo specifications look like the following:

MEMO FIREBIRD

MEMO SALESMTG

Introduction

In addition to the types of mail, there are also attributes that describe the origin of the mail. These are **Forwarded** and **Express** attributes.

Forwarded Mail

Forwarded mail is mail that the sender did not originate but received from someone else and then sent on to you.

Express Mail

EXPRESS mail is mail that has been sent with the express mail feature, instantly informing the recipient that it has arrived by displaying a message at the bottom of his or her screen.

Introduction

One of the features of the Electronic Mail System is its ability to select mail through the combination of qualifiers known as Search Descriptors. You can search for mail by type, source, date and time of mailing, or any combination of these qualifiers. These search descriptions can be used in conjunction with many of the EMS commands which are discussed later in this chapter.

Mail Type

When looking for specific mail, you must first describe the mail type, e.g., "DOCUMENTS AND FORWARDED MEMOS." This example would select memos that have been forwarded to you and documents.

Mail Origination

If you are looking for mail from a particular origination without regard to its type, a phrase such as "MAIL FROM MARK, PETE, OR MANAGEMENT CONSULTANTS" might find what you wish. In this case the word "MAIL" designates that either documents, files, letters, or memos qualify. The word "FROM" designates that the following list is where the mail originated. All mail is tagged as having come from a specific user. The user can be tagged according to username, a group that the user belongs to, or a combination of two groups (the user must belong to both groups to qualify (such as "SECRETARIES IN ACCOUNTING").

Mail Date

The final qualifier for finding specific mail is the date and/or time of mailing. The date can be specified by Year, Year and Month, or Year Month and Day. You can select mail that has been sent before, on, or after the specified date. The time may be specified in 12 or 24 hour style, with or without reference to minutes. If no date is specified, the current day is assumed. Generally, you will only use time comparisons when you have a specific date in mind.

Mail Date (cont.)

However, you can request a search for all mail sent after a specific time on any day (such as all mail sent after 11 pm in 1983) if you wish.

Time and Date Phrases

Time and date phrases are always keyed by one of the words MAILED, SENT, DATED, AFTER, BEFORE, DURING, IN, ON, or AT. Here are some examples:

MAILED DURING 1983

MAILED ON OCT 18, BEFORE 5 PM

MAILED AT 4:59 AM ON JANUARY 14, 1982

DATED SEPTEMBER

MAILED AFTER 14:10

MAILED AFTER 11 PM DURING 1983
(this is the uncommon example described above)

Search Descriptions

Search descriptions are used with commands to specify which mail the commands are to deal with. Here are some examples of how to search for mail:

OPEN FILES AND MEMOS FROM MARK AND VICE-PRESIDENTS

READ ALL MEMOS FROM MARK

CLOSE MAIL FROM VICE-PRESIDENTS

REMOVE FILES FROM MARK

LIST ARCHIVED MAIL FROM ANYONE IN ACCOUNTING

Search Descriptions (cont.)

OPEN FORWARDED DOCUMENTS, FORWARDED MEMOS, AND FILES FROM MIKE, ART, CHUCK, SECRETARIES IN ACCOUNTING, OR JIM

SHOW LETTERS FROM MANAGEMENT

READ ANYTHING

The words ALL, ANYTHING, ANYONE, etc., are predefined optional words that can be used wherever they improve the readability of the commands. They are not necessary but you may wish to use them if you prefer "FROM ANYONE IN ACCOUNTING" to "FROM ACCOUNTING".

Clarity of Commands

When you search for mail, be careful to construct your phrase so that it is not ambiguous. Any separation in a phrase by "AND", "OR", or commas will logically isolate items. As an example, consider the search phrase "OPEN ALL FORWARDED MEMOS AND FILES." This will cause the Mail system to open all forwarded memos and will also cause it to open ALL files. This may be quite different if what you wanted to be opened were all forwarded memos and all FORWARDED files.

Therefore, you must be specific with each item in the list. The phrase "REMOVE ALL EXPRESS MAIL, EXPRESS MEMOS, AND FILES" will remove ALL express mail, ALL express memos, and ALL files. This could be disastrous if you really wanted just express files to be removed.

Introduction

The EMS allows you to send mail to all the members of a user group. Additionally, you can specify a combination of two groups. For example, if you send mail to "SECRETARIES IN ACCOUNTING," then mail will be sent to all users that are members of BOTH groups (Secretaries and Accounting) only. A member of Accounting who is not a member of Secretaries will not receive the mail. This is convenient in an organization where people are distinguished by departments and job titles.

Groups relating to job titles should be defined in the plural (such a Secretaries) and groups relating to departments defined as singular (such as Marketing).

Example

Syntax	Description
All title IN/OF dept	IN/OF ALL CONSULTANTS OF SOFTWARE
title IN/OF dept	MANAGERS IN ENGINEERING
EVERYONE/ ANYONE in dept	ANYONE IN MARKETING
ALL dept title SECRETARIES	ALL MARKETING
dept title	SOFTWARE CONSULTANTS

Example (cont.)

In the previous example, Software, Consultants, Managers, Engineering, Marketing, and Secretaries are all user groups defined by the system supervisor. The word ALL causes no change in the meaning of the phrase.

Searching For Mail From Groups

The EMS allows you to search for mail from users that belong to specific groups or users that belong to a given group combination. See the previous examples for information on how to specify a group combination.

Command Entry

The screen display for the Electronic Mail System is very structured. The last four lines (lines 22 through 25) are used for command entry. When the system prompts you with a right angle bracket (>) on line 24, you may enter a command. The command may be two full lines long if you wish.

When entering a command, you are actually using a subset of the EMS Editor. The arrow keys move the cursor. The Delete and Backspace keys delete characters to the right and left of the cursor, respectively. Text insertion is automatic when you type. [Esc] and [Ctrl]C will erase the entire two-line command to allow you to start over. Pressing [F3] will cause the command line to show the previous command, which you can then edit.

Pressing [Retrn] will cause the command to execute. If the system cannot interpret your command, it will position the cursor where it thinks you made a mistake. You can then correct the mistake and press [Retrn] to try it again.

Information Window

The largest portion of the screen is the Information Window. This window displays the information you request with a command. For example, if the command was READ, the information window contains the document you are reading. The name of the mail you are looking at is at the top of the window and the text occupies the rest of the window. Anything displayed in the window may be scrolled up or down. The following keys manipulate the contents of the information window:

- o [Up Arrow] scrolls the line immediately above the information window into view.
- o [Down Arrow] scrolls the line immediately below the information window into view.
- o [PgUp] causes the window to display the preceding window-full of information, as in the Editor.
- o [PgDn] causes the window to display the next window-full of information, as in the Editor
- o [Home] positions the window at the beginning of the information being displayed.
- o [End] positions the window at the end of the information being displayed.

Command Entry

The screen display for the Electronic Mail System is very structured. The last four lines (lines 22 through 25) are used for command entry. When the system prompts you with a right angle bracket (>) on line 24, you may enter a command. The command may be two full lines long if you wish.

When entering a command, you are actually using a subset of the EMS Editor. The arrow keys move the cursor. The Delete and Backspace keys delete characters to the right and left of the cursor, respectively. Text insertion is automatic when you type. [Esc] and [Ctrl]C will erase the entire two-line command to allow you to start over. Pressing [F3] will cause the command line to show the previous command, which you can then edit.

Pressing [Retrn] will cause the command to execute. If the system cannot interpret your command, it will position the cursor where it thinks you made a mistake. You can then correct the mistake and press [Retrn] to try it again.

Information Window

The largest portion of the screen is the Information Window. This window displays the information you request with a command. For example, if the command was READ, the information window contains the document you are reading. The name of the mail you are looking at is at the top of the window and the text occupies the rest of the window. Anything displayed in the window may be scrolled up or down. The following keys manipulate the contents of the information window:

- o [Up Arrow] scrolls the line immediately above the information window into view.
- o [Down Arrow] scrolls the line immediately below the information window into view.
- o [PgUp] causes the window to display the preceding window-full of information, as in the Editor.
- o [PgDn] causes the window to display the next window-full of information, as in the Editor
- o [Home] positions the window at the beginning of the information being displayed.
- o [End] positions the window at the end of the information being displayed.

Information Window (cont.)

When you finish viewing the contents of the information window you should press [F2]. (Any other keys will be ignored.) You will then be returned to command entry mode.

If the EMS is processing a READ request for you, then [F2] can have a slightly different function. Pressing [F2] will terminate the current display and bring up the next piece of mail if you specified more than one piece of mail in your command.

Help

If you forget how to compose a specific command, even if you've already typed half of it, you may request help by pressing [F1]. A general help screen will appear, instructing you how to ask questions or select help according to the topical menus that are given on each help screen.

After you find your answer, pressing [Esc] will return you to the command line editor, and you can pick up where you left off. The last screen you were viewing in the help facility will remain visible while you continue editing.

Introduction The commands discussed in the next pages will allow you to take advantage of the flexibility of the EMS. You must enter these commands from the command line editor. If you need help while you are working, pressing [F1] will always provide you with on-line assistance.

Purpose To check for newly received mail.

Command Format CHECK[Retrn]

Synonym CHK

Remarks The CHECK command checks for new mail that has been received but not filed into your mailbox. This is done automatically when you enter the Mail System, but if you want to check for mail that you may have received while you were using the EMS, you can use the CHECK command. Any new mail is automatically opened.

Purpose To close open items of mail.

Command Format CLOSE <ID list>[Retrn]
CLOSE <search description>[Retrn]

Synonym C

Remarks The CLOSE command closes (or takes off the open list) any items of open mail which match the given description. The mail remains in your mailbox for later recall.

A CLOSE with no list or description will close all open mail.

Examples CLOSE 1, 5 THRU 10, AND 3

CLOSE 5

CLOSE

CLOSE FILES FROM SOFTWARE CONSULTANTS

CLOSE MEMOS AND DOCUMENTS DATED 10/23

Purpose Display a listing of files in a given directory.

Command Format DIRECTORY [pathspec\]<filename>[Retrn]

Synonyms D
DIR

Remarks This command will display a list of files in any directory to which you have search privileges. Wildcard characters (* and ?) may be used to specify the file name. DIR with no options will cause a listing of files in your default directory to be displayed.

Examples

```
DIR

DIR *.COM

DIR E:

DIR SYS:PUBLIC/L*.EXE
```

Purpose Create and edit your own memos.

Command Format EDIT [MEMO]<memo name>[Retrn]

Synonym E

Remarks The EDIT command invokes a simple word processor that you may use to create new memos or edit existing memos. The maximum size for a memo (not letters or files) is 30,000 characters.

When typing with the editor, you will notice that it is like using a typewriter except that it is easier. The editor inserts characters as they are typed and automatically wraps the words to the next line if you come to the margin. You can move the cursor to any position that you want to type and if there is a character there, the new character will be inserted with the old character moving to the right. If you want to end a line and go to the next line, then type [Retrn].

The cursor may be moved in any of four directions with the arrow keys on the keyboard. If typing the arrow keys causes numbers to be inserted, then press [Num Lock].

When you have finished editing your memo, press [F2] to save the memo. The memo can then be mailed with the SEND Command.

Remarks (cont.)

If you wish to cancel editing and abandon the edited version of the memo, press [Ctrl]Break. A question, "Cancel Memo Editing? (Y/N)" will appear. Press "Y" to cancel, or any other key to continue editing.The editing functions available are described below.

Cursor Movement

Arrow Keys	Move the cursor in the direction indicated by the arrow.
[F10]	Moves the cursor to the far right of the line.
[F9]	Moves the cursor to the far left of the line.
[PgUp]	Moves the cursor to the previous screen of text.
[PgDn]	Moves the cursor to the next screen of text.
[Ctrl][Home]	Moves the cursor to the very beginning of the memo.
[Ctrl][End]	Moves the cursor to the very end of the memo.

Delete Character	[Backspace]	Deletes a character to the left of the cursor.
	[Del]	Deletes a character to the right of the cursor.
Delete Line	[F5]	Deletes a line and causes the line below to move up into its place.
Delete to End of Line	[F6]	Deletes the text from the cursor to the end of the line that the cursor is on.
Scrolling	[Ctrl]Z	Causes the screen to scroll up. The cursor moves with the text.
	[Ctrl]W	Scrolls the text down. The cursor moves with the text.

Examples

EDIT DEMOSALE

EDIT MEMO MARKET3

Purpose To help you use the Electronic Mail System.

Command Format
HELP[Retrn]
HELP <question>[Retrn]

[F1] (you may press this key at any time during command editing)

Synonyms
H
WHAT
WHY
HOW

Remarks The EMS provides you with a comprehensive interactive help facility. If you ask a question from the EMS command interpreter, the help facility will display an appropriate help screen in the information window. If you do not ask a question, a screen will appear to explain how the help facility works.

Once you have entered the help mode, there are two methods you can use. First, you may press [F2] and then enter a question in English. Your question may be entirely free-form. The help facility scans words that it finds in the question to determine the subject of your question. Secondly, you may choose from several topics that will be highlighted on each help screen. The topic that is displayed in reverse video is the selected topic. You can use the cursor direction keys (up, down, left, and right arrows) to change topics. Press [Retrn] to display the information regarding the selected topic.

Remarks (cont.)

To exit the help command, press [Esc]

You may enter the interactive help mode by pressing [F1] at any time during the command entry. After you have located the information you need, you may press [Esc] to return to the EMS editor and continue editing your command right where you left off.

Purpose To display a list of open or filed items of mail.

Command Format
LIST[Retrn]
LIST <search description>[Retrn]
LIST FILED <search description>[Retrn]

Synonym L

Remarks The LIST command allows you to review what items of mail are currently open or find out what is archived in your mailbox. LIST with no parameters will list all open mail. Unless the word "FILED" or "ARCHIVED" is given after "LIST", only open mail will be considered.

The mail is described by a mail header--one or two lines of information describing the type, attributes, source, and time of distribution of the mail item. If the mail has been opened, its ID will appear at the beginning of the first line.

Examples
LIST ARCHIVED FILES FROM
MIKE OR ART

LIST FILED MAIL

LIST EXPRESS MEMOS FROM ACCOUNTING

Purpose To display a list of memos that you have created with the EMS editor.

Command Format LIST MY MEMOS [memo specification][Retrn]

Synonyms L MY MEMOS
MLIST
ML

Remarks The LIST MY MEMOS command allows you to display a list of memos that you have created using the EMS memo editor. If you wish, you may supply a wildcard memo specification (X*, SALES*, ??LIST, etc.) to list only certain memos.

Examples LIST MY MEMOS

MLIST SALES??

LIST MY MEMOS XYZ?AB*

Purpose

List network users, groups, and members of groups or group combinations.

Command Format

LIST USERS[Retrn]
LIST GROUPS[Retrn]

LIST USERS IN <group name>[Retrn]
LIST [GROUP] <group name>Retrn]

LIST<group combination>[Retrn]

Synonym

L

Remarks

This command allows you to find out what users and groups are defined on the network (1st and 2nd options). You can also list the members of any group (3rd and 4th options above) or users that belong to any two groups (5th option), i.e., the common members between any two groups, such as Secretaries in Accounting.

This command will let you know just who will receive mail that you send to groups or group combinations.

Example

LIST EVERYONE IN
ACCOUNTING

LIST ALL SOFTWARE CONSULTANTS

LIST USERS

Purpose Open specified items of mail for examination or manipulation.

Command Format OPEN <search description>[Retrn]

Synonym O

Comments The OPEN command searches your mailbox for all mail that matches the given description and places it in the open, or "active" list with an attached mail ID. This is a number between 1 and 127 inclusive by which you can reference that piece of mail with EMS commands.

Example

```
OPEN ALL MAIL

OPEN ANY MEMOS OR LETTERS SENT AFTER SEP 15

OPEN MAIL FROM ACCOUNTING SECRETARIES

OPEN EXPRESS MEMOS
```

Purpose

Copy a memo or other item of mail from your mailbox to a normal DOS file.

Command Format

PUT<mail>ID TO [pathspec/]<filename>[Retrn]
PUT MY MEMO <memo> TO [pathspec]<filename>[Retrn]

Synonyms

P
DUMP
COPY

Remarks

The PUT command gives you the ability to transfer an item of mail to a DOS file that can be printed, modified, or accessed outside of the Electronic Mail System. The mail item to be copied must first be opened so you can then identify it by its mail ID.

You may also transfer memos created with the EMS Editor by using the "PUT MY MEMO" variation shown above.

If the mail item is a memo it will be translated from the EMS format to a normal text file format. You must have Write, Create, and Delete privileges in the target directory area for the command to succeed.

Examples

PUT 3 TO TEST.FIL

PUT MY MEMO ARTIST TO SYS:MARK/MUSIC/ARTIST.DOC

Purpose Terminate the current EMS session and return to DOS.

Command Format QUIT[Retrn]

Synonym
Q
BYE
OUT
STOP
END
EXIT
ABORT
SYSTEM

Examples
Q

PLEASE STOP

Purpose Display the specified mail in the information window.

Command Format
READ[Retrn]
READ ID <list>[Retrn]
READ <search description>[Retrn]

Synonyms
R
SHOW
SCAN

Remarks The Read command causes the contents of all open mail which matches the given description to be displayed one at a time in the information window. If you specify an ID list, the mail items are displayed in the order specified. The ID is the number from 1 to 127 that is assigned to the mail when it is opened.

Items requested by a search description will be displayed in the order in which they occur in the list of open mail (by ascending ID number).

A READ with no list assumes that all open mail is to be shown.

Pressing [F2] will cause the current item to be removed from the information window and the next item to be loaded. [Ctrl]Break will cause a return to the command mode.

Examples

READ MAIL FROM MIKE

READ LETTER 16

READ 1 - 5, AND 10

READ EXPRESS MAIL

READ ANYTHING SENT BEFORE 1 PM

Purpose Remove filed items of mail from your mailbox.

>> CAUTION <<
This command has a potentially dangerous effect.

Command Format
REMOVE <ID list>[Retrn]
REMOVE <search description>[Retrn]

Synonyms
REM
ERASE
TRASH

Remarks The REMOVE command locates the mail which matches the given description and removes it from the mailbox. This command removes both open and unopened mail.

When used with a date/time qualifier, the Remove command can be very useful for purging old mail.

Examples

REMOVE ALL MEMOS FROM GLORIA

REMOVE ANY MAIL FROM MARK, JAMIE, OR MANAGEMENT

REMOVE MY MEMO FIREBIRD

REMOVE ALL MAIL

REMOVE ALL MAIL MAILED BEFORE OCTOBER

TRASH EVERYTHING MAILED BEFORE 10:30 AM

Purpose Remove memos that you have created with the EMS Editor.

Command Format REMOVE MY MEMOS [memo specification][Retrn]

Synonyms
MREMOVE
MR
REM MY MEMOS
ERASE MY MEMOS
TRASH MY MEMOS

Remarks The REMOVE MY MEMOS command allows you to discard old unwanted memos that you created with the EMS Memo Editor. Rather than removing them one at a time, you may include a wildcard specification to destroy whole sets of memos at the same time.

REMOVE ALL MY MEMOS will remove ALL your memos and should be considered carefully before being used.

Examples
REMOVE MY MEMOS
SALES??

MR XYZ?AB*

TRASH ALL MY MEMOS

Purpose

To send files, documents, letters or memos to other users or members of user groups.

Command Format

SEND <source list> TO <target list>[Retrn]

<source item> is
<file> | <document> | <letter> | <memo> | <open mail ID

<target item>is
<user name> | <user group> | <user & group combination>

Synonyms

MAIL

Remarks

With the SEND command, you can send mail from many different sources to many destinations with one command. File names for files, letters, and documents can be preceded by a full or pathspec. For any source specified, you must have Read, Search, and Open privileges to the directory that each file resides in.

After the complete command has been accepted, the mail distribution will begin. A distribution log will be displayed in the information window, identifying each user who successfully received mail. If for some reason a condition prevents a user from receiving the mail, you will be notified in the log. If the log exceeds one window in length, you may scroll the display as usual.

Remarks (cont.)

To inform the recipient(s) of the arrival of critical information, you may use the EXPRESS option. Just place the word "Express" before any source item that you wish to have marked as urgent. Each user that receives the mail will be notified immediately via the supervisor's broadcast facility.

Forwarding an item of mail that has been previously received is accomplished by opening it (see the OPEN command) and supplying its associated mail ID as a source in the send command. The item of mail will be sent to the target users, and it will have the attribute "Forwarded" associated with it. (See mail attributes)

Examples

SEND DOCUMENT ..\EUROPE.DOC TO JAMIE

SEND GL0583.DAT,EXPRESS MEMO SALESMTG
AND DOCUMENT TECHNOTE.15A
TO MIKE, ART, AND ENGINEERING

SEND EXPRESS MEMO READ THIS AND 7 TO GLORIA

SEND LETTER WORK:COMPLAINTS/BAD-NEWS
TO PRESIDENTS AND MANAGERS
IN MARKETING

Note: In the last example it is important to note that the target list is interpreted as "presidents" and "managers in marketing", NOT as "presidents and managers" of marketing.

Purpose View the contents of ordinary text files (files that are not related to the EMS).

Command Format VIEW [pathspec/]<filename>[Retrn]

Synonym V

Remarks The VIEW command allows you to view the contents of any files to which you have Read and Open privileges. Full or partial pathspecs may be specified in front of the file name, if desired.

Examples

```
VIEW MYFILE.DAT

VIEW E:TEXT.NEW

VIEW SYS:DEMO\SCREEN.X1
```

MANAGING DIRECTORIES

Overview

This chapter begins with a discussion of NetWare's directory structure, followed by listings of the NetWare and DOS commands related to directory management.

Be sure to review the Command Rules and Restrictions in the Preface to this manual before attempting to use the commands presented here.

Directory Commands

NetWare commands related to directory management and covered in this chapter are:

- o CHKVOL
- o ERASEDIR
- o FLAGDIR
- o GRANT
- o LISTDIR
- o MAKEDIR
- o NPATH
- o RENDIR
- o REVOKE
- o SHOWDIR
- o TLIST
- o UDIR
- o VOLINFO

DOS commands related to directory management and covered in this chapter are:

- o CHDIR
- o CHKDSK
- o DIR
- o MKDIR
- o PATH
- o RMDIR

What is a Directory?

A directory is a list of the names of a group of files on a drive or on a diskette. Directories also contain information about the size of the files, their location on the hard disk or diskette, and the dates and times they were created or updated.

In an office, you can separate files by putting them in different filing cabinets, in effect creating different directories of information. NetWare allows you to organize the files on your hard disk drive into directories. Directories are a way of dividing your files into convenient groups of files. Any one directory may contain any reasonable number of files, and it may also contain other directories, called subdirectories. A subdirectory is a directory you create to further organize a group of files.

Like a directory, a subdirectory doesn't hold the text of the files within it. It lists the names of both files and subdirectories. Say you have all your report files in a directory called REPORTS (which is, in fact, a subdirectory of the volume SYS:). You may want to divide these files into subdirectories such as MONTHLY and WEEKLY. In a directory listing, the subdirectory REPORTS would list the subdirectories MONTHLY and WEEKLY.

The directory structure grows as you create new directories for groups of files. Within each new directory, you can create subdirectories or files. The maximum number of network directories and files is determined at installation by the system supervisor.

You can think of this structure as a family tree. The volume, SYS:, is the original parent. The root directory then has offspring, which are called directories or subdirectories, and each of these can become the parent of another generation of subdirectories. A subdirectory has only one parent directory, but a parent directory can have several subdirectories descending from it.

Home Directories

During installation, the system supervisor mapped a network path to each user's "home" volume and directory. When you log in the system automatically places you in your home directory. Once you are logged in, you may create your own chain of subdirectories, and define paths to other directories to which you have access rights. You use the MAKEDIR command (described later in this chapter) to create new directories and subdirectories.

Defining Network Paths

Networking allows you to access information in any network directory to which you have the proper access rights.

To access information you must specify a directory "path" on some logical network drive. The path must identify -- or "point" to -- the directory where the information is stored.

You specify a full directory path by first specifying the root directory of the volume on which the directory resides, and then naming successive subdirectories below the root directory name which lead down the path to the subdirectory you need. The last subdirectory name in any path is always the name of the subdirectory you are accessing.

For example, let's suppose Mike's progress report is in the file PROGRESS.RPT, which is in the MONTHLY subdirectory of his REPORTS directory, which is a subdirectory of his home directory, MIKE, which is a subdirectory of the USERS directory in the volume SYS:. If you had access rights to Mike's subdirectories, you could define a path to take you to them.

If you were typing it, such a path would look like this:

```
NPATH SYS:USERS\MIKE\REPORTS\MONTHLY
```

Pressing [Retrn] would place you in subdirectory MONTHLY

Defining Network Paths (cont.)

Note that a backslash "\" is used to separate all subdirectories after the first directory.

Once in the MONTHLY directory, you could issue a command to open the file PROGRESS.RPT.

Directory-Related Commands

The following pages contain NetWare and DOS commands that will help you use your directory structure efficiently. You should review all of these commands for a thorough understanding of the network concepts involved.

Purpose Displays statistics for a specified volume.

Command Format CHKVOL [<volume> | <drive>][,...][Retrn]

Volume defaults to the current volume. You can use an asterisk (*) to request the statistics of several volumes.

Drive shows the statistics for the volume the drive is mapped to.

Multiple drives and/or volume names may be entered, separated by a comma or a space.

Remarks The CHKDSK command provided with DOS won't give the proper statistics for a network drive. The CHKVOL command must be used instead.

Examples If drive F: was mapped to SYS:LEVEL1\LEVEL2\LAST, you could obtain the statistics for volume SYS: by entering:

CHKVOL F:[Retrn]

If you type:

CHKVOL SYS:[Retrn]

the system will display something <u>similar</u> to:

```
Statistics for fixed volume FD2/SYS:
 17678336     bytes total volume space,
 12165120     bytes in 770 files,
  5513216     bytes available on volume,
      766     directory entries available.
```

Purpose Removes an empty subdirectory from your directory listing.

Command Format ERASEDIR [<pathspec>\]<subdirectory name>[Retrn]

Subdirectory name is any group of 14 characters (as explained in the MAKEDIR command).

The pathspec is required only if the subdirectory being erased is not an immediate subdirectory of the current default directory.

Abbreviations in the pathspec may be given as follows:

: / or ~	for current volume
.	for current directory

Wildcards (* and ?) can be used with ERASEDIR.

Remarks The ERASEDIR command removes the specified subdirectory from the network directory structure.

Subdirectories can't be erased if they contain files or other subdirectories, or if another user has an active path mapping to that subdirectory.

Also, you should never erase a directory which you--or any other users--are currently mapped to. If you should accidentally do this, (whether by typing "ERASEDIR ." or by specifying the pathspec), you should be able to recover normal operation by changing to another default drive.

A user attempting to remove a subdirectory must have Personal Ownership privileges to the directory above the subdirectory to be removed.

Remarks (Cont.)

If you want to delete a subdirectory of the directory that you are currently mapped to, you need only supply the subdirectory's name. In any case, you can always specify a full path, with the subdirectory to be deleted as the last name in the path specification.

Examples

To remove subdirectory SYS:FOUNDATION\FIRST\OLD, type the following (from any directory):

ERASEDIR SYS:FOUNDATION\FIRST\OLD[Retrn]

and the system responds with:

SYS:FOUNDATION\FIRST\OLD deleted.

Note that subdirectories SYS:, SYS:FOUNDATION, and SYS:FOUNDATION\FIRST are left unchanged. Only the last subdirectory in the list has been removed.

If your current drive was already mapped to SYS:FOUNDATION\FIRST, subdirectory OLD could be removed by entering:

ERASEDIR OLD[Retrn]

and the system would respond with:

SYS:FOUNDATION\FIRST\OLD deleted.

Purpose Displays or modifies the access attributes of a directory.

Command Format FLAGDIR [<pathspec> [</flag list>]][Retrn]

<flag list>
/flag/flag.../flag
ALL

<flag>

Read	**Delete**
Write	**Personal Ownership**
Open	**Search Directory**
Create	**Modify Flag**
NOflag	**~flag**
NONE	

NO or ~ in front of a flag will take away that privilege; for example, NOWRITE or ~WRITE.

The word NONE in the flag list removes all privileges <u>except</u> those specified in the flag list. If NONE is used, it must appear first in the flag list.

Remarks The directory listed will be assigned the same access privileges indicated by the flag list.

This means trustees may exercise their rights only if they coincide with that directory's rights. Changing the security of a directory doesn't affect the security of its subdirectories.

You do not need to be mapped to a directory to modify or display its attributes.

Remarks (cont.)

As explained in Chapter 1, directory security restricts the rights of all users. Therefore, directory rights should not usually be restricted--unless, for example, the directory contains public, read-only files.

Examples

Suppose you type:

FLAGDIR DIR1 NONE READ OWNERSHIP[Retrn]

and the system responds:

Directory DIR1 has been set to [R P]

Now, simply typing:

FLAGDIR DIR1[Retrn]

will produce this display:

Directory DIR1 is set to [R P]

To restore all rights, type:

FLAGDIR DIR1 ALL[Retrn]

and the system responds:

Directory DIR1 is set to [RWOCDPSM] (all)

If you only type:

FLAGDIR[Retrn]

then all subdirectories immediately below the default path and their flags are shown.

Examples (cont)

Typing this command:

FLAGDIR SALES /NONE CREATE[Retrn]

would assign only the create privilege to the directory SALES.

Typing:

FLAGDIR SALES /NOCREATE[Retrn]

would remove the create privilege.

Purpose

Grants trustee privileges for a particular directory to a user or user group.

Command Format

GRANT [ONLY] <privilege list> [EXCEPT <privilege list>] [FOR <pathspec> | <drive:>] TO [USER | GROUP] <user> | <group name>[Retrn]

<privilege list>
<privilege,privilege,...,privilege>
ALL RIGHTS
NO RIGHTS

privilege

Read	- ability to read files
Write	- ability to write files
Open	- ability to open files
Create	- ability to create files
Delete	- ability to delete files
Personal or **OW**nership	- ownership rights
Search	- ability to view the directory
Modify	- ability to modify the status of files (i.e., non-shareable, read/write)

If <pathspec> is not given, rights are granted for the current directory.

Remarks

A trustee is defined as any user or group that has been given any privileges for a directory on the network. So, anyone who has rights to a directory area has been "entrusted" with privileges.

GRANT is used to modify or set a trustee's privileges for a specified directory. REVOKE may also be used to modify privileges.

Remarks (cont.)

To GRANT or REVOKE trustee privileges in a directory, you must have personal ownership rights for that directory.

The rights a user has for a directory remain in effect for all of the subdirectories beneath that directory, unless privileges are specifically re-granted to the user somewhere in the directory structure.

Some users or user groups are entrusted with more privileges than others.

The network always attempts to give you the most privileges that it can; if you have no privileges to a directory as a user, but belong to a group that has been given all privileges, you will be a trustee with all privileges for that directory.

A directory can have an almost unlimited number of trustees.

Examples

To grant user Pat no rights in the SYS:SALES directory, type:

GRANT NO RIGHTS FOR SYS:SALES TO USER PAT[Retrn]

To grant Pat all rights in SALES, type:

GRANT ALL RIGHTS FOR SYS:SALES TO USER PAT[Retrn]

and the system acknowledges with:

Trustee's access rights set to [RWOCDPSM]

Examples (cont.)

Any rights you GRANT are added to the user's current rights unless you use the ONLY option. With ONLY, any existing rights are revoked, and only the rights listed will be granted. For example, although Pat may have all rights in SALES, the following command would grant the Read right but revoke all others:

GRANT ONLY READ FOR SYS:SALES TO PAT[Retrn]

Privileges can also be listed, separated by commas, as follows:

```
GRANT C,W,O,P TO PROGRAMMERS
GRANT CREATE,WRITE,OPEN TO PROGRAMMERS
```

You also have the option to:

GRANT ALL RIGHTS EXCEPT <list> TO PAT

Purpose Lists subdirectories directly beneath the specified path.

Command Format LISTDIR [<pathspec>][Retrn]

If no <pathspec> is given, the default path is assumed.

Abbreviations in the pathspec may be given as follows:

: / or ~	for current volume
.	for current directory

Remarks LISTDIR also lists the access privileges of the subdirectories.

LISTDIR only lists the subdirectories below the immediate level. Use the SHOWDIR command to display the directory structure of a volume or of another subdirectory.

Example If you were to type:

LISTDIR SYS:FOUNDATION[Retrn]

the screen would display something similar to the following:

```
Subdirectories in SYS:FOUNDATION
SECOND              [RWOCDPSM]
FIRST               [R OCD S ]
```

Purpose Makes a new subdirectory.

Command Format MAKEDIR [<pathspec>\]<subdirectory name>[Retrn]

Subdirectory name is any group of 14 characters with the following exceptions:

- o backslash (\)
- o slash (/)
- o colon (:)
- o semicolon (;)
- o asterisk (*)
- o question mark (?)
- o comma (,)
- o caret (^)
- o tilde (~)
- o percent sign (%)
- o left bracket ([)
- o right bracket (])

A period (.) may occupy any position of the subdirectory name except the first character.

If a <pathspec> is not given, the new directory will be a subdirectory of the current directory.

Remarks The MAKEDIR command doesn't require you to be mapped to the path where the new subdirectory is to be added. Any path may be specified and directories at all levels in that path will be created if they don't already exist.

Each new directory will be created with all directory access rights. To modify these access privileges, use the FLAGDIR command.

Examples To create the subdirectory SYS:UTILITIES/CRAIG, enter the following three commands:

NPATH D:=SYS:UTILITIES[Retrn]

D:[Retrn]

MAKEDIR CRAIG[Retrn]

You could also just type:

MAKEDIR SYS:UTILITIES\CRAIG[Retrn]

In either case, the system responds with:

SYS:UTILITIES\CRAIG created.

Now if you type:

MAKEDIR SYS:UTILITIES\CRAIG\SALES\OCT[Retrn]

SALES will be created as a subdirectory of CRAIG, and at the same time, OCT will be created as a subdirectory of SALES.

Purpose Assigns or maps a network drive to a directory path.

Command Format NPATH [[<drive>: [=]] [<drive>: | <pathspec>][Retrn]

The result of the command depends on the selection of parameters:

NPATH	Displays all of the current path mappings of the workstation.
NPATH <drive>:	Displays the specified drive's current mapping.
NPATH <drive>:=	Maps the specified drive to the directory the second drive is mapped to.
NPATH <drive>:= <drive>:	Maps the first drive to the directory the default drive is mapped to.
NPATH <drive>:= <pathspec>	Maps the specified drive to the given pathspec.
NPATH<pathspec>	Maps the current default drive to the given pathspec.

Abbreviations in the <pathspec> may be given as follows:

: / or ~ for current volume

One NPATH command may assign multiple drives. Separate each assignment with a semicolon. For example:

NPATH SEARCH1:=SYS:PUBLIC; E:=SYS:SALES;[Retrn]

Remarks

NPATH is a versatile combination of the DOS Path and CD commands.

The NPATH command allows users to point their network drives to directories in all areas of the network directory structure.

Twenty-six drives (A through Z) are available to each workstation as pointers to the directories that a user wishes to use. If you want to use all 26 drives, or just one drive, you can use the NPATH command to point or map any drive to any directory.

Since the system's security will only let you use files in directories to which you have been granted rights, it won't make much sense to point any drives to directories in which you don't have access privileges. NPATH allows more than one drive to be mapped to the same directory.

A local drive (such as drive A: or B:) can be mapped to a network directory. If a local drive is mapped to a network directory which is later deleted, then the local drive letter will be remapped back to the local drive.

Of the 26 available drives, up to 16 of these drives can be assigned as search drives (SEARCH1 through SEARCH16). They can be abbreviated (S1 through S16). NPATH assigns search drives starting at drive Z and working backwards toward drive A. Because of this, it makes sense for you to assign non-search drives from drive A and work toward drive Z.

Search drives created by NPATH will be checked whenever a command being executed requests a file to be opened that isn't on the current drive. The search drives will be checked in the order of their number (S1 through S16). Search drives are checked by any Open File request. (This is different from DOS 3.1, which only checks search drives for opens when a command is executed.)

Remarks (cont.)

Search drives can be mapped to any directory on the network.

NPATH allows you to assign multiple search paths in a particular sequence. However, the order in which they are assigned takes precedence over the drive number specified in the NPATH command. (If you attempt to assign S3 before S1 and S2 have been defined, the drive will be called S1.)

When you log out of the network, your search drives are deleted and will be reset during the next LOGIN by the NPATH commands in your login script.

Examples

The following command maps drive F: to subdirectory SALES of the root directory, which in this case is SYS:

NPATH F:=\SALES[Retrn]

System response:

Drive E: = SYS:SALES

As mentioned earlier, more than one drive may be mapped to the same directory. The following command will map drive C: to the same directory that drive F: is mapped to:

NPATH C:=F:[Retrn]

System response:

Drive C: = SYS:SALES

Examples (cont.)

If you attempt to map a local drive--for example, B:--to a network directory, you will receive the following prompt:

Drive B: currently maps to a local disk.
Do you want to assign it as a network drive? (Y/N)?

If you answer Y, the NPATH command will proceed as usual. To map it back to a local drive, use the Del option (explained later in this section).

The following command assigns the second search drive, which is drive Y:

NPATH SEARCH2:=SYS:SALES\DOC[Retrn]

System response:

SEARCH2: = Y:. [SYS:SALES\DOC]

This command maps Search3 to TEST, which is a subdirectory of the directory that drive Y is mapped to:

NPATH SEARCH3:=SEARCH2:\TEST[Retrn]

System response:

SEARCH3: = X:. [SYS:SALES\DOC\TEST]

Examples (cont.)

Now if you type:

NPATH[Retrn]

all of your workstation's drive mappings and search drive definitions will be shown on the screen. Note that only drives with assigned directories are listed.

```
Drive A:    is mapped to a local drive.
Drive B:    is mapped to a local drive.
Drive C: =  SYS:SALES
Drive D: =  SYS:ACCOUNTING
Drive F: =  SYS:SALES
Drive X: =  SYS:SALES\DOC\TEST
Drive Y: =  SYS:SALES\DOC
Drive Z: =  SYS:PUBLIC
  -------
SEARCH1: =  Z:. [SYS:PUBLIC]
SEARCH2: =  Y:. [SYS:SALES\DOC]
SEARCH3: =  X:. [SYS:SALES\DOC\TEST]
```

NPATH Options

The Insert option is used to insert a new search path in the ordered sequence of search paths. For example, if you typed:

NPATH INSERT SEARCH2:=SYS:SUPER[Retrn]

the search directories would now be:

```
  ------
SEARCH1:    =  Z:. [SYS:PUBLIC]
SEARCH2:    =  W:. [SYS:SUPER]
SEARCH3:    =  Y:. [SYS:SALES\DOC]
SEARCH4:    =  X:. [SYS:SALES\DOC\TEST]
```

Note that the drive order doesn't necessarily remain alphabetic, but the search order remains numeric (Search1, Search2 and so on).

NPATH Options (cont.)

The Remove (REM) option is used to remove an NPATH definition. For example, if you type:

NPATH REM F:[Retrn]

the system will respond:

Definition for drive F: removed.

If you attempt to remove a drive which is mapped as a search drive--for example, W:--you will receive a warning similar to the following:

Drive W: is used by the following search mapping(s):
SEARCH2:
Do you still want to remove it? (Y/N)

If you answer Yes, the system responds:

Mapping for SEARCH2: has been removed.
Definition for drive W: removed.

However, if instead of attempting to remove drive W: you simply remove Search2: then drive W: would be left pointing at the same directory. For example:

NPATH REM S2:[Retrn]

results in the removal of the mapping for SEARCH2:, but to remove drive W: completely, another remove will have to be issued, because W: is still mapped to SYS:SUPER.

A local drive that has been pointed to a network directory won't be removed with a REM option but will be pointed back to the local drive.

Purpose Renames a subdirectory (gives it a new name).

Command Format RENDIR <pathspec> [TO] <subdirectory name>[Retrn]

Subdirectory name is any group of 14 characters with the following exceptions:

- o backslash (\)
- o slash (/)
- o colon (:)
- o semicolon (;)
- o asterisk (*)
- o question mark (?)
- o comma (,)
- o caret (^)
- o tilde (~)
- o percent sign (%)
- o left bracket ([)
- o right bracket (])

The first character of the subdirectory name may not be a period.

Abbreviations in the <pathspec> may be given as follows:

: / or ~	for current volume
.	for current directory

Remarks The RENDIR command changes the old name of the subdirectory identified by the last level named in the pathspec to the new subdirectory name. You must have Ownership and Modify rights to rename a subdirectory.

Remarks (cont.)

Only one subdirectory may be renamed at a time. In other words, only the last subdirectory in the pathspec will be renamed. All preceding subdirectory names in the pathspec will not be changed. (They must be the current names of existing directories.)

Examples

If you type:

RENDIR JIM TO WORK[Retrn]

the system responds with:

Directory JIM renamed to WORK.

If you type:

RENDIR SYS:JIM\DOC TO MANUAL[Retrn]

the system responds with:

Directory SYS:JIM\DOC renamed to MANUAL.

If SYS:JIM\MANUAL is your default directory, typing:

RENDIR . TO REPORTS[Retrn]

will cause the following system response:

Directory SYS:JIM\MANUAL renamed to REPORTS.

Purpose

Revokes trustee privileges for a particular directory from a user or group.

Command Format

REVOKE <privilege list> [FOR <pathspec> | <drive>:]
FROM [GROUP | USER] <group> | <user name>[Retrn]

<privilege list>
<privilege,privilege...,privilege>
<privilege privilege... privilege>

<privilege>

Read	- ability to read files
Write	- ability to write files
Open	- ability to open files
Create	- ability to create files
Delete	- ability to delete files
Personal or OWnership	- ownership rights
Search	- ability to view the directory
Modify	- ability to modify the status of files

Each entry in the flag list is separated by a comma or a blank space. Flags can be abbreviated as indicated by the bold characters above.

Remarks

If a user with Personal Ownership rights to a directory has GRANTed trustee privileges for that directory to another user, the directory owner may then REVOKE any or all of those privileges.

The GRANT command can also be used to modify or set a trustee's privileges for a specified directory.

For more information, see the GRANT command, and refer to the NetWare Security discussion in Chapter 1.

Examples To revoke all rights from user PAT in the SALES subdirectory, enter:

REVOKE ALL RIGHTS FOR \SALES FROM USER PAT [Retrn]

and the system acknowledges with:

User Pat is no longer a trustee of the given directory.

or if the user Pat had all privileges, the command:

REVOKE SEARCH RIGHTS FROM USER PAT[Retrn]

would respond with:

Trustee's access rights set to [RWOCD PM]

Purpose

Shows the directory structure of the specified volume, drive, or path.

Command Format

SHOWDIR [<pathspec> | <drive>:][Retrn]

Remarks

SHOWDIR is useful for displaying the directory structure of a specified drive or path.

If drive: or pathspec is not specified, the default drive or pathspec is assumed.

Examples

To show the structure of SYS:FOUNDATION, type:

SHOWDIR SYS:FOUNDATION[Retrn]

Or to show the entire directory structure of volume SYS:, type:

SHOWDIR SYS:[Retrn]

and the system responds:

```
Subdirectory structure for SYS:
  DEMO
    TEST3
  PUBLIC
  INSTALL
  ARCHIVE
  MAIL
    70002
    60003
  SALES
  TEST
  DOC
  LOGIN
  SUPER
```

Examples (cont.)

If you are mapped to SYS:SALES and you just type:

SHOWDIR[Retrn]

the system defaults to the current path and shows:

```
Subdirectory structure for SYS:SALES
    TEST
    DOC
```

If drive D: is mapped to SYS:SALES and you type:

SHOWDIR D:[Retrn]

the system will respond with:

```
Subdirectory structure for SYS:SALES
    TEST
    DOC
```

Purpose

Displays user and group trustees and their privileges for a specified subdirectory.

Command Format

TLIST [<pathspec> | <drive>:][Retrn]

If no pathspec or drive: is given, the current path is assumed.

Examples

If you type:

TLIST SYS:PUBLIC[Retrn]

you will see something similar to the following:

```
SYS:PUBLIC
User trustees:
CRAIG (Craig Burton) [RWOCDPSM]
JIM (James Bills)    [R O    S ]
     -----
No group trustees.
```

Purpose Performs global file searching.

Command Format UDIR [<pathspec>] [<filename>][Retrn]

Wildcards (* and ?) may be used in the pathspec or filename. A full or partial pathspec may be given.

Remarks UDIR is similar to the DOS DIR command except that it searches the directory identified by the pathspec--and its subdirectories--for files matching the given filename. If the matching files are found, the full path name of the directory being searched is displayed, followed by the list of files.

If no pathspec is given, the current directory path and its subdirectories will be searched. If no filename is given, all of the files in the pathspec and its subdirectories will be displayed with their respective path names.

Directories for which you have no search privileges will be ignored.

Directories in which no requested files are found will not be listed.

An important difference between UDIR and DOS DIR command is that UDIR allows you to search for files on volumes other than your current volume without causing your current drive mapping to be altered.

UDIR also allows you to use directory names of up to 14 characters, while DIR only allows 8 characters in names.

Examples If you are mapped to SYS: and you type:

UDIR \SALES[Retrn]

the system might respond with:

```
SYS:SALES
  TEST
  OEM.DOC
SYS:SALES\DOC
  TEST1
```

Type:

UDIR * T*.*[Retrn]

to search all volumes and subdirectories for any files beginning with a T.

Purpose Displays disk allocation statistics for all volumes on the default file server.

Command Format VOLINFO[Retrn]

Remarks The CHKDSK command provided with DOS won't give the proper statistics for an network volume. The VOLINFO command should be used instead.

When you issue the VOLINFO command, your screen will display information about the file server's volumes. Information displayed include each volume's total disk space (in kilobytes) and total number of directories, as well as free disk space and free (remaining) directories.

This display remains on your screen and is periodically updated until you cancel the program. The display is updated approximately every five seconds. You may type "+" to increase the time between volume checks, or "-" to decrease it. Typing any other key will cause you to exit.

Example Type:

VOLINFO[Retrn]

A display similar to the following will appear on your screen:

	Total	Free	Total	Free
Volume name	SYS		VOL1	
KiloBytes	57816	10832	17264	4172
Directories	4992	1541	1536	850
Volume name	SALES		DOC	
KiloBytes	123912	79996	46064	16296
Directories	3840	922	2560	843

Thursday, 28 Feb 85 12 26 58 pm

Checking ..

Press'+' to increase or '−' to decrease time between volume checks

Press any other key to exit

Whenever the update causes any statistic to be changed, an arrow appears to the right of the number, pointing up if the value has increased, or down if the value has decreased.

Purpose Changes the current directory of the specified or default drive, or displays the current directory path of a drive.

Command Format CHDIR [<d>:][<path>] or CD [<d>:][<path>][Retrn]

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

```
CHDIR \bin\user\joe\forms[Retrn]
```

DOS will put you in the new directory. A shorthand notation for the CHDIR command is also available:

```
CD ..[Retrn]
```

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

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

```
CD[Retrn]
```

DOS displays: b:\bin\user\joe

Using the CHDIR command in this way is useful when you need to know the name of your working directory.

The command:

```
CD C:[Retrn]
```

will display the working directory on drive C.

Purpose

Analyzes the directories, files, and the File Allocation Table on the designated or default drive and produces a disk and memory status report.

>> CAUTION <<
Do not use CHKDSK on network drives! North Star workstation users should only run CHKDSK on diskettes (drives A or B). External PC users should only run CHKDSK on local drives.

Command Format

CHKDSK [<drive:>] [<pathname>] [/f] [/v][Retrn]

Remarks

CHKDSK should be run occasionally on each disk to check for errors. If any errors are found, Chkdsk will display error messages and then a status report.

A sample diskette status report looks like this:

```
362496 bytes total disk space
348160 bytes in 10 user files
 14336 bytes available on disk

524288 bytes total memory
455424 bytes free
```

Note that in this status report, the last two lines refer to the workstation's memory status. You should, however, use the CHKMEM command (see Chapter 8) rather than CHKDSK to find out about available workstation memory because you can run CHKMEM from network drives.

Remarks (cont.)

CHKDSK will not correct any errors found in your directory unless you specify the /f (fix) switch. If you specify the /v switch, CHKDSK will display messages while it is running.

If you type a filename after CHKDSK, DOS displays a status report for the disk and for the individual file.

You can redirect the output from CHKDSK to a file, if you want to save the status report for future use. Simply type:

CHKDSK a:>filename[Retrn]

The errors will be sent to the filename specified. Do not use the /f switch if you redirect CHKDSK output.

/F Switch

The following errors will be corrected automatically if you specify the /F switch:

- Invalid drive specification
- Invalid parameter
- Invalid subdirectory entry
- Cannot CHDIR to file
  Tree past this point not processed
- First cluster number is invalid, entry truncated
- Allocation error, size adjusted
- Has invalid cluster, file truncated
- Disk error reading FAT
- Disk error writing FAT
- File contains non-contiguous blocks
- All specified file(s) are contiguous

Errors You must correct the following errors returned by CHKDSK, even if you specified the /F switch:

Error	Explanation
Insufficient memory Processing cannot continue	There is not enough memory in your workstation to process CHKDSK for this drive. You must obtain more memory to run CHKDSK.
Invalid current directory Processing cannot continue	Reset the workstation and re-run CHKDSK.
Cannot CHDIR to root Processing cannot continue	The diskette or drive you are checking is bad. Try resetting your workstation. If you are using a diskette, try recovering it with RECOVER.
File is cross-linked on cluster	Make a copy of the file you want to keep, and then delete both files that are cross-linked.
X lost clusters found in y chains Convert lost chains to files (Y/N)?	If you respond Y to this prompt, CHKDSK will create a directory entry and a file for you to resolve this problem (files created by CHKDSK are named FILEnnnn.CHK). CHKDSK will then display: X bytes disk space freed If you respond N to this prompt or have not specified the /F switch, CHKDSK frees the clusters and displays: X bytes disk space would be freed

Errors (cont.)

Error	Explanation
Probable non-DOS disk Continue (Y/N)?	The diskette you are using is a non-DOS diskette You must indicate whether or not you want CHKDSK to continue processing
Insufficient room in root directory Erase files in root and repeat CHKDSK	CHKDSK cannot process until you delete files in the root directory
Unrecoverable error in directory Convert directory to file (Y/N)?	If you respond Y to this prompt, CHKDSK will convert the bad directory into a file You can then fix the directory yourself or delete it

Purpose Lists the files in a directory.

Command Format DIR [<drive:>][<pathname>][/p][/w][Retrn]

Remarks If you just type DIR[Retrn], all directory entries on the default drive are listed. If you include a drive name, such as "dir b:", all entries on the disk in the specified drive are listed. If only a filename is entered with no extension ("dir filename"), then all files with the designated filename on the disk in the default drive are listed. When you type a filename with a drive letter (for example, "dir b:filename.ext"), all files with the filename specified on the disk in the drive specified are displayed. In all cases, files are listed with their size in bytes and with the time and date of their last modification. The wildcards ? and * (question mark and asterisk) may be used in the filename option. Note that the following DIR commands are equivalent:

COMMAND	EQUIVALENT
dir	dir *.*
dir filename	dir filename.*
dir .ext	dir *.ext

Two switches may be used with DIR. The /p switch selects Page Mode. With /p, display of the directory pauses after the screen is filled. To resume display of output, press any key.

The /w switch selects wide display. With /w, only filenames are displayed, without other file information. Files are displayed five per line.

Remarks (cont.)

Note that if the COUNTRY command in the CONFIG.SYS file is set to a country other than the U.S., the directory date and time formats will differ. Refer to Appendix D, "How to Configure Your System," for more information on the CONFIG.SYS file.

If you are logged into a network volume, a DIR of another volume will log you to that volume, but the volume name on your display will not change. You should use the UDIR command because UDIR allows you to search for files on volumes other than your current volume without altering your current drive mapping.

DIR will only show you the first eight characters of your directory names. LISTDIR will show you the full names. You cannot refer to a directory with a name longer than eight characters by the first eight characters; you must use the full directory name.

Purpose Makes new directories and subdirectories.

Command Format

MKDIR [<drive:>]<pathname>[Retrn]

or MD [<drive:>]<pathname>[Retrn]

Remarks This command creates a multilevel directory structure. When you are in your root directory, you can create subdirectories by using the MKDIR command. The command:

```
MKDIR \user
```

will create a subdirectory \USER in your root directory. To create a directory named JOE under \USER, type:

```
MD \user\joe
```

When you create directories with MKDIR, they always appear under your working directory unless you explicitly specify a different pathname with MKDIR.

Messages The message:

Unable to create directory

means that DOS could not create the directory you specified. Check to see that there is not a name conflict. You may have a file by the same name, or the disk may be full.

Purpose Sets a command search path.

Command Format

PATH [[<drive:>][<pathname>];[<drive:>]
[<pathname>]...][Retrn]

Remarks You should use the NPATH command to set your search paths on the network. The PATH command still functions, however.

This command allows you to tell DOS which directories should be searched for external commands after DOS searches your working directory. The default value is no path.

To tell DOS to search the \USER\JOE directory for external commands, type:

```
PATH \user\joe
```

DOS now searches the \USER\JOE directory for external commands until you set another path or exit DOS.

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

```
PATH \user\joe;b:\user\sue;\bin\dev
```

tells DOS to search the directories specified by the above pathnames to find external commands. DOS searches the pathnames in the order specified in the PATH command.

Remarks (cont.)

The command PATH with no options will print the current path. If you specify,

```
path ;
```

DOS will set the NUL path. This means that only the working directory will be searched for external commands.

Messages

The message:

```
No path
```

means you typed PATH with no options but have not set a command search path.

Purpose Removes a directory from a multilevel directory structure.

Command Format

rmdir <pathname>[Retrn]

or RD <pathname>[Retrn]

Remarks This command removes a directory that is empty (except for the . and .. shorthand symbols. You must delete all files in the directory first.

Example To remove the \USER\JOE directory, first issue a DIR command for that path to ensure that the directory is empty. Then type:

```
RMDIR \user\joe
```

The directory will be deleted from the directory structure.

MANAGING FILES

Overview

This chapter presents the NetWare and DOS commands that related to file management.

Be sure to review the Command Rules and Restrictions in the Preface to this manual before attempting to use the commands presented here.

File Commands

NetWare commands related to file management and covered in this chapter are:

- o FLAG
- o HIDEFILE
- o LARCHIVE
- o LRESTORE
- o NARCHIVE
- o NCOPY
- o NRESTORE
- o PURGE
- o SALVAGE
- o SHOWFILE

DOS commands related to file management and covered in this chapter are:

- o ATTRIB
- o COPY
- o DEL
- o FC (File Compare)
- o RECOVER
- o RENAME
- o RESTORE

Purpose

Displays or modifies a file's non-shareable read/write status.

Command Format

FLAG [path\filename [/flag[...]]][Retrn]

path\filename
[pathspec/]filename

flag
Normal
Shareable
Non-Shareable
Read/Write
Read Only

Wildcards (* and ?) may be used in the filespec. Full or relative path specifications may precede the filename.

The default attribute of a file on the network is normal (which means Non-Shareable, Read/Write).

Remarks

To modify the attributes of a file, the user's effective rights must include Modify privileges for the directory containing the file.

If a file is in use by some other workstation, no change is made.

Omitting all flag specifications causes FLAG to display the current status of the given file(s).

Examples

To set all of the files in your default directory to Shareable and Read Only status, execute the following:

FLAG *.* SHAREABLE READ ONLY[Retrn]

All of the files in your default directory that aren't in use will be set to Shareable, Read Only status.

Flags may also be abbreviated. For example, if you type:

FLAG SYS:PUBLIC\TEST.EXE SRO[Retrn]

The system will respond with:

```
SYS:PUBLIC
  TEST.EXE set to shareable read only
```

Note: It's best to leave files found in the PUBLIC subdirectory with the status of SRO.

Typing only:

FLAG[Retrn]

gives the status of all the files in the directory:

```
SYS:PUBLIC
  TEST.      is non-shareable read/write
  OEM.DOC    is non-shareable read/write
```

Purpose Marks a specified file as hidden so that it will not show in a directory search and cannot be deleted or copied over.

Command Format HIDEFILE<filename>[Retrn]

Remarks Hiding a file prevents it from being copied over by any COPY command issuing the same filename. The COPY command will return a **file creation error**. Some supervisor files are hidden for this reason. For example, the network user and directory information is hidden so that an out-of-date directory is not inadvertently loaded during a restore.

A hidden file will still be accessible to many network commands, such as the TYPE command.

Use the SHOWFILE command to again reveal files hidden by HIDEFILE.

Examples Suppose you type DIR[Retrn].

The system responds with:

```
Volume in drive D is SYS
Directory of D:\JIM
TEST            128  8-02-84  3:53p
OEM     DOC    2432  8-02-84  3:53p
DOC     <DIR>        8-01-84  3:23p
    3 File(s)  5566464 bytes free
```

Now type:

```
HIDEFILE TEST[Retrn]
```

Examples (cont.)

The system responds with:

```
SYS:SALES
File(s) TEST hidden
```

Now if you enter:

```
TYPE TEST[Retrn]
```

the system will type the TEST file. But if you again type:

```
DIR[Retrn]
```

You will see that TEST is not listed:

```
Volume in drive D is SYS
Directory of D:\JIM

OEM    DOC 2432    8-02-84  3:53p
DOC    <DIR>       8-01-84  3:23p
    2 File(s)  5566464 bytes free
```

Purpose

Archives network files to the shared diskette drive in the central module, or to local drives on an external PC workstation.

To restore archived data, use LRESTORE.

To archive files to network disks, use NARCHIVE.

Command Format

LARCHIVE [pathspec][Retrn]

pathspec is any path specification that is valid with NPATH. For example,

```
LARCHIVE D: [Retrn]
LARCHIVE SYS:CRAIG/OUTLINES [Retrn]
LARCHIVE E:../MARK/DEMO [Retrn]
LARCHIVE SYSTEM [Retrn] (backs up the entire system)
```

LARCHIVE without the pathspec assumes the path of your current default drive as the initial path specification. LARCHIVE only archives files from the given, or assumed, path specification and its subdirectories.

Pressing [Retrn] in response to a Y/N question from LARCHIVE is interpreted as a No.

To cancel LARCHIVE at any time, press [Ctrl]C.

Remarks

When LARCHIVE searches directories to select files to be archived, files marked as System or Hidden will be uncovered along with the regular files.

Certain special system files--temporary spooler files, NET$SPL.QUE, and NET$MSG.SYS--will always be ignored by LARCHIVE.

Remarks (cont.)

A log file is created or updated on each target disk used for archiving. This file--ARCHIVE.LOG--must not be erased or altered; it must be present to use the LRESTORE command.

You may only archive files to which you have Read, Search, Open, and Modify File access rights.

While archiving to local disks, you can switch between the different local drives by pressing [Esc] when prompted to prepare a disk for archiving.

Beginning LARCHIVE

After displaying a version and copyright notice, LARCHIVE prompts you to enter the drive specification of a local disk drive; this is the letter by which the drive is referenced, e.g., A or C. A local disk may be either a floppy diskette or a hard disk.

Next, LARCHIVE asks whether you wish to save directory rights information, trustee lists, and user group definitions. This information should be archived only when the Supervisor runs a system backup.

Then LARCHIVE asks if the specified directories are to be scanned without interruption during the archiving process:

- Yes--indicates that you want to establish rules for selecting files before LARCHIVE starts to scan. Questions will not be asked about each specific directory.

- No--means that you want to indicate selected directories of the entire tree for archiving and/or specify different rules for selecting files in each directory.

Beginning LARCHIVE (cont.)

If you answered Yes and selected Uninterrupted Archiving, you are given two options:

1) Backup ALL qualified files in each directory.

2) Backup ONLY qualified files that have been modified since the last time they were archived.

Qualifying files means you are asked if you want to select or ignore files on the basis of certain filename specifications. Answering Yes means you can qualify only files which match specifications such as *.BAS, MX??0411.*, or LEDGER.LOG. After the > prompt, you can enter up to 50 such specifications. The specifications must be separated by commas. End each line by pressing [Retrn], and end the whole list by pressing [Retrn] on an empty line.

If, instead, you want to ignore specific files, press [Retrn] after the first > prompt. Then enter a list of specifications--in the same manner listed above--which will determine which files to ignore; i.e., all files will qualify except those which match a specification in the list.

If you answered No you selected Directory-by-Directory Archiving. In this case, LARCHIVE stops to query you about each directory.

LARCHIVE Process

The complete path name of the directory is printed, and you will be prompted with the question:

Backup? (Y/N)

Answering No causes LARCHIVE to go on to the next directory.

Answering Yes gives you three options:

1) Backup ALL qualified files in each directory.

2) Backup ONLY qualified files that have been modified since the last time they were archived.

3) Choose specific files during directory scan.

If option 1 or 2 is selected, you may define file specifications to qualify files as explained for Uninterrupted Archiving. Archiving of appropriate files will then proceed as described in the next section.

If option 3 is selected, LARCHIVE lists all the files in the directory and queries

Backup? (Y/N)

for each one. Pressing [Retrn] for an implied No saves some time. Archiving starts after you have selected from all of the files in the directory or after 250 files have been selected.

LARCHIVE will present the above questions and options for each directory under consideration. This is recommended for spot archiving, as opposed to the uninterrupted method for higher-volume global archiving sessions.

LARCHIVE Process (cont.)

You will be prompted to insert a backup diskette into the local drive or ensure that the target disk is on-line, and press the space bar when ready to continue. To change the target disk drive, an ESC can be entered here; LARCHIVE will then prompt for a destination drive and continue. The target disk is examined to determine the format and the amount of free space. If you are asked if the target drive deals in IBM-format diskettes or Victor 9000 diskettes, and your target is a hard disk, you must answer No. Hard disk formats and directory sizes vary from vendor to vendor; thus LARCHIVE may take time to determine the nature of the drive and available space before proceeding.

Note when using an IBM-PC: You may archive single- or double-sided diskettes if the target drive can handle double-sided diskettes. If the drive only handles single-sided diskettes, you must be careful not to use a previously-formatted, double-sided diskette.

The LARCHIVE command then attempts to arrange the selected files so that a minimum of space is wasted on the backup medium.

The message **Archiving** is displayed. A single plus sign (+) is shown at each directory to indicate that the trustee list has been archived. It is followed by an asterisk (*) for each file which is being copied. If it wasn't possible to put all of the selected files on the target disk, you will be prompted for another diskette. Press the space bar to continue archiving. If fixed Winchester space has been used up, press [Esc] in response to the drive-ready prompt.

LARCHIVE then allows the user to choose an alternate local target drive. The new drive can be a different type than the last one used; you may freely switch from one type of Winchester to another or to diskette drives.

LARCHIVE Process (cont.)

The remainder of the selected files will be packed onto this new diskette. This process will continue until all the files in the list have been archived.

You may archive several files from different directories, and they may all fit on the same diskette. You won't need to replace the backup diskette until the program rejects it as being too full.

Once you have begun scanning the files in a subdirectory, those files should not be modified. The packing precalculations are based on the file sizes as they were found during the initial directory scan. If a selected file is enlarged after that, the diskette may become prematurely full. This will cause a write error and termination of the LARCHIVE session.

If a file is locked by the network or another user when you try to archive it, you will be given the option to skip the file or to try again.

Archiving Volumes

You can specify in the command line that an entire volume or set of volumes is to be archived. Invoking LARCHIVE with:

LARCHIVE <volume:> [Retrn]

causes all directories on the given volume to be considered during the archiving session. Invoking LARCHIVE with:

LARCHIVE SYSTEM [Retrn]

causes LARCHIVE to query you as to which on-line volumes should be considered during the archiving session.

Archiving Volumes (cont.)

All the other options and questions described above will be presented; only the range of directories being considered has been expanded. These options are very useful when archiving an entire system or a major portion of the system. For example, it is convenient to LARCHIVE SYSTEM and also specify that all *.LST, *.BAK, and *.TMP files are to be ignored; only significant files will be archived. By routing the files to a hard disk system and selecting uninterrupted mode, you can do a complete backup without annoying pauses and interruptions.

Segmenting Large Files

Sometimes you may need to archive a file so large that it takes more than one disk to contain it. LARCHIVE provides a facility to segment the file over as many disks as necessary. The segments will be pieced together using LRESTORE. This feature is only useful if the local target disk drive is a floppy diskette drive or other type of device with removable media.

A file will only be segmented if it is so large that it can't be entirely contained on a single empty diskette of the size currently in the drive.

You will be presented with three choices:

1) Insert a higher-capacity diskette.

   This allows you to replace a single-sided diskette with a blank, double-sided diskette (drive permitting).

2) Skip the file.

   The file won't be archived.

3) Proceed with the file split.

   LARCHIVE will segment the file.

Segmenting Large Files (cont.)

The first segment of the file must be written to a blank diskette. Subsequent segments may be placed on any disk with free space. We recommend that all segments of an oversize file be archived to empty diskettes for ease in cataloging.

Archive Log

The archive log is a file created on every disk which contains archived files. It explains:

- What files are on the diskette
- What network directory they came from
- Whether they were system or hidden files
- In the case of split files, the segment and byte range of the file so that split files can be properly restored

In the event that two or more files from different directory areas have the same name, the extensions of all but the first file to be archived on that diskette will be assigned a numeric index to make their names unique. This essential information is also recorded in the archive log file.

The log file should never be modified. It is in a highly formatted form, and every byte is accounted for in the archiving process. Any change to this file may make it impossible to recover some archived material.

Archive Log (cont.)

LARCHIVE can generate a printed log report summarizing the session. Before archiving begins you will be asked:

Do you want to print a log report?

Press [Retrn] to answer No, or answer Yes to print the report. LARCHIVE will ask if the report should be printed on a local printer (answer L or Local), or sent to the network's print spooler (answer N or Network). Pressing [Retrn] indicates the network spooler by default. If you select the network spooler, you will be prompted to select the printer number.

Next you will be asked to indicate the desired number of copies. Press [Retrn] for one copy, or enter the number of copies, up to 9.

The log will be printed when the archiving session is completed.

Purpose

Restores files which were LARCHIVEd onto local diskettes.

See the LARCHIVE command for more information.

To restore files archived to network disks, see NRESTORE.

Command Format

LRESTORE [Retrn]

Remarks

The sequence of operation is similar to LARCHIVE but much simpler. You will be prompted to enter the letter of the local drive you will be restoring from.

Next you will be asked if you wish to restore security information with the directories. This information should be restored only by the Supervisor performing a full system restore.

If you need to restore only specific directories, you will be prompted at the beginning of LRESTORE to enter those directory names. Otherwise, LRESTORE will examine the file ARCHIVE.LOG on the source diskette, which contains a list of the LARCHIVED directories. If you had archived many directories but only want to restore one or two, it would be easier to enter the specific names rather than select them from the entire list.

To restore files back into a certain directory, you must have CREATE, DELETE, WRITE, OPEN and SEARCH privileges for the target directory.

You will be asked whether the directories should be restored without interruption. If you answer No, each directory in the archive log will be listed with a prompt:

Restore? (Y/N)

Remarks (cont.)

You may skip LRESTORE for any directory by answering No. If you answer Yes to the prompt, you will be asked if you want to restore all the files recorded for that directory automatically, or choose which files are to be restored. (If you chose earlier to restore without interruption, all files will be restored.)

Once you have selected the method of restoring the directories, you will be asked to insert the archive diskette in the source drive you identified earlier. When the diskette is in the drive, press the space bar to continue.

Restoration then proceeds in the manner chosen. As LRESTORE attempts to copy the LARCHIVEd file back into the directory from which it was originally LARCHIVEd, it may find an existing file of the same name. LRESTORE will ask if it should replace the existing file with the LARCHIVEd version, or skip restoring that file.

File restoration will continue until the log file has been completely examined or you cancel the session with [Ctrl]C. You then have the option of ending the LRESTORE session or inserting a new archive diskette to restore additional files.

Remarks (cont.)

When restoring segments of a file which was split by LARCHIVE, the pieces will go together properly, regardless of the sequence in which the segments are restored. If part of the file to which a segment belongs already exists, you have three choices:

1) Skip the file.

2) Create a new file and restore the segment.

3) Restore the segment as part of the existing file.

The second option might be used the first time a segment is restored to erase the existing old file, and the third option should then be used for all remaining segments of the file being restored.

Certain system files won't be restored without additional verification. These are the files which contain the network configuration information with regard to user security, subdirectory structure, etc. A brief message will be displayed explaining the potentially disastrous results of restoring such a file.

Purpose

Archives network files to network disks.

To restore archived data, use NRESTORE.

To archive files to local diskettes use LARCHIVE.

Command Format

NARCHIVE [<pathspec>] [Retrn]

<pathspec> is any path specification that is valid with NPATH. For example,

```
NARCHIVE D:[Retrn]
NARCHIVE SYS:CRAIG\OUTLINES[Retrn]
NARCHIVE E:..\MARK\DEMO[Retrn]
NARCHIVE SYSTEM[Retrn] (backs up the entire system)
```

NARCHIVE without the pathspec assumes the path of your current default drive as the initial path specification. NARCHIVE only archives files from the given or assumed path specification and its subdirectories.

Pressing [Retrn] in response to a Y/N question from NARCHIVE is interpreted as a No.

You can cancel NARCHIVE at any time by pressing [Ctrl]C.

Remarks

When NARCHIVE searches directories to select files to be archived, files marked as System or Hidden will be shown along with the regular files.

Remarks (cont.)

Certain special system files--temporary spooler files, NET$SPL.QUE, and NET$MSG.SYS--will always be ignored by NARCHIVE.

A log file is created or updated in each target directory or target disk used for archiving. This file--ARCHIVE.LOG--must not be erased or altered; it must be present to use the NRESTORE command.

You may only archive files from directories to which you have Read, Search, Open, and Modify File access rights.

Beginning NARCHIVE

After displaying a version and copyright notice, NARCHIVE asks you to enter path specifications for a list of target directory areas. You will be prompted with a caret (>) and should enter a list of volumes to which the files should be archived. If only a volume name is specified, the complete path name will be assumed to be <volume>:ARCHIVE. The ARCHIVE directory is system-defined and common to all volumes, fixed or removable.

Up to 20 paths may be specified; they must be separated by a comma. Multiple lines can be used. To terminate the list press [Retrn] on a blank line.

The list of target paths will be used in order until the list is exhausted. If there are still more files to be archived, you will be prompted to enter a new list of target directories.

If a given target volume isn't mounted, NARCHIVE will pause to allow you to mount the volume, or you may choose to have NARCHIVE advance to the next target path in the list.

Beginning NARCHIVE (cont.)

Here are some examples of possible target path lists:

> A1, A2, A3, A4, A5
> WORK4:MARK\TEST
> BACKUP_VOLUME1, BACKUP_VOLUME2

After the list of target paths is entered, NARCHIVE asks if you want to save directory rights, trustee lists, and the system's user and group definitions. This information should only be archived when the Supervisor runs a complete system backup.

Then NARCHIVE asks if the specified directories are to be scanned without interruption during the archiving process:

- o Yes indicates that you want to establish the rules for selecting files before NARCHIVE starts to scan. Questions won't be asked about each specific directory.

- o No indicates that you want to archive only from selected directories of the entire tree and/or specify different rules for selecting files in each directory.

If you answered Yes to select Uninterrupted Archiving, you are given two options:

1) Backup ALL qualified files in each directory.

2) Backup ONLY qualified files that have been modified since the last time they were archived.

Beginning NARCHIVE (cont.)

Qualifying files means you are asked if you want to select or ignore files on the basis of certain filename specifications. Answering Yes means you can qualify only files which match any of the listed specifications such as *.BAS, MX??0411.*, or LEDGER.LOG. After the > prompt, you can enter up to 50 such specifications. The specifications must be separated by commas. End each line by pressing [Retrn], and end the whole list by pressing [Retrn] on an empty line.

If, instead, you want to ignore specific files, press [Retrn] after the first > prompt. Then enter a list of specifications--in the same manner listed above--which will determine which files to ignore; i.e., all files will qualify except those which match a specification in the list.

If you answered No and selected Directory-by-Directory Archiving, NARCHIVE queries you about each directory. The complete path name of the directory is printed, and you will be prompted with the question:

Backup? (Y/N)

Answering No causes NARCHIVE to go on to the next directory.

Answering Yes gives you three options:

1) Backup ALL qualified files in each directory.

2) Backup ONLY qualified files that have been modified since the last time they were archived.

3) Choose specific files during directory scan.

Beginning NARCHIVE (cont.)

If option 1 or 2 is selected, you may define file specifications to qualify files as explained for Uninterrupted Archiving. Archiving of appropriate files will then proceed as described in the next section.

If option 3 is selected, NARCHIVE lists all the files in the directory and queries:

Backup? (Y/N)

for each one. Pressing [Retrn] for an implied No saves some time. Archiving starts after you have selected from all of the files in the directory or after 250 files have been selected.

NARCHIVE will present the above questions and options for each directory under consideration. This is recommended for spot archiving, as opposed to the uninterrupted method for higher-volume global archiving sessions.

NARCHIVE Process

NARCHIVE proceeds in the mode you've chosen. Whenever a new target volume is successfully mounted and available, a message indicates that the target volume and path is about to be used for archiving.

The NARCHIVE command then attempts to arrange the selected files so that a minimum of space is wasted on the backup medium.

The message **Archiving** is displayed. A single plus sign (+) is shown at each directory to indicate that the trustee lists have been archived. It is followed by an asterisk (*) for each file which is being copied.

The remainder of the selected files will be packed onto the volume. This process will continue until all the files in the list have been archived.

NARCHIVE Process (cont.)

Once you have begun scanning the files in a subdirectory, those files should not be modified. The packing precalculations are based on the file sizes as they were found during the initial directory scan. Enlarging a selected file after that will cause a write error and termination of the ARCHIVE session.

If a file is locked by the network or another user when you try to archive it, you will be given the option to skip the file or to try again.

If a given volume isn't mounted, NARCHIVE will pause to allow you to mount the volume, or you may choose to have NARCHIVE advance to the next target path in the list.

Note: A warning is displayed with the volume assignment message asking that no creative or destructive file activity be allowed on the target volume during archiving, or the space-optimizing calculations made by NARCHIVE will no longer be accurate. A prematurely-filled volume will cause a fatal error during file copying.

Archiving Volumes

You can specify in the command line that an entire volume or set of volumes is to be archived. Invoking NARCHIVE with:

NARCHIVE <volume:> [Retrn]

causes all directories on the given volume to be considered during the archiving session.

Invoking NARCHIVE with:

NARCHIVE SYSTEM [Retrn]

causes NARCHIVE to query you as to which on-line volumes should be considered during the archiving session.

Archiving Volumes (cont.)

All the other options and questions described above will be presented; only the range of directories being considered has been expanded. These options are very useful when archiving an entire system or a major portion of the system. For example, it is convenient to NARCHIVE SYSTEM and also specify that all *.LST, *.BAK, and *.TMP files are to be ignored; this way only significant files will be archived. By routing the files to a removable system volume and selecting uninterrupted mode, you can do a complete backup without annoying pauses and interruptions.

Segmenting Large Files

Sometimes you may need to archive a file so large that the remaining space of the target volume cannot contain it. NARCHIVE provides a facility to segment the file. The segments may be pieced together using NRESTORE.

A file will only be segmented if no file to be archived from a given directory is small enough to fit on the remaining space of the target volume, and more than 10% of the volume is empty.

If this is the case, NARCHIVE will want to split the file, which it will identify along with three choices:

1) Advance to the next target volume.

   This indicates that you would rather not split any files, but would like NARCHIVE to advance to the next target volume. If the target volume list is empty, you'll be prompted for a new list.

2) Skip the file.

3) Proceed with the file split.

Archive Log The archive log is a file created on every disk which contains archived files. It explains:

- o What files are on the disk
- o What network directory they came from
- o Whether they were system or hidden files
- o In case of split files, the segment and byte range of each file so that they can be properly restored

In the event that two or more files from different directory areas have the same name, the extensions of all but the first file to be archived on that diskette will be assigned a numeric index to make their names unique. This essential information is also recorded in the log file.

The log file should never be modified. It is in a highly formatted form, and every byte is accounted for in the archiving process. Any change to this file may make it impossible to recover some archived material.

NARCHIVE can generate a printed log report summarizing the session. Before archiving begins you will be asked:

Do you want to print a log report?

Press [Retrn] to answer No, or answer Yes to print the report. If you answer Yes, NARCHIVE will ask if the report should be printed on a local printer (answer L or Local), or sent to the network's print spooler (answer N or Network). Pressing [Retrn] indicates the network spooler by default. You must provide the printer number.

Next you will be asked to indicate the desired number of copies. Press [Retrn] for one copy, or enter the number of copies, up to 9. The log will be printed following the archiving session.

Purpose Provides a fast way to copy files within network directories.

Command Format

```
NCOPY  [<drive:>][<pathspec>/]<filename>
    [TO]  [<drive:>][<pathspec>/][<filename>]
  [/VERIFY][Retrn]
```

The first filename indicates the source file (the file being copied).

If you do not specify a source filename, then either drive: or pathspec must be included to indicate the source directory, and NCOPY will copy every file in the source directory.

If a second (destination) filename is specified, it will be assigned to the newly created copy.

If you do not specify the destination filename, the copy will have the same name as the source file. In that case, a destination directory must be specified, either by drive: or pathspec. Otherwise you will receive a message saying, "Can't copy file to itself."

If the pathspec is not specified in either case, the default (current) directory path is assumed.

You may specify a full or partial pathspec as required.

Wildcard characters (* and ?) may be included in the filename, but not in the pathspec.

Remarks

The NCOPY command is nearly identical to the DOS Copy command except that it is considerably faster when copying files between network drives. (The DOS Copy command will transfer the target file from the network disk to the memory of the workstation, and then to the destination file. NCOPY won't transfer the target file to the memory of the workstation, but will write it directly from the memory of the file server to the destination file.) NCOPY is also more descriptive than the DOS COPY command.

New copies will receive the original dates and times of the source files.

If multiple files are being copied, a failure to copy one file will not abort the function.

Examples

Suppose you are copying the file SAMPLE.FIL from subdirectory OLD to subdirectory NEW on VOL1:. If you type:

```
NCOPY VOL1:OLD\SAMPLE.FIL VOL1:NEW\
  SAMPLE.FIL[Retrn]
```

the system will respond:

```
From VOL1:OLD
To   VOL1:NEW
SAMPLE.FIL    to    SAMPLE.FIL

1 File copied.
```

Note that the screen display lists the source directory, destination directory, source file, and destination file.

Examples (cont.)

Now if you are mapped to VOL1:NEW and you type:

```
NCOPY SAMPLE.FIL TO COPY.FIL[Retrn]
```

the system will respond:

```
From VOL1:NEW
To   VOL1:NEW
SAMPLE.FIL    to    COPY.FIL

1 File copied.
```

The following command will copy all the files from your default drive to the PUBLIC subdirectory on volume SYS, retaining the original file names:

```
NCOPY  *.*  SYS:PUBLIC[Retrn]
```

The same action could be accomplished with the command:

```
NCOPY  .  SYS:PUBLIC[Retrn]
```

The following command will copy all the files from the directory drive K: is mapped to the local diskette drive A:

```
NCOPY  K:*.*  TO  A:[Retrn]
```

Purpose

Restores files which were NARCHIVEd to network disks.

See the NARCHIVE command for more information.

To restore files archived to local diskettes, see LRESTORE.

Command Format

NRESTORE[Retrn]

Remarks

The sequence of operation is similar to NARCHIVE but much simpler.

You will be asked if you wish to restore security information with the directories. This information should be restored only by the Supervisor performing a full system restore.

If you need to restore only specific directories, you will be prompted at the beginning of NRESTORE to enter those directory names. Otherwise, NRESTORE will examine the file ARCHIVE.LOG in the source volume, which contains a list of the NARCHIVED directories. If you have archived many directories but only want to restore one or two, it would be easier to enter the specific names rather than select them from the entire list.

To restore files back into a certain directory, you must have Create, Delete, Write, Open and Search privileges for the target directory.

Remarks (cont.)

You will be asked whether the directories should be restored without interruption. If you answer No, each directory in the archive log will be listed with a prompt:

Restore? (Y/N)

You may skip restoring any files in the directory by answering No. If you answer Yes to the prompt, you will be asked if you want to restore all the files recorded for that directory automatically, or choose which files are to be restored. (If you chose earlier to restore without interruption, all files will be restored.)

Once you have selected the method of restoring the directories, you will be asked to enter the volume or volume:path containing the NARCHIVEd files. If you enter only the volume, the directory <volume>:ARCHIVE will be assumed.

Restoration then proceeds in the manner chosen. As NRESTORE attempts to copy the NARCHIVEd file back into the directory from which it was originally NARCHIVEd, it may find an existing file of the same name. NRESTORE will ask if it should replace the existing file with the NARCHIVEd version, or skip restoring that file.

File restoration will continue until the log file has been completely examined or you cancel the session with [Ctrl]C. You then have the option of ending the NRESTORE session or advancing to a new source volume to restore additional files.

Remarks (cont.)

When restoring segments of a file which was split by NARCHIVE, the pieces will go together properly, regardless of the sequence in which the segments are restored. If part of the file to which a segment belongs already exists, you have three choices:

1) Skip the file.

2) Create a new file and restore the segment.

3) Restore the segment as part of the existing file.

The second option might be used the first time a segment is restored to erase the existing old file, and the third option should then be used for all remaining segments of the file being restored.

Certain supervisory files won't be restored without additional verification. These are the files which contain the network configuration information with regard to user security, subdirectory structure, etc. A brief message will be displayed explaining the potentially disastrous results of restoring such a file.

Purpose Permanently erases files which have been marked for deletion by a previous erase.

Command Format PURGE[Retrn]

Remarks The effect of the PURGE command is limited to the volume and the workstation from which it is executed.

Executing any application or command which causes any file to be created or erased have the same effect as executing PURGE.

See the SALVAGE command for more information.

Example If you type:

PURGE[Retrn]

the system will respond:

All your recoverable erased files have been purged from the network.

Now the deleted files are beyond recovery by the SALVAGE command.

Purpose

Recovers files after they have been erased from network drives.

Command Format

SALVAGE[Retrn]

Remarks

After files have been erased, either by an ERASE command or by being overwritten by a file rename, you may recover the files by issuing the SALVAGE command. The effect of SALVAGE is limited to the workstation from which it is executed.

Under certain circumstances this command may not be able to SALVAGE erased files.

If a PURGE command, or any command that creates or erases a file, is invoked from the same workstation, in the same volume, before a SALVAGE command is issued, then the erased files are permanently deleted and cannot be recovered.

SALVAGE cannot recover files erased from a local drive.

See the PURGE command for more information.

Examples

Suppose you delete from directory SYS: a file named SAMPLE.FIL. Now if you execute a directory search for it, the system will display:

File not found

The file has been marked for deletion. This marked file will not display in directory searches, but it is not yet permanently deleted.

Examples (cont.)

If you find that by mistake you erased a file you actually need, then you may type:

SALVAGE[Retrn]

to recover the erased file, and the system will list the name of the file as it is recovered. (However, if a PURGE, CREATE, or ERASE command was invoked from the same workstation, in the same volume, before the SALVAGE, then the erased file is permanently deleted.)

In this case, the system will respond as follows:

```
Salvaging files on volume SYS:
SAMPLE.FIL   recovered.
```

Purpose Makes visible a file hidden by HIDEFILE.

Command Format SHOWFILE<filename>[Retrn]

Examples Refer to the example under the HIDEFILE command. Remember that the file TEST was hidden in that example. Now, type:

```
SHOWFILE TEST[Retrn]
```

and the system will respond:

```
SYS:SALES
File(s) TEST set to non-hidden non-system.
```

Now if you type:

```
DIR[Retrn]
```

you will see:

```
Volume in drive D is SYS
Directory of D:\SALES

TEST       128      8-02-84  3:53p
OEM     DOC2432 8-02-84  3:53p
DOC     <DIR>    8-01-84  3:23p
  3 File(s)  5566464 bytes free
```

Purpose

Sets or resets the read-only attribute of a file; displays the attributes of a file.

Command Format

ATTRIB[+r|-r] [<drive:>] <pathname>[Retrn]

Remarks

+R sets the read-only attribute of a file.

-R disables read-only mode.

[<drive:>][<pathname>] specifies the file you want to reference.

If an application opens a file with read and write permission, ATTRIB forces read-only mode to allow file sharing over a network.

Although this command works on NetWare files, you should use the NetWare FLAG command instead. You will need to use ATTRIB to write protect files on diskettes because FLAG only works on network drives

To display the attribute of a specific file, type:

ATTRIB <pathname> [Retrn]

The wildcard characters *.* can be used to print the attributes of all files on a specific drive.

Example

The following example makes the file named MYFILE.TXT read-only:

ATTRIB +r myfile.txt [Retrn]

Purpose

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

Command Format

COPY [<drive:>]<pathname>
[<drive:>][<pathname>]
[/v] [/a] [/b] [Retrn] (to copy)

COPY <pathname> + <pathname> ... <pathname>[Retrn]
(to append)

Copying Files

If the source and destination files are both in the working directory, you may use filenames, rather than complete pathnames.

If the second pathname is not given, the copy will be on the default drive and will have the same name as the original file (first pathname). If the original file is on the default drive and the second pathname is not specified, the COPY command will quit. (Copying files to themselves is not allowed.) DOS will display the error message:

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

The second option may take three forms:

1. If the second option is a drive name only, the original file is copied with the original filename to the designated drive. (For example, "copy memo.doc b:" will make a copy on drive B named MEMO.DOC.)

Copying Files (cont.)

2. If the second option is a filename only, the original file is copied to a file on the default drive with the filename specified. (For example, "copy memo.doc letter.doc" will make a copy of MEMO.DOC, name it LETTER.DOC, and place it on the default drive.)

3. If the second option includes a drive name, the original file is copied to a file on the drive specified. (For example, "copy memo.doc b:memo.doc" will make a copy of memo.doc on the default drive, name the copy MEMO.DOC, and place the copy on the disk in drive B.)

The /v switch causes DOS to verify that the sectors written on the destination disk are recorded properly. although recording errors rarely occur. This option causes the COPY command to run more slowly because DOS must check each entry recorded on the disk. An error is displayed if a write is not verified.

The /a and /b switches mean that the files being processed are ASCII or binary files. Each switch applies to the filename preceding it, and to all remaining filenames in the command, until another /a or /b is encountered.

The following discussion applies to the /a and /b switches:

When used with a source filename:

/a Causes the file to be treated as an ASCII (text) file. Data in the file is copied up to but not including the first end-of-file mark (in EDLIN, this is Control-Z). The remainder of the file is not copied.

/b Causes the entire file to be copied, including any end-of-file mark.

Examples

COPY memo.txt /a letter.txt[Retrn]

COPY report.asm /b report2.asm[Retrn]

When used with a designation filename:

COPY memo.txt letter.txt /a[Retrn]

/a Causes an end-of-file character to be added as the last character of the file.

COPY report.asm report2.asm /b

/b Causes no end-of-file character to be added.

When you are appending files, as discussed below, the default switch is always /a.

Appending Files

The COPY command also allows you to append files. Simply list any number of files as options to COPY, separated by +, then specify a destination file to send the combined files to.

For example:

COPY intro.rpt + body.rpt + b:sum.rpt report[Retrn]

This command combines files named INTRO.RPT, BODY.RPT, and SUM.RPT (on drive B) and places them in the file on the default drive called REPORT.

To combine several files using wildcards into one file, you could type:

COPY *.lst combin.prn[Retrn]

Appending Files (cont.)

The command above takes all files with a filename extension of .LST and combines them into a file named COMBIN.PRN.

In the following example, each file found matching *.LST, is combined with the corresponding .REF file. The result is a file with the same filename but with the extension FILE1.REF to form FILE1.PRN; then XYZ.LST with XYZ.REF to form XYZ.PRN; and so on.

```
COPY *.lst + *.ref *.prn
```

The following command combines all files matching *.LST, with files matching *.REF, into one file named COMBIN.PRN:

```
COPY *.lst + *.ref combin.prn
```

Note: Do not try to append files 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 ALL.LST could have already been destroyed.

COPY compares the filename 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 joining 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, and will not produce an error message.

Messages

When you see this message displayed:

Cannot do binary reads from a device

it means that the copy cannot be done in binary mode when you are copying from a device. Remove the /b switch or specify an ASCII copy with the /a switch.

When you see this message:

Content of destination lost before copy

it means that a file to be used as a source file has been overwritten prior to completion of the copy. For example,

```
COPY x + y y
```

destroys Y before it can be copied.

Purpose Deletes all files with the designated file specification.

Command Format

DEL [<drive:>] <pathname>[Retrn]
<u>or</u> ERASE [<drive:>] <pathname>[Retrn]

Remarks If the pathname is *.*, the prompt "Are you sure?" appears. If you type Y as a response, then all files on the disk are deleted.

Purpose

Compares the contents of two text files (a complex command used primarily by programmers).

Command Format

FC [/# /b /w /c] <filename1> <filename2>[Retrn]

Remarks

It is sometimes useful to compare files on your disk. If you have copied a file and later want to compare copies to see which one is current, you can use the MS-DOS File Comparison Utility (FC).

The File Comparison utility compares the contents of two files. The differences between the two files can be sent to the screen or to a third file. The files being compared must be text files.

The comparisons are made in one of two ways:

- On a line-by-line basis
- On a byte-by-byte basis

The line-by-line comparison isolates blocks of lines that are different between the two files and prints those blocks of lines. The byte-by-byte comparison displays the bytes that are different between the two files.

FC uses a large amount of memory as buffer storage space to hold the text files. If the text files are larger than available memory, FC will compare what can be loaded into the buffer space. If no match is found in the portions of the files in the buffer space, FC displays the message:

*** Files are different ***

Remarks (cont.)

For binary files larger than available memory, FC compares both files completely, overlaying the portion in memory with the next portion from disk. All differences are output in the same manner as those files that fit completely in memory.

FC matches the first file against the second and reports any differences between them. Both filenames can be pathnames. For example:

```
FC b:\user\joe\file1.txt \user\sue\file2.txt[Retrn]
```

FC takes FILE1.TXT in the \USER\JOE directory of disk drive B and compares it with FILE2.TXT in the \USER\SUE directory. Since no drive is specified for filename2, FC assumes that the \USER\SUE directory is on the disk in the default drive.

FC Switches

There are four switches that you can use with the File Comparison Utility:

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

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

(where xxxxxxxx is the relative address of the pair of bytes from the beginning of the file). Addresses start at 00000000; yy and zz are the mismatched bytes from file1 and file2, respectively. If one of the files contains less data than the other, then a message is printed out. For example, if file1 ends before file2, then FC displays:

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

FC Switches (cont.)

/# (# stands for a number from 1 to 9). This switch specifies the number of lines required to match for the files to be considered as matching again after a difference has been found. If this switch is not specified, it defaults to 3.

/w Causes FC to compress whitespace (tabs and spaces) during the comparison. Thus, multiple contiguous white spaces in any line will be considered as a single whitespace. Note that although FC compresses whites, it does not ignore them. The two exceptions are beginning and ending whites in a line, which are ignored. For example (note that an underscore represents a white):

```
_____More___data_to_be_found______
```

will match with:

```
More_data_to_be_found
```

and with:

```
________More_____data_to_be_____found______
```

but will not match with:

```
______Moredata_to_be_found
```

/c Causes the matching process to ignore the case of letters. All letters in the files are considered uppercase letters. For example,

```
Much_MORE_data_IS_NOT_FOUND
```

will match:

```
much_more_data_is_not_found
```

FC Switches (cont)

If both the /w and /c options are specified, then FC will compress whites and ignore case. For example,

```
____DATA_was_found_____
```

will match:

```
data_was_found
```

FC Reports

FC reports the differences between the two files you specify by displaying the first filename, followed by the lines that differ between the files, followed by the first line to match in both files. FC then displays the name of the second file followed by the lines that are different, followed by the first line that matches. The default for the number of lines to match between the files is 3. (If you want to change this default, specify the number of lines with the /# switch.) For example:

```
...
...

----------<filename1>
<difference>
<1st line to match file2 in file1>

----------<filename2>
<difference>
<1st line to match file1 in file2>

-----------------------------------------

...
...
```

FC will continue to list each difference.

FC Reports (cont.)

If there are too many differences (involving too many lines), the program will simply report that the files are different and stop.

If no matches are found after the first difference is found, FC will display:

*** Files are different ***

and will return to the default drive prompt (for example, A>).

The differences and matches between the two files you specify will be displayed on your screen unless you redirect the output to a file. This is accomplished in the same way as DOS command redirection.

To compare File1 and File2 and then send the FC output to DIFFER.TXT, type:

```
fc file1 file2 >differ.txt
```

The differences and matches between File1 and File2 will be put into DIFFER.TXT on the default drive.

Examples

Example 1:

Assume these two text files are on disk:

ALPHA.DOC	BETA.DOC
A	A
B	B
C	C
D	G
E	H
F	I
G	J
H	1
I	2
M	P
N	Q
O	R
P	S
Q	T
R	U
S	V
T	4
U	5
V	W
W	X
X	Y
Y	Z
Z	

To compare the two files and display the differences on the screen, type:

```
FC alpha.doc beta.doc[Retrn]
```

Examples (cont.)

FC compares ALPHA.DOC with BETA.DOC and displays the differences on the screen. The other defaults (Do not compress whites, and Do not ignore case) remain intact.

The output will appear as follows on the screen (the notes do not appear):

```
----------ALPHA.DOC
D                    NOTE: ALPHA file
E                    contains DEFG,
F                    BETA contains G.
G

----------BETA.DOC
G
--------------------------------

----------ALPHA.DOC
M                    NOTE: ALPHA file
N                    contains MNO where
O                    BETA contains J12.
P

----------BETA.DOC
J
1
2
P

--------------------------------

----------ALPHA.DOC
W                    NOTE: ALPHA file
                     contains W where
----------BETA.DOC   BETA contains 45W.
4
5
W
```

Examples (cont.)

Example 2:

You can print the differences on the line printer using the same two files. In this example, four successive lines must be the same to constitute a match. Type:

```
FC /4 alpha.doc beta.doc >prn[Retrn]
```

The following output will appear on the line printer:

```
----------ALPHA.DOC
D
E
F
G
H
I
M
N              NOTE: P is the 1st of
O              a string of 4 matches.
P

----------BETA.DOC
G
H
I
J
1
2
P
-------------------------------

----------ALPHA.DOC
W                              NOTE: W is the 1st of a
                               string of 4 matches.
----------BETA.DOC
4
5
W
```

Examples (cont.)

Example 3:

This example forces a binary comparison and then displays the differences on the screen using the same two files as were used in the previous examples. Type:

FC /b alpha.doc beta.doc[Retrn]

The /b switch in this example forces binary comparison. This switch and any others must be typed before the filenames in the FC command line. The following display should appear:

```
--ADDRS-----F1---F2--
00000009     44   47
0000000C     45   48
0000000F     46   49
00000012     47   4A
00000015     48   31
00000018     49   32
0000001B     4D   50
0000001E     4E   51
00000021     4F   52
00000024     50   53
00000027     51   54
0000002A     52   55
0000002D     53   56
00000030     54   34
00000033     55   35
00000036     56   57
00000039     57   58
0000003C     58   59
0000003F     59   5A
00000042     5A   1A
```

Error Messages

When FC detects an error, one or more of the following error messages will be displayed:

Bad file

means one of the files you specified is defective.

Data left in (filename)

means after reaching the end of one of the files in a file comparison, the other file still has uncompared data left.

File not found:<filename>

means FC could not find the filename you specified.

Incorrect DOS version

means you are running FC under a version of MS-DOS that is not 2.0 or higher.

Internal error

this message indicates an internal logic error in the FC utility.

Invalid number of parameters

means you have specified the wrong number of options on the FC command line.

Invalid parameter:<option>

means one of the switches that you have specified is invalid.

Read error in:<filename>

means FC could not read the entire file.

Purpose

Recovers a file or an entire disk containing bad sectors. Do not use this command on network drives. It may be used only on the shared diskette drive or on local diskette drives at external workstations.

Command Format

RECOVER [<drive:>][Retrn]
<u>or</u> RECOVER <drive:>[<pathname>][Retrn]

Remarks

If a sector on a disk is bad, you can recover either the file containing that sector (without the bad sector) or the entire disk (if the bad sector was in the directory).

To recover a particular file, type:

RECOVER <filename>[Retrn]

This will cause DOS to read the file sector by sector and to skip the bad sector(s). When DOS finds the bad sector(s), the sector(s) are marked and DOS will no longer allocate your data to that sector.

To recover a disk, type:

RECOVER <drive:>[Retrn]

where <drive:> is the letter of the drive containing the disk to be recovered.

Messages

When you see the message:

File not found

it means that DOS cannot find the file that you specified. Check to see that the pathname is accurate and that the file exists in the directory you specified.

Messages (cont.)

When you see:

(xxxx) of (xxxx) bytes recovered

this message tells you the number of bytes that DOS was able to recover from the disk.

When you see:

Warning - directory full

it means that the root directory is too full for Recover processing. Delete some files in the root directory to free space.

Purpose Changes the name of a file.

Command Format REN [<drive:>]<pathname> <pathname>[Retrn]

Remarks You must supply a drive name if the file resides on other than the default drive. Any drive name for the third option (pathname) is ignored. You cannot rename files across drives.

Wildcards may be used in either option. All files matching the first filename are renamed. If wildcards appear in the second filename, corresponding character positions will not be changed. For example, the following command changes the extension of all filenames ending in .LST to .PRN:

```
REN *.lst *.prn
```

In the next example, REN renames the file CHAP10 on drive B to PART10:

```
REN b:chap10 part10
```

The file remains on drive B.

Messages The message:

```
File not found
```

means you tried to rename a file to a name already present in the directory.

Purpose

Restores files backed up using the DOS or the IBM backup program. This command is necessary when transferring data files from a North Star system which was operating under North Star DOS but is converting to North Star NetWare. The procedure for this particular file transfer is explained in the North Star NetWare Supervisor's Manual

Use the LRESTORE or NRESTORE to restore network drives

Command Format

RESTORE <drive:>[<drive:>][<pathname>]
[/s][/p][/i][/n][Retrn]

Remarks

The first option you specify is the drive name for the disk containing the backed up files. The second and third options are the drive and pathname of the files you want to restore.

This restore program and the one supplied by IBM are compatible except for the extra switches described below.

The following switches are used with the DOS RESTORE command:

/s Restore subdirectories also (from the root).

/p If any hidden or read-only files match the filespecification, prompt for permission to restore them.

/i Restore files regardless of their origin. For example, to restore files to drive D which were backed up on drive C.

/n Restore files starting at the current directory instead of starting at the root. This option is specific to transferring files from North Star DOS to North Star NetWare.

Remarks (cont.)

The backup program sets the ERRORLEVEL in the following manner:

0 Normal completion
1 No files were found to back up
3 Terminated by user
4 Terminated due to error

USING THE SHARED DISKETTE DRIVE

Introduction

This chapter covers the external DOS commands required to use the shared diskette drive located in the central module.

The diskette drive in the central module is considered a local drive for North Star workstations, just as a diskette drive in an external workstation is a local drive; therefore, this diskette drive is only available to North Star workstations and cannot be accessed from an external workstation.

The commands are covered in the order listed below:

- o REQUEST
- o RELEASE
- o FORMAT
- o DISKCOPY
- o DISKCOMPARE
- o DISKBOOT

Following the command descriptions, instructions are provided for using the simulated second diskette drive.

Purpose

Enables you to use the diskette drive. For North Star workstations only.

Command Format

REQUEST D[Retrn]

Remarks

To request the diskette drive, type:

REQUEST D[Retrn]

If the diskette drive is already assigned, a message appears with the username and workstation number of the person using it.

Purpose Releases the use of the diskette drive. For North Star workstations only.

Command Format RELEASE D[Retrn]

Remarks To release the use of the diskette drive, type:

RELEASE D[Retrn]

Fragmentation

Diskettes that have been subjected to considerable file creation and deletion become more difficult to access as the links connecting data to files become fragmented rather than sequential. Since DISKCOPY produces an exact copy of a diskette, it also duplicates the fragmentation. To more efficiently rearrange data from a fragmented diskette, use the COPY command to create your copy from the source diskette.

Purpose Compares any two DIMENSION diskettes.

Command Format DISKCOMP[Retrn]

Remarks If the formats are different this will be reported and you will be prompted to compare two more diskettes or to exit.

If the diskettes are identical this will be reported and you will be prompted to compare two more diskettes or to exit.

If the diskettes are different the differences will be reported by sector and byte. Sector numbers run from 0 to number of sectors for that type of media, and byte numbers run from 0 through 511.

[Ctrl]C and [Ctrl][Num Lock] are active and can be used during the execution of this program.

If you wish to cancel the display of the locations of the differences between two diskettes that are substantially different without exiting this program, press the space barand the current compare will cancel. You will then be prompted to compare two more diskettes or to exit.

Purpose Runs programs that do not operate under DOS and that must be executed from a diskette. For North Star workstations only.

Command Format DISKBOOT[Retrn]

Remarks This command should be used with caution. If it is used with an operating system not compatible with North Star NetWare, it could halt the entire system, causing other users to lose data.

With this command, you can run applications which are not designed to run under DOS and which start and run from a diskette. You must first request the diskette drive using the REQUEST D command.

DISKBOOT restarts the workstation using the operating system on the diskette in the diskette drive. Since NetWare will no longer be running on that workstation, none of the normal commands will be available. In order to return the workstation to normal use, you must ask the system supervisor to clear your station from the console.

The diskette drive is automatically released if you press [ESC] to cancel the command before proceeding, or after the supervisor clears the station.

Introduction

Even though this system only contains one diskette drive, you can perform operations with DOS commands that require two diskettes. This is possible through the simulated second diskette drive, or drive B.

The diskette drive is usually called drive A. When an operation requires two diskette drives, the diskette drive can be called A or B, depending on whether the system is seeking the first or second diskette in the operation.

Copying Files

If you specify drive B when the "A" diskette was last used, you are prompted to insert the diskette for drive B. For example, suppose you want to copy a file from one actual diskette to another. After you have requested the diskette drive, insert the diskette containing the file you want to copy (this is the source diskette) in the drive and type:

COPY <filename> B:[Retrn]

DOS reads the file from the diskette and prompts you with:

Insert diskette for drive B
and strike any key when ready

Remove the "A" diskette and replace it with the "B" (target or destination) diskette and press any key. DOS copies the file, then displays:

1 File(s) copied

If you specify drive A when the "B" diskette was last used, you are prompted again to change diskettes. This time DOS prompts you to insert the "A" Diskette.

Copying Files (cont.)

The same procedure is used if a command is executed from a batch file. DOS prompts you to insert the appropriate diskette and press any key before it continues.

Default Diskette Drive

The letter displayed in the system prompt represents the default drive, not the last diskette used. If you had previously typed DIR B:[Retrn], DOS believes the "B" diskette is still in the drive. However, the prompt is still A>, because A is still the default drive. If you then type DIR[Retrn] with no drive letter, DOS prompts you for the "A" diskette because drive A is the default drive and you did not specify another drive with the DIR command.

PRINTING

Overview

This chapter explains the two types of printers that can be used on the system, and presents the NetWare and DOS commands that relate to printing.

Types of Printers

The system has two types of printers. Up to three shared printers are attached to the central module. One local printer can be attached to each North Star workstation. This chapter explains how to use both types, beginning with shared printers.

Print Commands

The commands related to printing are covered in the order listed below:

- o PRINTER
- o LOCAL
- o NPRINT
- o Q
- o SPOOL
- o ENDSPOOL
- o DOS PRINT

Printer Names

As many as three shared printers may be connected to your system's central module. Each shared printer has a number to identify it. When you log in, you are assigned a connection to printer 0; the printer to which you are connected is called your "current printer." The PRINTER command will allow you to change your current printer. To find the name of your current printer, type:

PRINTER[Retrn]

Some reasons why you might want to change current printers are:

- You want to use your local printer
- Your current printer is busy and you want to try another printer
- You want to use a different type of printer. Keep in mind that application programs are usually installed for one type of printer. Make sure the new printer is the type that will work with your application.

You can use PRINTER as often as you like. For example, suppose you want to print two files on different printers. You can send the first file to one printer, use PRINTER to connect to a new printer, and then print the second file. After you use PRINTER, your files will be printed on the new current printer until you use PRINTER again or log out. When you log in again, your current printer will be printer 0.

Printing

The usual way to print a file is through an application such as a word processing program. You can also use the NPRINT command, explained later in this chapter.

Printing (cont.)

Most applications have their own print commands. When you give the command, the application asks for the filename and possibly how the print-out should be formatted. When you're using a shared printer, your files are not always printed immediately. A copy of the file is first placed in a storage area of the hard disk, called a spool file. The system keeps track of who sent the file and which printer the file is destined for. Any number of users can send files at the same time; the files are released from the storage area and printed on a first-in, first-out basis for each printer. Meanwhile, you may continue working on other files.

Print List

The list of files waiting for a particular printer is its print list. If there are files ahead of yours in the list, your file waits until the others are printed. If there aren't, your file is printed right away. You can use the Q command (explained later in this chapter) to display print lists and also to delete files from the lists.

Setting Up the Printer

You may connect any serial device to your North Star workstation; a printer, modem or mouse is most common. Before you can use the device, you must give the system information about how the device works. Use the LOCAL command (explained later) to do this.

Assigning the Printer

If the workstation you are using has a local printer, and that printer has been defined (using the LOCAL command), you can select that printer by typing:

PRINTER /P=L[Retrn]

Then, all files will be printed on this printer until you select another printer or log out.

Printing

Like shared printers, local printers maintain a print list. You can send several files to the printer, and they will be printed one after the other. Unlike shared printers, however, local printers don't require you to wait until a copy of the file has been sent to the storage area. There is no storage area for a local printer. When a file's turn comes up in the print list, it is simply "read" from the drive that contains it and then printed.

When using a local printer you must use either an application's print command, or the PRINT command. Do not use NPRINT to send files to a local printer.

There are two additional rules to remember with respect to local printers:

- Do not edit a file once you've sent it to the printer until printing has completely stopped.

- Do not log out until all files are printed. If you do, the print list will be deleted and the files won't be printed (to eliminate the list, do log out).

Purpose

Allows you to change your current shared printer to another shared printer or a local printer. You can also use it to set the default form, specify tab expansions and form feed options.

Command Format

PRINTER [[<printer name>][/<flag>/<flag>...]] [Retrn]

Remarks

If you simply type PRINTER[Retrn] the screen displays the command syntax and your current printer settings.

Flags

The PRINTER command has a number of flags. These flags or options can be set by typing the command format as described below. Flags may be set in series, separated by slashes.

/P

The /P (printer) flag selects your current printer. Printer names are 0, 1, 2, and [L]ocal. The default printer name is 0. The command format is :

PRINTER /P=<0|1|2|L>[Retrn]

/F

The /F (form) flag sets the default form for all files printed under your username. Form 0 is the default if no other form number is specified. The command format is:

PRINTER /F=<0-255>[Retrn]

The system supervisor uses the form numbers to define particular printing characteristics such as different color ribbons or paper, different paper types or sizes, or different character fonts. Ask your system supervisor for a list of the form definitions used on your system.

/F (cont.) As files are printed, the system checks the form specification. It halts the printer when it recognizes a form designation that is different from the one immediately preceding, except in the case of form 255, which will always assume a "go" condition.

/T The /T (tab) flag controls the number of spaces between tabs. Many applications, such as word processing, already do this for you. If you don't specify, the system will expand the tabs to eight spaces apart. The command format is:

PRINTER /T=<1-18>[Retrn]

/W The /W (wait) flag allows you to specify the number of seconds (from 2 to 255) that the system will wait before assuming that the file is completely copied to the print storage area. (The default is automatically set at 5 seconds.) When the file is determined to be complete, the file is listed in the print list. The command format is:

PRINTER /W=<2-255>[Retrn]

/B The /B (banner) flag allows you to enter up to twelve characters to be printed at the top of your files. The default banner is the username plus the filename. This switch allows you to specify a different name for the filename portion of the banner. If /B is used, all files in the path/file list will have the same banner name. The command format is:

PRINTER /B=<1-12 character filename>[Retrn]

/C

The /C (Copies) flag allows you to select the number of copies to be printed of each file in the print list. The default is 1. The command format is:

PRINTER /C=<0-255>[Retrn]

/NT

The /NT (No Tabs) flag cancels all tab expansion. Tab characters will be passed directly to the printer. The command format is:

PRINTER /NT[Retrn]

/NB

The /NB (No Banner) flag allows you to cancel the banner page for all files in the path/file list. The command format is:

PRINTER /NB[Retrn]

/I Switch

The /I (initialize) switch allows you to return the printer settings to the default settings. This switch is useful if you have changed settings for a special print job and wish to return to your standard settings without respecifying all options. The command format is:

PRINTER /I[Retrn]

/NE

The /NE (No Eject) flag allows you to turn off the automatic form feed between consecutive print jobs. The command format is:

PRINTER /NE[Retrn]

You cancel this flag by initializing the printer settings using the /I switch.

Purpose

Configures the protocol for the local serial port to respond correctly to signals from any serial device. Use this command only on North Star workstations. External workstations should use the MODE command described in their operating system manuals.

Command Format

See below.

Remarks

Before you can use a printer, modem or mouse connected to your workstation, you must supply some "configuration" information about it to the system using the LOCAL command.

The five items of information needed by LOCAL are explained briefly in this section. If there is anything you don't understand, consult your system supervisor and the printer/modem/or mouse manual before you begin.

Each item has a default value, which is the value assumed by LOCAL if you don't enter a new one.

Chart

Item	Explanation
Sending Baud Rate	The rate at which data should be *sent* to the printer or modem (or other receiving device) from any program you'll use it with. If you only have a mouse, you don't need this value. The choices are: 110 1200 150 2400 300 4800 600 9600 The default value is 2400 baud.

Chart (cont.)

Item	Explanation
Receiving Baud Rate	The rate at which data will be *received* by a program from the modem or mouse (or other sending device) you're using it with If you only have a printer, you don't need this value. The choices and the default value are the same as for the sending baud rate
Parity	Identifies the method of checking the integrity of the data that is sent or received. The choices are "odd," "even" or "none," and the default value is "none "
Data Word Length	Data is typically transmitted between the device and the workstation in chunks, or "data words," which are either 7 bits or 8 bits in length (The *bit* is the unit of measurement just below the byte.) The default length is 8 bits
Stop Bits	Stop bits relate to the synchronization of data being sent and received. The choice is 2, if the sending baud rate is 110, or 1, if the sending baud rate is *not* 110 The default value is 1.

Rules

You can enter only one value for each item. Therefore, these rules apply:

- o If you have a printer and a modem, they must have the same sending baud rate, parity, data word length and stop bits.
- o If you have a printer and a mouse, they must have the same parity, data word length and stop bits.
- o If you have a modem and a mouse, they must have the same receiving baud rate, parity, data word length and stop bits.

If the two devices don't have the same values, you can't use them without reconfiguring. You must configure with LOCAL for one device, then reconfigure when you want to use the other.

Ways to Use

There are two ways to use LOCAL. The first is to type:

LOCAL [Retrn]

LOCAL then prompts you for each item of information in the order shown in the previous chart. It normally displays the default value for each one. You can press [Retrn] to accept the default value or type a different value over it. When you finish the stop bits, press [Retrn].

Ways to Use (cont.)

If you invoke LOCAL again before you sign off, the values you entered the previous time will be displayed instead of the default values. The second way to use LOCAL is to enter the values all at once on the command line without waiting for the prompts. The order is the same: sending baud rate, receiving baud rate, parity, data word length and stop bits. For example:

LOCAL 1200 1200 none 8 1[Retrn]

Type a space between each value. If the sending and receiving baud rates are the same, or if you have only one device connected to the workstation, you can type only one value for the baud rate. LOCAL will then set both rates to that value. You also don't have to type the parity, data word length or stop bits if you want their default values.

If you make a mistake when using this method, LOCAL prompts you for the values again in order.

If you change the device connected at your workstation, use LOCAL again to configure the new device.

The values you set with LOCAL will remain in effect only until you log out or use LOCAL again, unless you include a LOCAL command line (see previous example) in an AUTOEXEC.BAT file. Then the values will be set automatically each time you log in. Refer to Chapter 10 for information about writing an AUTOEXEC.BAT file.

Purpose

Queues files for printing by a network--shared-- printer (makes a copy of each file in the print storage area and creates the print list).

Command Format

NPRINT [<pathspec>/] <filename> [,...] [</flag list>] [Retrn]

<filename> may include wildcard characters (* and ?).

</flag list>
</flag></flag...></flag>

<flag>
Tabs = A number from 0 to 18
Copies = A number from 0 to 255
Forms = A number from 0 to 255
Printer = A number from 0 to 2
Delete
Banner = ASCII string
No Tabs
No Banner

The default flags are:
/Tabs = 8 /Copies = 1 /Forms = 0 /Printer = 0
/Banner = filename

You can use a blank space or a slash character (/) as a delimiter between flags in the flaglist. There must be a blank space between the end of the <filename> and the beginning of the <flag list>.

The equal sign (=) is optional.

Definitions

The following is a brief description of what each flag represents:

Tabs = n
: When a tab character (ASCII Control/I) is encountered in a file, it is expanded to enough spaces to bring the print position to the next tab position (every n columns).

Copies = n
: This flag allows you to print multiple copies of the given file.

Forms = n
: This flag is designed for the environment where different types of forms are required for specific jobs. When a job is queued to be printed, the forms type is cross-checked with the currently assigned forms type for that printer. If the forms types don't match, a request is displayed on the system console to mount the proper form and restart the printer.

Printer = n
: This flag allows you to direct printing to a particular printer.

Delete
: The file, after being printed, is deleted from the network.

Banner = ASCII string (up to 12 characters)
: The banner name default is the user name plus the file name. This command allows you to specify a different name for the filename portion of the banner page. If banner is used, all files in the path/filename list will be printed with the same banner name.

Definitions (cont.)

No Tabs	This flag allows you to cancel all tab expansion. Tab characters will then be passed directly to the spooled device (shared printer).
No Banner	This flag allows you to cancel the banner page for all files in the path/file list.

Remarks

The network allows several users to share one or more network printers through what is called a print spooler.

(The DOS PRINT command only allows you to specify one file, and does not allow files to be queued to the printer. Printing using this command (PRINT) is normally done with a local printer, meaning a printer connected directly to your workstation.)

The spooler may be selected in two ways, with the NPRINT command or the SPOOL command (SPOOL is only for external PC workstations). You may queue one or more existing files to the spooler by indicating the file(s) in the path/filename list.

For example,

NPRINT JUNK,SYS:SALES/TEST/MYFILE[Retrn]

attempts to locate the two files indicated and put them in the queue for printer 0. If other print jobs are already in the queue, the new jobs are placed in the queue directly following the existing jobs. Files are printed in first-come, first-serve order.

Remarks (cont.)

An optional flag list can be associated with the NPRINT command. The flag indicators are very flexible and can be abbreviated by the experienced user. For example:

NPRINT MYFILE,JUNK /Copies=2/No Banner[Retrn]

could also be written as:

NPRINT MYFILE,JUNK C2 NB[Retrn]

This command prints two copies each of the files MYFILE and JUNK without a banner page.

Sometimes, the file you want to print doesn't exist yet, or the way you want it printed is created by a word processor only at print time. This case requires that the file be sent directly to a spool queue file that will be stored on the disk and subsequently placed in a print queue. To redirect any printouts which would normally go to a local printer to a shareable printer, use the SPOOL command. For example,

SPOOL /NO BANNER/PRINTER 1[Retrn]

sets the local LST: device in capture mode with the indicated flags activated. See the SPOOL command for further information.

Purpose

Displays network print spooler information, including a list of files in the queue. Also allows for deletion of unwanted entries from the print queue.

Command Format

Q [<entry>] [</flag list>][Retrn]

An <entry> may be the job number (#19, #76, etc.) as assigned by the spooler, or a filename (LST.001, MYFILE, etc.). Wildcard characters (* and ?) may be used to specify <entry>.

</flag list>
</flag></flag...></flag>

<flag>

Printer	=	n
Station	=	n
User	=	user name
Delete		
No List		

Each entry in the flag list is separated by the slash (/) character or a blank space. The first flag must be preceded by a slash. Flags can be abbreviated as indicated by the bold characters above.

Definitions

The following are brief descriptions of the Q flag list options.

Printer = n
: This flag allows you to view or select a specific printer queue. This defaults to print queue 0.

Station = n
: This flag allows you to view or select print jobs by the number of the station that made the request.

User = user name
: This flag allows you to view or select print jobs by the login name of the user who made the print request.

Delete
: This flag allows you to delete queue entries as selected by filename, job number, user name or station number. Only the print request in the queue is deleted, not the original file.

No List
: This flag allows you to view the status of a particular printer without displaying a detailed list of entries; only the number of entries will be displayed.

If the /Delete and /No List flags are given in the same command, only a summary of how many jobs deleted will be displayed.

Remarks

The ability to send files to a print spooler creates the need to examine these queues. The system console controls the spool devices. From a workstation, a user may only view a queue, and delete entries which that user placed in the print queue. This is done with the Q command.

Remarks (cont.)

The Q command displays spool information for a specific printer. This information includes the printer number, whether the printer is on-line and running, and the forms type currently selected.

Then, for each queue entry, or "job," the Q command displays the job sequence, job number, banner name, number of copies, form number, flags, station number, user name, and time queued.

Example

Suppose you use the NPRINT command to send the files TEST.1, TEST.2, TEST.3, and TEST.4 to printer 2 from station 17.

Now you can view the queue display by entering:

Q /P=2 S=17 [Retrn]

Your screen will display information about entries queued to printer 2 from station 17. If many entries are listed, you could view the status of only the TEST.* files by entering:

Q TEST.* /P=2 S=17 [Retrn]

Only information about the TEST.* files is displayed. You can then stop the files from printing by entering the following Q command with the delete flag:

Q TEST.* /P=2 S=17 D [Retrn]

To view all of the jobs that end with .EXE and start with the letter R, on printer 0, enter the following:

Q R*.EXE [Retrn]

Purpose

Reroutes the LST: device to the network print spooler (shareable printer). You do not need to use this command if you are using a North Star workstation. You may use this command (in conjunction with ENDSPOOL) to queue to the printer large jobs requiring special forms, so that you do not tie up a printer until a time that is convenient for most users. You must use this command to send files to shared printers from external PC workstations.

Command Format

SPOOL [</flag list>] [Retrn]

</flag list>
</flag></flag...></flag>

<flag> may be as follows:

Tabs	=	A number from 0 to 18.
Copies	=	A number from 0 to 255.
Forms	=	A number from 0 to 255.
Printer	=	A number from 0 to 2.
Banner	=	ASCII string
Create	=	pathspec/filename
No Tabs		
No Banner		

Flags may be abbreviated as indicated by the bold characters above.

Default flag list
/Tabs = 8 /Copies = 1 /Forms = 0 /Printer = 0
/Banner = LST:

You can use a blank space or a slash character (/) as a delimiter between flags. However, the first slash of the flag list is required.

The equal sign (=) is optional.

Definitions The following are brief descriptions of what each flag represents:

Tabs = n
When a tab character (ASCII Control/I) is encountered in a file, it is expanded to enough spaces to bring the print position to the next tab position (every n columns).

Copies = n
This flag allows you to print multiple copies of data about to be spooled. When creating disk files with the /Create option, you may specify 0 copies if you don't want it printed.

Forms = n
This flag is designed for the environment where different types of forms are required for specific jobs. When a job is queued to be printed, the forms type is cross-checked with the currently assigned forms type for that printer. If the forms types don't match, a message is given at the console to mount the proper form and restart the printer.

Printer = n
This flag allows you to direct spooling to a particular printer.

Banner = ASCII string (up to 12 characters)
The banner name default is the user name plus "LST:." This command allows you to specify a filename instead of LST: on the banner page.

Definitions (cont.)

Create = path/filename

This flag allows you to specify a file in which the spooler will place the data it receives. The banner name, by default, will be the name of the Create file. Specifying /Copies = 0 causes the data to be spooled to the file but not printed.

IMPORTANT: The path/filename must either be the last item on the command line or be separated from subsequent flags by at least one blank in order to be valid.

No Tabs — This flag allows you to cancel all tab expansion.

No Banner — This flag allows you to cancel the banner page.

Remarks

The network allows several users to share one or more printers through what is called the network print spooler. The spooler is selected in two ways, with the NPRINT command or the SPOOL command. Use the SPOOL command when the file you want to print doesn't exist yet, or the printed form is created by a word processor at print time.

Executing the SPOOL command sets the workstation's local LST: device in capture mode. This means that any printout that would normally go to a local printer is instead sent to a file server spool queue file, according to the flag list.

Remarks (cont.)

After executing SPOOL, you may run the application that outputs the data to the LST: device. Instead of going to the local printer, the output is placed into a temporary "bucket" (spool queue file or storage area). This bucket remains open until it is closed by an ENDSPOOL command, an NPRINT command, or another SPOOL command (which also opens another bucket). Once the bucket is closed, it is placed at the end of the print queue. It is printed when the preceding buckets, or "jobs," in the queue have been printed or removed.

Each SPOOL command opens only one job no matter how many applications send printouts to it.

For example, if a SPOOL command is issued and then two files are printed using a word processor, both files will be entered into the same job. When the job is closed, it will enter the queue, where it will be listed as one job, although it contains the printouts of both files printed by the word processor. See the ENDSPOOL command for details on terminating or canceling a job.

Example

Enter the following command:

SPOOL /NB [Retrn]

Now press [PrtSc] to generate a printout of your workstation screen.

Next enter:

ENDSPOOL [Retrn]

A printout of your workstation's screen will be spooled to print, without a banner, at network printer 0.

Examples (cont.)

As another example, suppose you enter the following command:

SPOOL /COPIES=2/CREATE SYS:SALES/MYFILE[Retrn]

The system responds:

Device LST: re-routed to spooler #0

Now if you run an application which generates printed output, the output data will not print immediately, but will be saved in the capture file. After you execute an ENDSPOOL command or another SPOOL, the job containing the printout will be sent to the proper queue. In this case, it will be queued to network printer 0, and two copies will be printed. It will also be saved in the file MYFILE in the directory FS1/SYS:SALES.

Purpose

Terminates the workstation's LST: device capture mode and place the captured file in the spool queue (i.e., print spool mode). (See the SPOOL command). This command must be used to print files from external PC workstations on the network shared printers.

Command Format

ENDSPOOL [/flag][Retrn]

flag is one of the following (the slash is optional):

Abort
Cancel
Stop

Remarks

The ENDSPOOL command performs the following:

- o Closes the capture file opened by a previous SPOOL command at the same workstation
- o Queues the job to be printed by the specified printer
- o Resets the LST: device to the local printer

If the LST: device isn't in print spool mode, then no change takes place.

Any of the options used will simply have the effect of canceling the queue entry created while in capture mode.

The LOGIN, LOGOUT, SPOOL and NPRINT commands all carry an implied ENDSPOOL.

If the workstation is cold-booted, anything that was in the spool file will be properly closed out and printed when the internetwork shell is reloaded, minus the data that was still in the memory of your workstation.

Example

Suppose you execute SPOOL at your workstation, and then perform a Print Screen. Now when you enter:

ENDSPOOL[Retrn]

the output will be queued to an internetwork printer, and your screen will display:

Device LST: set to local mode.

If you had typed:

ENDSPOOL /CANCEL[Retrn]

the printout would have been cancelled, and the system would have responded:

Device LST: set to local mode
Data spooled since last SPOOL command has been discarded.

Purpose

Prints a text file on a line printer while you are processing other DOS commands (usually called "background printing"). North Star workstation users should not need to use this command, except to print to a local device. If a user of an external PC workstation wants to use this command, he or she should first run the EOJOFF command to disable the End Of Job indication in the NetWare Shell to ensure that PRINT will work properly. Be sure to run EOJON after printing is complete. (Ask your system supervisor for more information about the EOJOFF and EOJON commands.)

Command Format

PRINT [<drive:>][<pathname>][[/d:<device>]
[/b:<size>][/u:<value>][/m:<value>]
[/s:<value>][/q:<value>][/t][/c][/p]

Remarks

You will use the PRINT command only if you have a line printer attached to your workstation. The following switches are provided with this command:

/d DEVICE. Specifies the print device. If not specified, the default device is PRN. PRINT will prompt for a print device.

The following switches are allowed only the first time you run the PRINT command after starting DOS.

/b BYTES. This switch sets the size in bytes of the internal buffer. Increasing the value of /b speeds up the PRINT command.

/u BUSY TICKS. Specifies the number of computer clock ticks that print will wait for until the printer is available. If PRINT waits longer than this value, it gives up its timeslice (see /s switch, below).

Remarks (cont.)

/m MAX TICKS. Specifies how many computer clock ticks PRINT can have to print a file. The default value is two (2) clock ticks. The value of /m can be from 1 to 255 clock ticks.

/s TIME SLICE. Specifies the time slice value. The default is eight.

/q QUEUE. Specifies the number of files allowed in the print queue if you want more than 10. The minimum value for the /q switch is 4; the maximum is 32

/t TERMINATE. This switch deletes all files in the print queue (those waiting to be printed). A message to this effect will be printed.

/c CANCEL. This switch turns on cancel mode. The preceding filename and all following filenames will be removed from the print queue.

Note: If there are no files in the queue to match the canceled filename, no error message will appear.

/p PRINT. This switch turns on print mode. The preceding filename and all following filenames will be added to the print queue.

PRINT, when used with no options, displays the contents of the print queue on your screen without affecting the queue.

Examples

The command:

```
PRINT /t[Retrn]
```

empties the print queue.

Examples (cont.)

The command:

PRINT temp1.tst /c temp2.tst temp3.tst[Retrn]

removes the three files indicated from the print queue.

The command:

PRINT temp1.tst /c temp2.tst /p temp3.tst[Retrn]

removes TEMP1.TST from the queue, and adds TEMP2.TST and TEMP3.TST to the queue.

Messages

The message:

All files canceled by operator

means that you specified the /t switch.

The message:

File canceled by operator

Means you used the /c switch to cancel the file currently being printed.

The message:

Cannot open (filename)

means either DOS can't find the specified file to print, or the file does not exist. Check for a valid filename.

Errors on list device indicate that it may be offline. Please check it.

means that your printer is turned off.

Messages (cont.)

The message:

(filename) file not found

means that you switched disks when a file was queued up, but before it started to print. Reissue the PRINT command for that file.

The message:

List output is not assigned to a device

is displayed if the "Name of list device" specified to the above prompt is invalid. Subsequent attempts will return the same message until a valid device is specified.

The prompt:

Name of list device [PRN:]

appears when PRINT is run the first time and the /d switch is not specified. Any current device may be specified, and that device then becomes the PRINT output device. Press [Retrn] to use the default list device, PRN.

The message:

No files match drive:xxxxxxxx.xxx

means that no files match the given drive and filename.

The message:

Print queue is empty

means there are no files in the print queue.

Messages (cont.)

If you see this message:

Print queue is full

it means you have attempted to put more than 10 files in the queue. There is room for only 10 files. To put more files in the queue, refer to the /q switch in the list above.

The message:

Resident part of PRINT installed

means that available memory has been reduced to several thousand bytes to process the PRINT command along with other processes.

OTHER DOS COMMANDS

Overview

The commands presented in this chapter are those DOS commands which are not used specifically in directory management, file management or printing, but which may prove useful in other ways. Some commands, such as CTTY, are for special applications, but simple commands such as CURSOR or PROMPT may be used to customize your video display.

The commands covered in this chapter are:

- o BREAK
- o CHKMEM
- o CLS
- o COMMAND
- o CTTY
- o CURSOR
- o DATE
- o EXIT
- o FIND
- o MORE
- o PROMPT
- o SET
- o SORT
- o TIME
- o TYPE
- o VER
- o VERIFY
- o VOL

All these DOS commands are available to North Star workstations on the network. The external commands (CURSOR, FIND, MORE, and SORT) may not function properly on external workstations running MS-DOS version 3.0 or lower.

Purpose Changes the size of the cursor.

Command Format CURSOR <1 to 8> <1 to 9>[Retrn]

Remarks The first number of the command determines the location of the top of the cursor relative to the space that a capital letter occupies. The second number specifies the location of the bottom of the cursor.

The top of a capital letter is represented by 1. The bottom of a capital letter is represented by 8.

Example CURSOR 4 8[Retrn] yields a cursor one half the height of a capital letter.

CURSOR 7 8[Retrn] yields the default cursor.

Specifying 9 as the bottom of the cursor gives a full-size, full intensity cursor regardless of the number entered for the top of the cursor.

CURSOR 1 9[Retrn] yields the full-size cursor.

Purpose

Displays the date known to the system. This date will be recorded in the directory for any files you create or change.

Under NetWare, the SYSTIME command (see Chapter 2) displays the system time, date, and day of the week.

Command Format

DATE [<mm>-<dd>-<yy>][Retrn]

Remarks

If you type:

DATE[Retrn]

DATE will respond with the message:

Current date is <weekday> <mm>-<dd>-<yy>

Press [Retrn] to return to the prompt.

You cannot change the system date. The system supervisor uses a special command to set the system time and date at installation.

It is possible to change the mm-dd-yy format in which the date is displayed and entered. The Country command in the CONFIG.SYS file allows you to change the date format to the European standard dd-mm-yy. Refer to Appendix D, "How to Configure Your System," for more information on the CONFIG.SYS file.

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

Command Format EXIT[Retrn]

Remarks This command can be used when you are running an application program and want to start the DOS command processor, then return to your program. For example, to look at a directory on drive B while running an application program, you must issue an EXEC of the command interpreter (system call 4BH). The system prompt will appear. You can now type the DIR command and DOS will display the directory listing. When you type "exit", you return to the previous level (your application program).

Purpose Searches for a specific string of text in a file or files.

Command Format FIND[/V][/C][/N] <string> [<pathname>...][Retrn]

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

If you don't specify any files, FIND will take the input on the video display and display all lines that contain the specified string.

/VSwitch The /V switch causes FIND to display all lines not containing the specified string.

The command:

```
DIR D: | FIND /V "DAT"[Retrn]
```

causes DOS to display all names of the files on drive D: which do not contain the string DAT.

/C Switch The /C switch causes FIND to print only the count of lines that contained a match in each of the files.

/N Switch The /N switch causes each line to be preceded by its relative line number in the file. (This switch cannot be used with the /C switch.)

Strings

The string must be enclosed in quotes. For example:

FIND "Fool's Paradise" BOOK1.TXT BOOK2.TXT[Retrn]

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

Error Messages

When an error is detected, FIND responds with one of the following error messages:

FIND: Invalid number of parameters

You did not specify a string when issuing the FIND command.

FIND: Command error

You typed an illegal string when issuing the FIND command.

FIND: File not found <filename>

The filename you have specified does not exist or FIND cannot find it.

FIND: Read error in <filename>

An error occurred when FIND tried to read the file specified in the command.

FIND: Invalid parameter <option-name>

You specified an option that does not exist.

Purpose Sends output to video display one screen at a time.

Command Format | MORE[Retrn]

Remarks MORE is a filter that reads from standard input (such as a command typed at your keyboard) and displays one screen of information at a time. The MORE command then pauses and displays the --More-- message at the bottom of the display.

Pressing any key will display another screen of information. This process continues until all the information has been displayed.

The MORE command is useful for viewing a long file one screen at a time. If you type:

TYPE MYFILES.COM | MORE[Retrn]

DOS will display the file MYFILES.COM (on the default drive) one screen at a time.

Note: MORE creates a temporary file on the current drive which it later erases. Therefore, you must have create, open, read, write, and search rights on the current drive.

Purpose Changes the DOS command prompt.

Command Format PROMPT [<text>][Retrn]

Remarks This command allows you to change the DOS system prompt (for example, A>). If no text is typed, the prompt will be set to the default prompt, which is the default drive designation. You can set the prompt to a special prompt, such as the current time, by using the characters indicated below.

The following characters can be used in the prompt command to specify special prompts. They must all be preceded by a dollar sign ($):

Specify This Character	To Get This Prompt:
$	The '$' character
t	The current time
d	The current date
p	The working directory of the default drive
v	The version number
n	The default drive
g	The '>' character
l	The '<' character
b	The '|' character
_	A carriage return line feed sequence
s	A space (leading only)
e	ASCII code X'1B' (escape)

Examples The command:

PROMPT $p[Retrn]

sets the drive prompt to <drive:><current directory>.

PROMPT Time = t_Date = $d[Retrn]

sets a two-line prompt which prints:

Time = (current time)
Date = (current date)

If your terminal has an ANSI escape sequence driver, then you can use escape sequences in your prompts. For example:

PROMPT $e[7m$n:$e[m[Retrn]

sets the prompts in inverse video mode and returns to video mode for other text. ANSI escape sequences are discussed in Appendix C of this manual.

Purpose Sets one string value equivalent to another string for use in later programs.

Command Format SET [<string=string>][Retrn]

Remarks This command is meaningful only if you want to set values that will be used by programs you have written. An application program can check all values that have been set with the SET command by issuing SET with no options.

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 DOS will use for that variable with the SET command. The command SET FILE=DOMORE replaces the %FILE% parameter with the filename DOMORE. Therefore, you do not need to edit each batch file to change the replaceable parameter names. Note that when you use text (instead of numbers) as replaceable parameters, the name must be ended by a percent sign.

Example The command:

```
SET TTY=VT52
```

sets your TTY value to VT52 until you change it with another Set command. If you just type:

```
SET[Retrn]
```

DOS displays the current environment strings.

Purpose Reads input from your keyboard, sorts the data, then writes it to your video display or files.

Command Format SORT [/R] [/+n][Retrn]

Remarks SORT is used to sort a file by a certain column. There are two switches which allow you to select options:

/R reverse the sort; that is, sort from Z to A.

/+n sort starting with column n where n is some number. If you do not specify this switch, SORT will begin sorting from column 1.

Examples This command will read the file UNSORT.TXT, reverse the sort, and then write the output to a file named SORT.TXT:

```
SORT /R <UNSORT.TXT >SORT.TXT[Retrn]
```

The following command will pipe the output of the directory command to the SORT filter:

```
DIR | SORT /+14[Retrn]
```

The result of this command is a directory sorted by file size: the SORT filter will sort the directory listing starting with column 14 (this is the column in the directory listing that contains the file size), then send the output to the display.

Examples (cont.)

The command:

DIR | SORT /+14 | MORE[Retrn]

will do the same thing as the command in the previous example, except that the MORE filter will give you a chance to read the sorted directory one screen at a time.

Purpose

Displays the known system time. Under NetWare, the SYSTIME command gives the system time, date and day of the week.

Command Format

time [<hours>:<minutes>][Retrn]

Remarks

If you type TIME[Retrn], the following message is displayed:

Current time is <hh>:<mm>:<ss>.<cc>

You cannot change the system time; it is set by your system supervisor during installation and maintained by a clock in the central module.

Like the Date command, the Time Command format can be changed with the Country command in the CONFIG.SYS file. Refer to Appendix D, "How to Configure Your System," for more information.

Purpose Shows the contents of a text file (such as one created with EDLIN) on the screen.

Command Format TYPE <pathname>[Retrn]

Remarks Use this command to examine a text file without modifying it. (Use DIR to find the name of a file and EDLIN to alter the contents of a file.) The only formatting performed by TYPE is that tabs are expanded to spaces consistent with tab stops every eighth column. Note that text files created with some text editors will contain control characters and may not be displayed as expected. Also note that a display of binary files causes control characters (such as [Ctrl] Z) to be sent to your computer, including bells, form feeds, and escape sequences.

Purpose Prints the version number of MS-DOS currently in use at your workstation.

Command Format VER[Retrn]

Remarks If you want to know what version of MS-DOS you are using, type:

VER[Retrn]

The version number will be displayed.

Purpose

Turns the verify switch on or off when writing to a drive. This command should be used only on local and diskette drives.

Command Format

VERIFY [ON|OFF][Retrn]

Remarks

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 the drive, you can use the VERIFY command to tell DOS to verify that your files are intact (no bad sectors, for example). DOS will perform a VERIFY each time you write data to a drive. You will receive an error message only if DOS was unable to successfully write your data to the drive.

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[Retrn]

with no options.

Purpose Displays disk volume label, if it exists.

Command Format VOL [<drive:>][Retrn]

Remarks This command prints the volume label of the disk in a specific drive. If you do not type a drive letter, DOS prints the volume label of the disk in the default drive.

Messages If you see:

Volume in drive x has no label

it means that the disk in the drive does not have a volume label.

USING THE SETLOGIN COMMAND

Overview

The NetWare system allows each user to maintain a special set of instructions that are carried out at login time. These instructions are specified using the SETLOGIN command, which lets you specify the drive mappings that will be set up automatically when you log into the network. By using SETLOGIN, you can avoid having to issue multiple NPATH commands to map drives to directories each time you log in.

When you log into the network, your network drive path mappings are initialized by a user-written login "script." The script is a set of commands kept in your mailbox directory. Included in the script may be commands to assign path mappings, check for new mail in your mailbox, and display miscellaneous information such as your station number and login time.

For example, when you log in, the following will usually occur:

- Your first network disk drive (the first drive letter after the local drives) is mapped to your home directory area. The supervisor has already assigned your home directory area.

- The drive that maps to your home directory will become the default drive. (However, this can be changed by using the DRIVE command in the login script.)

- Your first search drive will be mapped to SYS:PUBLIC.

You or the supervisor may alter your login script to perform differently. The SETLOGIN command has a built-in editor and interactive help to aid you in customizing your login script.

Example SETLOGIN

For example, suppose that each time you log in, you want drive C assigned to SYS:BILL, drive D assigned to SYS:ACCOUNTS, and a second search drive assigned to an area where a number of BASIC commands are written.

We'll assume you always want your first search drive mapped to SYS:PUBLIC, so you can use all of NetWare's commands.

To have these drives set up automatically each time you log in, you must make a text file spelling out these drive mappings.

The SETLOGIN command calls up a text editor which is used to create your text file.

To create a text file to specify the drive mappings you want, use the SETLOGIN command as follows:

Type:

```
SETLOGIN[Retrn]
```

An editing screen will appear. Type:

```
NPATH C := SYS:BILL (or D : if first drive)
NPATH D := SYS:ACCOUNTS (or E :)
NPATH Search1 := SYS:LOGIN
NPATH Search2 := SYS:BILL/BASIC
```

and finish by pressing [F2].

(If your workstation has five local drive designations, use F and G instead of C and D.)

That's it. Next time you log in, NetWare will automatically read the commands in this text file and will set your drives accordingly.

Further Exercises

Refer to your worksheet #3 to determine what directories in your directory tree you have access to and use NPATH to reassign a network drive to one of them. Then list the files in that directory.

After you have listed the files in one directory, map different drive paths to all the directories to which you have access rights, and list the files in all directories.

The remainder of this chapter is a detailed explanation of the SETLOGIN command.

Purpose

Edits the script that the system uses to set up your customized network environment automatically each time you log in.

Command Format

SETLOGIN[Retrn]

Remarks

The SETLOGIN command has a self-contained screen editor and interactive help with which you may edit your login script. Press [F1] at any time during editing to receive help. Help operations are self-explanatory.

Command lines in a script file must not be more than 150 characters long. We recommend that command lines not exceed 80 characters so they will be more readable.

When you finish editing, press [F2] to save the new script, or [Ctrl][Break] to cancel editing.

A number of commands can be used in your script file. They are described below.

NPATH

The use of SETLOGIN's NPATH command is virtually the same as use of the NetWare NPATH command.

The simple command NPATH causes a complete listing of all drive and search mappings (if path display has not been disabled using NPATH DISPLAY OFF).

NPATH Options

Other NPATH command options are as follows:

NPATH DISPLAY ON

Enables the display of status messages when path mappings are assigned.

NPATH Options (cont.)

NPATH DISPLAY OFF

Disables NPATH's display.

NPATH ERRORS ON

Enables the display of messages that report serious errors which may be encountered while making path assignments.

NPATH ERRORS OFF

Disables the display of error messages.

The four NPATH options above may be used anywhere in the script. The defaults are DISPLAY ON and ERRORS ON.

When specifying drive names (e.g., C:, E:, etc.), you may use an asterisk followed by a number n to represent the nth network drive. For example, *1: represents the first drive beyond your local drives. If there were two local drives on a workstation, A: and B:, then *2: would represent D:. It is generally best to use the relative specification *n: to allow you to log in to workstations with differing configurations of local drives. A local drive may be assigned to a network path, but it is no longer usable until the network path mapping is removed.

Remember that the supervisor may initially assign *1: to your home directory. You may redefine *1: in the script if you choose.

Search drives should be assigned in numerical order, beginning with S1:. For more information, refer to the NPATH command description in Chapter 4.

Examples

```
NPATH *1:=SYS:SALES; Q:=*1:TEST
NPATH SEARCH1:=SYS:PUBLIC
```

COMSPEC

This command is used to tell DOS where to find the COMMAND.COM program if it needs to be reloaded. This will allow you to reload the proper COMMAND.COM for the machine and operating system you are using. You may specify the location of the command interpreter in one of three ways:

COMSPEC = *n:[/]<filename>

Specifies that COMMAND.COM be loaded from the directory the nth internetwork drive maps to.

COMSPEC = drive:[/]<filename>

Specifies that COMMAND.COM be loaded from a particular drive (e.g., A:, X:).

COMSPEC = Sn:[/]<filename>

Specifies that COMMAND.COM be loaded from the directory pointed to by search drive n.

The filename will usually be COMMAND.COM. No more than 12 characters may appear following the drive specification.

Examples

COMSPEC = *3:COMMAND.COM

DOS is configured to load COMMAND.COM from the third internetwork drive.

COMSPEC = C:\MYCOMAND.COM

DOS loads command file MYCOMAND.COM from the root directory relative to drive C:.

Examples (cont.)

COMSPEC = A:COMMAND.COM

DOS loads COMMAND.COM from drive A:.

COMSPEC = S2:COMMAND.COM

DOS loads COMMAND.COM from the directory search drive #2 is mapped to when this script command is interpreted.

INCLUDE

The login script command INCLUDE is used to allow the login script interpreter to process "sub-scripts" which are external to the login script currently being processed. These sub-scripts are simply text files that contain valid script commands and can be created and edited using any text or word processor.

The INCLUDE command takes the following form:

INCLUDE [<pathspec>/]<filename>[Retrn]

This feature may be used to shorten the login scripts of many users by having a "boiler plate" script file that is then included by each user who needs it.

INCLUDE could also be used by the supervisor to post various messages relating to network needs or new features by having each user's script include a special script file kept in the PUBLIC area.

INCLUDE nesting may go to 10 levels deep. This means that one script file may INCLUDE yet another script file which, in turn, may INCLUDE yet another script file, and so on down to 10 levels. A user must, of course, have at least Open and Read access privileges to any directory containing a desired sub-script.

Example

INCLUDE SYS:PUBLIC/MESSAGES.INC

DRIVE

The DRIVE command allows you to select a new default drive other than the first network drive. That is, the command DRIVE F: would inform DOS that F: is the new default drive. When LOGIN has finished execution, the command prompt from DOS would be F>. This will only be successful, of course, if drive F: had been defined by the time the DRIVE command is executed.

If no DRIVE command is used, the default drive set by LOGIN is the first network drive, which was initially assigned to be your home directory area.

Examples

DRIVE F:

DRIVE *2:

WRITE

You can customize your login messages with the WRITE command. A list of text strings or identifiers separated by semicolons (;) may be given after any WRITE prefix. A new line follows each line of messages unless a semicolon appears at the end of the line. This allows multiple WRITE commands to generate a one-line display.

Text strings are enclosed in quotes (") and may include the following super-characters:

\r	for a carriage return
\n	for a new line
\"	for an embedded quote mark
\7	to sound a "bell" tone on the workstation

Identifiers

Identifiers used with WRITE are predefined to represent text information of a variable nature. When the WRITE subcommand is executed, the text is substituted and is displayed on the workstation screen. These identifiers are:

HOUR	Hour of day or night (1 - 12)
HOUR24	Hour (00 - 23)
MINUTE	Minute (00 -59)
SECOND	Second (00 - 59)
AM_PM	Day or night (am or pm)
MONTH	Month number (01 -12)
MONTH_NAME	Month name (January, June, etc.)
DAY	Day number (01 - 31)
NDAY_OF_WEEK	Weekday number (1 - 7, Sunday is 1)
YEAR	Year in full format (1983,2016)
SHORT_YEAR	Year in short form (83, 16, etc.)
DAY_OF_WEEK	Day of week (Monday, Tuesday, etc.)

Note: The identifiers above represent your login date and time.

Identifiers (cont.)

LOGIN_NAME	User's unique login name
FULL_NAME	User's full name (from system files)
STATION	Logical Workstation number (North Star stations are 1-12, PC's 16-66)
P_STATION	Physical station (North Star stations are 41-4C hex; PC's are 0-31 hex or 0-49 dec., equivalent to Omninet board addresses.
GREETING_TIME	Morning, afternoon, or evening
NEW_MAIL	YES or NO, indicating whether or not new mail has arrived. Your mailbox is checked automatically when you log in, and the appropriate value ("YES" or "NO") is assigned to this identifier.
SHELL_TYPE	Shell type number (0, 27, 3, etc.)
OS	The workstation's operating system, e.g., MSDOS.
OS_VERSION	The version of the workstation's DOS, e.g., V3.1.
MACHINE	The machine which the shell was written for, e.g., IBM_PC
SMACHINE	Short machine name, e.g., IBM.

Example

For example, if user Bill's login script contains the following WRITE command:

```
WRITE "Hello, "; LOGIN_NAME
WRITE "This is station "; STATION
```

the following would be displayed when Bill logs in to station 16:

```
Hello, Bill
This is station 16
```

Identifier Macros

The identifiers listed with the WRITE command may also be used in other login script commands if they are preceded by a percent sign.

These "identifier macros," as they are called, will be replaced by their values before the script line is interpreted.

These identifier macros may also be placed within literal text strings in a WRITE statement; however, the identifier name must be in uppercase. For example, the following two lines would result in the same output:

```
WRITE "Good "; greeting_time; ", John"
WRITE "Good %GREETING_TIME, John"
```

Identifier macros are extremely useful if you intend to use different PC operating systems concurrently on different network stations. Depending upon the operating system type, you may appropriately set search mappings and other OS-dependent paths for the operating system you are using when you log in. The same need may arise with machine-dependencies. Use the P_STATION identifier to identify workstations with the same machine identifier.

For example, if you are using an external IBM PC attached to the network and running PC-DOS Version 3.00, you could create a directory:

```
SYS:PUBLIC/IBM_PC/PCDOS/V3.00
```

and place in that directory a copy of the appropriate COMMAND.COM. Create a similar directory for each different machine and system type. Your login script could then contain the following command, which would access the proper login command no matter which machine you execute login from:

```
NPATHS2:=SYS:PUBLIC/%P_STATION/%OS/%OS_VERSION
```

Conditionals

LOGIN also allows you to take action conditionally, depending on the values of the identifiers listed above. The conditionals allow for detection of six relations: equal to, not equal to, greater than, less than, greater than or equal to, and less than or equal to.

Also, several conditionals may be compounded. Compound conditionals are joined by commas or the word AND. The compound conditional is true only if all of its constituent conditionals are true. Some valid conditionals are:

```
DAY_OF_WEEK = "Monday"
GREETING_TIME IS NOT "Evening"
HOUR IS GREATER THAN "08"
```

Compound conditionals look like this:

```
HOUR IS "08" AND DAY_OF_WEEK IS "Monday"
HOUR24="23", MINUTE="59", AND SECOND="59"
```

Notice that literal text is enclosed in double quotes (\"). An embedded quote may be represented by \".

Comparisons

Identifiers may be compared with literal strings, and literals may be compared with other literals (useful with command line parameter substitution as described later). Upper and lower case is not significant in any of the comparisons.

Comparisons (cont.) Possible ways of representing comparisons for equal and not equal conditions are:

Equal	Not Equal
IS	IS NOT
=	!=
==	<>
	#
EQUALS	DOES NOT EQUAL
	NOT EQUAL TO

Valid formats of the other conditions include:

>	IS GREATER THAN
<	IS LESS THAN
>=	IS GREATER THAN OR EQUAL TO
<=	IS LESS THAN OR EQUAL TO

IF ... THEN An IF ... THEN command is of the format: IF <conditional> THEN <command>. For example:

```
IF DAY_OF_WEEK="Monday" THEN WRITE "AARGH..."
```

The command following the THEN may be a single statement or the beginning of a block of commands. The command or block of commands will be interpreted only if all of the specified conditions were true.

A block of conditional commands is started by specifying BEGIN after the THEN. Subsequent lines make up the conditional block until a line with the command END is reached. An example will help illustrate this concept:

```
IF DAY_OF_WEEK = "Tuesday" THEN BEGIN
  WRITE "Staff meeting today at 10 a.m."
  INCLUDE SYS:PUBLIC/UPDATE
END
```

IF ...THEN (cont.)

Note that the BEGIN statement must be on the same line as the conditional IF. DO can also be used in place of BEGIN.

Nesting of IF statements isn't allowed; that is, an IF statement isn't allowed as the object of a conditional interpretation.

Introduction

This portion of the SETLOGIN command is only available on an external PC workstation. The North Star workstation login system does not permit you to enter additional arguments. External PC's using the LOGIN command, however, do support the capability to "pass in" parameters from the command line which invoked it. In this case, when a script command is interpreted, any percent sign (%) followed by a number from 1 to 19 will be replaced by the corresponding command line parameter.

The user's login name is %1.

If you really want to print %5, and not the corresponding substitution, then the script entry must be written as "\%5". The backslash (\) will cause the % to be treated as a normal character.

Example

Suppose you type the following LOGIN command line:

```
LOGIN Mark alpha beta gamma
```

and your login script contains this statement:

```
WRITE "%4, %3, and %2 follow \%1"
```

This script line then becomes:

```
WRITE "gamma, beta, and alpha follow \%1"
```

The following is written to the screen:

```
gamma, beta, and alpha follow %1
```

Command line parameters may be used in the conditional IF ... THEN as follows:

```
IF "%2" != "sales" THEN NPATH *3:=SYS:PROGRAMS/%2
```

Example (cont.)

Remember that parameter substitution takes place before the script command is scanned and interpreted; parameters can be substituted in any part of a script command line. That is why the %2 in the conditional part of the last example must be in quote marks--the %2 represents a literal text string, not an identifier.

The script commands in the last example accomplish the following: If parameter 2 is not "sales", then assign network drive 3 to the subdirectory beneath SYS:PROGRAMS as named by parameter 2.

Transient Commands

Commands external to the login script can be executed using a transient command as follows:

#[path]<command parameter line>[Retrn]

The first character on the script line is the pound sign (#). The command is the name of an .EXE or .COM program to be executed (excluding the extension). The command name may be preceded by a path specification. Following the command name, parameters may be added as they would normally appear if the command were being executed manually from the DOS prompt.

A transient command statement must appear on its own line, as in the following example. In this example, NSNIPES is executed when the user enters "snipes" as a LOGIN parameter after the login name:

```
IF "%2" = "snipes" THEN BEGIN
  #NSNIPES 4
END
```

The transient command should appear after NPATH assignments, since the program will execute in the environment of drive mappings, default drive, and search drives set up previously in the login script. The transient command function will fail if the given path is invalid, if proper network access privileges are lacking, if the command cannot be found, or if there is insufficient memory in the workstation to load the command.

Note: These parameters will not work on North Star workstations.

EXIT

The EXIT command causes LOGIN to terminate execution, ignoring the remainder of the script beyond the EXIT statement.

Example:

```
IF NEW_MAIL = "No" THEN EXIT
```

The EXIT command may optionally pass a short command line to COMMAND.COM. The format of this option is:

```
EXIT <command>
```

This allows you to execute not only .COM and .EXE files, but also .BAT files and COMMAND.COM intrinsics such as DIR. The command specified must be no longer than 14 characters.

Example:

```
IF NEW_MAIL = "Yes" THEN EXIT "MAIL"
```

BREAK

The BREAK command allows you to control whether or not LOGIN can be interrupted with [Ctrl]C or [Ctrl][Break] while interpreting the script. The interruption occurs between script lines. The two forms of this command are:

BREAK ON

BREAK OFF

The default form is BREAK OFF. When the BREAK option is ON, type-ahead keyboard buffering isn't possible. This side effect may not be desirable to some users.

DOS BREAK

DOS BREAK allows you to control whether or not any programs or DOS commands executed after logging in can be interrupted with [Ctrl][Break]. The two forms of this command are:

DOS BREAK ON

DOS BREAK OFF

The default form is DOS BREAK OFF. DOS BREAK ON will enable the check only when LOGIN terminates execution; this helps to prevent unexpected termination of LOGIN.

DOS BREAK OFF, however, disables [Ctrl][Break] immediately. Technically speaking, the command sets/resets the DOS [Ctrl][Break] check feature using DOS function 33 hex.

DISPLAY/ FDISPLAY

These commands allow you to display the contents of a text file on the workstation screen. The DISPLAY command will display the exact characters that are contained in the file, without modification. FDISPLAY, on the other hand, will "filter" the text in the file, stripping high-order bits and removing any non-printable characters. The resulting text is composed entirely of printable ASCII standard characters. The format of the command is:

[F]DISPLAY [<pathspec>/]<filename>[Retrn]

This command might be used in every user's script file to display the contents of a public "bulletin board" which the supervisor updates as necessary.

Example

FDISPLAY SYS:PUBLIC/MESSAGES/SYSNEWS.TXT[Retrn]

If the given path does not exist, or the given file is not found, no error will be given. Thus, there will be no annoying messages on the user's screen if there are no bulletins when the user logs in.

REMARK

The REMARK command allows you to insert explanatory text in your script file. Any text following the word REMARK will be ignored. The abbreviation "REM", an asterisk (*), and a semicolon (;) will also be accepted.

Examples

```
* * * * * * * * * * * * * * * *
*                             *
*       Sample Text           *
*                             *
* * * * * * * * * * * * * * * *
```

```
; assign OS-dependent Search mappings

REMARK:  Check for new mail.
```

Appropriate use of remarks in the script file will greatly improve the script's readability.

REMARK text must be the only thing on a line. Remarks will cause errors if placed on the same line with script commands.

FIRE PHASERS

The FIRE PHASERS command allows you to generate a shooting noise any number of times by specifying:

FIRE PHASERS n TIMES

where n is the number of times that you wish to hear this sound.

Sample Script

The sample script below shows a typical sequence of SETLOGIN command options.

```
NPATH *2:=SYS:PUBLIC
NPATH Q:=SYS:TEST
NPATH S1:=SYS:PUBLIC
WRITE "Hello  %LOGIN_NAME"
WRITE "Today is  %DAY_OF_WEEK"
IF DAY_OF_WEEK = "MONDAY" THEN BEGIN
  WRITE "Have a nice weekend?"
END
FIRE PHASERS 4 TIMES
```

Press [F2] to save your script to be used at login time.

CREATING AND USING BATCH FILES

Introduction

Often you may find yourself typing the same sequence of commands over and over to perform some commonly used task. With DOS, you can put the command sequence into a special file called a **batch file**, and execute the entire sequence simply by typing the name of the batch file. "Batches" of your commands in such files are processed as if you had typed them individually. Each batch file must be named with the .BAT extension, and is executed by typing the filename without its extension.

You can create a batch file by using the Line Editor (EDLIN) or by using the COPY command.

If you are not writing batch programs, you do not need to read this chapter.

Commands for Batch Files

Two DOS commands are available for use expressly in batch files: REM and PAUSE. REM permits you 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 later in this chapter.

Purpose Suspends execution of the batch file.

Command Format PAUSE [comment][Retrn]

Remarks 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]C.

When the command processor encounters PAUSE, it prints the optional comment and then prints:

Strike a key when ready . . .

If you press [Ctrl]C, another prompt will be displayed:

Terminate batch job (Y/N)?

If you type Y in response to this prompt, execution of the remainder of the batch command file will be cancelled and control will be returned to the operating system command level.

Therefore, PAUSE can be used to break a batch file into pieces, allowing you to end the batch command file at an intermediate point.

Remarks (cont.)

You can create a message that appears when the batch file pauses. For example, say you want to to create a prompt that instructs you to change diskettes in the disk drive. In the batch file you would type:

```
Pause Please change diskettes now.[Retrn]
```

When the batch file runs you will see:

```
D>PAUSE  Please change diskettes now.
Strike a key when ready...
```

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

Command Format SHIFT[Retrn]

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

```
if  %0 = "foo"
    %1 = "bar"
    %2 = "name"
    %3...%9 are empty
```

then a SHIFT will result in the following:

```
%0 = "bar"
%1 = "name"
%2...%9 are empty
```

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

Purpose

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

Command Format

REM [comment][Retrn]

Remarks

The only separators allowed in the comment are the space, tab and comma.

Example

```
REM  This file formats and checks new diskettes[Retrn]
REM  It is named NEWDISK.BAT[Retrn]
PAUSE  Insert new diskette in drive A[Retrn]
REQUEST D
FORMAT A:/S[Retrn]
DIR A:[Retn]
CHKDSK A:[Retrn]
RELEASE D
[Ctrl]Z[Retrn]
```

THE DOS LINE EDITOR (EDLIN)

Overview

In this chapter, you will learn how to use EDLIN, the line editor program. You can use EDLIN to create, change, and display files, whether they are program or text files.

You can use EDLIN to:

- o Create new files and save them.
- o Update existing files and save both the updated and original files.
- o Delete, edit, insert, and display lines.
- o Search for, delete, or replace text within one or more lines.

Text Placement

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 line 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.

Creating A File

To start EDLIN, type:

EDLIN <filespec>[Retrn]

If you are creating a new file, the <filespec> should be the name of the file you wish to create. If EDLIN does not find this file on a drive, EDLIN will create a new file with the name you specify. The following message and prompt will be displayed:

```
New file
*
```

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.

Editing an Existing File

If you want to edit an existing file, <filespec> should be the name of the file you want to edit. When EDLIN finds the file you specify on the designated or current drive, the file will be loaded into memory. If the entire file can be loaded, EDLIN will display the following message on your screen:

```
End of input file
*
```

Editing an Existing File (cont.)

You can then edit the file using EDLIN editing commands.

If the file is too large to be loaded into memory, EDLIN will load lines until memory is 3/4 full, then display 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 the drive to free memory, then EDLIN can load the unedited lines from the drive into memory. Refer to the Write and Append commands in this chapter for the procedure.

Saving the File

EDLIN automatically creates a copy of any file you edit. 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 will not be erased until the end of the editing session, or until space is needed by EDLIN.

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 DOS RENAME command discussed in Chapter 5, then start EDLIN and specify the new <filespec>.

Keys to Use

The special editing keys and template discussed in here can be used to edit your text files. These keys are discussed in detail in this section.

The table below summarizes the commands, codes, and functions. Descriptions of the special editing keys follow the table.

Key	Command	Function
[F1]	COPY1	Copies one character from the template to the new line
[F2]	COPYUP	Copies all characters from the template to the new line, up to the character specified
[F3]	COPYALL	Copies all remaining characters in the template to the display
[Del]	SKIP1	Does not copy (skips over) a character
[F4]	SKIPUP	Does not copy (skips over) the characters in the template, up to the character specified
[Esc]	VOID	Voids the current input; leaves the template unchanged
[Ins]	INSERT	Enters/exits insert mode
[F5]	NEWLINE	Makes the new line the new template

Purpose Copies one character from the template to the command line.

Remarks Pressing [F1] copies one character from the template to the command line. When [F1] is pressed, one character is inserted in the command line and insert mode is automatically turned off.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Pressing [F1] copies the first character (T) to the second of the two lines displayed:

```
1:*This is a sample file
[F1] 1:*T
```

Each time [F1] is pressed, one more character appears:

```
[F1] 1:*Th
[F1] 1:*Thi
[F1] 1:*This
```

and so forth.

Purpose Copies multiple characters up to a given character.

Remarks Pressing [F2] copies all characters up to a given character from the template to the command line. The given character is the next character typed after [F2]; it is not copied or displayed on the display. Pressing [F2] causes the cursor to move to the single character that is specified in the command. If the template does not contain the specified character, nothing is copied. Pressing [F2] also automatically turns off insert mode.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Pressing [F2] copies all characters up to the character specified immediately after [F2].

```
1:*This is a sample file
[F2]p 1:*This is a sam
```

Purpose Copies template to command line.

Remarks Pressing [F3] copies all remaining characters from the template to the command line. Regardless of the cursor position at the time [F3] is pressed, the rest of the line appears, and the cursor is positioned after the last character on the line.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Pressing [F3] copies all characters from the template (shown in the upper line displayed) to the line with the cursor (the lower line displayed):

```
1:*This is a sample file (template)
[F3] 1:*This is a sample file. (command line)
```

Also, insert mode is automatically turned off.

Purpose Skips over one character in the template.

Remarks Pressing [Del] skips over one character in the template. Each time you press [Del], one character is not copied from the template. The action of the [Del] key is similar to the [F1] (COPY1) key, except that [Del] skips a character in the template rather than copying it to the command line.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Pressing [Del] skips over the first character (T).

```
1:*This is a sample file
[Del] 1:*
```

The cursor position does not change and only the template is affected. To see how much of the line has been skipped over, press **[F3]** which moves the cursor beyond the last character of the line.

```
1:*This is a sample file.
[Del] 1:*
[F3] 1:*his is a sample file.
```

Purpose Skips multiple characters in the template up to the specified character.

Remarks Pressing [F4] skips over all characters up to a given character in the template. This character is not copied and is not shown on the display. If the template does not contain the specified character, nothing is skipped over. The action of the [F4] key is similar to the [F2] key, except that [F4] skips over characters in the template rather than copying them to the command line.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Pressing [F4] skips over all the characters in the template up to the character pressed after [F4]

```
1:*This is a sample file
[F4]p 1:*
```

The cursor position does not change. To see how much of the line has been skipped over, press [F3] to copy the template. This moves the cursor beyond the last character of the line:

```
1*This is a sample file:
[F4]p 1:*
[F3]  1:*ple file.
```

Purpose Quits input and empties the command line.

Remarks Pressing [Esc] empties the command line, but it leaves the template unchanged. [Esc] also prints a back slash (\), carriage return, and line feed, and turns insert mode off. The cursor is positioned at the beginning of the line. Pressing [F3] copies the template to the command line and the command line appears as it was before [Esc] was pressed.

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Assume that you want to replace the line with "Sample File:"

```
1:*This is a sample file.
1:*Sample File
```

To cancel the line you just entered (Sample File), and to keep "This is a sample file.", press [Esc]. Notice that a backslash appears on the Sample File line to tell you it has been cancelled.

```
1:*This is a sample file.
[Esc] 1:*Sample File\
1:
```

Press **[Retrn]** to keep the original line, or to perform any other editing functions. If you press [F3] the original template is copied to the command line:

```
[F3] 1: This is a sample file.
```

Purpose Enters/exits insert mode.

Remarks Pressing [Ins] causes EDLIN to enter or exit insert mode. The current cursor position in the template is not changed.

Normally, EDLIN is in "replace" mode, in which the characters you type overstrike and replace characters in the template.

Example Assume that the screen shows:

```
1:*This is a sample file.
1:*_
```

Assume that you then press **[F2]m, [Ins]lary, [Ins] tax,** and then **[F3]**:

	1:*This is a sample file.
[F2]m	1:*This is a sa_
[Ins]lary	1:*This is a salary_
[Ins] tax	1:*This is a salary tax_
[F3]	1:*This is a salary tax file._

You used [Ins] the first time to insert "lary" and then used it again to turn insert mode off in order to replace "mple" with " tax."

In insert mode, the cursor moves as each character is inserted. However, when you have finished inserting characters, the cursor will be positioned at the same character as it was before the insertion began. Thus, characters are inserted in front of the character to which the cursor points.

Example (cont.)

Here is another example of insert mode. Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Assume that you press **[F2]f**

```
                1:*This is a sample file
[F2]f           1:*This is a sample
```

Now press **[Ins]** and insert the characters "edit" and a space

```
                1:*This is a sample file.
[F2]f           1:*This is a sample
[Ins]edit       1:*This is a sample edit
```

If you now press **[F3]** the rest of the template is copied to the line:

```
                1:*This is a sample edit
[F3]            1:*This is a sample edit file.
```

If you pressed **[Retrn]** the remainder of the template would be truncated, and the command line would end at the end of the insert:

[Ins]edit [Retrn] 1:*This is a sample edit

To exit insert mode, simply press **[Ins]** again.

Purpose Creates a new template.

Remarks Pressing [F5] copies the current command line to the template. The contents of the old template are deleted. Pressing [F5] outputs an @ ("at sign" character), a carriage return, and a line feed. The command line is also emptied and insert mode is turned off.

Note: [F5] performs the same function as [Esc] except that the template is changed and an @ ("at sign" character) is printed instead of a \ (backslash).

Example Assume that the display shows:

```
1:*This is a sample file.
1:*
```

At the beginning of the editing session, the cursor is positioned at the beginning of the line. Assume that you enter **[F2]m**, **[Ins]lary** (to turn insert mode on and insert "lary"), **[Ins] tax** (to turn insert mode off and replace "mple" with " tax") and then **[F3]**:

```
                1:*This is a sample file.
[F2]m           1:*This is a sa_
[Ins]lary       1:*This is a salary_
[Ins] tax       1:*This is a salary tax_
[F3]            1:*This is a salary tax file._
```

At this point, assume that you want this line to be the new template, so you press the **[F5]** key:

```
[F5]            1:*This is a salary tax file.@
```

The @ indicates that this new line is now the new template. Additional editing can be done using the new template.

General Information

EDLIN commands perform editing functions on lines of text. The following list contains information you should read before you use EDLIN commands.

Pathnames

Pathnames are acceptable as options to commands. For example, typing EDLIN \BIN\ACCOUNTS\AP\OFFICE will allow you to edit the OFFICE file in the subdirectory AP.

Line Numbers

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. For example:

-10,+10L[Retrn]

This command lists 10 lines before the current line, the current line and 10 lines after the current line.

Multiple Commands

Multiple commands may be issued on one command line. When you issue a command to edit a single line using a line number (<line>), a semicolon must separate commands on the line. Otherwise, one command may follow another without any special separators. In the case of a Search or Replace command, the <string> may be ended by [Ctrl]Z instead of [Retrn]. For example, the following command line edits line 15 and then displays lines 10 through 20 on the screen.

15;-5,+5L[Retrn]

Multiple Commands

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.

SThis string[Ctrl]Z-5,+5L[Retrn]

Space

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.

Control Characters

It is possible to insert a control character (such as [Ctrl]C) into text by using the quote character [Ctrl]V before it while in insert mode. [Ctrl]V tells 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]VZ

will find the first occurrence of [Ctrl]Z in a file.

R[Ctrl]VZ[Ctrl]Z foo

will replace all occurrences of [Ctrl]Z in a file by "foo".

S[Ctrl]VC[Ctrl]Z bar

will replace all occurrences of [Ctrl]C by "bar".

It is possible to insert [Ctrl]V into the text by typing:

[Ctrl]VV

End-of-File

The [Ctrl]Z character ordinarily tells EDLIN, "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 switch to tell EDLIN to ignore any [Ctr]Z characters in the file and to show you the entire file.

EDLIN Command Summary

The EDLIN commands are summarized in the following table. They are also described in further detail following the description of command options.

Command	Purpose
<line>	Edits line number
A	Appends lines
C	Copies lines
D	Deletes lines
E	Ends editing
I	Inserts lines
L	Lists text
M	Moves lines
P	Pages text
Q	Quits editing
R	Replaces lines
S	Searches text
T	Transfers text
W	Writes lines

Command Options

Several EDLIN commands accept one or more options. The effect of a command option varies, depending on with which command it is used. The following list describes each option.

<line>

<line> indicates a line number that you type. Line numbers must be separated by a comma or a space from other line numbers, other options, and from the command.

<line> may be specified one of the following ways:

Number Any number less than 65534. If a number larger than the largest existing line number is specified, then <line> means the line after the last line number.

<line.>

Period If a period (.) is specified for <line>, then <line> means the current line number. The current line is the last line edited, and is not necessarily the last line displayed. The current line is marked on your screen by an asterisk (*) between the line number and the first character.

Pound The pound sign (#) indicates the line after the last line number. If you specify # for <line>, this has the same effect as specifying a number larger than the last line number.

A carriage return entered without any of the <line> specifiers listed above directs EDLIN to use a default value appropriate to the command.

?

The question mark option directs EDLIN to ask you if the correct string has been found. The question mark is used only with the Replace and Search commands. Before continuing, EDLIN waits for either a Y or [Retrn] for a yes response, or for any other key for a no response.

<string> <string> represents text to be found, to be replaced, or to replace other text. The <string> option is used only with the Search and Replace commands. Each <string> must be ended by a [Ctrl]Z or a [Retrn] (see the Replace command for details). No spaces should be left between strings or between a string and its command letter, unless you want those spaces to be part of the string.

Purpose

Adds the specified number of lines from the drive to the file being edited in memory. The lines are added at the end of lines that are currently in memory.

Command Format

[<n>]A[Retrn]

Remarks

This command is meaningful only if the file being edited is too large to fit into memory. As many lines as possible are read into memory for editing when you start EDLIN.

To edit the remainder of the file that will not fit into memory, lines that have already been edited must be written to the drive. Then you can load unedited lines from the drive into memory with the Append command. Refer to the Write command in this chapter for information on how to write edited lines to the drive.

Note: If you do not specify the number of lines to append, lines will be appended to memory until available memory is 3/4 full. No action will be 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

Copies a range of lines to a specified line number. The lines can be copied as many times as you want by using the <count> option.

Command Format

[<line>],[<line>],<line>,[<count>]C[Retrn]

Remarks

If you do not specify a number in <count>, EDLIN copies the lines one time. If the first or the second <line> are omitted, the default is the current line. The file is renumbered automatically after the copy.

The line numbers must not overlap or you will get an "Entry error" message. For example, 3,20,15C would result in an error message.

Example

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[Retrn]

Example (cont.)

The result is:

```
1:  This is a sample file
2:  used to show copying lines.
3:  See what happens when you use
4:  the Copy command
5:  (the C command)
6:  to copy text in your file.
7:  This is a sample file
8:  used to show copying lines.
9:  See what happens when you use
10: the Copy command
11: (the C command)
12: to copy text in your file.
```

Copy and Insert

If you want to place the text within other text, the third <line> option 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.
```

Copy and Insert (cont.)

The command **3,6,9C[Retrn]** 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.
```

Purpose Deletes a specified range of lines in a file.

Command Format [<line>][,<line>]D[Retrn]

Remarks If the first <line> is omitted, that option will default to the current line (the line with the asterisk next to the line number). If the second <line> is omitted, then just the first <line> will be deleted. When lines have been deleted, the line immediately after the deleted section becomes the current line and has the same line number as the first deleted <line> had before the deletion occurred.

Example 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

To delete lines 5 through 24, type:

5,24D[Retrn]

The result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7:*in your file.
```

To Delete a Single Line

To delete line 6, type:

6D[Retrn]

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.
```

To Delete A Range of Lines

Next, delete a range of lines from the following file:

```
1: This is a sample file
2: used to show dynamic line numbers.
3:*See what happens when you use
4:Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
```

To Delete A Range of Lines (cont.)

To delete a range of lines beginning with the current line 3, type:

,6D[Retrn]

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.

Purpose Edits line of text.

Command Format [<line>][Retrn]

Remarks 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 editing commands to edit the line. The existing text of the line serves as the template until the key is pressed.

If no line number is typed (that is, if only [Retrn] 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 [Retrn] to accept the line as is.

>> CAUTION <<
If [Retrn] 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:

```
1: This is a sample file.
2: used to show
3: the editing of line
4:*four.
```

Example (cont.)

To edit line 4, type:

4[Retrn]

The contents of the line are displayed with a cursor below the line:

```
4:* four.
4:*
```

Now type:

[Ins]number[space]	4: number
[F3][Retrn]	4: number four:*
	5:*

Purpose Ends the editing session.

Command Format E[Retrn]

Remarks This command saves the edited file on the drive, renames the original input file <filename>.BAK, and then exits EDLIN. If the file was created during the editing session, no .BAK file is created.

The E command takes no options. Therefore, you cannot tell EDLIN on which drive to save the file. The drive you want to save the file on must be selected when the editing session is started. If the drive is not selected when EDLIN is started, the file will be saved on the default drive. It will still be possible to COPY the file to a different drive using the DOS COPY command.

You must be sure that the drive contains enough free space for the entire file. If the drive does not contain enough free space, the write will be aborted and the edited file lost, although part of the file might be written out to the drive.

Example Type:

E[Retrn]

After the execution of the E command, the DOS default drive prompt (for example, C>) is displayed.

Purpose Inserts text immediately before the specified <line>.

Command Format [<line>]I[Retrn]

Remarks 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 [Retrn] is pressed.

EDLIN remains in insert mode until [Ctrl]C is typed. When the insert is completed and insert mode has been exited, the line immediately following the inserted lines becomes the current line. All line numbers following the inserted section are incremented by the number of lines inserted.

If <line> is not specified, the default will be the current line number and the lines will be inserted immediately before the current line. If <line> is any number larger than the last line number, or if a pound sign (#) is specified as <line>, the inserted lines will be appended to the end of the file. In this case, the last line inserted will become the current line.

Examples Assume that the following file exists and is ready to edit:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
```

Examples (cont.)

To insert text before a specific line, type <line>I[Retrn] as in:

7I[Retrn]

The result is:

7:*

Now, type the new text for line 7:

7:* and renumber lines[Retrn]

End Insert

Then to end the insertion, press **[Ctrl]Z** on the next line:

8:*[Ctrl]Z[Retrn]

Now type **L[Retrn]** to list the file. The result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: and renumber lines
8:*in your file.
```

Insert Before Current Line

To insert lines immediately before the current line, type:

I[Retrn]

The result is:

8:*

Insert Before Current Line (cont.)

Now, insert the following text and terminate with a [Ctrl]Z on the next line:

8:* so they are consecutive[Retrn]
9:* [Ctrl]Z[Retrn]

Now to list the file and see the result, type:

L[Retrn]

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 Inser
5: (the D and I commands)
6: to edit text
7: and renumber lines
8: so they are consecutive
9:*in your file.
```

To Append New Lines

To append new lines to the end of the file, type:

10I[Retrn]

This produces the following:

```
10:*
```

Now, type the following new lines:

10: The insert command can place new lines[Retrn]
11: in the file; there's no problem[Retrn]
12: because the line numbers are dynamic[Retrn];
13: they'll go all the way to 65533.[Retrn]

To Append New Lines (cont.)

End the insertion by pressing:

[Ctrl]Z

on line 14. The new lines will appear at the end of all previous lines in the file. Now type the List command:

L[Retrn]

The result is:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: and renumber lines
8: so they are consecutive
9: in your file.
10: The insert command can place new lines
11: in the file; there's no problem
12: because the line numbers are dynamic;
13: they'll go all the way to 65533.
```

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

Command Format [<line>][,<line>]L[Retrn]

Remarks Default values are provided if either one or both of the options are omitted. If you omit the first option, as in:

,<line>L[Retrn]

the display will start 11 lines before the current line and end with the specified <line>. The beginning comma is required to indicate the omitted first option.

Note: If the specified <line> is more than 11 lines before the current line, the display will be the same as if you omitted both options.

If you omit the second option, as in:

<line>L[Retrn]

23 lines will be displayed, starting with the specified <line>.

If you omit both parameters, as in:

L[Retrn]

23 lines will be displayed--the 11 lines before the current line, the current line, and the 11 lines after the current line. If there are less than 11 lines before the current line, more than 11 lines after the current line will be displayed to make a total of 23 lines.

Examples

Assume that the following file exists and is ready to edit:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
.
.
.
15:*The current line contains an asterisk
26:to edit text
27: in your file.
```

To list a range of lines without reference to the current line, type **<line>,<line>L** and press **[Retrn]**:

2,5L[Retrn]

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 **,<line> L** and press **[Retrn]**:

,26L[Retrn]

Examples (cont.)

The result is:

```
15:*The current line contains an asterisk.
.
.
.
26:to edit text
```

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

L[Retrn]

The result is:

```
4:Delete and Insert
5:(the D and I commands)
.
.
.
14: The current line remains unchanged.
15:*The current line contains an asterisk.
.
.
.
26:to edit text.
```

Purpose Moves a range of text to the line specified.

Command Format [<line>],[<line>],<line>M[Retrn]

Remarks Use the Move command to move a block of text (from the first <line> to the second <line>) to another location in the file. The lines are renumbered according to the direction of the move. For example:

,+25,100M[Retrn]

moves the text from the current line plus 25 lines to line 100. If the line numbers overlap, EDLIN will display an "Entry error" message.

To move lines 20-30 to line 100, type:

20,30,100M[Retrn]

Purpose Pages through a file 23 lines at a time.

Command Format [<line>][,<line>]P[Retrn]

Remarks If the first <line> is omitted, that number will default to the current line plus one. If the second <line> is omitted, 23 lines will be listed. The new current line becomes the last line displayed and is marked with an asterisk.

Purpose

Quits the editing session, does not save any editing changes, and exits to DOS.

Command Format

Q[Retrn]

Remarks

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 except Y 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 Abort edit (Y/N)? message, your previous backup copy will no longer exist.

Example

```
Q
Abort edit (Y/N)?Y[Retrn]
C>
```

Purpose

Replaces all occurrences of a string of text in the specified range with a different string of text or blanks.

Command Format

[<line>][,<line>][?]R<string1>[Ctrl]Z
<string2>[Retrn]

Remarks

As each occurrence of <string1> is found, it is replaced by <string2>. Each line in which a replacement occurs will be displayed. If a line contains two or more replacements of <string1> with <string2>, then the line will be 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 reappears.

If a second string is to be given as a replacement, then it must be separated from <string2> with a [Ctrl]Z. <String2> must also be ended with a [Ctrl]Z [Retrn] combination or with a simple [Retrn].

Omitting Strings

If <string1> is omitted, then Replace will take the old <string1> as its value. If there is no old <string1>, i.e., this is the first replace done, then the replacement process will be terminated immediately. If <string2> is omitted, then <string1> may be ended with a [Retrn]. If the first <line> is omitted in the range argument (as in ,<line>) then the first <line> will default to the line after the current line. If the second <line> is omitted (as in <line> or <line>,), the second <line> will default to #. Therefore, this is the same as <line>,#. Remember that # indicates the line after the last line of the file.

Omitting Strings (cont.)

If <string1> is ended with a [Ctrl]Z and there is no <string2>, <string2> will be taken as an empty string and will become the new replace string. For example,

R<string2>[Ctrl]Z[Retrn]

will delete occurrences of <string1>, but

R<string1>[Retrn]

will replace <string1> with the old <string2> and

R[Retrn]

will recall the last values entered as <string1> and <string2>. Note that "old" here refers to a previous string specified either in a Search or a Replace command.

?Option

If the question mark (?) option is given, the Replace command will stop at each line with a string that matches <string1>, the line with <string2> in place, and then display the prompt O.K.?. If you press Y or [Retrn] then <string2> will replace <string1>, and the next occurrence of <string1> will be found. The O.K.? prompt will be displayed. This process will continue 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 besides Y or [Retrn] after the O.K.? prompt, the <string1> will be left as it was in the line, and Replace will go to the next occurrence of <string1>. If <string1> occurs more than once in a line, each occurrence of <string1> will be replaced individually, and the O.K.? prompt will be displayed after each replacement. In this way, only the desired <string1> will be replaced, and you can prevent unwanted substitutions.

Examples Assume that the following file exists and is ready for editing:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
8: The insert command can place new lines
9: in the file; there's no problem
10: because the line numbers are dynamic;
11: they'll go all the way to 65533.
```

To replace all occurrences of <string1> with <string2> in a specified range, type:

2,12 Rand[Ctrl]Z or[Retrn]

The result is:

```
4: Delete or Insert
5: (the D or I commors)
8: The insert commor can place new lines
```

Note: In the above replacement, some unwanted substitutions have occurred. To avoid these and to confirm each replacement, the same original file can be used with a slightly different command.

In the next example, to replace only certain occurrences of the first <string> with the second <string>, type:

2? Rand[Ctrl]Z or[Retrn]

Examples (cont.)

The result is:

```
4: Delete or Insert
O.K.? Y
5: (The D or I commands)
O.K.? Y
5: (The D or I commors)
O.K.? N
8: The insert commor can place new lines
O.K.? N
*
```

Now type the List command:

L[Retrn]

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
```

Purpose

Searches the specified range of lines for a specified string of text.

Command Format

[<line>][,<line>][?]S<string>[Retrn]

Remarks

The <string> must be ended with a [Retrn]. The first line that matches <string> is displayed and becomes the current line. If the question mark option is not specified, the Search command will terminate when a match is found. If no line contains a match for <string>, the message "Not found" will be displayed.

? Option

If the question mark option (?) is included in this command, EDLIN will display the first line with a matching string; it will then prompt you with the message OK ?. If you press either the Y or [Retrn] key, the line will become the current line and the search will terminate. If you press any other key, the search will continue until another match is found, or until all lines have been searched (and the Not found message is displayed).

Omitting Strings

If the first <line> is omitted (as in ,<line> <string>), the first <line> will default to the line after the current line. If the second <line> is omitted (as in <line> S<string> or <line>, S<string>), the second <line> will default to # (line after last line of file), which is the same as <line>,# S<string>. If <string> is omitted, Search will take 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 (i.e., no previous search or replace has been done), the command will terminate immediately.

Examples Assume that the following file exists and is ready for editing:

```
1: This is a sample file
2: used to show dynamic line numbers.
3: See what happens when you use
4: Delete and Insert
5: (the D and I commands)
6: to edit text
7: in your file.
8: The insert command can place new lines
9: in the file; there's no problem
10: because the line numbers are dynamic;
11:*they'll go all the way to 65533.
```

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

2,12 Sand[Retrn]

The following line is displayed:

4: Delete and Insert

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

[Del][F3],12 Sand[Retrn]

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)

Examples (cont.)

To search through several occurences of a string until the correct string is found, type:

1, ? Sand[Retrn]

The result is:

4: Delete and Insert
O.K.?

Examples

If you press any key (except Y or [Retrn]), the search continues, so type N here:

OK.? N

Continue:

5: (the D and I commands)
OK.?

Now press Y to terminate the search:

O.K.? Y
*

To search for string XYZ without the verification (O.K.?), type:

SXYZ[Retrn]

EDLIN will report a match and will continue to search for the same string when you issue the S command:

S[Retrn]

EDLIN reports another match.

Examples (cont.)

S[Retrn]

EDLIN reports the string is not found.

Note: <string> defaults to any string specified by a previous Replace or Search command.

Purpose

Inserts (merges) the contents of <filename> into the file currently being edited at <line>. If <line> is omitted, then the current line will be used.

Command Format

[<line>]T<filename>[Retrn]

Remarks

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 at the line number specified by <line> and the lines are renumbered.

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.

Command Format

[<n>]W[Retrn]

Remarks

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 three-quarters full.

To edit the remainder of your file, you must write edited lines in memory to disk. Then you can load additional unedited lines from disk into memory by using the Append command.

Note: If you do not specify the number of lines, lines will be written until memory is 3/4 full. No action will be taken if available memory is already more than 3/4 full. All lines are renumbered, so that the first remaining line becomes line number 1.

Introduction When EDLIN finds an error, one of the following error messages is displayed:

Error Message	Cause	Cure
Cannot Edit .BAK file--Rename File	You attempted to edit a file with a filename extension of .BAK. The .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.
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 drive or an illegal filename.	Check the command line that started EDLIN for illegal filename and illegal drive entries. If the command is no longer on the display and if you have not yet typed a new command, the EDLIN start command can be recovered by pressing [F3] If this command line contains no illegal entries, run the CHKDSK program for the specified drive. If the status report shows that the directory is full, replace the diskette or use another drive.

Error Message	Cause	Cure
Line Too Long	During a Replace command, the string given as the replacement caused the line to expand beyond the limit of 253 characters. EDLIN canceled the Replace command	Divide the long line into two lines, then try the Replace command twice
Disk Full--File Write Not Completed	You gave the End command, but the drive did not contain enough free space for the whole file EDLIN canceled the E command and returned you to the operating system Some of the file may have been written to the drive	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 will not be available after this error Always be sure that the drive has sufficient free space for the file to be written to the drive before you begin your editing session
Invalid Drive Name Or File	You have not specified a valid drive or filename when starting EDLIN.	Specify the correct drive or filename
Filename Must Be Specified	You did not specify a filename when you started EDLIN.	Specify a filename
Invalid Parameter	You specified a switch other than /B when starting EDLIN.	Specify the /B switch when you start EDLIN

Error Message	Cause	Cure
Insufficient Memory	There is not enough memory to run EDLIN.	You must free some memory by writing files to another drive or by deleting files before restarting EDLIN
File Not Found	The filename specified during a Transfer command was not found.	Specify a valid filename when issuing a Transfer command
Must Specify Destination Number	A destination line number was not specified for a Copy or Move command	Reissue the command with a destination line number
Not Enough Room To Merge The Entire File	There was not enough room in memory to hold the file during a Transfer command.	You must free some memory by writing some files to the drive or by deleting some files before you can transfer this file
File Creating Error	The EDLIN temporary file cannot be created	Check to make sure that the directory has enough space to create the temporary file. Also, make sure that the file does not have the same name as a subdirectory in the directory where the file to be edited is located.

PREPARING DOS PROGRAMS

Introduction This chapter introduces you to two methods of preparing DOS programs, the linker program, known as LINK and the EXE2BIN Command. The chapter is designed to aid programmers and other computer professionals. The Microsoft Macro Assembler and Utilities manuals referred to in this chapter are not a part of North Star NetWare.

Overview

If you want to compile and link programs, you can use the linker program, called LINK.

LINK:

- o Combines separately produced object modules into one relocatable load module--a program you can run
- o Searches library files for definitions of unresolved external references
- o Resolves external cross-references
- o Produces a listing that shows both the resolution of external references and error messages

Note: LINK commands are optional and they may not have been installed on your system. Check with your system supervisor.

Description

When you write a program, you write it in source code. This source code is passed through a compiler which produces object modules. The object modules must be passed through the link process to produce machine language that the computer can understand directly. This machine language is in the form required for running programs.

You may wish to link (combine) several programs and run them together. Each of your programs may refer to a symbol that is defined in another object module. This reference is called an external reference.

Description (cont.)

LINK combines several object modules into one relocatable load module, or Run file (called an .EXE or Executable file). As it combines modules, LINK makes sure that all external references between object modules are defined. LINK can search several library files for definitions of any external references that are not defined in the object modules.

LINK also produces a List file that shows external references resolved, and it also displays any error messages.

LINK uses available memory as much as possible. When available memory is exhausted, LINK creates a temporary file named VM.TMP.

DOS Memory Divisions

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 a high-level language, you will not need to know these terms. If you are writing and compiling programs in assembly language, however, you will need to understand LINK and the the definitions described below. The section LINK in the Macro Assembler Manual also contains useful information on how LINK works.

In DOS, memory can be divided into segments, classes, and groups.

Example

Segment Number	Segment Name	Segment Class Name
Segment 1	PROG.1	CODE
Segment 2	PROG.2	CODE
Segment 12	PROG.3	DATA

Note: 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 (i.e., 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.

Example (cont.)

During the processing, LINK references segments by their addresses in memory (where they are located). LINK does this by finding groups of segments.

DOS Groups

A group is a collection of segments that fit within a 64K byte area of memory. The segments do not need to be contiguous to form a group. The address of any group is the lowest address of the segments in that group. At link time, LINK analyzes the groups, 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.

Features LINK:

- o Works with one or more input files
- o Produces two output files
- o May create a temporary disk file
- o May be directed to search up to eight library files

Input File Extensions If no filename extensions are given in the input (object) file specifications, LINK will recognize the following extensions by default:

Output File Extensions LINK appends the following default extensions to the output (Run and List) files:

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 will create a temporary file, name it VM.TMP, and put it on the default drive. If LINK creates VM.TMP, it will display the message:

```
VM.TMP has been created.
Do not change diskette in drive, <d:>
```

If you are using a diskette, once this message has been displayed, you must not remove the diskette from the default drive until the link session ends. If the diskette is removed, the operation of LINK will be unpredictable, and LINK might display the error message:

```
Unexpected end of file on VM.TMP
```

VM.TMP (Temporary) File (cont.)

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 will delete the VM.TMP already on disk and create a new VM.TMP. Thus, the contents of the previous VM.TMP file will be lost.

Types of Input

LINK requires two types of input: a command to start LINK and responses to command prompts. In addition, seven switches control LINK features. Usually, you will type all the commands to LINK on the keyboard. As an option, answers to the command prompts and any switches may be contained in a response file. Command characters can be used to assist you while giving commands to LINK.

Methods of Starting

LINK may be started in any of three ways. The first method is to type the commands in response to individual prompts. In the second method, you type all commands on the line used to start LINK. To start LINK by the third method, you must create a response file that contains all the necessary commands and tell LINK where that file is when you start LINK.

Summary of Methods to Start LINK:

Method 1	LINK
Method 2	LINK <filenames>[/switches]
Method 3	LINK @<filespec>

Method 1 Prompts

To start LINK with Method 1, type:

LINK[Retrn]

LINK will be loaded into memory. LINK will then display four text prompts that appear one at a time. You answer the prompts to command LINK to perform specific tasks.

At the end of each line, you may type one or more switches, preceded by the switch character, a forward slash.

The command prompts are summarized below and 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 blank spaces or plus signs (+) If a plus sign is the last character typed, the prompt will reappear There is no default, a response is required
Run File [Object-file.EXE]·	Give filename for executable object code The default is the first object-filename EXE (You cannot change the output extension)
List File [Run-file.MAP]	Give filename for listing The default is NUL MAP. If you want to create a list file, you must specify a filename
Libraries [ ]:	List filenames to be searched, separated by blank spaces or plus signs (+) If a plus sign is the last character typed, the prompt will reappear. The default will search for default libraries in the object modules (Extensions will be changed to LIB)

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 this syntax:

LINK(object-list>,<runfile>,<listfile>,
<lib-list>[/switch...][Retrn]

object-list--a list of object modules, separated by plus signs

runfile--the name of the file to receive the executable output

listfile--the name of the file to receive the listing

lib-list--a list of library modules to be searched, separated by plus signs

/switch--refers to optional switches, which may be placed following any of the response entries (just before any of the commas or after the <lib-list>, as shown)

The entries following LINK are responses to the command prompts. The entry fields for the different prompts must be separated by commas.

To select the default for a field, simply type a second comma with no spaces between the two commas.

Example LINK FUN+TEXT+TABLE+CARE/P/M,FUNLIST, COBLIB.LIB[Retrn]

This command causes LINK to be loaded, then the object modules FUN.OBJ, TEXT.OBJ, TABLE.OBJ, and CARE.OBJ are loaded. LINK then pauses (as a result of using the /P switch). LINK links the object modules when you press any key, and produces a global symbol map (the /M switch); defaults to FUN.EXE Run file; creates a List file named FUNLIST.MAP; and searches the Library file COBLIB.LIB.

Method 3 Response File

To start LINK with Method 3, type:

LINK @<filespec>[Retrn]

where: filespec is the name of a response file. A response file contains answers to the LINK prompts (shown in Method 1) and may also contain any of the switches. 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 switches and command characters in the response file the same way as they are used for responses typed on the keyboard.

When the LINK session begins, each prompt will be 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 [Retrn]), LINK will display the prompt which does not have a response, then waiting for you to type a legal response. When a legal response has been typed, LINK continues the link session.

Method 3	FUN TEXT TABLE CARE
Response	/PAUSE/MAP
File	FUNLIST
(cont.)	COBLIB.LIB

This response file tells LINK to load the four object modules named FUN, TEXT, TABLE, and CARE. LINK pauses before producing a public symbol map to permit you to swap disks (see discussion under /PAUSE in the "Switches" section before using this feature). When you press any key, the output files will be named FUN.EXE and FUNLIST.MAP. LINK will search the library file COBLIB.LIB, and will use the default settings for the switches.

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 blank space may be used to separate object modules.) To type a large number of responses (each may be very long), type a plus sign[RETRN] at the end of the line to extend it. If the plus sign[RETRN] is the last entry following these two prompts, LINK will prompt you for more module 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 have been listed, be sure the response line ends with a module name and a [RETRN] and not a plus sign[RETRN]. For example:

```
Object Modules [.OBJ]:FUN TEXT TABLE CARE+[Retrn]
Object Modules [.OBJ]:FOO+FLIPFLOP+JUNQUE+[Retrn]
Object Modules [.OBJ]:CORSAIR[Retrn]
```

Semicolon

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

Note: Once the semicolon has been typed and entered (by pressing [Retrn]) 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 [Retrn]. For example, if you use:

```
Object Modules [.OBJ]: FUN TEXT TABLE CARE[Retrn]
Run Module [FUN.EXE]: ;[Retrn]
```

no other prompts will appear, and LINK will use the default values.

[Ctrl]C — Use [Ctrl]C 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]C to exit LINK then restart LINK. If the error has been typed but you have not pressed the [Retrn] key, you may delete the erroneous characters with [Del], but for that line only.

Introduction

LINK asks you for responses to four text prompts. When you have typed a response to a prompt and pressed [Retrn], the next prompt appears. When the last prompt has been 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 will display the appropriate error message.

LINK prompts you 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.

Object Modules Prompt

The Object Modules: prompt has no preset filename response and requires you to type a filename.

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 (+).

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 will be read by LINK. Refer to the Macro Assembler Manual for more information on this process.

Run File Prompt

Typing a filename will create 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. For example:

Run File [FUN.EXE]: A:PAYROLL

This response directs the LINK to create the Run file PAYROLL.EXE on drive A.

List File Prompt

The List file contains an entry for each segment in the input (object) modules. Each entry also shows the addressing in the Run file.

The default response is NULL.MAP

Libraries Prompt

The valid responses are up to eight library filenames or simply a [Retrn]. (A [Retrn] means default library search.) Library files must have been created by a library utility. (Consult the LIB section of the Macro Assembler Manual for information on library files.) LINK assumes by default that the filename extension is .LIB for library files.

Library filenames must be separated by blank spaces of 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.

Libraries Prompt (cont.)

If LINK cannot find a library file on the drives, it will display the message:

Cannot find library <library name>Type new drive letter:

Press the letter for the drive (for example, D).

Introduction

The seven LINK switches control various LINK functions. Switches must be typed at the end of a prompt response, regardless of which method is used to start LINK. Switches may be grouped at the end of any response, or may be scattered at the end of several. If more than one switch is typed at the end of one response, each switch must be preceded by a forward slash (/).

All switches 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, in the /DSALLOCATE switch:

Legal	Illegal
/D	/DSL
/DS	/DAL
/DSA	/DLC
/DSALLOCA	/DSALLOCT

/DSALLO-CATE

Using the /DSALLOCATE switch tells LINK to load all data at the high end of the Data Segment (DS). 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 switch in combination with the default load low (that is, the /HIGH switch is not used) permits the user application to dynamically allocate any available memory below the area specifically allocated within the group of data segments, 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 the group of data segments.

/HIGH

Use of the /HIGH switch 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 switch with Pascal or FORTRAN programs.

/LINE-NUMBERS

/LINENUMBERS switch 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 (global) symbols defined in the input modules. If /MAP is not given, LINK will list only errors (including undefined globals).

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 switch causes LINK to pause in the link session when the switch is encountered. Normally, LINK performs the linking session from beginning to end without stopping. This switch allows you to swap diskettes (if you're using diskettes) before LINK outputs the Run (.EXE) file.

When LINK encounters the /PAUSE switch, it displays the message:

```
About to generate .EXE file
Change diskettes <hit any key>
```

LINK resumes processing when you press any key.

>> CAUTION <<

Do not remove the diskette which will receive the List file, or the diskette used for the VM.TMP file, if one has been created.

/STACK: <number>

Number represents any positive numeric value (in hexadecimal radix) up to 65536 bytes. If a value from 1 to 511 is typed, LINK will use 512. If the /STACK switch is not used for a link session, LINK will calculate 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.

/STACK: <number> (cont.)

At least one object (input) module must contain a stack allocation statement. If not, LINK will display the following error message:

WARNING: NO STACK STATEMENT

/NO

/NO is short for NODEFAULTLIBRARYSEARCH. This switch tells LINK to not search the default (product) libraries in the object modules. For example, if you are linking object modules in Pascal, specifying the /NO switch tells LINK to not automatically search the library named PASCAL.LIB to resolve external references.

Introduction This sample shows you the type of information that is displayed during a LINK session.

In response to the prompt, type:

LINK[Retrn]

This is an example of the messages and prompts the system might display (user responses are printed in boldface):

```
Microsoft 8086 Object Linker
Version 2.30  (C) Copyright Microsoft Corp 1983, 1984

Object Modules [.OBJ]: IO PROG[Retrn]
Run File [IO.EXE]:[Retrn]
List File [NUL.MAP]: IO /MAP[Retrn]
Libraries [.LIB]:;[Retrn]
```

Note: 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 switch, LINK gives you a listing of all line numbers for all modules. (Note that the /LINE switch can generate a large volume of output.)

By pressing [Retrn] in response to the Libraries: prompt, an automatic library search is performed.

Linker Map Once LINK locates all libraries, the linker map displays a list of segments in the order of their appearance within the load module. Type:

IO.MAP[Retrn]

The result is:

Start	Stop	Length	Name	Class
00000H	00001H	0002H	BASE	PROG
00002H	04495H	4494H	PROG	PROG
044A0H	06C6BH	27CCH	DATA	DATA
06C70H	06D6FH	0100H	STACK	DATA

Origin	Group
044A:0	DGROUP
0000:0	PGROUP

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 Macro Assembler and Utilities Manual for information on how to determine where relative zero is actuallly located, and also on how to determine the absolute address of a segment.

Public Symbols

Because the /MAP switch was used, LINK displays the public symbols by name and value. For example:

Address	Publics By Name
044A:254C	ARGC
0000:0002	C
0000:0617	Get_UIT
0000:04EA	HEAD_QUE
044A:046E	LOCAL_QU
0000:1F29	MOVMEN
0000:1CE6	READ_UIT
0000:0136	XCOVF

Address	Publics by Value
0000:0002	C
0000:0136	XCOVF
0000:04EA	HEAD_QUE
0000:0617	GET_UIT
0000:1CE6	READ_UIT
0000:1F29	MOVMEN
044A:046E	LOCAL_QU
044A:254C	ARGC

Error Messages Display

All errors cause the link session to be cancelled. After the cause has been found and corrected, LINK must be rerun. The following error messages are displayed by LINK:

Error Message	Explanation
Attempt to access data outside of segment bounds possibly bad object module	There is probably a bad Object file.
Bad numeric parameter	Numeric value is not in digits.
Cannot open temporary file	LINK is unable to create the file VM.TMP because the disk directory is full. Delete other unneeded files on the diskette or in the drive to make more space.
Error: DUP record too complex	DUP record in assembly language module is too complex. Simplify DUP record in assembly language program.
Error: 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).

Error Message	Explanation
Symbol defined more than once	LINK found two or more modules that define a single symbol name.
Program size or number of segments exceeds capacity of linker	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 greater than or equal to 64K bytes with the /STACK switch.
Segment size exceeds 64K	64K bytes is the addressing system limit.
Symbol table capacity exceeded	Very many and/or very long names were typed, exceeding the limit of approximately 25K 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 256 (segments and classes taken together).
Unresolved externals: <list>	The external symbols listed have no defining module among the modules or library files specified.

Error Message	Explanation
Too many segments or classes	The limit is 256 (segments and classes taken together).
Unresolved externals· <list>	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 you typed the /STACK switch.
Warning. segment of absolute or unknown type	There is a bad object module or an attempt has been made to link modules that LINK cannot handle (e.g., an absolute object module).
Write error in TMP file	No more space remains to expand the VM.TMP file
Write error on run file	Usually, there is not enough space for the Run file

Purpose

Converts .EXE (executable) files to binary format. This results in a saving of disk space and faster program loading.

Command Format

EXE2BIN <filespec>[<filespec>][Retrn]

Remarks

This command is useful only if you want to convert .EXE files to binary format. The file named by filespec is the input file. If no extension is specified, it defaults to .EXE. The input file is converted to .COM file format (memory image of the program) and placed in the output file. If you do not specify a drive, the drive of the input file will be used. If you do not specify an output filename, the input filename will be used. If you do not specify a filename extension in the output filename, the new file will be given an extension of .BIN.

The input file must be in valid .EXE format produced by the linker. The resident, or actual code and data part of the file must be less than 64K. There must be no STACK segment.

Kinds of Conversions

Two kinds of conversions are possible, depending on whether the initial CS:IP (Code Segment:Instruction Pointer) is specified in the .EXE file:

- o If CS:IP is not specified in the .EXE file,a pure binary conversion is assumed. If segment fixups are necessary (i.e., the program contains instructions requiring segment relocation), you will be prompted for the fixup value. This value is the absolute segment at which the program is to be loaded. The resulting program will be usable only when loaded at the absolute memory address specified by a user application. The processor will not be capable of loading the program.

- o If CS:IP is specified as 0000:100H, it is assumed that the file is to be run as a .COM file with the location pointer set at 100H by the assembler statement ORG; the first 100H bytes of the file are deleted. No segment fixups are allowed, as .COM files must be segment relocatable; that is, they must assume the entry conditions explained in the Macro Assembler Manual. Once the conversion is complete, you may rename the resulting file with a .COM extension. Then the command processor will be able to load and execute the program in the same way as the .COM programs supplied on your drive.

- o If CS:IP does not meet either of these criteria, or if it meets the .COM file criterion but has segment fixups, the following message will be displayed:

  File cannot be converted

  This message is also displayed if the file is not a valid executable file.

Explanation If EXE2BIN finds an error, one or more of the following error messages will be displayed:

Error Message	Explanation
File not found	The file is not on the drive specified.
Insufficient memory	There is not enough memory to run EXE2BIN
File creation error	EXE2BIN cannot create the output file Run CHKDSK to determine if the directory is full, or if some other condition caused the error.
Insufficient disk space	There is not enough disk space to create a new file
Fixups needed — base segment (hex):	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
File cannot be converted	The input file is not in the correct format.
WARNING — Read ERROR on EXE file	Amount read less than size in header This is a warning message only

THE DOS DEBUG COMMAND

Overview

This chapter is for programmers and computer professionals.

The DEBUG program (DEBUG) provides a controlled testing environment for binary and executable object files. Note that EDLIN is used to alter source files; DEBUG is EDLIN's counterpart for binary files. DEBUG eliminates the need to reassemble a program to see if a problem has been fixed by a minor change. It allows you to alter the contents of a file or the contents of a CPU register, and then to immediately reexecute a program to check on the validity of the changes.

All DEBUG commands may be cancelled at any time by pressing [Ctrl]C. [Ctrl]C suspends the display, so that you can read it before the output scrolls away. Entering any key other than [Ctrl]C or [Ctrl]S restarts the display. All of these commands are consistent with the control character functions available at the DOS command level.

Methods

DEBUG may be started two ways. By the first method, you type all commands in response to the DEBUG prompt (a hyphen). By the second method, you type all commands on the line used to start DEBUG.

Method 1
DEBUG

To start DEBUG using method 1, type:

DEBUG[Retrn]

DEBUG responds with the hyphen (-) prompt, signaling that it is ready to accept your commands. Since no filename has been specified, current memory, disk sectors, or files can be worked on by using other command.

>> CAUTION <<

When DEBUG is started, it sets up a program header at offset 0 in the program work area. On previous versions of DEBUG, you could over-write this header. You can still 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 will terminate.

Do not restart a program after the "Program terminated normally" message is displayed. You must reload the program with the N and L commands for it to run properly.

Method 2: Command Line (cont.)

To start DEBUG using a command line, type:

DEBUG [<filespec> [<arglist>][Retrn]

For example, if a <filespec> is specified, then the following is a typical command to start DEBUG:

DEBUG FILE.EXE[Retrn]

DEBUG then 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.

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

<filespec> <arglist>

Here, <filespec> is the file to be debugged, and the <arglist> is the rest of the command line that is used when <filespec> is invoked and loaded into memory.

Delimiter

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

```
DCS:100 110
D CS:100 110
D,CS:100,110
```

Table of Parameters

Parameter	Definition
<drive>	A one-digit hexadecimal value to indicate which drive a file will be loaded from or written to The valid values are 0-F hex These values designate the drives as follows 0=A, 1=B, 2=C, 3=D, ..0F=P
<byte>	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.

Table of Parameters (cont.)

Parameter	Definition
<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.

Table of Parameters *(cont.)*

Parameter	Definition
<range>	<address> <address>, or <address> [L <value>] where <value> is the number of lines the command should operate on, and L80 is assumed The L and value cannot be omitted 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 For example fcs·100 42 45 52 54 41

Table of Parameters (cont.)

Parameter	Definition
<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 string is legal 'This "string" is okay ' However, this string is illegal 'This 'string' is not ' Similarly, these strings are legal "This 'string' is okay " "This z ""string"" is okay " However, this string is illegal "This "string" is not " **Note:** The ASCII values of the characters in the string can be used as a <list> of byte values

Purpose Assembles 8086/8087/8088 mnemonics directly into memory.

Command Format A[<address>][Retrn]

Remarks If a syntax 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 cannot be abbreviated.

Remarks (cont.)

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

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

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

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

Psuedo-Instructions

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]
```

Pseudo-Instructions (cont.)

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
LDTBYTEPTR[BX]        ; This line will not
```

Purpose

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

Command Format

C<range> <address>[Retrn]

Remarks

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

Purpose Displays the contents of the specified region of memory.

Command Format D[<range>][Retrn]

Remarks 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[Retrn]

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[Retrn]

Remarks (cont.)

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[Retrn]

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

If then you type the command:

DCS:100 115[Retrn]

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.

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

Command Format E<address>[<list>][Retrn]

Remarks 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 to perform one of the following actions:

- o Replace a byte value with a value you type. 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.

- o Press the space bar to advance to the next byte. To change the value, simply type the new value as described above. If you space beyond an 8-byte boundary, DEBUG starts a new display line with the address displayed at the beginning.

- o 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.

- o Press [Retrn] to terminate the Enter command. [Retrn] may be pressed at any byte position.

Example Assume that the following command is typed:

ECS:100[Retrn]

DEBUG displays:

04BA:0100 EB._

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

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 [Retrn] ends the Enter command and returns to the DEBUG command level.

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

Command Format F<range> <list>[Retrn]

Remarks If the <range> contains more bytes than the number of values in the <list>, the <list> will be used repeatedly until all bytes in the <range> are filled. If the <list> contains more values than the number of bytes in the <range>, the extra values in the <list> 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[Retrn]

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.

Command Format G[=<address>[<address>...]][Retrn]

Remarks 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 DEBUGcan distinguish thestart =<address> from the breakpoint <address>es.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 G command uses an IRET instruction to cause a jump to the program under test. The user stack pointer is set, and the user flags, Code Segment register, and Instruction Pointer are pushed or is too small, the operating system may crash.) An interrupt code (0CCH) is placed at the specified breakpoint address(es).

Remarks (cont.)

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[Retrn]

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.

Purpose Performs hexadecimal arithmetic on the two parameters specified.

Command Format H<value> <value>[Retrn]

Remarks First, DEBUG adds the two parameters, then subtracts the second one 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[Retrn]

DEBUG performs the calculations and then displays the result:

02A9 0095

Purpose Inputs and displays one byte from the port specified by <value>.

Command Format I<value>[Retrn]

Remarks A 16-bit port address is allowed.

Example Assume that you type the following command:

I2F8[Retrn]

Assume also that the byte at the port is 42H. DEBUG inputs the byte and displays the value:

42

Purpose Loads a file into memory.

Command Format L[<address> [<drive> <record> <record>]][Retrn]

Remarks Set BX:CX to the number of bytes read. The file must have been named either when DEBUG was started or with the N command. Both the DEBUGinvocation and the N command format a filename properly in the normal format of a file control block at CS:5C.

If the L 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 L command is typed with an address parameter, loading begins at the memory <address> specified. If L 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[Retrn]
-NFILE.COM[Retrn]

Now, to load FILE.COM, type:

L[Retrn]

Example

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 0F 6D[Retrn]

DEBUG then loads 109 (6D hex) records 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 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 L command with no parameters causes DEBUG to load the file beginning at the address specified in the .HEX file. If the L command includes the option <address>, DEBUG adds the <address> specified in the L command to the address found in the .HEX file to determine the start address for loading the file.

Purpose

Moves the block of memory specified by <range> to the location beginning at the <address> specified.

Command Format

M<range> <address>[Retrn]

Remarks

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: 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 M 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[Retrn]

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 D command, using the <address> typed for the M command, to review the results of the move.

Purpose Sets filenames.

Command Format N<filename>[<filename>...][Retrn]

Remarks 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 file to be debugged, then the N<filename> 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
```

The results will be:

- o (N)ame assigns the filename FILE1.EXE to the filename to be used in any later Load or Write commands.
- o (N)ame also assigns the filename FILE1.EXE to the first filename parameter used by any program that is later debugged.
- o (L)oad loads FILE1.EXE into memory.
- o (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).

Chain of Commands

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 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 FILE2.DAT FILE3.DAT had been typed at the DOS command level.

Note: 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.

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.

Chain of Commands (cont.)

Note: 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[Retrn]
-NPARAM1 PARAM2/C[Retrn]
-G[Retrn]
-
```

In this case, the Go command executes the file in memory as if the following command line had been typed:

```
PROG PARAM1 PARAM2/C[Retrn]
```

Testing and debugging therefore reflect a normal runtime environment for PROG.COM.

Purpose Sends the <byte> specified to the output port specified by <value>.

Command Format O<value> <byte>[Retrn]

Remarks 16-bit port address is allowed.

Example Type:

O2F8 4F[Retrn]

DEBUG outputs the byte value 4F to output port 2F8.

Purpose Terminates the DEBUG utility.

Command Format Q[Retrn]

Remarks The Q command takes no parameters and exits DEBUG without saving the file currently being operated on. You are returned to the DOS command level.

Example To end the debugging session, type:

Q[RETURN]

DEBUG has been terminated, and control returns to the DOS command level.

Purpose Displays the contents of one or more CPU registers.

Command Format R[<register-name>][Retrn]

Remarks 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 [Retrn] 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 alphabetic code. To alter any flag, type the opposite two-letter code. The flags are either set or cleared.

The flags and their codes for SET and CLEAR are listed in the table on the next page.

Remarks

Flag Name	Set	Clear
Overflow	OV	NV
Direction	DN Decrement	UP Increment
Interrupt	EI Enabled	DI Disabled
Sign	NG Negative	PL Plus
Zero	ZR	NZ
Auxiliary Carry	AC	NA
Parity	PE Even	PO Odd
Carry	CY	NC

Whenever you type the 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 as 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:

[Retrn]

Flags for which new values were not entered remain unchanged.

Remarks (cont.)

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.

Example

Type:

R[Retrn]

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=0E00BX=00FFCX=0007 DX=01FFSP=039D BP=0000
SI=005C  DI=0000  DS=04BA ES=04BA SS=04BA  CS=O4BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21 INT 21
```

If you type:

RF[Retrn]

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[Retrn]**

Example (cont.)

DEBUG responds only with the DEBUG prompt. To see the changes, type either the R or RF command:

RF[Retrn]
NV UP EI PL NZ AC PE CY - _

Press:

[Retrn]

to leave the flags this way, or to specify different flag values.

Purpose

Searches the <range> specified for the <list> of bytes specified.

Command Format

S<range> <list>[Retrn]

Remarks

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[Retrn]
```

DEBUG will display a response similar to this:

```
04BA:0104
04BA:010D
-
```

Purpose

Executes one instruction and displays the contents of all registers and flags, and the decoded instruction.

Command Format

T[=<address>][<value>][Retrn]

Remarks

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:

[Retrn]

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=0E00BX=00FFCX=0007 DX=01FFSP=039D BP=0000
SI=005C  DI=0000  DS=04BA ES=04BA SS=04BA  CS=O4BA
IP=011A  NV UP DI NG NZ AC PE NC
04BA:011A  CD21          INT     21
```

If you type:

T=011A 10[Retrn]

Example (cont.)

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 [Ctrl]S suspends the display at any point, so that you can study the registers and flags for any instruction.

Purpose Disassembles bytes and displays the source statements that correspond to them, with addresses and byte values.

Command Format U[<range>][Retrn]

Remarks 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[Retrn]

DEBUG disassembles 16 bytes beginning at address 04BA:0100:

```
04BA:0100   206472   AND   [SI+72],AH
04BA:0103   69DB69
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
```

Example (cont.)

If you type:

U04ba:0100 0108[Retrn]

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
```

If the bytes in some addresses are altered, the disassembler alters the instruction statements. The U command can be typed for the changed locations, the new instructions viewed, and the disassembled code used to edit the source file.

Purpose Writes the file being debugged to a disk file.

Command Format W[<address>[<drive> <record> <record>]][Retrn]

Remarks If you type W 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 W command is typed with just an address, then the file is written beginning at that address. If a G or T command has been used, BX:CX must be reset before using the Write command without parameters.

Note: 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 N command (refer to the Name command earlier in this manual). 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 W 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 here--0=A:, 1=B;, 2=C:, etc.); DEBUG writes the

Remarks (cont.)

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.

>> CAUTION <<
Writing to absolute sectors is EXTREMELY dangerous because the process bypasses the file handler.

Example

Type:

W[Retrn]

DEBUG will write the file to disk, display the number of bytes written and then display the DEBUG prompt. The result might be

```
Writing 4A0F bytes
-
```

If you type:

WCS:100 1 37 2B[Retrn]

Example (cont.)

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:

```
Writing 1580 bytes
-
```

Chart 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	Explanation	Cause
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 G command Retype the Go command with ten or fewer breakpoints
BR	Bad register	You typed the R 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

GLOSSARY

access--the ability to work with files on the network disks; there are several access rights which may be assigned to network users

access time--the amount of time that the system takes to read data from or write data to disk storage

address--an identifying number for a starting location in computer memory; also a unique number which identifies a particular network station

application--a program that performs a specific task, such as word processing or accounts payable

architecture--the particular selection, design, and interconnection of the main components of a hardware/software system

artificial intelligence--a hardware/software system that is designed to emulate human thought processes; current AI research includes the area of "expert systems," or programs that can approximate the methodology of humans who are experts in a particular field

background writing--functions with disk caching to improve the speed of the network; instead of writing file changes directly to disk, so that a user has to wait for the disk write to end, changes to files are placed into dynamic memory then written to the hard disk, while the workstation user is working on some other task

backup--a piece of hardware or a stored file to be used in case of a malfunction or loss of data; also, to copy files onto a separate media in order to recopy the data from the backup source if the data is accidentally destroyed on the original source

banner--a one-page identification printed as the first page of a printout that tells who created the printout; banner information includes the user's name, the filename, the station number, and the date

binary--the numerical base by which a computer operates - base 2; since a computer's electrical circuitry only understands the on and off states, it operates on strings of on and off signals, which represent the numbers 1 and 0; the number twenty-nine is represented as 11101 in binary code.

binary number--method of representing a decimal number by using only two characters, 0 and 1; for instance, the number 2 in binary numbering is written as 10 (a 1 in the "two's place" and a 0 in the "one's place")

BIOS--basic input/output system; the part of the operating system in each workstation which performs the actual communication with input and output devices such as printers, disks, keyboards, and consoles

BIT--a binary digit; must be either a 0 or a 1; indicates one of two states--off or on

booting--to load a computer program into the random memory of a computer

bootstrap--a small program that starts a "cold" computer; generally the bootstrap program tells the computer where to start so that it can clear the machine memory, load the operating system, and set up the machine to begin work on a particular application

boot up--to load a computer program into the random memory of a computer

boot without floppy--ability to boot up without using a floppy diskette; can be from the network disk or from the local hard disk; see also REMOTE SYSTEM RESET.

buffer--a storage area in random access memory where the computer temporarily places data when it must transfer data from one device to another (usually during I/O operations)

byte--the basic unit of storage in a computer system, capable of storing one character (letter, digit or symbol)

cache buffer--an area in the memory of the file server into which large portions of disk files are read; increases the chance that a data request from a workstation can be read from the file server memory; since memory is much faster than disk, performance is greatly increased

central module--the main unit of the computer system

coaxial cable--a connecting cable consisting of two insulating layers and two conductors, with a central conductor wire surrounded by the first layer of insulation; an outer shielding conductor is layed over the first layer of insulation and then covered with the second layer of insulation.

cold boot--a complete reload of the operating system into memory

command--an instruction you enter which specifies the computer operation to be performed

communication--the act of sending and receiving data from one station to another

communication medium--the wiring or other means used to allow transmission of data; it may be wiring such as coaxial cable, twisted pair cable, or dual twisted pair cable; it may also be a non-wire type medium such as fiber optics, infrared, or microwave

concatenate--to link together, as in joining two word processing files

concurrent--true concurrency refers to programs (such as the NetWare Operating System) which allow many users to operate simultaneously, sharing network resources; such programs allow users to take turns using the resources--this happens so rapidly that the users perceive no delay; true concurrency requires a special operating system like the NetWare Operating System

also: referring to programs that share resources on single-user, stand-alone stations; such programs "co-operate" by taking turns using the single-station resources, as the user requests

configuration, hardware--the equipment to be used in a particular system and the way it is to be connected

configuration, software--a procedure performed to define the workstation's resources to the network

cursor--the "=" character that marks your place in a screen display

database--a collection of data organized and stored on disk by network users, usually through use of a special applications program

data buffer--area of computer memory used to store received data

default--a value or option assumed by the network system, when no other value is specified

default directory--the directory in which the operating system or application will search for or store a file or program unless some other directory is specified

default drive--the drive in which the operating system or application will search for or store a file or program unless some other directory is specified

destination--the network station to which data is sent

diagnostic--a procedure used to detect and isolate a malfunction or mistake

direct memory access--a technique which dramatically increases the speed of the network; the network processor passes parameters to a special DMA integrated circuit (chip) which controls the reading and writing of memory independent of the network processor; the network processor is left free to perform other tasks

directory caching--a method of decreasing the time it takes to determine a file's disk location; the file allocation table and directory entry table are written into the file server's memory; a file location is obtained by a memory read rather than by the slow disk read method

Note: Directory caching is not as efficient a method of locating files as directory hashing, particularly with large directories (with large directories hashing can be more than 10 times faster); directory caching also takes more memory (32 bytes a file, compared to 8 bytes for hashing); when faced with making a choice between the two, always choose hashing

directory hashing--a method of indexing disk file locations, thus decreasing the time needed to obtain the location of any one file; the system looks at only a few directory entries rather than searching sequentially through all entries (as it does in MS-DOS); the greater the number of files, the more this is necessary for speed

disable--preventing certain interrupts from occurring in a processing unit (such as a network communications card) by setting a switch, or a jumper, or through some other means

disk cache block--data written into the Cache Buffer during a disk read (see also CACHE BUFFER above and DISK CACHING below)

disk caching--reading portions of requested files into the file-server memory (cache buffer); this decreases the number of disk reads and increases the speed of access (Material placed in memory can be accessed up to 100 times faster than material on disk, greatly increasing network efficiency)

diskette--a thin, flexible magnetic storage medium.

diskette drive--a device that reads information from and writes information onto diskettes; the central module contains one diskette drive

DMA--see **direct memory access**

download--to call for, and receive, a file from another computer

dynamic memory--the same as RANDOM ACCESS MEMORY; this is the primary type of memory used in computer hardware; it is measured in kilobytes; when a program is loaded into the computer, it goes into RAM, from which it then operates; RAM is temporary--turning off the computer erases it

dynamic memory workspace--a predetermined amount of dynamic memory which is automatically reserved for internal operations when the NetWare Operating System is started; when you install NetWare you may specify that you need additional working memory to support unusual working conditions; see the installation chapter for further details

DOS--disk operating system

drive--a path which has been assigned to a drive letter as a faster method of specifying where files are located

drive letter--a letter between A and Z which may be assigned to represent a path

elevator seeking--a method of decreasing file access time when there are file requests from several users; greater speed of access is obtained by decreasing the amount of head movement in the disk; the system determines the closest read point on the disk, reads this first, then proceeds to the next closest read, and so on

enable--allowing certain interrupts to occur in a processing unit (such as a network communications card) by setting a switch, or a jumper, or through some other means

external DOS commands--these commands are not part of the operating system's internal command processor. To use an external DOS command, you must be logged onto the drive where the command resides or have a path set up to that drive.

field--a category of information in a data file; also, a prompt on a data entry screen; an employee records file might include three fields: name, address, and employee number.

file--a set of related information stored as one unit; a file might contain many separate records, such the addresses of all customers, or a continuous text record, such as a letter; a file might be stored on hard disk or magnetic tape and must always have a unique filename

file allocation table (FAT)--information on the hard disk that tells NetWare where a file is stored on the disk

file extension--a file extension consists of one to three characters which are separated from a filename by a period; file types are optional; if used, they follow a filename

filename--a filename consists of one to eight characters, optionally followed by a period and a file extension of up to three characters

file server--the personal computer (with hard disk) in which the NetWare Operating System is loaded; the file-server processor maintains complete control over the network files and shareable resources

fixed disk--see hard disk

floppy disk--another name for a flexible diskette, a removable magnetic storage medium

format--the act of preparing a floppy diskette, or a hard disk to accept data according to the tracking and sectoring method required by the operating system

gateway--an interface between two different network systems

group access--access to protected files which is available only to users belonging to a group that has been granted access

handshaking--the signals used by two communications systems to control the starting and stopping of data exchanges

hard copy--a paper printout; (it is "hard" in the sense that it can actually be held, as opposed to a screen display)

hard disk drive--a high-capacity magnetic storage device that allows a user to read, write, or erase data

hardware--physical equipment; the electronic and mechanical components of a network, such as the personal computers, network interface cards, cables, cable connectors, hub devices, floppy disk drives, and hard disk drives

hierarchy--a structure with several levels arranged in a tree-like form; the relative priority assigned to disk volumes

hub--a mechanical or electrical junction box used for connecting the workstations in some networks; hubs may be active or passive; active hubs condition, boost, and relay the network signal; passive hubs merely relay the signal

home directory--the directory to which a user's first network drive is automatically mapped when the user logs in to the network; the supervisor usually assigns this directory when setting up the network; the user may change this directory using the NPATH command

interface--as a verb: making two devices capable of communicating; used most often to refer to the design of hardware and software which allows connection of network components and transfer of information; as a noun: a specific hardware or software connection

internal DOS command--the most commonly used DOS commands; they are part of the operating system's command processor, so the system always know where to find them

I/O--input/output; the process of transmitting data into or out of storage for processing

K--kilo; a unit of measurement for memory or disk storage capacity; two to the tenth power (1024 in decimal)

local area network (LAN)--a collection of computers and computer equipment connected to communicate, share resources, and possibly to access remote hosts or other networks

lst: device capture mode--when a workstation sends a file to the NetWare file server to be printed on a shared printer, the workstation's LST: DEVICE (its printer) is in the capture mode (sent to file server for printing)

M--mega; a unit of measurement for memory or disk storage capacity; two to the twentieth power (1,048,576 in decimal)

macro--a command defined by the operator which allows the operator to use a few keystrokes to reproduce a long string of frequently repeated characters; for example, you might define a macro that allows you to print your name, address, and phone number by pressing one function key

map or **mapping**--setting any NetWare logical drive to point to a chosen directory; this is done with the NPATH command; see the NetWare User's Manual for details on the NPATH command

memory (also **internal memory**)--one of the essential components of a computer's central processing unit, memory is the area where information and programs are actively stored and processed; memory includes both ROM (Read Only Memory, which is permanent memory) and RAM (Random Access Memory which is temporary memory, the contents of which can be modified as needed)

menu--a screen produced by a program that presents a choice of operations to perform

modem--a contraction of modulator/demodulator; a hardware device which sends data via telephone lines from one computer station to another or to a network resource, such as a file server

mouse--a piece of hardware shaped to fit the hand comfortably; it eliminates the need to use the keyboard for cursor control; by using buttons on the face of the mouse, and sliding the mouse across a flat surface in the direction of desired cursor movement, the cursor can be positioned over objects on the computer screen

multitasking--operating systems that allow a user to initiate concurrent tasks; see also, **concurrent**

multi-user--used to describe operating systems that allow multiple users to share a computer's central processing unit, storage, memory, and file resources from separate workstations

NetWare--the multi-user, multitasking, local area network operating system by Novell Inc., which allows network users access to files created by different operating systems without requiring separate disk partitions

NetWare Shell (NET20? or NET30?)--the special NetWare software program which is loaded into the memory of each workstation and which works with DOS to enable the workstation to communicate with the file server

network--a system which sends and receives data and messages through a cable

network communication--data transmission between network stations; requests for services and data are passed from one network station to another through a communications medium

network console--the CRT (display screen) and keyboard of a DIMENSION workstation when it is operating in a special mode

network station--an individual personal computer connected within a network by a communications card and some communications medium

operating system--the most important piece of software; it controls the execution of all other programs

packet--the unit of information in the network; the method of storing and sending messages and requests for services in the network; NetWare features a 508-byte packet

parent directory--the directory above another directory; for example, SYS:ACCOUNTING\RECEIVABLES is a subdirectory of (part of) the SYS:ACCOUNTING directory; Thus, directory SYS:ACCOUNTING is the parent of subdirectory SYS:ACCOUNTING\RECEIVABLES

password--A confidential name that permits a person to log into the system.

password protection--a method of limiting login access to the network by requiring users to enter a password; unless the password is entered correctly, access to the network will be denied

path specification--the combination of directories and subdirectories that create a path to the location that a file is stored

port--an opening in the back of the central module to which a device, such as a printer, is connected

print spooler--the program in the file server that controls the sharing of printers and the priorities of printouts

program--A series of instructions to the computer to carry out one or more tasks

prompt--a character or message from the software which appears on the display screen and requires a user response; the DOS prompt ">", and "Type H for Help" are examples

public access--access to the network volumes which is available to all network users

random access memory--this is the primary type of memory used in computer hardware; it is measured in kilobytes; when a program is loaded into the computer, it goes into RAM, from which it then operates; RAM is temporary: turning off the computer erases it

RAM--another name for random access memory (see the definition above)

read-after-write verify--immediately after a file write, the network automatically reads whatever was written to disk and compares the contents of the new disk file with the contents of the portion of memory from which the file was written; if a discrepancy is encountered between the disk file and the memory file, the comparison is retried 5 times; if further comparisons indicate that the two files do not match, the network operating system informs the user that the disk file is bad

read-only--a type of data protection that allows data to be read but not modified

record--a collection of related information that is treated as a unit within a file

record locking--an important feature of the NetWare Operating System which prevents different users from gaining simultaneous access to the same record in a shared file, thus preventing overlapping disk writes and insuring data integrity

redundant directory structures--a method of insuring that disk files are not lost because a file allocation table is damaged; the disk file allocation table and the directory entry table are each written in two separate locations on the disk; if one of the tables is damaged, the system automatically warns the user and searches the second table for the correct file location

remote system reset--ability to boot a remote workstation onto the network using the network hard disk, instead of a diskette

resource--any computer or device that provides a service to the network

ROM--"Read Only Memory"; storage place for frequently used computer instructions and data; this is permanent memory; the data stays in ROM when the power is turned off; also, a ROM integrated circuit (or "IC" or "ROM chip")

scrolling--moving the content of a CRT (display screen) up or down, through the use of directional command keys

sectors--physical locations on disk storage media; the logical divisions of a disk's tracks; each track is divided into several sectors; in hard sectoring, the sectors are marked by holes in the floppy disk; in soft sectoring, a single hole enables the computer to read or write sector boundaries on the disk medium

shared printer--A printer connected to the central module.

software--computer programs; a set of stored instructions and procedures that can be recalled as needed for the computer processor to execute; contrast with "hardware"

stand-alone--a computer that operates without connection to another computer.

station--an individual personal computer connected within a network by a communications card and some communications medium

station number--a unique number which identifies a workstation to the central module

storage--a device or medium (such as a disk) which allows the storage and retrieval of data

subdirectory--any directory that is part of a higher directory; a directory may be a parent and subdirectory simultaneously (being part of a directory above yet having subdirectories below)

system monitor display--a formatted screen display shown on the network file-server CRT (display screen); the system monitor shows network activity and allows the management of network resources

system supervisor--the person(s) responsible for installing and maintaining the system.

token--a special burst of electricity that travels around to each network station on some networks; the "permission to send" required before transmitting messages on some networks

tracks--physical locations on disk storage media; concentric circles on a disk; the smallest physical division of data storage on a disk; (sectors are not actually physical divisions)

transaction--the read-modify-write cycle used to change shareable files; the three operations are performed as one indivisible operation in order to assure that multiple users may work on the same file without file contamination; the NetWare Operating System supports transactions as one means of operating on shareable files

transmit--to send electronic signals from station to station through the network communications medium (coaxial cable, twisted pair cable, fiber optics, infrared, microwave, etc.)

transparent--not noticed or seen by the user; refers to functions which are automatically executed by the network hardware and software

virtual screen formatting--a method of handling data and displaying it on a monitor screen; this method allows the user to scroll the data back and forth on the screen, as the user wishes

volume--a fixed area of a hard disk; the size of each volume is determined during disk initialization

Winchester disk--a fixed rigid disk that is sealed within the computer or within a separate peripheral; this type of disk is more expensive than a floppy disk, but offers greater storage capacity and more rapid data access

workstation--any individual personal computer except the file-server, which is connected within a NetWare network by a communications card and some communications medium

workstation address--a unique number assigned to each network station; this number is usually assigned by the network installer, by setting a switch, or a jumper on the network interface card, or through some other method; usually this number must be between 1 and 255, depending on the type of network in use; see also: ADDRESS

write--to record data onto a hard disk or diskette

DOS DISK AND DEVICE ERRORS

Introduction If a disk or device error occurs at any time during a command or program, DOS displays an error message in the following format:

```
<type> <action> drive <x>
Abort,Ignore,Retry:_
```

In this message, <type> may be one of the following:

Write protect error
Bad unit error
Not ready error
Bad command error
Data error
Bad call format error
Seek error
Non-DOS disk error
Sector not found error
No paper error
Write fault error
Read fault error
General failure
Sharing violation
Lock violation
Invalid disk change
FCB unavailable

The <action> may be either of the following:

READING
WRITING

The drive indicates the drive in which the error occurred.

Responses DOS waits for you to respond in one of the following ways:

A Abort. End 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. This response should be used if you have corrected the error (such as with NOT READY or WRITE PROTECT errors).

Usually, you will want to recover by typing responses in this order:

R (to try again)
A (to terminate program and try a new disk)

One other error message might be related to faulty disk read or write:

FILE ALLOCATION TABLE BAD FOR DRIVE x

This message means that the copy in memory of one of the allocation tables has pointers to nonexistent 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.

ANSI ESCAPE SEQUENCES

Introduction

An ANSI escape sequence is a series of characters (beginning with an escape character or keystroke) that you can use to define functions to DOS. Specifically, you can reassign keys, change graphics functions, and affect cursor movement.

This appendix explains how the ANSI escape sequences are defined for DOS. Examples of how to use ANSI escape sequences are included at the end of this appendix.

Note: DOS uses a default value when you do not specify a value or when you specify zero.

Pn represents "numeric parameter." This is a decimal number specified with ASCII digits.

Ps represents "selective parameter." This is any decimal number that is used to select a subfunction. Multiple subfunctions may be selected by separating the parameters with semicolons.

Pl represents "line parameter." This is a decimal number specified with ASCII digits.

Pc represents "column parameter." This is a decimal number specified with ASCII digits.

Cursor Functions

The following escape sequences affect the cursor position on the screen:

CUP - Cursor Position

ESC [Pl ; Pc H

HVP - Horizontal & Vertical Position

ESC [Pl ; Pc f

CUP and HVP move the cursor to the position specified by the parameters. The first parameter specifies the line number, and the second parameter specifies the column number. The default value is 1. When no parameters are specified, the cursor moves to the home position.

CUU - Cursor Up

ESC [Pn A

This sequence moves the cursor up one line without changing columns. The value of Pn sets the number of lines moved. The default value for Pn is 1. DOS ignores the CUU sequence if the cursor is already on the top line.

CUD - Cursor Down

ESC [Pn B

This sequence moves the cursor down one line without changing columns. The value of Pn sets the number of lines moved. The default value for Pn is 1. DOS ignores the CUD sequence if the cursor is already on the bottom line.

Cursor Functions (cont.)

CUF - Cursor Forward

ESC [Pn C

The CUF sequence moves the cursor forward one column without changing lines. The value of Pn sets the number of columns moved. The default value for Pn is 1. DOS ignores the CUF sequence if the cursor is already in the far right column.

CUB - Cursor Backward

ESC [Pn D

This escape sequence moves the cursor back one column without changing lines. The value of Pn sets the number of columns moved. The default value for Pn is 1. DOS ignores the CUB sequence if the cursor is already in the far left column.

DSR - Device Status Report

ESC [6 n

The console driver will output a CPR sequence (see below) on receipt of the DSR escape sequence.

CPR - Cursor Position Report (from console driver to system)

ESC [Pn ; Pn R

The CPR sequence reports current cursor position using standard input. The first parameter specifies the current line and the second parameter specifies the current column.

Cursor Functions (cont.)

SCP - Save Cursor Position

ESC [s

The current cursor position is saved. This cursor position can be restored with the RCP sequence (see below).

RCP - Restore Cursor Position

ESC [u

This sequence restores the cursor position to the value it had when the console driver received the SCP sequence.

Erasing

The following escape sequences affect erase functions.

ED - Erase Display

ESC [2 J

The ED sequence erases the screen, and the cursor goes to the home position.

EL - Erase Line

ESC [K

This sequence erases from the cursor to the end of the line (including the cursor position).

Modes of Operation

The following escape sequences affect screen graphics.

SGR - Set Graphics Rendition

ESC [Ps ; ... ; Ps m

The SGR escape sequence calls the graphic functions specified by the parameters described on the following page. The graphic functions remain until the next occurrence of an SGR escape sequence.

SGR Escape Sequence

Parameter	Parameter Function
0	All Attributes off
1	Bold on
2	Faint on
3	Italic on
5	Blink on
6	Rapid blink on
7	Reverse Video on
8	Concealed on (ISO 6429 standard)
30	Black foreground (ISO 6429 standard)
31	Red foreground (ISO 6429 standard)
32	Green foreground (ISO 6429 standard)
33	Yellow foreground (ISO 6429 standard)
34	Blue foreground (ISO 6429 standard)
35	Magenta foreground (ISO 6429 standard)
36	Cyan foreground (ISO 6429 standard)
37	White foregound (ISO 6429 standard)
40	Black background (ISO 6429 standard)
41	Red background (ISO 6429 standard)
42	Green background (ISO 6429 standard)
43	Yellow background (ISO 6429 standard)
44	Blue background (ISO 6429 standard)
45	Magenta background (ISO 6429 standard)
46	Cyan background (ISO 6429 standard)
47	White background (ISO 6429 standard)
48	Subscript
49	Superscript

Modes of Operation (cont.)

SM - Set Mode

ESC [= Ps h
or ESC [= h
or ESC [= 0 h
or ESC [? 7 h

The SM escape sequence changes the screen width or type to one of the following:

Parameter	Parameter Function
0	40 x 25 black and white
1	40 x 25 color
2	80 x 25 black and white
3	80 x 25 color
4	320 x 200 color
5	320 x 200 black and white
6	640 x 200 black and white
7	wrap at end of line

RM - Reset Mode

ESC [= Ps l
or ESC [= l
or ESC [= 0 l
or ESC [? 7 l

Parameters for RM are the same as for SM (Set Mode), except that parameter 7 will reset the wrap at the end of line mode.

Keyboard Reassignment

Although not part of the ANSI 3.64-1979 or ISO 6429 standard, the following keyboard reassignments are compatible with these standards.

The control sequence is:

```
ESC [ Pn ; Pn ; ... Pn p
  or  ESC [ "string" ; p
  or  ESC [ Pn ; "string" ; Pn ; Pn ; "string" , Pn p
  or  any other combination of strings and decimal
       numbers
```

The final code in the control sequence (p) is one reserved for private use by the ANSI 3.64-1979 standard.

The first ASCII code in the control sequence defines which code is being mapped. The remaining numbers define the sequence of ASCII codes generated when this key is intercepted. Note that there is one exception: if the first code in the sequence is zero (NUL), then the first and second code make up an extended ASCII redefinition. For example:

Reassign the Q and q key to the A and a key (and vice versa):

```
ESC [ 6 5 ; 8 1 p      A becomes Q
ESC [ 9 7 ; 1 1 3 pa becomes q
ESC [ 8 1 ; 6 5 p      Q becomes A
ESC [ 1 1 3 ; 9 7 pq becomes a
```

Reassign the F10 key to to a DIR command followed by a carriage return:

```
ESC [ 0 ; 6 8 ; " d i r " ; 1 3 p
```

The 0;68 is the extended ASCII code for the F10 key; 13 decimal is a carriage return.

HOW TO CONFIGURE YOUR SYSTEM

Introduction

The DOS configuration file (CONFIG.SYS) allows you to add device drivers which will be activated for you when you log into the system. The configuration file is simply an ASCII file that has certain commands for DOS start-up.

A system initialization routine reads the configuration file (CONFIG.SYS), if it exists, to perform device installation and other user options. Its final task is to execute the command interpreter, which finished the DOS start-up (boot) process.

Any CONFIG.SYS file in the user's home directory will be executed. The presence of an AUTOEXEC.BAT will not affect the execution of CONFIG.SYS.

Changing the File

The default CONFIG.SYS file is located in SYS:PUBLIC. Copy it from SYS:PUBLIC to your home directory, then choose the commands to add or change from the list below:

BUFFERS=<number>	The number of sector buffers DOS will use for disk buffering.
FILES = <number>	The maximum number of files that DOS allows to be open; the default is 8.
DEVICE = <filename>	Installs the device driver in <filename> into the system list.
BREAK=<ONorOFF>	If ON is specified (default is OFF), a check for [Ctrl]C as input will be made each time the system is called. ON improves the ability to cancel programs.

Changing the File (cont.)	SHELL = <filename>	Begins execution of the shell (top level command processor) from <filename>.

Example

A typical configuration file might look like this:

```
Device = ZSTAR.DRV
Break  = ON
Device = ANSI.SYS
```

COMMAND INDEX

Overview

This section contains a list of all NetWare and DOS commands, a short statement of each command's purpose, and a reference to the chapter where the complete command description may be found.

Note: Some of the DOS commands do not work over a computer network. If you try to use these commands, DOS displays this error message:

Cannot <command> to a network device

where <command> is the name of the command you typed. The commands that do not work over a network are:

- o Chkdsk
- o Diskcopy
- o Format
- o Recover

All commands are external except as noted.

ATTRIB	Sets or displays attributes of a file (Chapter 5).
BREAK	Sets [Ctrl]C check (internal--Chapter 8).
CASTOFF	Disables message interrupts from other stations (Chapter 3).
CASTON	Enables message interrupts from other stations (Chapter 3).
CHDIR (CD)	Changes directories; prints working directory (internal--Chapter 4).
CHKDSK	Scans the directory of the default or designated drive and checks for consistency (Chapter 4).
CHKVOL	Displays disk allocation statistics for a specified volume (Chapter 4).
CLS	Clears the screen (internal--Chapter 8).
COMMAND	Processes internal DOS commands (internal--Chapter 8).
COPY	Copies the file(s) specified (internal--Chapter 5).
CTTY	Changes the console TTY (internal--Chapter 8).
CURSOR	Changes the size of the cursor (Chapter 8).
DATE	Displays and sets the date (internal--Chapter 8).
DEL (ERASE)	Deletes the file(s) specified (internal--Chapter 5).
DIR	Lists requested directory entries (internalChapter 4).
DISKBOOT	Allows programs that must be run from diskette to operate (Chapter 6).
DISKCOMP	Compares diskettes (Chapter 6).

DISKCOPY	Copies disks (Chapter 6).
ECHO	Turns batch echo feature on and off (internal--Chapter 10).
ENDSPOOL	Terminates the LST: device capture mode and places the captured file in the spool queue (Chapter 7).
ERASEDIR	Removes an empty subdirectory from the system directory tree (Chapter 4).
EXE2BIN	Converts executable files to binary format (Chapter 12).
EXIT	Exits command processor and returns to previous level (internal--Chapter 8).
FC	Compares two text files (Chapter 5).
FIND	Searches for a constant string of text (Chapter 8).
FLAG	Displays or modifies a file's access attributes (Chapter 5).
FLAGDIR	Displays or modifies a subdirectory's access attributes (Chapter 4).
FOR	Batch command extension (internal--Chapter 10).
FORMAT	Formats a disk to receive DOS files (Chapter 6).
HELP	Provides on-line help for the networking commands in NetWare (Chapter 2).
GRANT	Grants trustee privileges for a directory to a user or group (Chapter 4).
GOTO	Batch command extension (internal--Chapter 10).
IF	Batch command extension (internal--Chapter 10).

LARCHIVE	Archives system files to local disk(s) (Chapter 5.)
LINK	Links object modules (Chapter 12).
LISTDIR	Lists the names of subdirectories beneath the specified path (Chapter 4).
LOCAL	Configures printers, modems, or "mice" for use at a workstation (Chapter 7).
LOGOUT	Logs a user out of the network (Chapter 2).
LRESTORE	Restores files that were archived to local disks using the LARCHIVE command (Chapter 5).
MAKEDIR	Makes a new subdirectory on the network (internal--Chapter 4).
MKDIR (MD)	Makes a directory on a non-network drive (internal--Chapter 4).
MORE	Displays output one screen at a time (Chapter 8).
NARCHIVE	Archives system files to system volumes and subdirectories (Chapter 5).
NCOPY	Copies files between network drives (Chapter 5).
NPATH	Assigns or maps a drive to a directory path (Chapters 2, 4, and 9).
NPRINT	Queues network files for printing at shared printers (Chapter 7).
NRESTORE	Restores files that were archived to system volumes (Chapter 5).
NSNIPES	Provides a game for entertainment (Chapter 2).

PATH	Sets a command search path (Chapter 4).
PAUSE	Pauses for input in a batch file (internal--Chapter 10).
PRINT	Prints text files (Chapter 7).
PRINTER	Configures the shared printers (Chapter 7).
PROMPT	Assigns a default prompt (internal--Chapter 8).
PURGE	Removes files from the buffer where they are stored after the DEL command (Chapter 5).
Q	Displays the network print spooler information, including a list of files in the queue; also deletes unwanted entries from the print queue (Chapter 7).
RECOVER	Recovers a bad disk or file (Chapter 5).
RELEASE	Releases temporary "ownership" of the shared diskette drive (Chapter 6).
REM	Displays a comment in a batch file (internal--Chapter 10).
REN (Rename)	Renames first file as second file (internal--Chapter 5).
RENDIR	Renames first subdirectory as second subdirectory (internal--Chapter 4).
REQUEST	Requests temporary "ownership" of the shared diskette drive (Chapter 6).
RESTORE	Restores backed up files (Chapter 5).
REVOKE	Revoke trustee privileges for a directory from a user or group (Chapter 4).
RIGHTS	Displays the rights you have in a particular directory (Chapter 2).

RMDIR (RD)	Removes a DOS directory (internal--Chapter 4).
SALVAGE	Recovers files deleted by the DEL command (Chapter 5).
SEND	Sends messages to other workstations (Chapter 3).
SET	Sets one string value to another in the environment, or displays the environment (internal--Chapter 8).
SETLOGIN	Customizes your log in script (Chapter 9).
SETPASS	Allows you to change your password (Chapter 2).
SHIFT	Increases number of replaceable parameters in batch process (internal--Chapter 10).
SHOWDIR	Displays the directory structure of the specified volume, drive, or path (Chapter 4).
SORT	Sorts data forward or backward (Chapter 8).
SPOOL	Routes the LST: device to the system print spooler (Chapter 7).
SYSTIME	Displays and sets the workstation clock to the current system time and date (Chapter 2).
TIME	Displays and sets the time (internal--Chapter 8).
TLIST	Displays the user and group trustees and their privileges for the specified directory (Chapter 4).
TYPE	Displays the contents of a file (internal--Chapter 8).
UDIR	Makes a global file search (Chapter 4).
USERLIST	Displays a list of users currently logged into the system (Chapter 2).

VER	Prints DOS version number (internal--Chapter 8).
VERIFY	Verifies all writes to a disk (internal--Chapter 8).
VOL	Displays the volume ID, if any (internal--Chapter 8).
VOLINFO	Displays disk allocation statistics for all volumes on the server (Chapter 4).
WHOAMI	Displays log in information for the requesting station (Chapter 2).

USING AN EXTERNAL PC WORKSTATION ON THE NETWORK

Introduction

The North Star NetWare Operating System features the ability to incorporate IBM PC/XT or compatible workstations into the network through the use of additional cabling and circuit boards. External workstations are subject to some special considerations when used in a North Star network.

There are four major differences in operation between external and North Star workstations. These are:

- External workstations boot (start up) differently--you must prepare a special boot disk as described on the next page
- External workstations cannot use the shared diskette drive in the central module--it is a local resource for North Star workstations only
- There is no implicit spooling from external workstations --you must use the SPOOL and ENDSPOOL commands described in Chapter 7 if you want to print to network printers
- The CONSOLE commands used by the system supervisor will not run from external workstations, only from North Star workstations

Instructions for installing hardware for external workstations are contained in the North Star NetWare Supervisor's Manual.

Instructions for booting an external workstation begin on the next page.

Procedure: Creating a NetWare Boot Diskette

1. Follow the instructions in your computer's User's Manual to make a copy of your boot diskette. Be sure you know which version of PC- or MS-DOS you will be running in your workstation (you'll need this information in step #5 below.

2. Obtain your worksheet #3 from your system supervisor. This worksheet gives you your network username and password and shows your network drive connections.

3. Go to a North Star workstation near the central module (for easy access to the shared diskette drive). Enter your username and password.

4. At the C>, type:

   REQUEST D[Retrn]

   and insert your copy of your boot diskette into the diskette drive in the central module.

5. Add the network shell to your boot diskette by copying the appropriate network COM file from the public drive. If your workstation will be running:

   PC- or MS-DOS 2.xx copy NET20O.COM
   PC- or MS-DOS 3.0x copy NET30O.COM
   PC- or MS-DOS 3.1x copy NET31O.COM

   A sample copy routine is:

   C> NCOPY SYS:LOGIN\NET31O.COM A:[Retrn]

--->

Procedure: Creating a NetWare Boot Diskette

6. Copy the LOGIN.EXE file from the public drive in the same manner.

   C> NCOPY SYS:LOGIN\LOGIN.EXE A:[Retrn]

7. Type

   RELEASE D[Retrn]

   and remove your boot diskette from the diskette drive.

8. Type

   LOGOUT[Retrn]

   and go to your external workstation.

9. Insert your boot diskette into your local diskette drive and boot as described in your User's Manual. At the DOS prompt, type

   NET31O[Retrn]

   (Be sure to type the same filename that you copied. Do not type the COM extension.) This loads the network shell into your workstation's memory.

Logging In

When you boot your workstation, DOS automatically assigns drive designations for your available local drives.

Regardless of the total number of drives your workstation has, DOS always assigns a number of local drives, depending on the type of workstation. If you have an IBM-PC/XT, DOS will assign drives A and B as local drives, and the local hard disk as drive C. DOS 3.0 always assigns drives A through E as local drives.

In order to work on the network you must change your default (current) drive to a network drive.

Network drives begin as the next letter in sequence after your local drives -- if you use DOS 3.0, which defines five local drives, the first network drive would be drive F.

To change to any drive, simply type the drive letter (C, F, etc.) followed by a colon (:), then press [Retrn].

The DOS drive prompt will change to the letter of the drive you typed, such as **C>** or **F>**.

If your workstation does not have more than two floppy drives and does not have a local hard disk, you should change to drive C (which will be your first network drive), by typing

C:[Retrn]

If your workstation has no more than two floppy drives and it has a local hard disk, you should change to drive D (which will be your first network drive), by typing

D:[Retrn]

Logging In (cont.)

If your workstation has five floppy drive designations and it has a local hard disk, you should change to drive G (which will be your first network drive), by typing

G:[Retrn]

Now, to log in to the network, type the following:

LOGIN <*your username*>[Retrn]

If the supervisor assigned you a password, the following message will appear:

Enter your password:

If the supervisor did not assign you a password, the message will not appear.

If the password message appears, type in your password (which the supervisor supplied you), then press [Retrn].

If your supervisor set up a search path for you when he installed the system, a message similar to the following will appear on the screen:

Search1 := Z:. [SYS:PUBLIC]

Note: Usually the supervisor will set up two drive mappings for you when the network is installed. One of these drive mappings is for the Search drive [Z:] just mentioned above. The other drive mapping should be for your first network drive. The supervisor will usually map your first network drive to a home directory which was created for you.

To display a full listing of your drive mappings, type:

NPATH[Retrn]

Logging In (cont.)

NetWare will respond with a list of your mapped network drives, showing the directories the drives are mapped to.

As already mentioned, your first network drive should be mapped to your home directory, and your first network Search drive (drive Z) should be mapped to SYS:PUBLIC. If this is not the case, check with your supervisor to find out why your drives are mapped as they are.

Automatic Booting

If you must boot from a floppy diskette because you do not have a workstation which is equipped with Remote System Reset hardware, you may set up your own "Automatic Boot" batch file on your boot diskette.

If you set up an Automatic Boot batch file, you will be able to load DOS and the network shell into your station automatically, just as if you had booted by Remote Reset.

You can even run the LOGIN command as part of your batch file.

First, read the instructions in your DOS manual on setting up an AUTOEXEC.BAT file with the COPY CON command. Then set up an AUTOEXEC.BAT file for the version of DOS and the network shell which you will be using.

For example, if you will be running DOS 2.X on your workstation, you would type the following:

COPY CON: AUTOEXEC.BAT (press [Retrn])
NET20O (press [Retrn])
^Z (press [Retrn])

Note: The ^Z at the end of the above file is a Control/Z. Type it by holding down [Ctrl] and pressing the Z key at the same time. On an IBM station this is the same as pressing [F6], as explained in your DOS manual.

Automatic Booting (cont.)

After you have created an AUTOEXEC.BAT file on your boot diskette, each time you turn on the power to your station or do a system reset (by simultaneously pressing [Ctrl], [Alt], and [Del]) the station will load DOS and the network shell automatically.

If you also want to run LOGIN, you could simply include a change drive command and the LOGIN command in your AUTOEXEC.BAT file. For example:

COPY CON: AUTOEXEC.BAT (press [Retrn])
NET20O (press [Retrn])
C: (press [Retrn])
LOGIN your username
^Z (press [Retrn])

Note: The file above is set up to change to network drive C. If your first network drive is some other drive, substitute that drive designation in the AUTOEXEC.BAT file.

If you set up the above batch file, each time you powered on your station it would load DOS, load the network shell, change to the first network drive, and run LOGIN (from the PUBLIC directory), all automatically.

Logging Out

When you log out, all network drive mappings are removed except that the default drive is mapped to SYS:LOGIN so that you may log in without re-booting your workstation.

After you have logged out, the NetWare commands are not available to you. You can still operate your workstation using DOS commands on your local drives.

When you issue the LOGOUT command, the system tells you that you are logged out from a specific workstation and gives you a reading of the time you last logged in as well as your logout time.

Commands to Avoid

If you are familiar with DOS commands, you must be careful not to use a few of them now that you are actually logged in on the network.

Most DOS 2.X commands will work normally on network drives. However, there are some exceptions. In order to ensure optimum system performance, follow these important command guidelines:

o BACKUP and RESTORE

Don't use the PC/MS-DOS BACKUP or RESTORE commands to back up or restore files created on network drives.

In particular the DOS BACKUP command will not automatically back up NetWare chained disks. Use of BACKUP may also cause other problems.

You must use the NetWare LARCHIVE/LRESTORE and NARCHIVE/NRESTORE commands to back up and restore files created on network drives.

Use the PC/MS-DOS BACKUP and RESTORE commands to back up and restore files created on local DOS partitions of a hard drive. Another valid use of BACKUP and RESTORE is to transfer extremely large files from a stand-alone machine with a hard disk to a network at a different location.

o PATH

Although NetWare does support the DOS PATH command, it is unnecessary. We recommend that you make it a habit to use the NetWare NPATH command for setting search drives. NPATH will allow you to specify local drives as part of the search drive list. You must use PATH on local drives when you are not logged into the network.

Commands to Avoid (cont.)

- PRINT

  To print any file on a shared network printer, you must use the NPRINT command.

  You may use PRINT to send local or network files to a local printer.

- CHKDSK

  Use the NetWare CHKVOL command instead for NetWare volumes. CHKDSK is still used for local drives.

- DIR

  The DOS DIR command will not work properly if a volume name other than the default volume is specified. Use the UDIR command instead.

- COPY

  The DOS COPY command may also work improperly if a volume name is specified. NCOPY should be used in these cases.

- CD

  If CD is invoked from a local drive and a volume name is specified, the network shell will automatically change that local drive to a network drive.

Accessing Network files, 4-3
Applications, 1-13
Assigning the printer, 7-4
Batch commands,
    ECHO, 10-11
    FOR, 10-12
    GOTO, 10-14
    IF and THEN, 10-15
    IF, 10-15
    introduction, 10-1, 10-10
    PAUSE, 10-16
    REM(ark), 10-19
    SHIFT, 10-18
    THEN, 10-15
    AUTOEXEC.BAT, 10-4
    executing, 10-9
    general information, 10-2
Batch files,
    introduction, 10-1
    replaceable parameters, 10-7
    uses for, 10-2
BREAK Command, 8-2
CASTON, 3-3
Central module, 1-3
CHECKMEM Command, 8-3
CHKVOL Command, 4-5
Clarity of commands, 3-14
CLS Command, 8-4
COMMAND Command, 8-5
Communicating with other users, 3-1
Create rights, 1-25
Creating and sending a memo, 3-5
CTTY Command, 8-7
CURSOR Command, 8-8
Date and time, 2-6
DATE Command, 8-9

DEBUG commands,
(A)ssemble, 13-10
(C)ompare, 13-13
(D)ump, 13-14
(E)nter, 13-16
(F)ill, 13-18
(G)o, 13-19
(H)ex, 13-21
(I)nput, 13-22
(L)oad, 13-23
(M)ove, 13-25
(N)ame, 13-26
(O)utput, 13-29
(Q)uit, 13-30
(R)egister, 13-31
(S)earch, 13-35
(T)race, 13-36
(U)nassemble, 13-38
(W)rite, 13-40
DEBUG utility,
command format, 13-4
command parameters, 13-6
error messages, 13-43
introduction, 13-1
starting, 13-2
Defining Network paths, 4-3
Delete rights, 1-25
Determining rights, 1-29
Directory,
names, 1-17
security, 1-22, 1-28
structure, 4-2
open access, 1-28
Disabling message receipt, 3-2
Diskette drive, 1-3

Diskette drive commands, 6-2
DISKBOOT Command, 6-10
DISKCOMP Command, 6-9
DISKCOPY Command, 6-6
FORMAT Command, 6-4
RELEASE Command, 6-3
REQUEST Command, 6-2
DOS commands,
ATTRIB Command, 5-36
CHDIR Command, 4-34
CHKDSK Command, 4-35
COPY Command, 5-37
DEL Command, 5-42
DIR Command, 4-39
FC (File Compare) Command, 5-43
line editor, introduction, 11-1
MKDIR Command, 4-41
PATH Command, 4-42
PRINT Command, 7-26
RECOVER Command, 5-53
REN (Rename) Command, 5-55
RESTORE Command, 5-56
RMDIR Command, 4-44
DOS Line editor, introduction, 11-1
DOS programs, preparing, 12-1
DOS utilities,
DEBUG, 13-1
EXE2BIN, 12-29
LINK, 12-1
Drive letter, pathspec, 1-20
EDLIN
commands,
(A)ppend, 11-20
(C)opy, 11-21
(D)elete, 11-24
(E)nd, 11-29
(I)nsert, 11-30
(L)ist, 11-34
(M)ove, 11-37

EDLIN, commands, continued
(P)age, 11-38
(Q)uit, 11-39
(R)eplace, 11-40
(S)earch, 11-44
(T)ransfer, 11-48
(W)rite, 11-49
<line> Edit, 11-27
how to use, 11-15
error messages, 11-50
files, restrictions, 11-2
how to start, 11-3
introduction, 11-1
special editing keys, 11-5
[Del], 11-9
[Esc], 11-11
[F1], 11-6
[F2], 11-7
[F3], 11-8
[F4], 11-10
[F5], 11-14
[Ins], 11-12
Effective rights, 1-29
Electronic mail, 3-4
system commands, 3-20
Electronic mailbox, 3-8
Electronic mailbox, rights, 3-8
Enabling message receipt, 3-3
ERASEDIR Command, 4-6
EXIT Command, 8-10
Experimenting on the Network, 2-14
Express mail, 3-11
Extended rights, 1-26
File attributes security, 1-22
FIND Command, 8-11
Fixed disk, 1-3
FLAG Command, 5-2
FLAGDIR Command, 4-8
Forwarded mail, 3-11

Function keys, 1-10
GRANT Command, 4-11
HIDEFILE Command, 5-4
Home directories, 4-3
How printing works, 7-2
Identifying mail, 3-10
Information window, 3-18
Key conventions, 1-7
Keyboard, 1-6
    calculator mode, 1-11
    cursor control, 1-12
    function keys, 1-10
    key conventions, 1-7
    typewriter and control keys, 1-8
LARCHIVE Command, 5-6
LISTDIR Command, 4-14
Local device, 1-6
Local printers, 7-4
Logged drive, 1-20
Logging in, 2-2
Logging out, 2-10
Login security, 1-22
Login/password security, 1-23
LOGOUT Command, 2-10
LRESTORE Command, 5-15
Mail attributes, 3-11
Mail commands,
    CHECK Command, 3-21
    CLOSE Command, 3-22
    DIRECTORY Command, 3-23
    EDIT Command, 3-24
    HELP Command, 3-27
    help, 3-19
    LIST Command, 3-29
    LIST MY MEMOS Command, 3-30
    LIST USERS AND GROUPS Command,
    OPEN Command, 3-32
    PUT Command, 3-33
    QUIT Command, 3-34

Mail commands, continued,
    READ Command, 3-35
    REMOVE Command, 3-37
    REMOVE MY MEMOS Command, 3-38
    search descriptions, 3-13
    SEND Command, 3-39
    VIEW Command, 3-41
Mail, types, 3-9
MAKEDIR Command, 4-15
Managing directories, 4-1
Managing files, 5-1
Managing files, file commands, 5-1
Mapping and remapping drives, 2-13
Modify file attribute flag rights, 1-26
modifying, 4-11
MORE Command, 8-13
Multi-user system, 1-3
Names, 1-17
NARCHIVE Command, 5-18
NCOPY CCommand, 5-26
Network interface board, 1-3
NPATH Command, 4-17
NPATH, 2-13
NRESTORE Command, 5-29
NSNIPES, 2-9
Open access, 1-28
Open rights, 1-24
Operating system, 1-13
    file server, 1-13
    shell, 1-13
Optional devices, 1-6
    local device, 1-6
Passwords, 1-23
    changing, 2-8
Pathspec, 1-18
    drive letter, 1-20
Personal ownership, 1-25

Print commands, 7-1
    DOS PRINT Command, 7-26
    ENDSPOOOL, 7-24
    LOCAL Command, 7-8
    NPRINT Command, 7-12
    PRINTER Command, 7-5
    Q Command, 7-16
    SPOOL Command, 7-19
Printing, 7-1
Privileges
PROMPT Command, 8-14
PURGE Command, 5-32
Read rights, 1-24
RENDIR Command, 4-23
REVOKE Command, 4-25
RIGHTS Command, 2-7
Rights, 1-24
    combinations, 1-26
    create, 1-25
    delete, 1-25
    modify file attribute flags, 1-26
    open, 1-24
    personal ownership, 1-25
    read, 1-24
    search, 1-26
    write, 1-24
SALVAGE Command, 5-33
Screen display, command entry, 3-17
Search
    descriptions, 3-4
    Search drive, 1-21
    Search rights, 1-26
Searching
    for mail, 3-12
    for mail from groups, 3-16
Security features, 1-22
Sending mail to groups, 3-15
    Sending messages, 3-2

SET Command, 8-16
SETLOGIN Command, 9-4
    NPATH, 9-5
    using conditionals, 9-13
    BREAK, 9-20
    COMSPEC, 9-6
    Displaying files, 9-21
    DRIVE, 9-9
    EXIT, 9-19
    External program execution9-18
    Firing phasers, 9-23
    INCLUDE, 9-8
    Inserting explanatory text, 9-22
    login script parameters, 9-16
    sample script, 9-2
    WRITE, 9-10
SETPASS Command, 2-8
Shared printers, 1-3, 7-2
    printer names, 7-2
SHOWDIR Command, 4-27
SHOWFILE Command, 5-35
Simulated second diskette drive, 6-11
Software, 1-13
    electronic mail, 1-13
    operating system, 1-13
SORT Command, 8-17
Storing information, 1-14
    directories, 1-14
    directory names, 1-17
    file names, 1-17
    files, 1-14
    organization, 1-15
    subdirectories, 1-14
    volume names, 1-17
    volumes, 1-14
System supervisor, 1-1
SYSTIME, 2-6

TIME Command, 8-19
TLIST Command, 4-29
Trustee security, 1-22, 24
TYPE Command, 8-20
Types of printers, 7-1
UDIR Command, 4-30
USERLIST, 3-2
Usernames, 1-23
Using DOS commands, 2-11
Using HELP, 2-4
Using the SETLOGIN command, 9-1
Using the shared diskette drive, 6-1
VER Command, 8-21
VERIFY Command, 8-22
Video display, 1-6
VOL Command, 8-23
VOLINFO Command, 4-32
Volume names, 1-17
WHOAMI Command, 2-5
Workstation cable, 1-6
Workstation, 1-5
Write rights, 1-24

Reader Comments

Please use this post-paid mailer to make us aware of the strengths and weaknesses of this North Star Manual. We will carefully consider your comments and try to incorporate your suggestions in future editions.

Date _____

Please indicate the type of reader you most closely represent:

____ first time user ____ programmer
____ experienced user ____ other _____

What type of work do you do on the computer? _____

_____

Do you think the level of the manual is appropriate for you? _____
Did you find the information well-organized? _____
Could you follow the procedures? _____
Were there enough procedures? _____
Please explain any "no" answers: _____

_____

_____

_____

_____

Please list specific points that need clarification or any errors found in the manual

Page Problem

____ _____

____ _____

____ _____

____ _____

Additional Comments _____

_____

_____

_____

Please make any further comments on additional pages We appreciate your help

Name of Manual _____ Part Number _____
Your Name _____ Job Title _____
Company Name _____
Street Address _____
City _____ State _____ ZIP _____

FOLD FOLD

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 2106 SAN LEANDRO, CA

POSTAGE WILL BE PAID BY ADDRESSEE

North Star Computers, Inc.
ATTENTION: Technical Publications
14440 Catalina Street
San Leandro, CA 94577