

FlexOS™ User's Guide

Version 1.3

COPYRIGHT

Copyright © 1986 Digital Research Inc. All rights reserved. No part of this publication may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language, in any form or by any means, electronic, mechanical, magnetic, optical, chemical, manual, or otherwise, without the prior written permission of Digital Research Inc., 60 Garden Court, Box DRI, Monterey, California 93942.

DISCLAIMER

DIGITAL RESEARCH INC. MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE. Further, Digital Research Inc. reserves the right to revise this publication and to make changes from time to time in the content hereof without obligation of Digital Research Inc. to notify any person of such revision or changes.

NOTICE TO USER

This manual should not be construed as any representation or warranty with respect to the software named herein. Occasionally changes or variations exist in the software that are not reflected in the manual. Generally, if such changes or variations are known to exist and to affect the product significantly, a release note or README.DOC file accompanies the manual and distribution disk(s). In that event, be sure to read the release note or README.DOC file before using the product.

TRADEMARKS

Digital Research and its logo are registered trademarks of Digital Research Inc. FlexOS, DR EDIX, and FlexNet are trademarks of Digital Research Inc.

First Edition: November 1986

FOREWORD

INTRODUCTION TO FLEXOS™

FlexOS is an advanced multitasking operating system for microcomputers. What does that mean for you?

In our daily work we rarely do only one thing at a time, finishing one task and putting it away before picking up the next task. Usually we're doing several things at once: preparing reports, answering the telephone, jotting down notes, and calculating columns of figures.

We certainly don't want our computers to be limited to doing only one task at a time. But that is exactly the limitation of most microcomputer systems. For example, if you are preparing a report with a word processing program and need some figures from your spreadsheet program, you would have to close and save your report file, exit the word processor, start the spreadsheet program, retrieve the figures, exit the spreadsheet program, start the word processor again...you get the idea.

With multitasking built into your operating system, you can keep several programs (processes) running simultaneously, and you can hop back and forth between them as you need to. The advantages are clear: faster, more efficient work and none of the frustration of constantly stopping and starting each program.

That's what FlexOS will do for you.

WHAT THIS GUIDE CONTAINS

The FlexOS User's Guide provides a full description of the FlexOS operating system, including the following:

- the command line and some basic FlexOS commands
- directories and subdirectories
- wildcard and logical names
- windows
- input/output redirection
- DR EDIX^{TM*}, the FlexOS text editor
- batch processing
- how FlexOS handles errors

Section 7 is a reference summary of FlexOS commands. Appendix A describes the role of the system manager, the person who sets up and manages a FlexOS system with more than one user.

ON-LINE ERROR MESSAGES

FlexOS displays an on-line error message if an error occurs while you are working with the operating system.

FlexOS lets you set your own "help level," the amount of information displayed in the error message.

If you are a beginning user, you can set a help level so that FlexOS describes each error in detail. At this level FlexOS usually suggests a solution for the error encountered. You can also refer to Appendix B for information on common error situations.

* EdixTM is a trademark of Emerging Technology Consultants, Inc.

If you are an experienced user, you can set your help level so that FlexOS displays a briefer message.

Setting the help level is described under the DEFINE command in Section 7.

OTHER BOOKS ON FLEXOS

The User's Guide is one of several Digital Research... books about FlexOS. The other books in this set include:

- FlexOS Programmer's Guide. This book identifies the C-based software tools and techniques used to write programs that can run on FlexOS.
- FlexOS System Guide. This book describes the architecture of the microchip processor that supports FlexOS.

Digital Research also publishes microprocessor-specific supplements and programmer's utilities guides to FlexOS.

TERMS AND CONVENTIONS

Hardware

This guide uses the terms "diskette," "hard disk," and "disk" in the following manner:

diskette The 3 1/2-inch, 5 1/4-inch or 8-inch magnetic disks that can be inserted into and removed from your microcomputer. Your system must have at least one diskette drive.

hard disk An internal or external fixed magnetic disk. Your system might not have a hard disk.

disk The term used when information applies to both diskettes and hard disks. For example, "disk drive" means either a diskette drive or a hard disk drive.

User Input and FlexOS Output

This guide uses boldface and regular type to distinguish between user input and system output, as in the following example. The system prompt (A>) is in regular type. Following that is the user input in boldface type. The next line shows what the system returns to the user. This is also shown in the regular typeface.

```
A>DIR | FIND "ACCOUNTS PAY"  
  
ACCOUNTS PAY            1429    8-18-86    1:42P
```

The text shows user input in all uppercase letters, but you can actually enter commands in uppercase, lowercase, or a combination of the two.

CONTENTS

1 Basic FlexOS Operation

Initial LOGON Instructions	1-1
System Prompt and Command Line	1-2
Built-In and External Commands	1-3
Editing a Command Line	1-3
Interrupting Command Execution	1-5
Formatting Disks	1-5
Formatting a Diskette	1-6
Formatting a Hard Disk	1-7
Basic File Operations	1-8
Directories	1-9
Accessing Subdirectories	1-10
Wildcards	1-11
Placing Wildcards	1-11
Star (*) Wildcard	1-11
Question Mark (?) Wildcard	1-12
Caret (^) Wildcard	1-13
Logical Names	1-13
Creating and Using Logical Names	1-14
Logical Devices	1-14
Logical Filenames	1-15
Logical Information	1-16
Reserved Logical Names	1-16

2 Working with Windows

FlexOS Windows	2-1
Windowing Keys	2-1
Status Window and Message Window	2-2
Status Window	2-2
Status Lines	2-3
Command Menu	2-4
Command Prompt	2-5
Message Window	2-5
Sample Use of Windows	2-8

3 I/O Redirection, Chaining, and Filtering

I/O Redirection.	3-1
I/O Redirection Symbols	3-1
Redirecting Output to a File or Device	3-2
Adding Output to a File or Device	3-2
Getting Input from a File or Device	3-3
Piping Commands	3-3
Chaining Commands on a Command Line	3-4
Filtering Data	3-5

4 DR EDIX Text Editor

Introduction to DR EDIX	4-1
DR EDIX Help Facility	4-1
DR EDIX Files	4-2
Learning DR EDIX	4-2
Starting DR EDIX	4-3
The DR EDIX Screen	4-3
Text Area	4-4

Contents

Prompt Area	4-4
Message Area	4-5
Status Area	4-5
Entering and Editing Text	4-6
Inserting and Overtyping	4-6
Moving The Cursor	4-6
Editing Text	4-7
Opening and Saving Files	4-9
DR EDIX Backup Files	4-11
Exiting DR EDIX	4-12

5 Advanced DR EDIX Features

Buffers	5-1
Windows	5-2
Buffers and Files	5-3
Status Table	5-4
Block Commands	5-5
Marking Blocks	5-5
Copying Blocks	5-7
Moving Blocks	5-7
Deleting Blocks	5-7
Search and Replace	5-8
Searching	5-8
Recovering Search Patterns	5-9
Searching With Wildcards	5-9
Replacing	5-10
Going to a Line by Number	5-12
Merging Files	5-12
Changing the Filename	5-13

6 DR EDIX Commands and Messages

DR EDIX Command Summary	6-1
DR EDIX Messages	6-6

7 FlexOS Commands

Command Summary	7-1
ASSIGN	7-5
BACK	7-7
BACKUP	7-9
BREAK	7-12
CANCEL	7-13
CHDIR	7-16
CHKDSK	7-18
COMMAND	7-23
COMP	7-26
CONFIG	7-29
COPY	7-32
Combining Files	7-36
Appending Files	7-36
Using Device Names with COPY	7-38
CTTY	7-39
DATE	7-40
DEFINE	7-42
DIR	7-46
DISKCOMP	7-50
DISKCOPY	7-53
DISKSET	7-56
ERASE	7-59
EXIT	7-61
FDISK	7-62
FIND	7-63

Contents

FORMAT	7-66
Formatting a Diskette	7-68
Formatting a Hard Disk	7-68
FSET	7-72
Changing the Attributes of a File	7-73
Setting Protection on Files	7-74
LIST	7-78
LOGOFF	7-79
LOGON	7-80
MKDIR	7-81
MORE	7-83
ORDER	7-85
PASSWORD	7-87
PATH	7-89
PRINT	7-91
PROCESS	7-96
PROMPT	7-103
RECDIR	7-105
RECFILE	7-107
RENAME	7-110
RESTORE	7-112
RMDIR	7-115
SECURITY	7-116
SORT	7-117
SYS	7-122
TIME	7-124
TREE	7-126
TYPE	7-128
VER	7-129
VOL	7-130

8 Batch Processing

Using Parameters in Batch Files	8-3
Chaining Batch Files	8-5
Nesting Batch Files	8-6
Commands for Batch Files	8-7
BATCH	8-9
CLS	8-10
ECHO	8-11
FOR	8-13
GOTO	8-14
IF	8-16
PAUSE	8-21
REM	8-23
SHIFT	8-25
VERIFY	8-28

A System Manager A-1

What is a System Manager?	A-1
System Manager's Tools	A-1
Setting up LOGON/LOGOFF	A-2
Creating User Accounts	A-3
Creating Passwords	A-5
Creating the System Directory	A-5
Installing Devices	A-7
DVRLOAD	A-10
DVRUNIT	A-13
DVRLINK	A-15
DVRUNLK	A-16
Installing a Diskette Drive	A-17
Installing a Hard Disk Drive	A-18
Preparing a Hard Disk with FDISK	A-18

Contents

Starting FDISK	A-19
Creating a Partition	A-21
If No Partitions Exist	A-21
If Partitions Exist	A-24
Other FDISK Options	A-26
Changing the Active Partition	A-26
Deleting a FlexOS Partition	A-28
Displaying Partition Data	A-30
Selecting the Next Hard Disk Drive	A-30
Editing CONFIG.BAT	A-31
Logically Formatting The Partition	A-32
Formatting With System Tracks	A-33
Formatting Without System Tracks	A-34
Installing Parallel Printers	A-34
Defining Consoles	A-35
Adding an Application to the System	A-35
Installing the Print Spooler	A-36
How to use the Print Spooler	A-37
Using the Print Spooler with DR NET	A-38
B Error Handling	B-1
Troubleshooting Checklist	B-1
Disk Handling	B-3
Corrupted Files	B-4
Disk Error Messages	B-4
Insufficient Memory	B-5
Unresolvable Errors	B-5
C ASCII Conversion Tables	C-1

Index

Tables

1-1	Command Line Editing Keys.	1-4
1-2	Keys For Interrupting Commands.	1-5
1-3	FlexOS File Operations.	1-8
1-4	Reserved Logical Names.	1-16
4-1	DR EDIX Files.	4-2
4-2	DR EDIX Cursor Control Keys.	4-7
4-3	DR EDIX Editing Keys.	4-8
4-4	DR EDIX File Commands.	4-10
5-1	DR EDIX Window Commands.	5-3
6-1	DR EDIX Line-Editing Commands.	6-2
6-2	DR EDIX File/Buffer Commands.	6-3
6-3	DR EDIX Windowing Commands.	6-4
6-4	DR EDIX Block Commands.	6-4
6-5	DR EDIX Search Commands.	6-5
6-6	DR EDIX Status/Help Commands.	6-5
7-1	Working with Files.	7-1
7-2	Working with Directories.	7-2
7-3	Working with Disks.	7-2
7-4	Data Protection.	7-3
7-5	Modifying Your User Environment.	7-3
7-6	Additional Commands.	7-4
7-7	BACKUP Options.	7-10
7-8	CANCEL Options.	7-14
7-9	CHKDSK Options.	7-18
7-10	COMMAND Options.	7-24
7-11	CONFIG Options.	7-30
7-12	COPY Options.	7-34
7-13	Error Message Help Levels.	7-43
7-14	DIR Options.	7-47
7-15	FIND Options.	7-64
7-16	FORMAT Options.	7-67

Contents

7-17	PRINT Options.	7-93
7-18	PROCESS VIEW Options.	7-97
7-19	RECFILE Options.	7-108
7-20	RESTORE Options.	7-113
7-21	SORT Options.	7-117
A-1	Driver Commands.	A-8
A-2	Access Level Options.	A-11
C-1	ASCII Symbols.	C-2
C-2	ASCII Conversion Table.	C-3

Figures

1-1	Directory Structure.	1-9
2-1	FlexOS Status Window.	2-3
2-2	Help Information Message Window.	2-6
4-1	DR EDIX Screen Areas.	4-4
4-2	DR EDIX Status Area.	4-5
5-1	DR EDIX Status Table.	5-4
8-1	Chained Batch Files.	8-5
8-2	Nested Batch Files.	8-7

BASIC FLEXOS OPERATION

INITIAL LOGON INSTRUCTIONS

When you receive it, FlexOS is configured to run the LOGON command as the last command of your CONFIG.BAT file. LOGON prompts you for the following:

Username:

Password:

To enter the system, enter a carriage return at each prompt.

Valid user accounts (usernames and passwords) are defined in the file USER.TAB. The procedure for creating and modifying USER.TAB, including a full explanation of the structure of a user account entry, is described in Appendix A.

The USER.TAB provided on the disks has the following usernames, passwords, and other values defined.

username	password	user ID	group ID	home	wmanager	shell	access
SYSTEM	SYSTEM	0	0	fd0:	system: wmex.286	boot: command.286	1
USER	<CR>	100	0	fd0:	shell	boot: command.286	1
NOWIND	<CR>	0	0	fd0:	shell	boot: command.286	1
<CR>	<CR>	0	0	fd0:	system: wmex.286	boot: command.286	1

<CR> = carriage return

SYSTEM PROMPT AND COMMAND LINE

Much of your interaction with FlexOS involves listing directories, copying or deleting files, starting applications, and the like. To perform these tasks, you type a "command line" after the system prompt and then press the Enter key.

On many computers, the system prompt is a letter (usually A, B, or C, depending on your current, or "default," drive) followed by a greater-than symbol (>) and a blinking cursor.

The FlexOS command line has the following components:

A>COMMAND FILESPEC -OPTION

COMMAND includes FlexOS commands, executable files that start application programs like word processing programs or spreadsheets, and batch files. (Batch files are discussed in Section 11, "Batch Processing.") A command line must always include a command.

FILESPEC stands for "file specification". It is the name and, occasionally, the location in your system of a file on which you want FlexOS to perform an operation (like COPY). Some commands require a filespec; others do not. Some commands accept more than one filespec.

A complete filespec looks like this:

DRIVE:/DIRECTORY_PATH/FILENAME.EXT

The DRIVE:, DIRECTORY_PATH and .EXT (file extension) components are optional with most commands. See Section 7 for the requirements of each command.

An OPTION lets you modify what a FlexOS command does. Multiple options are separated from each other by hyphens (-). The options are described in Section 7 under the commands that accept them.

Built-In and External Commands

FlexOS commands fall into the following categories:

- | | |
|----------|--|
| built-in | Commands loaded into your computer's memory when you start FlexOS; they remain in memory until you terminate the operating system. You can enter a built-in command at any time from the FlexOS prompt. |
| external | Commands contained in disk files and not loaded into memory at startup; they must be accessed from the disk. If the command is on a diskette, you must insert the diskette into your computer before entering the command. |

Editing a Command Line

You can correct or cancel a command line before you press the Enter key by using several keys or key combinations.

The key combinations involve holding down the Control key (Ctrl -- represented by a caret "^") while you press a letter key. For example, the notations "Ctrl-X" and "^X" both indicate that you hold down the Ctrl key while pressing the X on your keyboard.

Table 1-1 summarizes the command line editing keys and key combinations.

Table 1-1. Command Line Editing Keys

Key	Description
left-arrow	Moves cursor one character to the left without affecting characters in the command line.
right-arrow	Moves cursor one character to the right without affecting characters in the command line.
Backspace	Moves cursor one character to the left, deleting the character at that position. Characters following the cursor move one position to the left.
Delete	Erases the character under the cursor. The cursor does not move, but characters following the cursor move one position to the left.
Ctrl-X	Deletes the entire command line. The cursor appears immediately after the system prompt.
Ctrl-R	Redisplays the most recently executed command line. You can then view, change, or reenter that command line.
Enter	Causes FlexOS to execute the command line.

Interrupting Command Execution

The Ctrl-S, Ctrl-Q, and Ctrl-C key combinations let you suspend, resume, and cancel commands that have started executing. These keys are summarized in Table 1-2.

Table 1-2. Keys For Interrupting Commands

Key	Description
Ctrl-S	Suspends the execution and output of a running command. The command runs until it is supposed to display output and then is suspended. Command and output remain suspended until you cancel the command with Ctrl-C or resume the command with Ctrl-Q.
Ctrl-Q	Causes a command suspended with Ctrl-S to resume running where it left off. Any output in progress when you pressed Ctrl-S also resumes.
Ctrl-C	Deletes a command line before you press Enter, or stops a running command from further execution.

When you cancel a FlexOS command with Ctrl-C, command execution is terminated and the system prompt is redisplayed. However, if you use Ctrl-C to stop a running program or application, results can be unpredictable. FlexOS has no way of controlling the program or application once it has started.

FORMATTING DISKS

Before you can use a disk, you must "format" it with the FORMAT command. Formatting creates a directory on the disk and divides the disk into storage areas called "sectors."

The **FORMAT** command deletes everything on the disk. For this reason, do not use **FORMAT** on a disk containing files you want to keep. In addition, it is a good practice always to specify a drive letter ("**FORMAT B:**", for example) to avoid accidentally formatting the wrong disk.

For more information on the **FORMAT** command and what it does, see Section 7.

Formatting a Diskette

To format a diskette on a system that has only one diskette drive, do the following:

1. Insert the diskette that contains the **FORMAT** command into drive A.
2. Type the following:

```
A>>FORMAT A:
```

FlexOS responds:

```
Insert diskette into drive A:
```

```
WARNING, ALL DATA ON DRIVE A: WILL BE LOST!
```

```
Proceed with FORMAT (Y/N)? _
```

3. Remove the diskette with **FORMAT** from drive A and insert the diskette you want to format.
4. Type **Y** (for Yes) next to the "Proceed with **FORMAT**" prompt. FlexOS responds:

```
Formatting:
```

The colon (:) blinks while FlexOS formats the diskette in drive A.

When FlexOS finishes formatting the diskette, it displays this message:

```
Formatting:  Formatting complete.  
          XXXXXX bytes of total disk space  
          XXXXXX bytes available on disk
```

Format another (Y/N)? _

The number of bytes ("XXXXXX") varies according to the number of tracks and the size of the sectors on the diskette.

5. Remove the newly formatted diskette from drive A. If you want to format another diskette, type Y. If you are done formatting, type N.

Formatting a diskette on a system with two diskette drives is similar, with the following variations:

- Insert the diskette that contains the FORMAT command into drive A and the diskette you want to format into drive B.
- Enter **FORMAT B:**
- You don't have to remove the diskette with FORMAT from drive A. Simply type Y (for Yes) after the "Proceed with FORMAT" prompt.

Formatting a Hard Disk

Some hard disks are formatted by the manufacturer. If you have a hard disk that is not formatted, you must prepare it with the FORMAT and FDISK commands. This is described in detail under "Preparing a Hard Disk with FDISK" in Appendix A, "System Manager."

BASIC FILE OPERATIONS

Table 1-3 describes four commands that perform basic FlexOS file operations.

Table 1-3. FlexOS File Operations

Command	Description
COPY	<p>Copies one or more files from disk to disk, or from directory to directory, either on the same disk or on different disks. The COPY command takes this form:</p> <p>COPY SOURCE_FILESPEC DESTINATION_FILESPEC</p> <p>DESTINATION_FILESPEC can be simply a drive identifier, as in the command line "COPY MYFILE.TXT B:".</p>
TYPE	<p>Displays the contents of a text (ASCII) file on the screen. The TYPE command uses this form:</p> <p>TYPE FILESPEC</p>
RENAME	<p>Gives one or more files new names, removing the original names from the directory. The RENAME command uses this form:</p> <p>RENAME OLD_FILESPEC NEW_FILESPEC</p>
ERASE	<p>Deletes one or more files from the default or specified directory. The ERASE command uses this form:</p> <p>ERASE FILESPEC</p>

Note: The COPY, RENAME, and ERASE commands accept wildcards, which are described later in this section. All four commands are described in detail in Section 7.

DIRECTORIES

A "directory" is a collection of filenames. Directory information includes the names of the files on your disk, how big they are, and when you last worked on them.

To make organizing your data easier and more coherent, FlexOS supports an inverted tree structure of directories. The topmost directory on each disk is called the "root directory," and each directory contained within the root is called a "subdirectory." Figure 1-1 illustrates a typical directory structure.

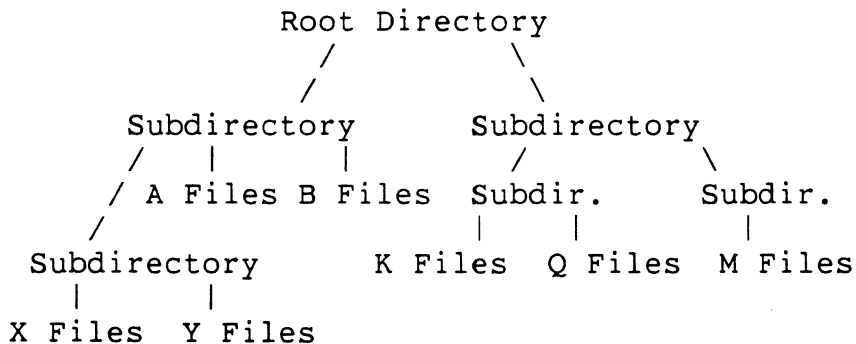


Figure 1-1. Directory Structure

When you format a disk, it initially contains only a single directory, the root. To create subdirectories, you must use the MKDIR command, which is described in detail in Section 7.

Accessing Subdirectories

When giving a filespec after a command like DIR, you must separate directory names with a slash (/) or backslash (\). With DIR as an example, here are the two rules for accessing subdirectories:

- If you are in the root directory, or if you are in a subdirectory and want to access a subdirectory below your current level, begin the filespec with the name of the next subdirectory in the path, not preceded by a backslash, as in the following:

```
C>DIR SUBDIRECTORY/SUBDIRECTORY/FILENAME.EXT
```

- If you are in a subdirectory and want to access a subdirectory above your current level, or if you want to access a subdirectory in another path entirely, begin the filespec with a backslash (representing the root directory) and then give the complete path specification, as in the following:

```
C>DIR /SUBDIRECTORY/SUBDIRECTORY/FILENAME.EXT
```

Be accurate and complete when you enter your filespec. Specify all the subdirectories in the directory path in order, as well as the name of the file itself. Your path or filespec, from the root directory to the level you want to access, must be no more than 128 characters.

You can change your current directory path by using the CHDIR command described in Section 7.

WILDCARDS

FlexOS uses three "wildcard" characters to give you greater flexibility in your file specifications: the asterisk or "star" (*), the question mark (?), and the caret (^).

Wildcards can take the place of one or more characters in a directory name, filename, or file extension; they make it possible for a command to affect more than one file or directory at the same time. You cannot use wildcards in a device name.

Note: Some FlexOS commands restrict how you can use wildcards. Refer to the command descriptions in Section 7 before you use wildcards.

Placing Wildcards

You can only use a wildcard in the rightmost (last) component of a filespec. The following is a valid use of wildcards:

```
A>DIR B:/GRADES/MATH/*.*
```

This is not a valid use of a wildcard:

```
A>DIR B:/GRADES/?ATH/PERIOD1.DAT
```

Star (*) Wildcard

The star wildcard takes the place of a variable number of characters in a filename, directory name, or extension. How many characters the star can represent depends on the specific case, as the following examples illustrate:

*.DAT Any filename or directory name with the extension DAT. The star takes the place of from one to eight characters. S.DAT and MONDAYS.DAT both match this specification.

- B*.*T** Any file or directory whose name begins with B and whose extension ends with T. The star before the dot (period) takes the place of from zero to seven characters; the star after the dot takes the place of up to two characters. BIOLAB.TXT, B.DAT, and BARGAINS.T all match this specification.
- M*N.TXT** Any file or directory whose name begins with M and ends with N and whose extension is TXT. The star can take the place of from zero to six characters. MOON.TXT, MN.TXT, and MASON.TXT all match this specification.

As you will see below, you can combine the star and question mark wildcards in a single specification.

Question Mark (?) Wildcard

The question mark wildcard takes the place of a single character at the wildcard's position.

The following examples illustrate the question mark wildcard.

- TEST?.DAT** Any five-character file or directory name whose first four characters are TEST and whose extension is DAT. TEST1.DAT and TESTG.DAT both match this specification; TESTNEW.DAT does not.
- TE?TS.*** Any five-character file or directory name that begins with TE and ends with TS, and with any extension. TEXTS.BAK and TESTS.TXT both match this specification.

DI??ONE.C Any seven-character file or directory name that begins with DI and ends with ONE, with the extension C. DIALONE.C and DISKONE.C both match this specification.

Caret (^) Wildcard

The caret wildcard excludes any files or directories that match the specification from the action on the command. The caret must be the first character in the specification. The following example

```
A>RENAME ^*.DAT *.DAT
```

says "Rename all files or directories that do not have the extension DAT to have the extension DAT." The result of this command is that all files or directories in the current directory have the extension DAT.

Note: Be very careful using the caret with the ERASE command; you can inadvertently wipe out entire directories. For example, the command ERASE ^*.TXT on a directory that contains no files with the TXT extension has the same effect as the command ERASE *.* , except that you do not receive the "Are you sure?" warning message.

LOGICAL NAMES

Logical names allow you to substitute a shorter name for all or part of a complex device name or filespec. You can also substitute one logical name for another.

For example, you could substitute the logical name CONIFER for the directory pathname A:/FLORA/TREES/CONIFER/. Whenever you need to invoke that path, you would simply enter CONIFER.

There are three types of logical names: logical devices, logical filenames, and logical information. Each is described later in this section.

Creating and Using Logical Names

To create a logical name, use the DEFINE command described in Section 7. For example, to create the logical name CONIFER:, you would issue the following command:

```
A>DEFINE CONIFER:=A:/FLORA/TREES/CONIFER/
```

Logical names have the following requirements:

- They can be up to nine characters. The original "long form" of your path or filespec cannot exceed 128 characters.
- They must be the first name following the command.
- There can only be one logical name on a command line.

When FlexOS finds a logical name on the command line, it replaces the logical name with the original name and then executes the specified command.

Logical Devices

A logical device name takes the place of a device or directory path name. Logical device names end in a colon (:).

The CONIFER: example given previously is a logical device name. If you have a text file in the CONIFER directory called PINES.DAT, use the following command line to display it:

```
A>TYPE CONIFER:PINES.DAT
```

FlexOS accesses drive A, the directory path /FLORA/TREES/CONIFER/, and the file PINES.DAT. Because the

logical name includes a drive identifier, you can enter the command line from any disk drive.

You can use a logical device name to replace part of a directory path. For example, you could substitute FTREES: for A:/FLORA/TREES/. To access PINES.DAT, enter the following:

```
A>TYPE FTREES:CONIFER/PINES.DAT
```

This would allow you to access other subdirectories of TREES, perhaps one named FRUIT, with a file called APPLE.DAT:

```
A>TYPE FTREES:FRUIT/APPLE.DAT
```

Logical Filenames

Logical filenames can completely replace a filespec or just a filename and extension. When you replace only the filename and extension, FlexOS assumes the file is in the current directory of the current drive. Logical filenames do not have to include a colon.

One use of a logical filename could be to invoke an executable file from anywhere in the system. For example, if you have a text editing program called TXTEDIT.EXE in the root directory of drive B, you can give it the logical filename EDIT with the following command:

```
A>DEFINE EDIT=B:TXTEDIT.EXE
```

If you are on drive C and want to edit a file called REPORT.TXT in the current directory, type the following:

```
C>EDIT REPORT.DAT
```

Note: Remember you can only use one logical name per command line. You cannot combine a logical device name and a logical filename like this:

```
A>EDIT CONIFER:PINES.DAT
```

Logical Information

“Logical information” is a catch-all term for the other kinds of logical names you can define. For example, you can change your system prompt by redefining the logical name PROMPT, as in the following example:

```
A>DEFINE PROMPT=Your wish is my command>
```

```
Your wish is my command>_
```

Reserved Logical Names

FlexOS reserves certain logical names for its own use. You must use them as intended or you may get unpredictable results. Table 1-4 lists the reserved logical names and how they should be used.

Table 1-4. Reserved Logical Names

Logical Name	Definition
a:, b:, ... p:	device
addmem	additional memory request
bgprn:	background printer
boot:	system boot directory path
bottom	FlexOS internal use only
con	console (keyboard and screen)
con:	physical console

Table 1-4. (Continued)

Logical Name	Definition
default:	device
groupid	group identification number
helpvl	message help level
home:	current default home directory path
left	FlexOS internal use only
lptX: [*]	parallel device
mouse:	console pointer device
name	user LOGON name
nul	null device
null:	null device
order	command type search order
path	search path for commands
pi:	pipe device
prn	printer
prn:	printer
prompt	system prompt
protect	system protection flag
right	FlexOS internal use only

^{*} X = 0, 1, 2, etc.

Table 1-4. (Continued)

Logical Name	Definition
ser:	serial device
shell	default shell
stdcmd	standard input/output (I/O) command stream path name
stderr	standard output error message stream path name
stdin	standard input stream path name
stdout	standard output stream path name
switchar	options separator
system:	default system directory path
tempdir:	temporary directory for print spooler
top	FlexOS internal use only
userid	user identification number
vc00X*	window number
vdiXX*	virtual device drivers
version	operating system and hardware identification
wmanager	default window manager program
wmessage	message pipe for window manager

* X = 0, 1, 2, etc.

End of Section 1

WORKING WITH WINDOWS

FLEXOS WINDOWS

FlexOS is a multitasking windowing operating system, which means that several processes or applications can run at the same time, each in its own full-screen window. You can only work in one window at a time (the current window), but while you are working in the current window, the processes in the other windows are either busy performing tasks or waiting for you to return and do something. You do not have to exit a process just because you aren't working in it at the moment. Furthermore, you can switch freely between windows, and thus between processes.

You create the windows for the programs you're working with. When you are finished with a program, you can start another program in the same window or you can delete the window. Each user on the system can have up to six windows running at the same time.

Windowing Keys

FlexOS uses three different key combinations for various windowing operations. The following descriptions identify the most common way you produce these key combinations, but you should check your computer documentation to be sure. To produce the key combinations, hold down the Alt or Ctrl key and then press the other key named.

<WINDOW> (Alt-plus)

Displays the Status Window, described below.

<NEXT> (Ctrl-plus)

Switches to the next higher window number. If you are already in the highest window number, pressing <NEXT> switches you to the lowest window number.

<PREV> (Ctrl-minus)

Switches to the next lower window number. If you are already in the lowest window number, pressing <PREV> switches you to the highest window number.

You can use all three key combinations from the Message Window (described later) or from the system prompt. You can also use the <PREV> (Ctrl-plus) and <NEXT> (Ctrl-minus) keys from the Status Window.

STATUS WINDOW AND MESSAGE WINDOW

In addition to the six windows you can create, there are two permanent windows, the Status Window and the Message Window, which you cannot create or delete. Their function is to provide you with information and help.

These windows overlay (but do not change) the information in the current window. When you are through with the Status or Message Windows, you are automatically returned to the underlying window.

Status Window

To display the Status Window, press the <WINDOW> key combination, described previously.

Figure 2-1 shows a sample Status Window.

```

===== [ S T A T U S   W I N D O W ] =====
#                                                                 #
# WNDW#  FMVID  Process ID    Command Line                #
# =====  =====  =====  =====  #
#      1      1          2    COPY CON FILE1.TXT                    #
#      2      3          4    COPY CON FILE2.TXT                    #
#=====  #
# Commands:  C - CREATE Window                                     #
#            D - DELETE Window                                     #
#            HELP - HELP INFORMATION                               #
#            <number> - Window number to select                  #
#                                                                 #
# Command?_#
=====

```

Figure 2-1. FlexOS Status Window

The Status Window has three parts: status lines, a command menu, and a command prompt.

Status Lines

The status lines display a line of information for each window currently running on your system. This section has the following components:

- WNDW#** Window number. Each window is identified by a number. You can switch windows by entering the appropriate window number.
- FMVID** Family identifier. All the programs in a window share a common family ID number, including those programs invoked by batch files. Programs running in other windows have different family IDs.

- Process ID** A unique process identifier for each program. If you run several programs sequentially in the same window (as when you chain commands together), each has its own process ID.
- Command Line** The last command line you entered in the window.

The family ID, process ID, and window number fields are important because they can all be used with the CANCEL command to stop running programs. For more information on the CANCEL command, see Section 7.

If the Status Window does not have enough space to show all your status lines, FlexOS displays the line "...more... (type M to see more)". Type M to view the status of additional windows. If you press M when the status of the last window is displayed, the status of the first window is redisplayed.

Command Menu

The command menu has four commands:

- **C (Create Window)** - Creates a window, gives it the lowest available window number, and automatically switches you to the new window. When the C command is finished, the system prompt appears in your new window.
- **D (Delete Window)** - Deletes a window. After you enter the D command, the following prompt appears:

Select a Window Number to Delete: _

When you enter a number, the program or command running in that window is immediately stopped, the file is closed, and you are automatically returned to the previous window. Changes to your file may or may not be saved,

depending on the program or command.

The remaining windows are not renumbered, but the Status Window is updated to show that a window has been deleted.

- **HELP INFORMATION** - Displays information on how to use windows. When you enter **HELP**, the Message Window appears. The Message Window is described below.
- **<number>** - Switches you to the specified window number. **<number>** can only be used from the Status Window. For example, to switch from Window 1 to Window 4, press **<WINDOW>** to display the Status Window and then, once you are in the Status Window, press 4 to switch to Window 4.

Note: If there is an active Message Window, the **<PREV>** and **<NEXT>** keys and the **<number>** command can be used to switch to it. The Status Window indicates if there is an active Message Window.

Command Prompt

Type your window menu commands at the command prompt. Press Enter to execute the commands.

MESSAGE WINDOW

FlexOS uses the Message Window to display help information and to notify you of situations that require your immediate attention. The Message Window displays information about any process that has a problem, not just the process in the current window. The Message Window "pops up" when a problem is detected.

Figure 2-2 shows the Message Window that appears when you select Help Information from the Status Window.

```

===== [ M E S S A G E   W I N D O W ] =====
#
#   After selecting a window, the <PREV>, <NEXT>
#   and <WINDOW> keys can be used as follows:
#
#       <PREV> select the next lower Window Number
#       <NEXT> select the next higher Window Number
#       <WINDOW> return to the Status Window
#
#
#   Press Enter to acknowledge:
#   Press <WINDOW> to delay answer.
=====
    
```

Figure 2-2. Help Information Message Window

The following example illustrates how the Message Window informs you of an error situation. Suppose you are updating CALENDAR.TXT in Window 1 while you have other programs running in other windows. FlexOS detects a problem and displays the following:

My appointments for this week:

Monday, 9:00 - Staff Meeting
 Monday, 3:00 - Meeting with Marketing Department

```

===== [ M E S S A G E   W I N D O W ] =====
#   Message from PAYROLL.BAS in Window 5:
#
#       *** Printer device not currently available***
#       Send listing to output file (Y/N)?_
#
#   (Press <WINDOW> to delay answering this question)
#####
    
```

The Message Window indicates which window has the problem, and the nature of the problem. In this example, the printer is not available to print the results of your payroll program. The Message Window also offers you a solution: You can send the listing to an output file instead of waiting for the printer.

You must respond to this Message Window before you can continue your work. When you do, the Message Window disappears. You remain in the window where you were updating your appointment calendar.

If you are not sure how to respond to the Message Window, you can press <WINDOW> to look at the Status Window and evaluate the status of your windows. In this case, the Status Window looks like this:

```

===== [ S T A T U S   W I N D O W ] =====
#
# WNDW#   FMVID   Process ID   Command Line
# =====   =====   =====   =====#
#       1       1         3     COPY CON CALENDAR.TXT
#       4       2         6     PRINT FORMLETR.PRN
#       5       3         7     RUN PAYROLL.BAS
#       6       4         8     Active Message Window
#=====
# Commands:  C - CREATE Window
#            D - DELETE Window
#            HELP - HELP INFORMATION
#            <number> - Window number to select
#
# Command?_
=====

```

The Status Window shows that Window 4 is using the printer to print a form letter. If you would rather print your payroll program, you can do the following:

1. Enter D to delete Window 4. This cancels the PRINT command, and FORMLETR.PRN is filed away. The printer is now available to the payroll program.
2. Enter 6 to return to the Message Window. The error message and prompt are still there.

3. Type N next to the prompt and press Enter. Your problem is resolved and the Message Window disappears. Your payroll program starts to print and you can go back to your work in Window 1.

Your response to the Message Window prompt can be up to one line long.

SAMPLE USE OF WINDOWS

The following shows you how to create, work with, and delete a window. The exercise assumes that you are starting with a single window; there are no other windows active.

1. Enter the following command line in the current window and press Enter:

```
A>COPY CON CALENDAR.TXT
```

2. Type the following and press Enter:

```
My appointments for this week:
```

3. Press the <WINDOW> key to display the Status Window. Your screen should look like this:

```
===== [ S T A T U S   W I N D O W ] =====
#                                                                 #
# WNDW#   FMVID   Process ID   Command Line           #
# =====   =====   =====   =====                   #
#         1         1           3   COPY CON CALENDAR.TXT      #
# =====   =====   =====   =====                   #
# Commands:  C - CREATE Window                               #
#             D - DELETE Window                               #
#             HELP - HELP INFORMATION                         #
#             <number> - Window number to select            #
#                                                                 #
# Command?_                                                #
=====
```

```
A>COPY CON CALENDAR.TXT
My appointments for this week:
```

4. Type C at the command prompt and press Enter to create a new window. FlexOS automatically switches you to the new window.
5. Create a new file in this window using the **COPY CON** command as you did in step 1:

```
A>COPY CON NOTES.TXT
This is my second file. This file
appears in its own window.
```

6. Use the <PREV> key to switch back to CALENDAR.TXT in Window 1.
7. Type another line of text and press Enter:

```
Monday, 9:00 - Staff Meeting
```

8. Use the <NEXT> key to switch to Window 2 and confirm that NOTES.TXT is still there.
9. Press the <WINDOW> key to look at the Status Window. It lists two windows:

```
===== [ S T A T U S   W I N D O W ] =====
#                                                                 #
# WNDW#   FMYID   Process ID  Command Line           #
# =====   =====   =====   =====#
#         1       1           3  COPY CON CALENDAR.TXT       #
#         2       3           4  COPY CON NOTES.TXT           #
#=====#
# Commands:  C - CREATE Window                               #
#            D - DELETE Window                               #
#            HELP - HELP INFORMATION                         #
#            <number> - Window number to select            #
#                                                                 #
# Command?_#
=====
```

```
This is my second file. This file
appears in its own window.
```

10. To delete Window 2, type D at the command prompt and press Enter. When you are prompted for the window

number, type 2 and press Enter. Window 2 is deleted and since NOTES.TXT is an open file, it is deleted. If we had closed this file, it would have been saved.

11. FlexOS automatically returns you to Window 1. To file away CALENDAR.TXT, type Ctrl-Z and then Enter. Window 1 displays the system prompt. You are back where you started. The file CALENDAR.TXT has been saved.

End of Section 2

I/O REDIRECTION, CHAINING, AND FILTERING

I/O REDIRECTION

Most of the time you enter input at the keyboard and see output displayed on the screen. With input/output (I/O) redirection, you can use data from files or devices as input, and you can send output to a file, a device, or another command.

I/O Redirection Symbols

The following shows the I/O redirection symbols and describes what they do. Each description is accompanied by an example of how you use the symbol.

- > Redirects the output from a command to a file or device.
 COMMAND > FILESPEC

- >> Adds the output from a command to the end of a file or sends to a device.
 COMMAND >> FILESPEC

- < Uses a file or device as input to a command.
 COMMAND < FILESPEC

- | "Pipes" the output of the first command to become the input for the second command.
 COMMAND1 | COMMAND2

- ! "Chains" a series of commands on a single command line. The commands are executed in order.

COMMAND ! COMMAND ! COMMAND

>* Redirects the command's standard error file to a file or device.

COMMAND >* FILESPEC

>>* Adds the command's standard error file to the end of a file or device.

COMMAND >>* FILESPEC

|* Uses the standard error file of the first command as the input for the second command.

COMMAND1 |* COMMAND2

You can combine two or more ways of redirecting I/O on the same command line.

You cannot use I/O redirection symbols within a filespec.

REDIRECTING OUTPUT TO A FILE OR DEVICE

The **>** symbol directs the output from a command to a specified file or device. If the file does not exist, FlexOS creates it. The following command line sends the output from the DIR command to the file PROGRAMS.LST.

A>DIR > PROGRAMS.LST

ADDING OUTPUT TO A FILE OR DEVICE

The **>>** symbol takes the output from a command and adds it to the end of an existing file or device. The following command line adds the directory of drive B to the end of the file PROGRAMS.LST on the default drive (drive A):

A>DIR B: >> PROGRAMS.LST

GETTING INPUT FROM A FILE OR DEVICE

The < symbol uses a file or device as the input to a command. In the following example, the MORE command displays information from the file QUOTAS.LST. (MORE is one of the "filters" described later in this section and in Section 7. It allows you to display output one screen at a time.)

```
A>MORE < QUOTAS.LST
```

```
SALES QUOTAS FOR FEBRUARY
-----
Anderson, R.      $ 5,000.00
Blake, J.         $12,500.00
.
.
.
Whitfield, D.    $ 3,000.00
-- More --
```

Piping Commands

"Piping" commands takes the output of one command and makes it the input to another command. To pipe between commands, place the vertical bar (|) symbol between the command names on the command line.

For example, the DIR command might produce this directory of files on one of your disk drives:

```
CLEANUP  BAT          25    6-06-1986    9:34a
MEMOS           25984   3-23-1986   11:03a
CATALOG  LST         17664   6-01-1986   10:13p
LETTERS        16256   4-16-1986    5:57p
```

By piping the output from DIR to the SORT filter (described later in this section and in Section 7), you can sort this directory alphabetically. The command line takes this form:

A>DIR B: | SORT

It produces the following directory listing:

%P00001	\$\$\$	0	6-08-1986	10:18a
%P00002	\$\$\$	0	6-08-1986	10:18a
CATALOG	LST	17664	6-01-1986	10:13p
CLEANUP	BAT	25	6-06-1986	9:34a
LETTERS		16256	4-16-1986	5:57p
MEMOS		25984	3-23-1986	11:03a

The two new files with the \$\$\$ extension are temporary files created by FlexOS. They are discussed under the SORT command in Section 7.

CHAINING COMMANDS ON A COMMAND LINE

"Chaining" allows you to enter more than one command in a single command line. The ! (exclamation mark) symbol separates the commands. FlexOS executes the commands in the order in which they appear in the command line.

The following command line displays the current directory of drive A, renames a file, and displays the contents of another file. Note that FlexOS displays each command line it runs.

A>DIR ! RENAME NOW.EVT THEN.OLD ! TYPE MSG.TXT

Volume in Drive A: has no label
Directory of A:

JANUARY	<DIR>	1-06-1986	9:34a
APRIL	<DIR>	3-30-1986	11:03a
ACCOUNTS PAY	1429	1-18-1986	1:42p
ACCOUNTS REC	13421	1-03-1986	8:46a

6 Files 324256 bytes free

A>RENAME NOW.EVT THEN.OLD

A>TYPE MSG.TXT

To All Employees:

Please turn in your monthly status reports by noon on Friday.

Thank you.

FILTERING DATA

"Filters" are FlexOS commands that receive input from a file or command, perform a special operation, and send the resulting output to the screen, a file, device, or command.

Note: Filters alter the way information is displayed; they do not alter the contents of your files, or affect the output of your commands.

The following describes the FlexOS filters briefly. Detailed descriptions of the filters are in Section 7.

FIND Searches one or more files for a specified text string; searches the output of a command for a specified string. FIND accepts options that display the number of matches found or the line number of the matches, as well as an option that excludes the string from the output.

MORE Displays a file or the output of a command one screen at a time.

SORT Sorts file input or command output alphabetically. SORT accepts options that sort in reverse alphabetical order and that sort on the characters in a specified column.

End of Section 3

DR EDIX TEXT EDITOR

INTRODUCTION TO DR EDIX

DR EDIX is a text-editing program with which you can create a variety of text files, including memos, reports, and computer programs. DR EDIX is a "what-you-see-is-what-you-get" editor -- your screen shows how the printed page will look.

Unlike some text-editing programs, DR EDIX does not read from files or write to files automatically. Everything you type is stored in a block of memory called a "buffer" until you tell DR EDIX what to do with it. The beginning and end of the buffer are equivalent to the beginning and end of a file.

Most DR EDIX commands are invoked by holding down the Ctrl key -- represented in this guide by a caret (^) -- as you press one or two letter keys. Thus, ^G means you press the Ctrl key and the G key; ^Z means you press the Ctrl key and the Z key. (The text uses uppercase letters for readability, but you can use lowercase letters.)

DR EDIX Help Facility

DR EDIX has an on-line help facility that you invoke by pressing ^J.

The help facility is context-sensitive. For example, when you are editing text, ^J produces a summary of all the DR EDIX commands. After you choose a command and DR EDIX displays a prompt, ^J provides specific instructions on what to do next.

When you want to leave the help facility and return to your session, press the Spacebar.

DR EDIX Files

Table 4-1 lists the DR EDIX files.

Table 4-1. DR EDIX Files

File	Purpose
DREDIX.286	DR EDIX program.
HELP.EDX	Help text.
LSN.EDX	DR EDIX tutorial.
STUDENT.BAT	Starts the DR EDIX tutorial.
TEXT	Sample text used by the tutorial.

The two files you need to run DR EDIX are DREDIX.286 and HELP.EDX. STUDENT.BAT, LSN.EDX, and TEXT are tutorial files that can be removed from your disk once you have learned to use DR EDIX.

LEARNING DR EDIX

If you are not familiar with text-editing programs, the DR EDIX on-line tutorial teaches you the basics of DR EDIX and gives you the background to understand the rest of this section. You should go through the tutorial before reading the rest of this section.

If you are already familiar with text-editing programs, you can turn to "Starting DR EDIX" in this section.

The tutorial requires that the files DREDIX.286 and STUDENT.BAT be on your current or system drive, and that they be accessible from your current directory. The files HELP.EDX, LSN.EDX, and TEXT must be in your current directory.

To start the tutorial, type the following command line:

```
A>STUDENT
```

From this point the tutorial tells you what commands to enter and when to type.

You can exit from the tutorial at any time by pressing ^KQ. This returns you to the system prompt.

STARTING DR EDIX

To start DR EDIX, make sure DREDIX.286 and HELP.EDX are on the current or system drive and then type the following command and press Enter:

```
A>DREDIX
```

THE DR EDIX SCREEN

The DR EDIX screen is divided into four areas:

- text area
- prompt area
- message area
- status area

Figure 4-1 shows these four areas.

MEMORANDUM

To: T.W. Smith
 From: B.J. Jones
 Date: 9/28/86
 Re: Quarterly Review

Please note that the quarterly review has been rescheduled due to the move.

The meeting will now be held in the Polynesian Room of Lenny's Restaurant, Friday evening, October 24th. ←

Please notify your staff that attendance is strongly encouraged as an important announcement will be made.

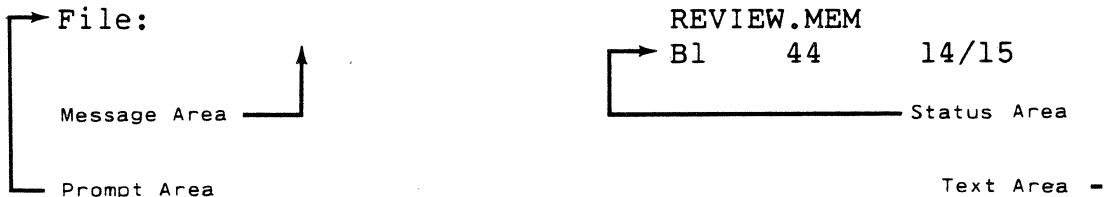


Figure 4-1. DR EDIX Screen Areas

Text Area

The text area is where you enter text, move your cursor, and enter DR EDIX commands.

Prompt Area

Some DR EDIX commands prompt you for additional information. For example, when you use ^KE to read a file into your current buffer, DR EDIX prompts you for a filename.

Any response to a prompt must be 50 characters or fewer,

and you must press the Enter key to transmit your response to DR EDIX.

To cancel a command after getting a prompt, press ^KQ. If you need help after getting a prompt, press ^J.

Message Area

The message area displays DR EDIX messages in reverse video. For a complete list of these messages, see Section 6.

Status Area

The status area gives you continuous information about the current window. Figure 4-2 shows a typical status area.

```
REVIEW.MEM
B1      44  14/15
```

Figure 4-2. DR EDIX Status Area

This status area contains the following information:

REVIEW.MEM	The name of the file in the current buffer.
B1	Buffer 1 – the current buffer.
43	The column in which the cursor is located.
14	The line on which the cursor is located.
15	The length (in lines) of the current buffer.

When the cursor changes windows, the status area is immediately updated to reflect the status of the new current window.

ENTERING AND EDITING TEXT

To enter text, you simply type at your keyboard. DR EDIX echoes each character to the screen and stores it in the current buffer. At the end of each line, press the Enter key. This inserts a carriage return at the end of the line and moves the cursor to the start of the next line.

Inserting and Overtyping

DR EDIX has two modes of entering text: Insert and Overtyping. Insert Mode is the default when you start DR EDIX. You can alternate between Insert and Overtyping modes by pressing ^V.

Insert mode allows you to insert one or more characters into a line without otherwise changing the line. The character you type is inserted at the cursor's position, and the cursor and all trailing characters on the line move to the right.

In Overtyping mode, typed character replace the characters at the cursor's position. The cursor moves to the right, but the trailing characters on the line do not move. When you are in Overtyping Mode, the status line at the bottom of your screen displays the letter O in reverse video.

MOVING THE CURSOR

Table 4-2 on the following page summarizes the function of the cursor control keys.

Table 4-2. DR EDIX Cursor Control Keys

Keys	What They Do
^S	Moves cursor one space to the left without erasing characters.
^D	Moves cursor one space to the right without erasing characters.
^A	Moves cursor one word to the left.
^F	Moves cursor one word to the right.
Tab	Moves cursor to the next tab stop. Tab stops are initially set eight spaces apart in columns 9, 17, 25, and so on.
^QS	Moves cursor to the beginning of the line.
^QD	Moves cursor to the end of the line.
^E	Moves cursor to the same position in the line above.
^X	Moves cursor to the same position in the line below.
^R	Moves cursor up one page.
^C	Moves cursor down one page.
^QR	Moves cursor to the top of the buffer.
^QC	Moves cursor to the bottom of the buffer.
^QE	Moves cursor to the top of the window.
^QX	Moves cursor to the bottom of the window.

EDITING TEXT

Table 4-3 on the following page lists and describes the DR EDIX editing command keystrokes.

Table 4-3. DR EDIX Editing Keys

Keys	What They Do
^G	Deletes the character at the cursor's position.
Del	Deletes the character at the cursor's position.
Backspace	Deletes the character to the left of the cursor and moves the cursor and all trailing characters one position to the left. If the cursor is at the beginning of a line, Backspace deletes the carriage return at the end of the previous line.
^Y	Deletes the current line.
^QY	Deletes the current line from the cursor's position to the end of the line.
^ZU	Recalls the last line you deleted with ^Y . (^ZU does not work for characters deleted with ^QY .) DR EDIX remembers the last 50 non-blank lines deleted, and with each additional ^ZU , DR EDIX restores an earlier line. The lines reappear in reverse deletion order (last in-first out).
Enter	Splits a line of text, moving the character under the cursor and any trailing text down to a new line. Any lines below the split line move down too.
^ZI	("i", not "I"). Inserts a blank line <u>before</u> the current line; the cursor appears at the beginning of the new line.
^ZA	Inserts a blank line <u>after</u> the current line; the cursor appears at the beginning of the new line.

To join two consecutive lines, put the cursor at the end of the upper line and press ^G, or put it at the beginning of the lower line and press Backspace. DR EDIX joins the two lines and puts a blank space between the two sections. If you do not want the blank space, press ^G again.

OPENING AND SAVING FILES

Table 4-4 on the following page lists and describes the commands for opening and saving files.

Table 4-4. DR EDIX File Commands

Keys	What They Do
^KS	<p>Saves your text to a file. DR EDIX writes the contents of your buffer to a file.</p> <p>If the text in the buffer does not have a name, DR EDIX prompts you for one:</p> <p>File:</p> <p>Type a filename for your text and press Enter.</p> <p>^KS does not change the current buffer in any way. After saving the buffer to a file, you can continue editing the buffer.</p>
^KE	<p>Clears the buffer so you can start a new file or read in an existing file for editing. DR EDIX prompts you for a filename. If you want to open a file on a different drive, include the drive letter and a colon (:) before the filename.</p> <p>If you have not saved the contents of the buffer, DR EDIX asks if you want to lose the changes you have made to the buffer.</p>
^KQ	<p>Cancel a command (like ^KE) that produces a prompt. The main function of ^KQ is to exit DR EDIX (see "Exiting DR EDIX, later in this section).</p>

DR EDIX has the same rules for naming files as FlexOS. Do not use a filename that already exists unless you want to replace the contents of the old file with new material.

Note: You should save your files often during lengthy editing sessions. If the power goes off or your computer fails, edits made after the last ^KS are lost.

DR EDIX BACKUP FILES

Before DR EDIX writes a file to disk, it takes the previous version of the file and makes a "backup" version with the file extension BAK.

Backup files guard against losing the file entirely because of an error during the write operation. If DR EDIX detects an error while writing your file, it displays a message and restores the original file. If DR EDIX does not detect the error, as can happen in a power failure, you might end up with two very different versions of the file, one with extension BAK and the other with the original extension. (Sometimes you'll see significant differences in the sizes of the two files in your directory listing.) In that case, when your computer is running again, you should compare the two versions of the file, deleting the more corrupted version and giving the other the name you want to save it under.

You should never lose data because of a write error, provided you heed the messages from DR EDIX and use the Help command (^J) whenever you are unsure of what to do.

Backup files are also valuable if you accidentally erase the current version of your file, but they do take up valuable disk space. If space is an issue, you can copy your current backup files onto a diskette and then erase the backup files from your current directory.

EXITING DR EDIX

To end a session in DR EDIX, press `^KQ`. If you have been working in a buffer, you would typically press `^KS` to save before you press `^KQ`.

If you press `^KQ` before saving the buffer, DR EDIX displays the following prompt:

```
OK to lose changes (y/n)?
```

If you press `Y`, you exit DR EDIX and return to the system prompt without your changes being written to disk. If you press `N`, you can continue editing, or you can choose another DR EDIX command (like `^KS`).

You can use `^ZS` to review the status display to see which buffers have unwritten changes in them. The status display is described in Section 5 under "Status Table."

As noted previously, `^KQ` also cancels commands that produce prompts.

End of Section 4

ADVANCED DR EDIX FEATURES

This section teaches you some advanced DR EDIX features, including:

- how DR EDIX handles buffers and windows
- how to work with two or more buffers at once
- the DR EDIX status table
- how to move, copy, or delete blocks of text
- how to merge one file into another

BUFFERS

DR EDIX stores your files in blocks of memory called "buffers." For all practical purposes, once you read a file into a buffer or create new text in an empty buffer, you can treat the buffer as a file, moving from the beginning to end, adding and deleting lines, writing it to a disk, and so forth.

A line in a DR EDIX buffer can be up to 255 characters long. Because some of these characters can be tabs, you can actually enter lines that take over 429 columns to display. This is useful for editing very wide spreadsheets.

DR EDIX does not limit the number of lines in a buffer, but adds lines to the buffer until the available memory is filled. If you fill a buffer, use the ^KS command to write the buffer contents to a file.

DR EDIX can divide its memory into as many as four buffers, which means you can work on as many as four documents at once, switching back and forth among the buffers.

WINDOWS

You can split your DR EDIX screen horizontally into two "windows," which are effectively the same as having two separate physical screens.

When you begin a DR EDIX session, Window 1 occupies the entire screen; it is the only window allowed to occupy the whole screen. The command ^Z2 splits the text area horizontally to create Window 2, which can occupy only the bottom half of the screen.

The commands ^Z1 and ^Z2 move the cursor to Window 1 and Window 2, respectively. If the screen has not already been split into two windows, ^Z2 first splits it and then moves the cursor.

The command ^ZO (the letter O, not zero) restores the full text area to Window 1.

Table 5-1 on the next page summarizes the DR EDIX windowing commands.

Table 5-1. DR EDIX Window Commands

Command	Meaning
^Z1	Window 1. Switches cursor to Window 1. Window 1 is always present on the screen.
^Z2	Window 2. If necessary, splits the screen to show Window 2, then switches cursor to Window 2.
^ZO	One window. Restores the screen to one window (Window 1). Use the letter O, not zero.

Buffers and Files

When you start DR EDIX, Buffer 1 is assigned by default to Window 1 and Buffer 2 by default to Window 2, even though Window 2 is not visible.

If you start DR EDIX without specifying a filename, Buffer 1 is empty and has no name until you do one of the following:

- You write its contents to a disk using ^KS. DR EDIX prompts you for a filename.
- You assign the buffer a filename using ^KF.
- You explicitly read a file into the buffer using ^KE. DR EDIX prompts you for a filename, which then becomes the name of your buffer.

You can display any buffer on the screen by assigning it to a DR EDIX window. To assign a buffer to a window, use the ^ZB command. DR EDIX displays the following prompt:

Buffer number or file:

If you respond with a number, DR EDIX displays the specified buffer in the current window. If you give a filename, the buffer associated with that filename is displayed in the current window.

STATUS TABLE

The status table displays additional information about each of your files, windows, and buffers. To display the status table, press ^ZS, the status command. Figure 5-1 shows a sample status table.

File	Bfr	Lines	Wnd	Chg	Mrk
REVIEW.MEM	1	18	1	Y	
EXPENSES.MEM	2	13	2		
GUESTS.LST	3	25		Y	Y
	4	1			

Figure 5-1. DR EDIX Status Table

The status table lists all four buffers, whether or not they contain anything. The first column (File) shows the filename, if any, associated with each buffer. The remaining columns display the buffer number (Bfr), the number of lines in each buffer (Lines), the window number in which it appears (Wnd), whether the buffer has unwritten changes (Chg), and whether the buffer contains a marked block (Mrk).

For example, REVIEW.MEM is in Buffer 1, has 18 lines of text, is displayed in Window 1, and contains changes that have not been written to disk.

Notice that GUESTS.LST contains a marked block of text. (Block commands are described later in this section.) You can

only have one marked block at a time, and the status table shows its location.

Notice also that although Buffer 4 is empty, the status table displays one line of information for it. A buffer is always at least one line long, even when empty.

BLOCK COMMANDS

A "block" is a line or a group of lines that you mark for copying, moving, or deletion. DR EDIX supports the following block commands:

- `^KC` Copies a block of text within a buffer or between buffers.
- `^KV` Moves a block of text within a buffer or between buffers.
- `^KY` Deletes a block of text from a buffer.

Note: You cannot mark only a portion of a line. The smallest possible block is a single line.

Marking Blocks

Before you use a block command, you must mark the block with the command `^KB`. If you try to execute a block command without marking a block, DR EDIX displays the following message:

```
No source block
```

The block must be entirely within the same buffer (you cannot mark across buffers), and the buffer must be displayed in a window when you mark the block. However, once the block is marked, you can put this buffer in the background.

To mark a single line, press ^KB twice while the cursor is anywhere in the line. To mark a group of lines, press ^KB on the first line of your block. DR EDIX displays the following message:

```
Waiting for second mark ...
```

Move the cursor to the last line of the intended block and press ^KB again. DR EDIX highlights the marked lines by displaying the first character of each marked line on the screen in reverse video.

Once you have marked the block, you can execute one of the block commands.

You can unmark a block by doing any of the following:

- Pressing ^KB a third time while the cursor is anywhere in the buffer that contains the marked block.
- Pressing ^KY, which deletes the block.
- Pressing ^ZG and then entering 0 (zero) or pressing the Enter key in response to the prompt. See "Going to a Line by Number," later in this section.

Note that there can be only one marked source block at a time. If there is a marked block and you press ^KB in the same buffer, you unmark the block. If you press ^KB in a different buffer, DR EDIX displays the following message:

```
Source block marked in some other buffer
```

To mark another block, you must first unmark the current marked block. If the marked block is not visible on the screen, use the ^ZS command to display the status table. Then assign the buffer with the marked block to a window and press ^KB to unmark the block.

Copying Blocks

To copy a block you have marked, locate the cursor where you want the block copied and press ^KC. You can copy a block within its own buffer or to any other buffer that is displayed in a window. Unless you are copying to the same buffer, the source block and its buffer do not have to be in a window.

After you copy a block, the source block stays marked, and you can use the ^KC command to copy the same block repeatedly.

Moving Blocks

To move a block you have marked, locate the cursor where you want the block moved and press ^KV. The marked block is inserted after the line containing the cursor. You can move a block within its own buffer or to any other buffer that is displayed in a window. Unless you are moving within the same buffer, the source block and its buffer do not have to be in a window.

After you move a block, it stays marked, and you can use the ^KV command to move it again. Each time you move a block, it is deleted from its original location.

Deleting Blocks

To delete a marked block, press ^KY. The buffer containing the block does not have to be visible in a window. Deleting a marked block changes the status of its buffer from marked to unmarked.

You can undo a block deletion by pressing ^ZU command for every line that was deleted, up to a maximum of 50 lines. You cannot recall an entire block of deleted lines with one command.

SEARCH AND REPLACE

Using the `^QF` command, you can search for and move the cursor to a "pattern" in your text. A pattern can be a single character or a string of characters (like a word or sentence) of 50 characters or fewer. Using `^QA`, you can search for a pattern and replace it with one you specify. You can create complex search patterns with the wildcard characters `*` and `?`, as described later.

Searching

To search for a pattern in a buffer, press `^QF` and enter the pattern after the prompt, as in the following example:

```
Pattern:  Buffalo Bill
```

DR EDIX makes a "circular" search of the buffer. It starts at the cursor's position, searches to the end of the buffer, jumps back to the top of the buffer, and continues searching until it returns to the original cursor position.

If DR EDIX finds a pattern that matches what you've specified, it reports "Found", moves the cursor to the first occurrence of the pattern in the buffer, and stops the search. You can continue the search by pressing `^L` for each additional occurrence of the pattern.

If DR EDIX does not find a matching pattern, it reports "Not found" and returns the cursor to its original position.

Because DR EDIX retains a pattern until you replace it with a new `^QF` or `^QA` command or until you quit the editor, you can continue a search with `^L` at any time, not just immediately after the initial search. For example, you can search on a string, edit the text at the first occurrence, and then continue the search.

Recovering Search Patterns

You can recover the pattern from the previous search by pressing ^E and the current pattern by pressing ^X, as in the following example.

1. You press ^QF, enter the pattern "hard disk drive", and press the Enter key.
2. After DR EDIX finishes searching, you press ^QF, enter the pattern "floppy disk drive", but do not press the Enter key.
3. You press ^E. "Hard disk drive" (the previous search pattern) replaces "floppy disk drive".
4. You press ^X. "Floppy disk drive" (the current search pattern) replaces "hard disk drive".

Searching With Wildcards

You can use the star (*) and question mark (?) wildcard characters in your search patterns.

The * wildcard stands for any string of any length in a line. For example, the pattern "a*p*" matches the following strings:

```
application
astute programmer
antipasto
```

The ? wildcard stands for any single character (including a tab or a space), but not the end-of-line character. For example, the pattern "c?re" matches all of the following:

```
care
c re
c:re
```

The * and ? wildcards can appear any number of times in a

pattern. However, several * wildcards will make your search very general, and it might take DR EDIX a long time to find all the possible matches.

To search for a pattern that contains a star or question mark as a literal character instead of a wildcard, precede the star or question mark with a backslash (\). Similarly, if the string contains a backslash, precede it with a second backslash.

<u>string</u>	<u>search pattern</u>
3 * 2 = 6	3 \ <u>*</u> 2 = 6
Yes/No?	Yes/No\ <u>?</u>
hello\goodbye	hello\ <u>\</u> goodbye

Replacing

The command ^QA replaces the search string with a string you specify. For example, to replace the word "microprocessor" with "chip", you would do the following:

1. Press ^QA. At the pattern prompt, type:

Pattern: **microprocessor**

Press the Enter key.

2. DR EDIX displays the replacement prompt. Type this:

Replacement: **chip**

The replacement text does not have to be the same length as the matched text, but it cannot be more than 50 characters. To send the command to DR EDIX, press Enter again.

DR EDIX makes a circular search of the buffer, just as it did for the ^QF command, ultimately returning to the original cursor position. If a match is found, this message is displayed:

l=local; g=global; ^KQ=exit; other=skip

You have the following options:

- | | |
|------------|--|
| l=local | Replaces only the current string and then continues the search for the next occurrence of the search pattern. If one is found, DR EDIX again displays the replacement options. |
| g=global | Replaces all matching strings in the buffer. Once you press g, you cannot cancel the replacement operation. |
| ^KQ=exit | Cancels the search-and-replace operation. |
| other=skip | Does not replace the current matching string, but continues the search for the next occurrence of the pattern. "Other" can be the spacebar or any key other than L, G, or ^KQ. |

If no match is found, DR EDIX displays the following message:

Not found

You can replace the search pattern with a "null string" -- a string containing nothing, not even a blank space. To do so, simply press Enter at the replacement prompt. This is a fast way of doing global deletions in the buffer.

If you use the * and ? in the replacement pattern, DR EDIX places them in the text as literal characters, not as wildcards.

As with searching, DR EDIX remembers the pattern and replacement string from the previous ^QA command. If a previous pattern or replacement string exists, you can bring it to the prompt response area by pressing ^E.

GOING TO A LINE BY NUMBER

DR EDIX assigns a number to each line in the buffer. Although these numbers are not shown on the display, the number of the line in which the cursor is located is shown in the status area. (Line number support is provided for use by other programs, like language compilers, used in programming.)

You can use the ^ZG command to go to a specific line. DR EDIX prompts you for a line number, which must be between 1 and the number of lines in the buffer. (If you enter a number that is out of this range, DR EDIX prompts you again.) After you enter the line number, DR EDIX moves the cursor to the first column of that line.

If you enter 0 for the line number or only press Enter, DR EDIX returns the cursor to its location before you pressed ^ZG. As noted previously, this is also a way of unmarking a marked block of text.

MERGING FILES

Using the ^KR command, you can read the contents of an existing file into the current buffer.

Locate the cursor on the line immediately before where you want the file inserted. When you press ^KR, DR EDIX prompts you for a filename. After you respond and press Enter, DR EDIX inserts a copy of the file into the buffer after the current line.

Reading in a file does not affect the name (if any) associated with the current buffer, and it does not alter the original of the file that was read in.

If you want to read in only part of a file into a buffer, you have two choices:

- Read in the whole file and then delete the part you do not need.
- Read the file into another buffer, mark the block you want, and use ^KV to move it into the target buffer.

CHANGING THE FILENAME

You can change the current filename with ^KF. DR EDIX prompts you to enter a new filename. Changing the filename does not otherwise alter the file.

End of Section 5

DR EDIX COMMANDS AND MESSAGES

DR EDIX COMMAND SUMMARY

Most DR EDIX commands are invoked by pressing a combination of the Ctrl key, represented by a caret (^), and another key.

The following tables summarize the DR EDIX commands. The tables group the commands into the following categories:

- line editing commands
- file/buffer commands
- windowing commands
- block commands
- search commands
- status/help commands

Table 6-1. DR EDIX Line-Editing Commands

Command	Description
^A	Places cursor on first character of previous word.
^F	Places cursor on first character of next word.
^QE	Places cursor at first non-blank character of first line of window.
^QX	Places cursor at first non-blank character of last line of window.
^Y	Deletes current line.
^\	Inserts blank line before current line.
^ZA	Inserts blank line after current line.
^ZG	Prompts for line number and places cursor at the start of that line. Responding 0 or Enter returns cursor to its original position and unmarks a marked block (if any).

Table 6-2. DR EDIX File/Buffer Commands

Command	Description
^KS	Saves buffer under its current filename or, if buffer does not yet have a name, prompts for filename and then writes buffer to a file under that name.
^KE	Prompts for filename, clears current buffer, and then either reads specified file into buffer or assigns specified filename to empty buffer.
^KF	Prompts for filename and then assigns that filename to contents of current buffer.
^KR	Prompts for filename and then reads that file into current buffer, following the current line. Places cursor at first character of inserted text.
^KQ	Exits from DR EDIX back to FlexOS or, if issued in response to a prompt, cancels command that caused prompt.
^ZB	Prompts for buffer number or filename, then brings specified buffer or file into current window.

Table 6-3. DR EDIX Windowing Commands

Command	Description
^Z1	Switches to Window 1 and its associated buffer. (Use the 1 key on the top row of your keyboard, not the keypad key.)
^Z2	Switches to Window 2 (if open) and its associated buffer. If Window 2 is not open, first splits the screen to show Window 2. (Use the 2 key on the top row of your keyboard, not the keypad key.)
^Z0	Restores the screen to one window (Window 1).

Table 6-4. DR EDIX Block Commands

Command	Description
^KB	Marks first or last line of block. If block is already marked, unmarks block.
^KC	Inserts copy of marked block after current line.
^KV	Moves marked block to follow current line. Places cursor at first character of moved block.
^KY	Deletes marked block from buffer.

Table 6-5. DR EDIX Search Commands

Command	Description
^QF	Prompts for pattern and then searches for text that matches it.
^QA	Prompts for pattern and then prompts for replacement string. Searches for pattern and then gives you choice of local replacement, global replacement, canceling the replacement, or skipping to next match.
^L	Searches for next occurrence of current pattern.

Table 6-6. DR EDIX Status/Help Commands

Command	Description
^ZS	Displays status screen.
^J	Provides help information.

DR EDIX MESSAGES

This section describes the messages that DR EDIX displays on your screen. The messages are listed in alphabetical order.

Bad pattern

The specified pattern contains an invalid combination of characters, usually a wildcard. Refer to "Search and Replace" in Section 5 or use ^J for help.

Bad target

The cursor was still inside the marked block when you issued the ^KC or ^KV command. Move the cursor out of the marked block.

Can't create <filename.ext>

A new file cannot be created with a ^KS command, probably because of a full directory or insufficient disk space. Try writing the file to another drive, or change disks and reissue the command.

Can't delete <filename.ext>

The backup file cannot be deleted, perhaps because of a defective disk, a problem with your disk drive hardware, or because you have protected your backup diskette with the FSET command.

Can't open <filename.ext>
Can't open <filename.ext>; new file assumed

The file you are trying to read with ^KR or ^KE cannot be opened, perhaps because you misspelled the filename or because you have not given the full directory pathname. The second message, resulting from a ^KE command, assumes you are opening a new file.

Device full

The disk to which you are writing with ^KS is full. The file has been restored to the way it was before the write. You must write to a different drive or make room on the disk by erasing unneeded files. Press ^J for instructions.

Empty buffer

You have attempted to delete the only line in an empty buffer. An empty buffer always contains an end-of-line character, which cannot be deleted.

Empty delete stack

You have issued a ^ZU (undo) command when there are no deleted lines in the delete stack.

Invalid command

You have issued a command that is invalid in this context. An example is ^ZG given in response to a pattern prompt. To cancel the prompt, use ^KQ.

Invalid response

Responses to a (Y/N)? query must be an uppercase or lowercase Y or N.

No buffer has that file

No buffer is associated with the filename you entered. Use ^ZS to see which files are associated with which buffers, or invoke a buffer by number.

No help file

DR EDIX cannot find the file named HELP.EDX. This file must be on the current drive or on drive A.

No pattern

You have pressed ^L to continue a search, but there is no search pattern stored.

No source block

You have issued a DR EDIX block command, but there is no marked block.

Not found

DR EDIX could not match the pattern you entered.

Number required

After pressing ^ZG, you must type a number following the line prompt. To cancel the ^ZG, press ^KQ.

Out of range

The line number you gave in response to a ^ZG command must be between 1 and the number of lines in the buffer. Press ^KQ to cancel the ^ZG command or type a valid line number.

Read error

A read error occurred while DR EDIX was reading a file. This message can occur if a bad filespec is specified or if the file that DR EDIX is reading is not terminated with an end-of-file character (decimal 26).

Response too long

The response you typed after a prompt has too many characters for that prompt. Press ^J to see which responses are appropriate.

Source block marked in some other buffer

You have tried to mark a block while there is still a block marked in another buffer. Use ^ZS to see which buffer has the marked block. Make that buffer current (with ^ZB or a window-switching command) and press ^KB to clear the marks. You can also use ^ZG to go directly from your current position to the start of the source block.

Text capacity exceeded

You have filled all allocated memory with text. You should write all buffers to disk, exit from DR EDIX, and reinvoke DR EDIX. Although DR EDIX usually maintains the integrity of internal storage structures when memory is exhausted, it is safer to get a fresh start.

Unknown command. Press ^J for help.

You have typed a command that is unknown to DR EDIX.

Write Error

A write error occurred while DR EDIX was writing a file, probably as the result of a defective disk. This error is different from the "device full" condition.

End of Section 6

FLEXOS COMMANDS**COMMAND SUMMARY**

This section describes each of the FlexOS commands and illustrates them with selected examples.

FlexOS commands let you do things like copy and rename your files, look at your directory, and print your data. Unless otherwise noted, you may not abbreviate command names.

This section lists the FlexOS commands in alphabetical order for easy reference; Tables 7-1 through 7-6 organize the commands according to the task you wish to perform.

Table 7-1. Working with Files

Task	Command
Compare two files	COMP
Copy a file	COPY
Erase a file	ERASE
Define search order of file extensions	ORDER
Send files to print spooler for background printing	PRINT
Analyze and repair a file	RECFILE
Rename a file	RENAME
Display text	TYPE

Table 7-2. Working with Directories

Task	Command
Change the current directory	CHDIR
Display files in a directory	DIR
Make a directory	MKDIR
Set up a search path	PATH
Analyze and repair a directory	RECDIR
Remove an empty directory	RMDIR
Display subdirectories	TREE

Table 7-3. Working with Disks

Task	Command
Back up a disk	BACKUP
Display disk statistics	CHKDSK
Compare two diskettes	DISKCOMP
Copy a diskette	DISKCOPY
Partition a hard disk	FDISK
Prepare a disk	FORMAT
Restore backed up files to disk	RESTORE
Verify information written to disk	VERIFY
Display the volume label	VOL

Table 7-4. Data Protection

Task	Command
Write a volume and enable or disable disk protection	DISKSET
Change the protection or attributes of a file	FSET
Log off the system	LOGOFF
Log onto the system	LOGON
Change a password	PASSWORD
Display or change file access privileges	SECURITY

Table 7-5. Modifying Your User Environment

Task	Command
Reroute drive requests	ASSIGN
Create a new shell	COMMAND
Configure a serial port	CONFIG
Change the standard input and output devices	CTTY
Create a logical name	DEFINE
Exit a shell you have created	EXIT
Change the system prompt	PROMPT

Table 7-6. Additional Commands

Task	Command
Start a command that runs in the background	BACK
Enable or disable control breaks	BREAK
Cancel a running program	CANCEL
Set or display the date	DATE
Search files or command output for text strings	FIND
List the built-in FlexOS commands	LIST
Display a file or command output one screen at a time	MORE
Display information on running programs	PROCESS
Sort a file or command output	SORT
Transfer operating system files to another disk	SYS
Set or display the time	TIME
Display the version of FlexOS	VER

ASSIGN

Forms Give a drive a new assignment:

```
ASSIGN original_drive=new_drive
```

Reset a drive to its original assignment:

```
ASSIGN  
ASSIGN original_drive=  
ASSIGN original_drive=original_drive
```

Explanation

The ASSIGN command routes requests for one drive to another drive.

Changes to drive assignments only affect the program you are running in your current window. They do not affect other programs, commands you have already entered, or other users on the system.

ASSIGN also permits applications designed to run only on drive A or B to use other drives, such as hard disk drives C or D.

Colons after the drive letters are optional.

Examples

```
A>ASSIGN A=C
```

Routes requests for drive A to drive C.

```
A>ASSIGN
```

Resets all drives to their original assignments.

A>ASSIGN B=

Resets drive B to its original assignment. After the drive is reset, the command DIR B: displays the directory of drive B.

A>ASSIGN A=A

Explicitly resets drive A to itself.

A>ASSIGN A=C B=C

Combines assignments on a single command line. This example routes requests for both drive A and drive B to drive C. You must separate the assignments with blank spaces.

A>ASSIGN A=C B=

Routes requests for drive A to drive C and resets drive B to its original assignment.

BACK

Forms BACK command
 BACK batchfile

Explanation

The BACK command starts a command or batch file and runs it in the background. The background command continues to run even after you have invoked LOGOFF.

BACK invokes the command or batch file and then displays its process ID. To stop the background command or batch file, use the process ID with the PROCESS or CANCEL command.

After invoking BACK, you can run other programs from the shell from which you invoked BACK. However, the background command cannot get access to the shell from which it was started.

You can redirect a background command's input and output as described in Section 3. Redirecting a background command's output to a file is one way of determining whether the command ran successfully.

Examples

A>BACK DIR > LOGFILE.DOC

Writes the output of DIR to the file LOGFILE.DOC and displays the following message:

```
Process id = 0007 started Mon 08-26-86 08:05:00.00
```

A>BACK BATFILE

Runs all commands in the batch file BATFILE.BAT in the background and displays the following message:

```
Process id = 0023 started Fri 10-04-86 17:12:16.04
```

BACKUP

Forms Back up the current directory:

BACKUP source_drive: backup_drive:

BACKUP source_drive: backup_drive: -option

Back up one or more filespecs:

BACKUP source_drive: filespec backup_drive:

BACKUP source_drive: filespec backup_drive: -option

Explanation

The BACKUP command makes archival copies of one or more files from a source disk onto a backup diskette. You can then restore the backed up files to the source drive with the RESTORE command. You might typically back up your files in preparation for formatting your hard disk.

Caution: Unless you specify the -A option (see Table 7-7), BACKUP erases the backup diskette before writing files to it from the source disk.

Table 7-7 summarizes the BACKUP options. You can use these options singly or in combination.

Table 7-7. BACKUP Options

Option	Description
-A	Adds files to the backup disk without erasing files already on the disk.
-D:mm-dd-yy	Only backs up files modified on or after the specified date. The date must be specified in month, day, and year form. -D can be specified with the -A option.
-M	Only backs up files modified since the last backup.
-S	Backs up the current directory of the source disk, as well as all the subdirectories attached to the current directory.

If you do not specify a directory path, files are backed up from the current directory. If you do not specify a filespec, all the files in the directory are backed up. Files are backed up to the root directory of the backup diskette.

As BACKUP fills each diskette, it prompts you to insert another, until the directory has been completely backed up.

BACKUP displays the name of each file as it backs it up.

Examples

A>BACKUP C: A:

Backs up the files from the current directory of the source disk in drive C onto the backup disk in drive A. FlexOS displays the following message:

```
Insert the backup diskette #01 in drive A:  
Warning! All files on diskette in drive A: will be erased.  
Press any key when you are ready ...
```

A>BACKUP D: A: -A

Uses the -A option to add files to the backup disk. Files already on the backup disk are not erased, and the warning message is not displayed.

A>BACKUP C: *.TXT A:

A>BACKUP C: *.TXT A: -A -M

Specifies a filespec with the BACKUP command. This enables you to back up selected files rather than the entire disk. The combination of the -A and -M options backs up only those .TXT files modified since the last backup and adds them to the files already on the backup disk.

BREAK

Forms BREAK
 BREAK ON
 BREAK OFF

Explanation

The BREAK command has been included with FlexOS for compatibility with other DOS operating systems.

In some operating systems the BREAK command turns the Ctrl-C function on and off. Ctrl-C stops running programs. In FlexOS, Ctrl-C always stops a running program whether BREAK is set to ON or OFF.

If you use any of the command forms shown above, either from the command line or within a batch file, FlexOS ignores the command and continues processing.

CANCEL

Forms CANCEL process_ID
 CANCEL process_name
 CANCEL process_name window_number
 CANCEL -F family_ID

Explanation

The CANCEL command stops a running program and removes it from the system. A program is defined as any user application or FlexOS command. You can only cancel your own programs; however, a system manager can cancel any program running on the system.

When you run a program, it is identified by a process name and three ID numbers: a process ID, a window number, and a family ID. To determine these values for a particular program, use the PROCESS command described later in this section.

Under certain circumstances the CANCEL command does not return an error message when it fails to cancel a process. To determine if the process was in fact canceled, you must issue a PROCESS VIEW command after the CANCEL command. See the description of the PROCESS command later in this section.

Note: You can also stop a running program by using Ctrl-C. However, you can only do this if the program is running in the current window. For more information on using FlexOS windows, see Section 2.

Table 7-8 summarizes the options you can use with CANCEL.

Table 7-8. CANCEL Options

Option	Description
process_ID	Decimal number up to 10 digits long. FlexOS automatically assigns each running program a Process ID number.
process_name	Name up to eight characters long.
window_number	Decimal number up to three digits long.
-F family_ID	Decimal number up to five digits long.

Examples

A>CANCEL 6

Identifies the program to be canceled by its Process ID. This example stops a running program whose process ID is 6.

A>CANCEL ERASE

Cancels the specified program, regardless of the window in which it is running. If more than one program has the specified name, CANCEL terminates only the program most recently started. This example cancels the ERASE command.

A>CANCEL WPROC 3

Cancels the specified program in the specified window. Use this form if two or more programs have the same process name. In this example, the program WPROC is running in both Window 3 and Window 5. This command line cancels WPROC in Window 3.

A>CANCEL -F 3

Uses the -F family ID option to cancel all the programs in a process family. Process families include all the commands in a batch file or commands chained together using I/O redirection. This form cancels a process family in any window. This example cancels process family 3.

CHDIR

Forms CHDIR
 CHDIR directory_path

Abbreviation CD

Explanation

CHDIR lets you change the current directory. The current directory is the directory FlexOS looks in first when you enter a filename without specifying which directory it is in.

Changing the current directory implies that you have another directory on your disk to make current. If you try to change your current directory to a directory that does not exist, you receive an error message. For information on how to create a directory, see the MKDIR command in this section.

If you do not specify a drive with the CHDIR command, the current drive is assumed.

Examples

A>CHDIR

A:

Displays the current directory path of the current drive. In this example, because the current directory is the root directory, CHDIR displays only a drive letter.

A>CHDIR B:/INFO

Changes the current directory of drive B to the subdirectory INFO. If you change your default drive to drive B, your current directory will be INFO. If you copy a file from drive A to drive B, it will be copied to INFO.

A>CHDIR BASEBALL/NATIONAL

Changes the current directory of drive A to the path root/BASEBALL/NATIONAL. (The subdirectories BASEBALL and NATIONAL must already exist or you receive an error message.) If you are changing the directory path all the way back to the root directory, you must include a slash or back slash for the root directory.

You can confirm your change by doing the following:

A>CD

A:BASEBALL/NATIONAL

A>CHDIR GIANTS

Attaches additional directory levels to your current path, provided these directories already exist and are logically related to the current directory. This example adds the subdirectory GIANTS to the current directory of drive A.

Again, CHDIR confirms your change:

A>CHDIR

A:BASEBALL/NATIONAL/GIANTS

CHKDSK

Forms CHKDSK
 CHKDSK -F
 CHKDSK -V
 CHKDSK filespec
 CHKDSK filespec -F -V

Explanation

CHKDSK (check disk) displays statistics about disks, directories, files, and memory. You should run CHKDSK periodically for each disk to ensure the integrity of the file structures.

To prevent accidental changes to your disk, all Yes or No (Y/N) prompts from CHKDSK require you to press the Enter key after entering your response.

If you specify a filespec, CHKDSK reports the number of non-contiguous blocks. Non-contiguous blocks in a file can degrade system performance.

Table 7-9 describes the CHKDSK options.

Table 7-9. CHKDSK Options

Option	Description
-F	Causes CHKDSK to correct the errors it finds in the directory or File Allocation Table and write the corrections to the disk. Requires a Yes response to its prompt before it actually makes the corrections.
-V	Causes all pathnames on the specified or default drive to be displayed.

Examples

A>CHKDSK

Displays the following status report on the disk in drive A:

```
Volume in A: is DISKETTE1           Created MAY 10, 1986 11:25

1228800 bytes total disk space.
   512 bytes in boot area.
   7168 bytes in two FATs.
   7168 bytes (224 entries) in root directory.
1213952 bytes total file space.
266752 bytes in 2 hidden files.
160256 bytes in 12 user files.
   512 bytes in 1 bad clusters.
786432 bytes available on disk.

MEMORY:
913776 bytes total memory.
148970 bytes system memory.
206806 bytes allocated memory.
558000 bytes free memory.
```

The CHKDSK memory status report provides the following information:

total memory	Total memory available <u>after</u> FLEXOS.SYS has been loaded into memory.
system memory	Memory allocated by the system.
allocated memory	Memory allocated by a process.
free memory	Memory currently not in use: the Transient Program Area (TPA).

The sum of system memory, allocated memory, and free memory equals the total memory.

Note: CHKDSK reports two files that have the "hidden" attribute set. Some application programs create hidden files. Hidden files are not displayed in directory listings.

A>CHKDSK -F

Analyzes the directories and File Allocation Table on your disk and corrects any errors it finds.

If CHKDSK finds any lost allocation units (clusters) on the disk, it asks if you want to recover the lost data. If you answer yes, each chain of lost allocation units is recovered into its own file. The name of each file is FILEnnnn.CHK, where nnnn is a sequential number starting with 0000. The files are stored in the root directory of the current or specified drive. You can look at these files to see if they contain any useful information. If they do not, you can erase them.

A>CHKDSK -V

Displays a list of all files, including hidden ones, and their pathnames in addition to the memory status report.

CHKDSK C:*.*.TXT

Analyzes all TXT files in the current directory of drive C. In this case, CHKDSK's report includes the number of non-contiguous areas occupied by the specified file(s). By entering a wildcard in the filespec, many files can be analyzed at once.

Badly fragmented files (files with many non-contiguous blocks) can slow down system performance when they are accessed; the disk drive head has to move more often to find parts of the file. To determine how fragmented your files are, use CHKDSK periodically with a filespec of *.*. You can correct fragmented files by copying them to another disk.

```
A>CHKDSK *.* -F -V
```

Displays the number of non-contiguous areas occupied by the specified filespec and corrects any errors in the directory or File Allocation Table. CHKDSK only looks for the filespec in the current directory. In this example, the system responds by displaying all file pathnames with non-contiguous block information, as well as the memory status report.

```
Directory A:/
  A:/FLEXOSLDR.IMG
  A:/FLEXOS.SYS
  A:/RAMDSK.DRV
  A:/ATHD.DRV
Contains 1 non-contiguous blocks.
  A:/CONFIG.SAV
  A:/CONFIG.BAK
  A:/PRINTER.DRV
  A:/LOGON.286
Contains 2 non-contiguous blocks.
  A:/COMMAND.286
Contains 5 non-contiguous blocks.
  A:/WMEX.286
Contains 1 non-contiguous blocks.
  A:/AUTOEXEC.BAT
  A:/USER.TAB
  A:/CONFIG.BAT
Contains 1 non-contiguous blocks.
  A:/CONFIG2.BAT
Scanning FAT for lost chains.

DISK:
  Volume NONE   Created January 1, 1980   00:00:00

  1228800 bytes total disk space.
    512 bytes in boot area.
    7168 bytes in two FATs.
    7168 bytes (224 entries) in root directory.
  1213952 bytes total file space.
    266752 bytes in 2 hidden files.
    160256 bytes in 12 user files.
    512 bytes in 1 bad clusters.
  786432 bytes available on disk.
```

MEMORY:

913776 bytes total memory.
148970 bytes system memory.
206806 bytes allocated memory.
558000 bytes free memory.

COMMAND

Forms Invoke a copy of the primary shell:

```
COMMAND  
COMMAND <command>  
COMMAND -option <command>
```

Invoke a different shell:

```
COMMAND filespec -option
```

Explanation

The "shell" is a program that determines your user environment--what system prompt FlexOS displays, how the system interprets the command line, what command abbreviations it recognizes, and so on. The shell invoked when you start FlexOS is called the "primary" shell.

COMMAND lets you change from the primary shell to a secondary shell that can be a copy of the primary shell or different from the primary shell. The secondary shell might display a different prompt or use a set of specially defined commands. Note, however, that a secondary shell must be able to recognize the commands you are using. Some FlexOS commands are not compatible with other shells.

To return to the primary shell from a secondary shell, use the EXIT command. If you have used the -P option with COMMAND (see Table 7-10), EXIT has no effect.

Table 7-10 summarizes the options you can use with COMMAND.

Table 7-10. COMMAND Options

Option	Description
-P	Makes the secondary shell permanent in memory. You cannot EXIT from this shell. To leave this shell, you must LOGOFF or turn off your computer.
-C <command>	Runs a command on the secondary shell, then causes the secondary shell to terminate automatically when the command is finished. Substitute the command for <command> when you use this option.

Examples

A>COMMAND

Invokes a secondary shell that is a copy of your primary shell. The secondary shell has the same system prompt and recognizes the same commands as the primary shell.

A>COMMAND DIR

Accepts a command to run in the copy of the primary shell.

A>COMMAND -C DIR

Invokes a copy of the primary shell, runs the DIR command in the new shell, then automatically terminates the new shell.

You can precede -C with -P, but this has the same effect as using -C by itself. However, you cannot have -P following -C, because the -C option causes COMMAND to treat everything after the -C as input to another shell, not as input to COMMAND.

A>**COMMAND SHELL2.EXE**

%EXIT

Invokes the shell identified by SHELL2.EXE. The file containing your shell must be an executable file, which includes files with the extension 286, 68K, CMD, COM, or EXE. In this example, the secondary shell contained in SHELL2.EXE uses a percent sign as the system prompt. Because the secondary shell is not permanent in memory, EXIT returns you to the primary shell.

You can also use one or both of the COMMAND options following the filespec designating the secondary shell.

COMP

Forms COMP
 COMP filespec_1
 COMP filespec_1 filespec_2

Explanation

COMP compares the contents of two files of equal size. For example, you can use COMP after copying a file to make sure the copy is identical to the original.

The COMP command lets you compare files as follows:

- on the same or different drives
- in the same or different directories
- with the same path name or filename (if the files are on different drives)

You can use wildcards in the filespecs to compare two groups of files in which each pair of files is of equal size. COMP compares all the files matching filespec_1 with the corresponding files in filespec_2.

COMP displays the path names and filenames as each file is compared. An error message is displayed if:

- COMP cannot find a file matching filespec_1.
- filespec_1 and filespec_2 are different sizes.
- A specified directory path is invalid.

If COMP finds mismatching information in the files, it displays the offset of the mismatching bytes in hexadecimal and decimal, as follows:

```
Compare error at offset 17H (decimal 23)
File 1 = 49H (decimal 73)
```


File 2 = 4FH (decimal 79)

After 10 mismatches, COMP concludes that further comparison is unnecessary and displays:

10 Mismatches - ending compare

If the two files are identical, COMP displays:

Files compare ok

COMP continues comparing pairs of files that match the two filespecs until no more files can be found that match filespec_1. COMP then displays:

Compare more files (Y/N)? _

To compare two more files, type Y. You do not have to press Enter. You are immediately prompted for a new filespec_1 and filespec_2.

If you are done comparing files, type N to return to the system prompt. You do not have to press Enter.

Note: To compare diskettes instead of files, use the DISKCOMP command.

Examples

A>COMP

Enter first filespec :
GUESTS.LST

Enter second filespec or drive name :
B:

```
A:GUESTS .LST and B:GUESTS .LST
Files compare ok
```

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

Prompts you for filespec_1 and for filespec_2 or a drive name. This example compares GUESTS.LST on drive A with GUESTS.LST on drive B.

```
A>COMP INIT.BAT
```

Specifies filespec_1 only. COMP prompts you for filespec_2 or a drive name and then compares the files and displays the appropriate messages.

```
A>COMP ANSI.SYS B:
```

Specifies both filespecs. If you only specify a drive name for filespec_2, COMP assumes that filespec_2 has the same filename and file extension as filespec_1 and looks for it in the default directory. This example compares ANSI.SYS on drive A with ANSI.SYS on drive B.

```
A>COMP A:ORIGINAL.TXT B:ARCHIVE.TXT
```

Compares ORIGINAL.TXT on drive A with ARCHIVE.TXT on drive B.

```
A>COMP A:/DIR1 A:/DIR2
```

Compares all the files in one directory with all the files in another directory. The files are listed as they are compared:

```
A:/DIR1/FILE1.DAT and A:/DIR2/FILE1.DAT
Files compare ok
```

```
A:/DIR1/FILE2.DAT and A:/DIR2/FILE2.DAT
Files compare ok
```

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

CONFIG

Form CONFIG [devname [baudrate [wordlength
 [parity [stopbits]]]]]

Explanation

The CONFIG command lets you specify the parameters of a serial port on your command line. FlexOS can support as many serial ports as its loaded drivers can handle.

If your system does not have a serial port, you receive an error message when you try to use CONFIG.

Table 7-11 defines the CONFIG parameters and their acceptable values.

Table 7-11. CONFIG Options

Parameter	Definition
devname	Any known serial device name or serial port driver.
baudrate	The transmission rate in bits per second. Acceptable values are 50, 75, 100, 134.5, 150, 300, 600, 1200, 1800, 2000, 2400, 3600, 4800, 7200, 9600 and 19200.
wordlength	The number of data bits for a transmitted character. Acceptable values are 5, 6, 7, or 8.
parity	The type of parity, if any, to be used. Acceptable values are O(DD), E(VEN), or N(ONE).
stopbits	The number of stopbits added to the end of a transmitted character. Acceptable values are 0, 1, 1.5 or 2.

Note: When you specify a baudrate value, you need only specify the first two digits. You can specify 11 for a baudrate of 110, 15 for a baudrate of 150, and so on. When specifying a parity value, you need only specify the first character of the value: O for ODD, E for EVEN, and N for NONE.

Examples

A>CONFIG P0:

Displays the values assigned to the specified device as follows:

```
Device:
Baudrate: 9600
Wordlength: 7
Parity: EVEN
# of Stopbits: 2
```

Note: In this example you specify only the device name with CONFIG.

A>CONFIG P0: 9600 8

Changes the wordlength to 8 data bits. Notice that you must specify all parameters preceding the one you want to change. You do not have to change these values, but you must include them on the command line.

A>CONFIG P0: 9600 8 EVEN 1.5

Changes the number of stopbits to 1.5.

COPY

Forms Copy a single file:

COPY source
COPY source destination

Combine files:

COPY source_1+source_2+...source_last destination

Add one or more files to the end of another file:

COPY source_1+source_2+...source_last

Explanation

The COPY command lets you:

- Copy one or more files to another disk.
- Copy files to the same disk, provided you rename the copy or specify a different directory.
- Combine files while copying.
- Add one or more files to the end of another file.
- Transfer data between any of the system devices.

"Source" is the filespec you want to copy; "destination" is the filespec of your copy. If you do not specify a drive with the source or destination, the current drive is assumed. If the destination is a directory path without a filename, files are copied into that directory without their names being changed. To copy a file to the same directory, you must change its name.

When you copy a file, the current date and time are recorded in the destination file's directory.

You can use drive names, device names, and directory paths with the source and destination files. You can use wildcards

with the source files, but not in the destination. Notice that you can only use the asterisk (*) and the question mark (?) wildcards with COPY. If you use wildcards, the names of the files are listed as they are copied. The rules for wildcards are described in Section 1.

Files are usually copied in text (ASCII) mode. To copy or combine binary files, use the -B option. You can combine ASCII and binary files by using the -A and -B options on the same command line.

When you copy a file marked Read-Only, the copy does not have the Read-Only attribute.

Table 7-12 summarizes the options you can use with COPY.

Table 7-12. COPY Options

Option	Description
-A	Copies an ASCII (text) file. -A applies to the file preceding it and all remaining files until you specify another -A or -B. When files are being combined, -A is the default for the source and destination files.
-B	Copies a binary file. -B applies to the file preceding it and to all remaining files until you specify another -A or -B. When files are being copied, -B is the default for the source and destination.
-S	Copies hidden and system files in addition to the specified file or files. You only need to specify -S once and it can appear anywhere on the command line.
-V	Verifies that sectors are properly written on the target disk. Although -V slows down performance by forcing FlexOS to verify your file twice, it is good practice to verify an important copy with -V. You need only specify -V once; it can appear anywhere on the command line.

Examples

A>COPY B:MYFILE

Copies MYFILE.TXT from drive B to the current drive (drive A). The file is not renamed.

If you use this form of COPY on a single-drive system, COPY prompts you first to insert the diskette containing your source file, and then the diskette that is to contain the copy. Depending on the size of your computer's memory, you might be prompted to insert your diskettes more than once.

A>COPY B:SCORES.DAT -A

Copies SCORES.DAT as an ASCII file from drive B to the current drive, drive A. You might do this if you had previously combined several binary files into SCORES.DAT using the -B option. This would leave the end-of-file character in each source file intact in SCORES.DAT. When you copy with the -A option, COPY reads only to the first end-of-file character it encounters, and thus only copies the first source file contained in SCORES.DAT to drive A.

A>COPY A:CURRENT.TXT B:ARCHIVE.TXT

Copies CURRENT.TXT to drive B and renames it ARCHIVE.TXT. Because the file was renamed, the destination could have been the same drive as the source.

A>COPY *.* B: -V

Copies all the files on drive A to drive B. COPY lists the files as it copies them. The files are not renamed. The -V option verifies the copies.

Copying files in this manner is useful if the source files are fragmented, which means that parts of the file are written on non-contiguous (physically separate) areas of the disk. When a file is copied, these fragments are consolidated and stored as one block in the new file (assuming there is contiguous space available on the destination disk). To determine if your files are fragmented, use the CHKDSK command described in this section.

A>COPY B:/EXPENSES.DAT B:/BUDGET

Copies EXPENSES.DAT from the root directory of drive B to the subdirectory root/BUDGET, also on drive B. The copy is not renamed.

CAUTION: This command assumes the directory BUDGET exists on drive B. If the directory does not exist, EXPENSES.DAT is copied into a file named BUDGET in the root directory of drive B.

Combining Files

In combining files, COPY adds a copy of one or more source files to the end of source_1, and then copies the combined files to a destination file. If the destination filespec already exists, the new source_1 overwrites it. If the destination filespec does not exist, COPY creates it. The contents of the source files themselves are unaffected by this operation.

Example

```
A>COPY ABC.DAT+XYZ.DAT B:OLD.DAT
```

Adds a copy of XYZ.DAT to the end of ABC.DAT on the current drive. The result is copied to drive B. The destination file is named OLD.DAT.

COPY lists the source files as it combines them and then displays the message "1 File combined." to indicate that one destination file has been created.

You can also combine files from different drives, and use the -A and -B options with different files on the same command line.

Appending Files

In appending files, COPY adds copies of one or more source files to the end of source_1. This allows you to create one large file from several smaller, related files.

COPY adds the source files to the end of source_1 in the order you specify them. However, if COPY cannot find

source_1, it adds the remaining source files to source_2. If it cannot find source_2, it adds the remaining source files to source_3, and so on. If COPY cannot find any source files, it displays an error message.

This process does change the file to which the other source files are appended.

Examples

```
A>COPY A.DAT+B:B.DAT
```

Adds a copy of B.DAT from drive B to the end of A.DAT. The message "1 file combined." means COPY found at least one source file to add to A.DAT.

```
A>COPY BASIC.DAT+*.DAT
```

The next example adds all the DAT files, except BASIC.DAT itself, to the end of BASIC.DAT. COPY displays the message "1 file combined."

```
A>COPY *.PRT+*.FUL B:
```

Creates multiple destination files. In this example, the PRT files (DAY.PRT, SWING.PRT, and GRAVEYRD.PRT) are lists of part-time employees for different work shifts, and the FUL files are lists of full-time employees. The wildcards make COPY add each FUL file to the end of the corresponding PRT file, producing PRT files on drive B that are lists of all employees on each shift. The files can then be renamed to something like DAY.EMP, etc.

As it combines, COPY displays the following message:

```
Combining -  
  
DAY.PRT  
SWING.PRT  
GRAVEYRD.PRT  
  
3 files combined.
```

Using Device Names with COPY

You can also use device names with COPY. FlexOS device names are included in the list of logical names at the end of Section 1.

When you copy a file to or from a device, the file is copied in ASCII (-A) mode.

Example

```
A>COPY CON SCREEN.TXT  
I'm using the COPY command  
to copy text from my keyboard  
to a file. [Ctrl-Z]
```

```
1 file copied.
```

Copies what you enter at the keyboard into a file. The name of the source file is CON, which is another name for your console (keyboard and screen).

After each line of text you type, press Enter. When you have typed your last line of text, press Ctrl-Z to insert an end-of-file character in the file, and then press Enter. COPY puts the text in the destination file and returns you to the system prompt. In this example, the destination file is SCREEN.TXT.

CTTY

Forms CTTY
 CTTY device_name

Explanation

On most computers, the console (the keyboard and the screen) is the standard input and output device. Unless you specify otherwise, input is entered at the keyboard and output goes to the screen. The CTTY command lets you define any character-oriented device that can handle both input and output as the standard input and output device. Examples of these devices include terminals and teletypes.

CTTY also lets you reset the standard input and output device to its default value.

If you want to change the standard input device to one device and the standard output device to another, you must use the DEFINE command.

Examples

A>CTTY

Resets the standard input and output device to its default value.

A>CTTY AUX

Defines AUX as the standard input and output device.

A>CTTY CON

Defines the console as the standard input and output device.

DATE

Forms Set the date:

DATE

DATE mm-dd-yy

Display the date:

DATE -D

Explanation

The DATE command displays, sets, or changes the date known to the system. When you create or modify a file, FlexOS records the current date in the file's directory entry.

Note: If you are on a multi-user system, only the system manager can set the date. If you are on a single-user system, you can use any form of the DATE command.

All dates must be in the form mm-dd-yy or mm/dd/yy, where:

- mm (month) is a number 1-12.
- dd (day) is a number 1-31.
- yy (year) is a number 80-99 (the 19 is assumed) or 1980-2099.

DATE does not accept invalid dates such as 2-30-86 (February 30, 1986) or 9/31/86 (September 31, 1986).

If you are a privileged user and you enter a valid date with the DATE command, FlexOS accepts it and displays the system prompt. If you enter an invalid date or separator, DATE displays an error message.

Examples

A>DATE

Current date is Tue 10-14-1986
Enter new date : 10-15-1986

Displays the current date known to the system and prompts you for a new one. You can enter any valid date as the new date. This example changes the date to October 15, 1986. To leave the date unchanged, press Enter.

A>DATE 5-14-1986

Changes the date from the command line.

A>DATE -D

Current date is Wed 5-14-1986

Displays the current date but does not prompt you for a new one. This form is useful if you want to use the DATE command in a batch file; the date is displayed and then the next batch command (if there is one) is immediately executed.

DEFINE

Forms Create a logical name:

```
DEFINE logical_name=value  
DEFINE -S logical_name=value
```

List the logical names that have been created:

```
DEFINE -N  
DEFINE -S -N  
DEFINE -N logical_name  
DEFINE -S -N logical_name
```

Delete a logical name:

```
DEFINE -D logical_name  
DEFINE -D -S logical_name
```

Explanation

The DEFINE command lets you substitute a logical name for a more complex device name, directory name, filename, or other logical information.

In addition, DEFINE lets you:

- List your logical names and the names they replace.
- List any drive assignments you have changed with the ASSIGN command.
- Delete any of the logical names you created.
- Define the error message help level. (The help level is described in detail below.)

The help level you specify determines the amount of information FlexOS includes with its error messages. The higher the help level, the more detailed the error message.

The help level can have a value between 1 and 4. Your computer manufacturer sets a default help level which you can override with DEFINE.

Table 7-13 summarizes the type of information you get with each help level.

Table 7-13. Error Message Help Levels

Help	Description Level
1	Displays the FlexOS function, the error source module, and the return code.
2	Identifies the command and type of error in one sentence. An example of a level 2 error message is "COPY: Write error."
3	Expands on the level 2 message and includes more specific information. An example of a level 3 message is "COPY: An error occurred writing report.txt on a:".
4	Expands on the level 3 message and often suggests a possible solution to the error. An example of a level 4 message is "COPY: An error occurred writing report.txt on a: The disk a: is full. You can erase unnecessary files to free up space."

You can create a logical name at the process or system level. A process level logical name can only be used by the particular program it was created for; it does not have any meaning to programs you may be running on other screens, commands you have already started, or programs being run by other users.

A system level logical name can be used by all the programs

and users on the system. You can create a system level logical name only if you are on a single-user system, or if you are the system manager on a multi-user system.

You can only specify one logical name on a command line.

For more information on logical names, see Section 1.

Examples

A>DEFINE WP=A:/WORK/WORDPROC.EXE

Defines a process level logical name. In this example, the command line defines the filespec A:/WORK/WORDPROC.EXE as WP. This logical name can only be used by the program running in the current window. WP does not have any meaning to programs running in other windows.

A>DEFINE HELPLVL=2

Sets the error message help level to 2. If an error occurs, you receive a brief error message that does not include a possible solution.

A>DEFINE -S DATA=C:/CLIENTS/DATABASE.EXE

Uses the -S option to define DATA as a system level logical name that can be used by all users and programs on the system to access the file DATABASE.EXE.

A>DEFINE -N

Displays all process level logical names, their equivalent names, and any drive assignment names created with the ASSIGN command. The display takes the following form:

Process level logical names:

WP = A:/WORK/WORDPROC.EXE
INT = A:/PROGRAMS/BASIC/INTEREST.BAS

```
AVE = A:/PROGRAMS/BASIC/AVERAGE.BAS
```

```
A>DEFINE -S -N
```

Displays all system level logical names and their equivalent names, as follows:

System level logical names:

```
DATA = C:/CLIENTS/DATABASE.EXE
```

```
A: = FLPY0:
```

```
B: = FLPY1:
```

```
C: = HD1:
```

```
A>DEFINE -N WP
```

```
WP = A:/WORK/WORDPROC.EXE
```

Displays the equivalent name of the specified process level logical name, in this case WP. This form also displays any drive assignments you have created with the ASSIGN command.

```
A>DEFINE -S -N DATA
```

```
DATA = C:/CLIENTS/DATABASE.DAT
```

Displays the specified system level logical name and its equivalent name.

```
A>DEFINE -D WP
```

Deletes the process level logical name WP.

```
A>DEFINE -D -S DATA
```

Deletes the specified system level logical name.

DIR

Forms List all the files in a directory:

DIR

DIR -option

List specific files in a directory:

DIR filespec

DIR filespec -option

Explanation

The DIR command lists files in the current or specified directory. Depending on the form and options you use, you can list some or all of the files in a directory, and any hidden or system files.

Regardless of the form you use, DIR displays the name of each file, volume identification, and the amount of free space left on the disk. In each form of the DIR command entries that refer to subdirectories are identified with <DIR> in the file size column.

If you do not specify the -W (wide display) option, the listing includes the size of each file in decimal bytes and the date and time you last modified it.

If you do not specify the -H (hidden files) option, DIR does not list hidden and system files, even if they are present.

You can use wildcards with DIR.

Unless you specify a path with DIR, the command lists the files in the current directory.

If either filename or file extension is omitted, an * is assumed.

You can also use the DIR options in any combination. Table 7-14 summarizes these options.

Table 7-14. DIR Options

Option	Description
-H	Displays all hidden and system files in addition to the usual listing.
-P	Pauses after each screenful of directory listings. Press any key to display the next screenful of data.
-W	Produces a wide display of the directory listing.

Examples

A>DIR

Lists the current directory of the current drive. This form is equivalent to entering a DIR *.* command. The display takes the following form:

```

Volume in drive A: is DISKETTE1
Directory of A:

FILE1      A          10368   9-04-1986   1:19p
FILE3      A           1613   6-27-1986  12:14p
BUDGET    DAT              31   2-15-1986   8:52a
LEVEL2          <DIR>       5-09-1986  12:10p
FILE1      2288       7-02-1986   3:25p

          5 Files      141312 bytes free

```

A>DIR -W

The -W option produces a wide display of the current directory. Each line can contain five directory names or filenames. Use -W only if you have an 80-column display.

Volume in drive A: is DISKETTE1
Directory of A:

FILE1	TXT	FILE3	DOC	BUDGET	DAT	LEVEL2
EXAMPLE2	FIL	EXAMPLE3	FIL	GOFORIT	EXE	LEVEL3

8 Files 141312 bytes free

A>DIR B:

Lists the files in the current directory of drive B.

A>DIR BUDGET.DAT

Lists the file, if any, in the current directory with the filename and file extension BUDGET.DAT.

A>DIR B:* .DAT

Uses a wildcard to list all the DAT files in the current directory of drive B.

A>DIR FILE1

Lists all entries in the current directory of the current drive with the filename FILE1, regardless of their file extension. Notice that the filename is entered without a file extension. This form is the same as **DIR FILE1.*** and lists filenames like FILE1.TXT, FILE1.BAS, and FILE1.

A>DIR FILE1.

Lists the directory entry FILE1 with no file extension, if present. Notice that in this form the filename is followed by a period and the file extension does not default to *.

A>DIR LEVEL2

Lists the files in the subdirectory LEVEL2 on drive A, as follows:

```
Volume in drive A: is DISKETTE1
Directory of A:/LEVEL2/

.                <DIR>          5-09-1986  11:45a
..               <DIR>          5-05-1986   9:00a
MYPROG  COM      2463      7-30-1986   8:55a
HOMEWORK TXT          1      9-18-1986   2:20p

                3 Files      141312 bytes free
```

Note: All subdirectory listing include a single period and a double period. The single period represents the subdirectory being listed (LEVEL2). The double period represents the directory one level above LEVEL2, called the "parent" directory. In this example, the parent directory is also the root directory.

A>DIR ..

Lists the files in the parent directory of your current directory.

DISKCOMP

Forms Compare two diskettes on a single-drive system:

DISKCOMP

Compare the diskette in the specified drive with the one in the current drive:

DISKCOMP drive_1

Compare the diskettes in two specified drives:

DISKCOMP drive_1 drive_2

Explanation

DISKCOMP compares the contents of two entire diskettes.

This command is used only for comparing diskettes, not individual files. Use the COMP command to compare individual files. If you specify a hard disk with DISKCOMP, you receive an error message.

DISKCOMP determines the number of sides and sectors per track to be compared based on the diskette in drive_1.

If you use DISKCOMP on a single-drive system, you are prompted to insert the diskettes at the appropriate time. DISKCOMP waits for you to press any key before it continues.

If an error occurs while DISKCOMP is comparing the diskettes, it issues a message indicating the track and side on which the error occurred. Then DISKCOMP continues to compare the rest of the diskette.

When DISKCOMP is finished and if it has found no errors, it prompts:

Diskettes compare ok

Compare more diskettes (Y/N)?_

If you type Y, DISKCOMP prompts you to insert another pair of diskettes. It compares these diskettes on the drives you originally specified.

If you do not want to compare any more diskettes, type N.

Examples

A>DISKCOMP

Performs a single-drive comparison on the current drive. DISKCOMP displays the following prompt:

```
Insert first diskette in drive A:  
Press any key when you are ready
```

It reads the first disk and then prompts you for the second disk:

```
Insert second diskette in drive A:  
Press any key when you are ready
```

Depending on the amount of available memory, DISKCOMP might prompt you to insert your diskettes more than once.

A>DISKCOMP B:

Performs a two-drive comparison, with drive B as the first drive and the current drive (drive A) as the second drive. DISKCOMP displays the following prompt:

```
Insert first diskette in drive B:
```

```
Insert second diskette in drive A:  
Press any key when you are ready
```

Note: Specifying the current drive in the command line causes DISKCOMP to perform a single-drive comparison.

A>DISKCOMP A: B:

Compares the diskette in drive A with the diskette in drive B. DISKCOMP displays the following prompt:

Insert first diskette in drive A:

Insert second diskette in drive B:
Press any key when you are ready

Note: If you specify the same drive twice, DISKCOMP performs a single-drive comparison.

DISKCOPY

Forms DISKCOPY
 DISKCOPY source
 DISKCOPY source destination

Explanation

DISKCOPY copies the entire contents of one diskette to another. In the forms above, the "source" drive contains the diskette you want to copy; the "destination" drive contains the diskette you are copying to.

Caution: DISKCOPY writes over any information already on the destination diskette.

DISKCOPY only copies diskettes. If you specify a hard disk, DISKCOPY displays an error message. If you wish to copy individual files, use the COPY command.

Note: If the destination diskette is not formatted with the same number of sides and sectors per track as the source diskette, DISKCOPY formats it while copying.

DISKCOPY prompts you to insert the diskettes at the appropriate times and waits for you to press a key before continuing.

After copying, DISKCOPY prompts:

Copy another (Y/N)?_

If you press Y, DISKCOPY prompts you to insert another pair of diskettes. The copy is performed on the drives you originally specified.

Press N to end the command.

Examples**A>DISKCOPY**

Performs a single-drive copy on the current drive. FlexOS first displays this prompt:

Insert source diskette in drive A:

Strike any key when ready

After reading from the source diskette, FlexOS displays this prompt:

Insert target diskette in drive A:

Strike any key when ready

Depending on the size of your computer's memory, FlexOS may prompt you to insert your diskettes more than once.

A>DISKCOPY B:

Copies the diskette in drive B to the current drive, drive A. When you specify only the source drive, DISKCOPY uses the current drive as the destination drive. If you specify the current drive as the source drive, a single-drive copy takes place. DISKCOPY displays the following prompt:

Insert source diskette in drive B:

Insert target diskette in drive A:

A>DISKCOPY A: B:

Copies the diskette in drive A to the diskette in drive B. If you specify the same drive for the source and destination, DISKCOPY performs a single-drive copy. DISKCOPY displays the following prompt:

Insert source diskette in drive A:

Insert target diskette in drive B:

DISKSET

Forms Protect and/or label the current disk:

```
DISKSET
DISKSET -P=ON|OFF
DISKSET -L=volume_label
DISKSET -M=ON|OFF
DISKSET -P=ON|OFF -L=volume_label -M=ON|OFF
```

Protect and/or label a specified disk:

```
DISKSET drive:
DISKSET drive: -P=ON|OFF
DISKSET drive: -L=volume_label
DISKSET drive: -M=ON|OFF
DISKSET drive: -P=ON|OFF -L=volume_label -M=ON|OFF
```

Explanation

The DISKSET command lets you enable or disable the protection on a disk, and it also lets you label a disk. Disk protection cannot be enabled on a disk that does not have a label.

When protection is enabled, the options you specified with the FSET command determine who may access a particular file on your disk and how they may work with it. In addition, your User ID and Group ID are written on the disk label. Your User ID and Group ID determine which users have owner, group, and world privileges when working with your files.

Caution: When protection is disabled, all users can read, write, delete, or execute the files on your disk, regardless of the restrictions you specified with FSET.

You can use DISKSET on either a single- or multi-user system. On a multi-user system, the system manager or

anyone whose User and Group IDs match your own can modify the protection of your disk.

The `-P` value is recorded on the disk and does not need to be set again unless you want to change it. To change the protection, you must be the only person using that disk when you invoke DISKSET. If anyone else is working with files on the disk, DISKSET refuses to operate.

The label is also recorded on the disk. The label can be changed only by the person who set it initially or the system manager.

Examples

A>DISKSET

Shows the volume label of the disk and whether protection is on or off.

A>DISKSET -P=ON

Enables protection on the current disk, which must have a volume label. When you enable protection, access to the files on your disk is based on the options you specified with the FSET command.

When you disable protection (`-P=OFF`), all users on your system can read, write, delete, or execute any file on your disk.

When you enable mixed mode media (`-M=ON`), both upper and lowercase characters can be used in filenames. When you disable mixed mode media (`-M=OFF`), all filenames are created in upper case only.

A>DISKSET -L=DISKETTEL

Writes a volume label of 11 characters or fewer on the disk. The volume label can contain only valid filename characters and cannot contain any blank spaces. When you create or change your volume label, your User ID and Group ID are written on the disk label. Only you or a privileged user can create or change your volume label.

A>DISKSET B: -P=OFF -L=EVERYBODY -M=ON

Disables the protection on the diskette in drive B, gives it the volume label EVERYBODY, and turns on mixed mode filenames.

ERASE

Forms ERASE filespec
 ERASE filespec -Q

Abbreviation DEL

Explanation

ERASE deletes the specified filespec. You can also use the alternate form DEL. ERASE and DEL function the same way.

If you enter ERASE by itself, you receive an error message. If you do not specify a drive with the ERASE command, ERASE deletes the file from the current drive. If you do not specify a path, the ERASE deletes the file from the current directory.

If you use the filespec *.* to erase all the files in a directory or on a disk, the ERASE command issues the following message to make sure you want to erase them:

Are you sure (Y/N)? _

If you want to erase all the files on the disk, type Y and press Enter; otherwise, type N and press Enter.

You cannot erase the system file and the two special directory entries (. and ..) in each subdirectory.

You can use the wildcard characters *, ?, and ^ in the filename and file extension. You should use wildcards with caution, however, because several files can be erased with a single command. For more information on wildcards, see Section 1.

Examples

A>ERASE MYFILE.1

Erases MYFILE.1 from the current directory of drive A.

A>ERASE B:*. *

Are you sure (Y/N)? **Y**

Erases all files in the current directory of drive B.

A>DEL B:/ACCOUNTS

Are you sure (Y/N)? **Y**

Erases all files in the subdirectory /ACCOUNTS on drive B.

A>ERASE *.DAT -Q

Queries you before erasing each DAT file in the current directory of drive A. Press Enter after you type each response. The -Q option helps you avoid erasing important files. The display looks like this:

FLPY0:LIST.DAT, delete (Y/N)? N

FLPY0:MARKET.DAT, delete (Y/N)? Y

FLPY0:REPORT.DAT, delete (Y/N)? Y

EXIT**Form** EXIT**Explanation**

The EXIT command returns you from a secondary shell to the primary shell. The secondary shell is invoked with COMMAND. Notice that if you used the -P option with COMMAND, EXIT will not return you to the primary shell; the secondary stays in effect until you use LOGOFF or turn off your computer.

Example**%>EXIT**

Returns you to the primary shell. The secondary shell was not permanent in memory.

FDISK

Form FDISK

Explanation

The FDISK command prepares a hard disk.

Preparing a hard disk with the FDISK command is an essential step in setting up the computer system. Because the system manager usually performs this task, using FDISK is described in Appendix A, "System Manager."

FIND

Forms Search one or more files:

```
FIND "string" filespec
FIND -option "string" filespec
FIND "string" filespec_1 filespec_2...filespec_last
FIND -option "string" filespec_1 filespec_2... filespec_last
```

Search command output:

```
COMMAND | FIND "string"
COMMAND | FIND -option "string"
```

Explanation

FIND searches files or command output for a specified character string, and sends lines that include the string to the standard output device or to a file.

The string you want to search for must be enclosed in quotation marks. If the string contains a quotation mark, use double quotation marks as string delimiters.

You can specify more than one filespec on the command line. Separate filespecs with blank spaces. FIND searches for the string in each file, in the order the files appear on the command line.

Table 7-15 summarizes the options you can use with FIND.

Table 7-15. FIND Options

Option	Description
-C	Displays the matched string once, followed by total number of matches in the specified file.
-N	Displays the line number of each match, followed by the line in which the string occurs.
-V	Displays each line that <u>does not</u> include the string.

Examples

A>FIND "Twinkle, Twinkle," STAR

Searches the file STAR for the string "Twinkle, Twinkle," and displays each line that contains the string. The resulting display looks like this:

```
----- STAR
Twinkle, Twinkle, little star
Twinkle, Twinkle, little star
```

A>FIND -C "Nevermore" /POETS/POE/RAVEN.TXT

Uses the -C option to count the number of times the string "Nevermore" appears in the file A:/POETS/POE/RAVEN.TXT, with this result:

```
----- POETS/POE/RAVEN.TXT: 8
```

A>FIND -N "Twinkle, Twinkle," STAR

Displays the line number of each match:

```
----- STAR
[1]Twinkle, Twinkle, Little star
```

[5]Twinkle, Twinkle, Little star

A>FIND -V "BRONTE, EMILY" AUTHORS.TXT

Lists the lines in AUTHORS.TXT that do not contain the string "BRONTE, EMILY".

A>DIR | FIND "ACCOUNTS PAY"

Searches the output of the DIR command for the string "ACCOUNTS PAY" and displays the following:

ACCOUNTS PAY 1429 1-18-1985 1:42p

FORMAT

Forms Format the specified drive:

 FORMAT drive:

 FORMAT drive: -option

 Format the current drive:

 FORMAT

 FORMAT -option

Explanation

FORMAT prepares a disk in a recording format acceptable to FlexOS. It analyzes the entire disk for any defective tracks, and prepares the disk to accept FlexOS files.

Caution: Formatting destroys any existing data on the disk. Because of this, you should always specify the disk you want to format. Forms 3 and 4 leave more room for error because FORMAT uses a default value for the current drive. If the default value is different from what you expect, you could mistakenly format the wrong disk.

You must format all new disks before they can be used by FlexOS. You may format your disk with the default values supplied by your manufacturer or, if you are formatting a diskette, you can also use the -P option to select a standard format from the Format Menu.

Table 7-16 summarizes the options you can use with FORMAT. Unless otherwise noted, you can use these options in any combination on the same command line.

Table 7-16. FORMAT Options

Option	Description
-B	Copies the boot loader, but not the system files to the newly formatted disk. If you do not use -B, the disk can only be used to store data files. To make your disk a boot disk, use -B.
-P	Displays the Format Menu and prompts you for a diskette format. -P can only be used to format diskettes. -P overrides -1 and -8 if they appear on the same command line.
-S	Copies the operating system file, the boot image, and all hidden, system and Read-Only files to the newly formatted disk. If you specify -S, it is unnecessary to also specify -B.
-V	Prompts you for a volume label and whether you want to enable protection on the disk. A volume label must be 11 characters or fewer. All characters acceptable in filenames are acceptable in volume labels. The disk must have a volume label for you to enable disk and file protection. Volumes labels are also used to identify your disks.
-1	Selects a single-sided format from the internal library of standard formats. -1 can only be used with diskettes.
-8	Selects an 8-sectored format from the internal library of standard formats. -8 can only be used with diskettes.

Formatting a Diskette

When you use `FORMAT` to prepare a diskette, `FORMAT` prompts you to insert a diskette in the current or specified drive, warns you that any data already on the diskette will be lost, and asks you if you want to continue the `FORMAT` operation.

If you decide not to continue, type `N` next to the prompt and press `Enter`. `FORMAT` returns you to the system prompt.

If you decide to continue, type `Y` and press `Enter`. `FORMAT` then displays the message "Formatting:" while it formats the disk. The colon blinks while the disk is being formatted.

When `FORMAT` has finished, it displays the following:

```
Format complete.  
  XXXXXX bytes of total disk space  
  XXXXXX bytes used by system  
  XXXXXX bytes available on disk
```

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

The total disk space and space available on disk vary according to the options you use to format your disk and the type of disk you have. Enter `Y` to format another diskette. Enter `N` to end the `FORMAT` program.

Formatting a Hard Disk

Formatting a hard disk is a three-step procedure:

1. If the hard disk has never been formatted, format the entire hard disk (this is called a "physical" format) with the driver installed as unpartitioned. Use Form 1 of the `FORMAT` command. If the hard disk has been formatted, skip to step 2.

2. Create a partition for FlexOS using the FDISK command. This procedure is described in Appendix A.
3. Format the partition (this is called a "logical" format) with the driver installed as partitioned. Use Form 2 of the FORMAT command with the -S and -V options.

When you format a hard disk, the responses you receive from FORMAT are slightly different from those you receive when you format a diskette:

- You are not prompted to insert a diskette. However, you are warned that any data already on the disk will be lost, and you are asked if you want to continue the FORMAT operation.
- When you physically format a hard disk, FORMAT prompts you to enter defective track information provided by the hard disk manufacturer.
- When FORMAT is done, you do not receive the prompt "Format another (Y/N)?".

Examples

A>FORMAT A:

Formats the disk in drive A with the default values supplied by your manufacturer. When you format a hard disk for the first time (physical format), use this form of the command.

A>FORMAT B: -P

Uses the -P option to display a list of formats, as follows:

Select a format [1 - 6], or else press 'ESC' to cancel

- 1) 8 sector, 40 track, double-sided (ff)
- 2) 8 sector, 40 track, single-sided (fe)
- 3) 9 sector, 40 track, double-sided (fd)
- 4) 9 sector, 40 track, single-sided (fc)

- 5) 15 sector, 80 track, double-sided (f9)
- 6) Manufacturer's diskette format

Select a format [1 - 6], else 'ESC' to exit:

The letters in parentheses after each format are media codes, which are used by FlexOS and certain programs to identify the disk format you are using.

A>FORMAT B: -S

Uses the -S option to format the disk in drive B and copies the operating system file, boot image, and all hidden, system, and Read-Only files to the newly formatted disk. When formatting is completed, FlexOS displays the following message:

```
System transferred
```

A>FORMAT -SC:FLEXOS.SYS

Uses the -S option with a filespec to format the diskette in drive A and copy the operating system file FLEXOS.SYS from drive C. The boot image and all hidden, system, and Read-Only files are also copied. Do not put a blank space between the -S and the filespec.

When formatting is complete, FlexOS displays the "System transferred" message.

A>FORMAT B: -V

Formats the diskette in drive B and then prompts you for a volume label:

```
Enter VOLUME label (11 chars. max. or press Enter for none)?
```

A volume label longer than 11 characters results in an error message, and you are prompted to enter another one. If you

enter a volume label, FlexOS asks if you want to enable file security:

Enable file security (Y/N)?

A>FORMAT B: -1

Formats your diskette for single-sided use. FORMAT does not change the default values for the number of sectors and tracks if it has a single-sided format with those values in its library.

A>FORMAT B: -1

Formats your diskette at 8 sectors per track. FORMAT does not change the default values for the number of sectors and sides if it has a 8-sector format with those values in its library.

FSET

Forms View the attributes of a file:

FSET filespec

Change the attributes of a file:

FSET filespec -H=ON|OFF

FSET filespec -A=ON|OFF

FSET filespec -S=ON|OFF

FSET filespec -R=ON|OFF

Set the protection on a file:

FSET filespec -O=[R][W][E][D]

FSET filespec -G=[R][W][E][D]

FSET filespec -W=[R][W][E][D]

Explanation

The FSET command lets you:

- View the attributes of a file.
- Change the attributes of a file. You can change certain characteristics of your file without changing the information in it.
- Set the protection on a file. You can decide who has access to your file and how people may work with it. This protection will not go into effect until you use the DISKSET command to turn on disk protection, or use the FORMAT command to put a volume label on your disk.

You can use wildcards with FSET to change the attributes of or set the protection on several files at once.

Changing the Attributes of a File

Your FlexOS files can have one or more of the following attributes:

- H Hidden file. If you specify -H=ON in your FSET command, that particular file becomes a hidden file. It is not shown in your directory unless you use the DIR command with the -H option.

If you specify -H=OFF, the file appears in your directory. If you do not specify a value for -H, -H=OFF is the default. However, if your file is also a system file (-S=ON), it is not shown in your directory even if -H is off.

- A Archive file. If this attribute is off, it indicates the file has been archived since it was created or last modified. -A is automatically turned on when you modify your file. Programs that back up your files can turn off archiving so they can back up files that have changed since the last backup.
- S System file. If you specify -S=ON, your file becomes a system file. A system file may be accessed by any user on the system without the user needing a copy of the file in his or her directory. Applications that are useful to many people on the system make good system files. A system file is not shown in directory listings unless you use the DIR command with the -H option.

If you specify `-S=OFF`, your file is not a system file and it appears in the directory listing. If you do not specify a value for `-S`, `-S=OFF` is the default. Note however, that if your file is also a hidden file (`-H=ON`), it is not shown in your directory listing even if `-S` is off.

- R Read-Only file. If you specify `-R=ON`, your file becomes a Read-Only file. You can only read the file, even if your User or Group ID give you other privileges. `-R=OFF` turns off the Read-Only attribute. If you do not specify a value for `-R`, `-R=OFF` is the default value.

Setting Protection on Files

FSET lets you decide who may have access to each of your files and how people may work with them. You can set different levels of protection for different files.

There are three classes of user on the FlexOS system. They are:

- O = owner
- G = group
- W = world

The class of user you are depends on your User and Group ID numbers. On a multi-user system, your system manager assigns you a User and Group ID when he sets up your account. If you are on a single-user system, you assign yourself a User and Group ID. For more information on User and Group IDs, see Appendix A.

You are the owner of every file you create. FlexOS keeps a record of this ownership information and uses it to determine who may have access to your file.

When you try to access a file, FlexOS compares your User and Group IDs with the file's ownership information. If your User and Group IDs match the owner information, you are given the owner privileges of the file. If only your Group ID matches, you are given group privileges. If neither your User or Group IDs match the owner information, you are given world privileges.

You can assign each class of user one or more of the following privileges:

R = Read the specified file.

W = Write to the specified file. To modify the contents of a file, you must have the write privilege.

E = Run the specified file (if the file is a program).

D = Delete part or all of the specified file. To change the security of a file, you must either have owner access or have the delete privilege.

Examples

```
A>FSET PERSONAL.TXT
```

Displays the file attributes of the file PERSONAL.TXT:

```
A:PERSONAL.TXT          A=ON,O=RWED
```

```
A>FSET BUDGET86.DAT -H=ON
```

Uses the -H option to set the hidden attribute ON for the file BUDGET86.DAT. Files with the hidden attribute on are not shown in a directory listing unless you use the -H option of the DIR command.

A>FSET BUDGET86.DAT -A=ON

Uses the **-A** option to set the archive attribute **ON** for the file **BUDGET86.DAT**.

A>FSET BUDGET86.DAT -S=ON

Uses the **-S** option to set the system attribute **ON** for the file **BUDGET86.DAT**. Files with the system attribute are not shown in a directory listing unless you use the **-H** option of the **DIR** command.

A>FSET BUDGET86.DAT -R=ON

Uses the **-R** option to set the Read-Only attribute **ON** for the file **BUDGET.DAT**. The Read-Only attribute overrides the access rights that are user/group based. A process cannot delete or write to a Read-Only file even if it has write and delete privileges for the file.

A>FSET DOCUMENT.PVT -O=R

Sets the security level for the owner category of your account. In this example, **-O** is set to read, which means that the owner can read from this file, but cannot write to it, execute it, or delete it. Set **-O** to any combination of the four settings, **R**, **W**, **E**, or **D** (Read, Write, Execute, or Delete).

A>FSET GAMEFILE -G=RWED

Sets the security level for the group category of your account. In this example, **-G** is set to read, write, execute, and delete. This means any other user in your group category has maximum security privileges to the file called **GAMEFILE**. Set **-G** to any combination of the four settings, **R**, **W**, **E**, or **D**.

A>FSET FORMULA.BAT -W=RW

Sets the security level for the world category of your account. In this example, -W is set to read and write. This means any user on the system can read from or write to this file, but cannot execute or delete it. Set -W to any combination of the four settings, R, W, E, or D.

LIST

Form LIST

Explanation

The LIST command displays a list of the FlexOS commands which are built-in to the system. Your computer manufacturer determines which commands are built-in.

LOGOFF

Forms LOGOFF
 LOGOFF node::

Explanation

LOGOFF ends your current session, either on a local computer or over a network on a remote computer. Use form 1 to end the session on your computer and all remote computers you have logged on. Use form 2 to end the session on the the remote computer designated by "node::" only. See the DR Net User's Guide for the description of the network and remote computer use.

After you see the logoff message, LOGOFF prompts you to begin another session.

Example

```
A>LOGOFF
Jones logged off 05-15-1986 08:30:00.00
Press any key to begin._
```

In this example, user Jones ended his session at 8:30 A.M. on May 15, 1986.

Note: If you did not begin your session with a LOGON command, LOGOFF uses "noname" as a default name. In the example above, "noname" would have been displayed instead of "Jones". Setting up the LOGON/LOGOFF procedure is described in Appendix A, "System Manager."

LOGON

Forms LOGON
 LOGON node::

Explanation

LOGON tells FlexOS you want to start a session. Use form 1 to start a session on your computer. Use form 2 to start a session over the network with the remote computer designated by "node::". See the DR Net User's Guide for the description of the network and remote computer use.

When you type LOGON, FlexOS prompts you for your username and password. When you type your password, the letters are not displayed on the screen. When you see the system prompt, you know you have logged on successfully and you can begin your work session. The PASSWORD command is discussed later in this section.

LOGON

Username:

Password:

The LOGON/LOGOFF procedure is an effective way to protect your data from unauthorized access; however, this procedure is optional. The system manager of your computer system decides whether or not users must use LOGON and LOGOFF. Setting up the LOGON/LOGOFF procedure is described in Appendix A, "System Manager."

MKDIR

Forms MKDIR directory_path
 MKDIR drive:directory_path

Abbreviation
 MD

Explanation

MKDIR creates a subdirectory on the current or specified disk.

The number of subdirectories you can create is limited only by the available disk space. However, the maximum length of a directory path from the root directory to the level you want to access must be 128 characters or fewer.

Examples

The first example of the MKDIR command creates a new subdirectory called FOREST that is attached to the root directory.

```
A>MD /FOREST
```

To create a subdirectory beneath FOREST, use one of the following examples:

If your current directory path is root/FOREST, use this form:

```
A>MD TREES
```

The absence of a leading slash mark (/) tells FlexOS to create a subdirectory TREES and add it to the current directory.

The next form also creates TREES and adds it to the path

root/FOREST, but you can use this command line regardless of your current directory path.

A>MD /FOREST/TREES

In this form of the command, the first / tells FlexOS to begin its directory search with the root directory.

The second form of the MKDIR command creates a directory on a drive other than the current drive. The following example creates a subdirectory called PHONE in the root directory on drive B:

A>MD B:/PHONE

MORE

Forms Display a file one screen at a time:

```
MORE < filespec
```

Display command output one screen at a time:

```
COMMAND | MORE
```

Explanation

MORE displays the contents of a file or the output from a command one screen at a time. You can also direct MORE's output to a file.

MORE displays a screenful of data and then displays the message:

```
-- More --
```

If you press any character key, another screenful of data is written to the screen or file. This process continues until all input data is read.

Examples

The first form of MORE uses a file for input. The following example displays a screenful of data from the file TEST.HIS:

```
A>MORE < TEST.HIS
```

```
HISTORY 170 TEST ANSWERS
```

```
-----
```

1. Louisiana Purchase
2. Jefferson

```
3.  Monroe
.
.
.
20. Lincoln
-- More --
```

The second form of MORE uses the output from a command as input. The following example displays a screenful of output from the DIR command:

```
A>DIR | MORE
```

```
Volume in drive A: is DISKETTEL
Directory of  A:

SALES      JAN      2015    1-15-1985    1:19p
EXPENSES JAN      1613    1-31-1985    12:14p
SALES      FEB      3256    2-15-1985    9:00a
EXPENSES FEB      1300    2-21-1985    2:45p
SALES      MAR      8788    3-15-1985    10:21a
EXPENSES MAR      1525    3-25-1985    12:14p
.
.
.
EXPENSES OCT      2691    10-29-1985    8:30a
-- More --
```

ORDER

Forms ORDER
 ORDER EXT
 ORDER EXT;EXT; ... EXT;

Explanation

The ORDER command defines the search order of specified file extensions. ORDER functions the same way as the PATH command except that it searches for file extensions rather than subdirectories.

ORDER is most often used to search for the file extensions of executable files when FlexOS is loaded. These files extensions include 286, 68K, CMD, COM, and EXE.

If you specify a file extension that does not exist, FlexOS displays an error message when it tries to load the file.

If you specify more than one file extension with ORDER, you must separate file extensions with a semicolon, blank, or tab.

Examples

The ORDER command without a file extension displays the search order currently defined to FlexOS. In the following example, ORDER reports that, if you do not specify a file extension, FlexOS looks for files with the extension EXE.

```
A>ORDER
ORDER = EXE
```

The ORDER command with one or more file extensions tells FlexOS to look for files with one of those extensions if an

extension is not supplied by the user. The following example tells FlexOS to look for files with the extension EXE or COM:

```
A>ORDER EXE;COM
```

PASSWORD

Forms **PASSWORD**
 PASSWORD username

Explanation

The **PASSWORD** command lets you or the system manager change your LOGON password. Passwords are primarily used on multi-user systems to prevent unauthorized access to files. However, if you are on a single-user system you can still use a password to prevent others from accessing your work. Setting up the LOGON/LOGOFF procedure and working with passwords are explained in Appendix A, "System Manager."

A password can be any string of ASCII characters. The first eight characters of the password are significant. When you attempt to log-on to your computer, FlexOS asks you to type your username and then your password. Although your username is displayed at the screen as you type it, your password, for security reasons, is not. If you correctly enter your username and password, FlexOS lets you access your files.

Examples

A>PASSWORD

Sets or changes your password. **PASSWORD** displays the following prompts:

Old password:
New password:
Confirmation:

Enter your old password, then your new password, and then enter your new password a second time to make sure you entered it correctly. None of your input is displayed on the screen.

If you decide not to change your password after you enter the **PASSWORD** command, press **Ctrl-C** to return to the system prompt.

A>PASSWORD CATHERINE

Specifies a username. To use this form of the command, you must be logged into a privileged account. The FlexOS system manager uses this form to set the initial passwords for each user. After the initial assignment, a user can change his or her password without consulting anyone.

This form displays the following prompts:

New password:
Confirmation:

By not prompting for the old password in this case, **PASSWORD** enables the system manager to assign a new password to a user who has forgotten the old one.

PATH

Forms PATH
 PATH directory_path
 PATH directory_path;directory_path...
 PATH drive:
 PATH drive:directory_path
 PATH drive:directory_path;drive:directory_path...

Explanation

PATH defines the search path of specified directories for commands or batch files that were not found by searching the current directory. PATH does not change your current directory.

If the command you entered is not found in any of the directories you specified, you receive an error message.

If you specify a path that no longer exists, FlexOS ignores that path and goes on to the next one, if one exists.

Note: When you specify one or more search paths, you must type the words DEFAULT: and SYSTEM: in front of the first path.

Subdirectory names in a directory path must be preceded, separated, and followed by slashes (/), in the following manner:

```
/subdirectory/subdirectory/.../subdirectory/
```

If you specify more than one path with the PATH command, you must separate paths with a blank, semicolon, or tab.

Examples

C>PATH

PATH = C:/

Displays the search path currently defined to FlexOS.

C>PATH DEFAULT: SYSTEM:/COMMANDS/

Specifies a search path on the current drive. FlexOS searches these directories in the order they appeared in the PATH command. Note that, for FlexOS to search the current ("default") directory, you must explicitly name it in the PATH command. In the example, FlexOS first looks in the default directory on drive C, then in the directory /COMMANDS.

A>PATH DEFAULT: SYSTEM:/ACCOUNTS/;/BASIC/

Specifies more than one path on the current drive. In the example, FlexOS looks for a command in the current directory and in the subdirectories /ACCOUNTS and /BASIC.

The remaining forms let you invoke commands from the root directory or one or more subdirectories of a drive other than your current drive.

PRINT**Forms** Print One or More Files:

```
PRINT filespec_1 filespec_2 ...  
PRINT filespec -d=device
```

Print a File on a Server Node:

```
PRINT filespec -n=node::  
PRINT filespec -n=node:: -d=device
```

Display Printer Status:

```
PRINT -s  
PRINT -s -n=node:: -d=device
```

Cancel Print Jobs:

```
PRINT -t  
PRINT -c=job_id
```

Put Priority Job at Head of Print Queue:

```
PRINT filespec -h
```

Load Print Queue:

```
PRINT -l
```

Explanation

You must have a print spooler to use this command. See "How to use the Print Spooler" in Appendix A. If your system does not include a spooler, use the COPY command to send files to your printer. See "Using Device Names with COPY" under the COPY command earlier in this section.

You can only print files on nodes if your system is part of a network. See the FlexNET User's Guide for further information on network operations.

PRINT sends your files to a print spooler to be printed in the background, freeing your system for other work. You can combine any and all options listed above in any sequence. You can also have more than one filespec (but less than 20) on the command line. Wildcards may be used. A space must separate each element of the command line. Every character string that does not begin with a hyphen is treated as a filespec. For instance, "PRINT s" will try to print a file named "s" but "PRINT -s" shows printer status.

Use Form 1 to send one or more files to the local default printer's print queue. (A print queue is a lineup of files waiting to be printed.) The system confirms job entries by returning the status of the jobs entered. (However, when you cancel print jobs or load print queues PRINT does not return confirmation.)

You can override the system printer definitions and specify a local printer (Form 2) or a printer on another network node (Form 4). A node is a system in a network of systems. If you do not specify a printer, the job will be sent to the local default printer or the default printer on the node you specified (Form 3).

If you do a lot of printing on a specific server node, use the DEFINE command to identify that node as the default so that you do not have to enter its name each time. Define the default printer (PRN) as follows: "DEFINE prn:=node::spldrv:". Replace "node" with your node name.

Form 5 is used to show the status of all print devices on the local node. You can also specify other nodes and devices

with Form 6. The system displays a status report when either of these options are used. Form 7 is used to terminate all your print jobs. (If the `-t` option is used in any entry, it overrides all other options.) You can cancel specific print jobs by using the `-c` option and specifying the job id as shown in Form 8. The job id is listed in the status report; it is also displayed by the system when the job is entered.

Use Form 9 to enter a priority job at the head of the print queue so that it is next in line to print. (A priority job does not interrupt a job in progress.) Form 10 must be used to reload the print queue if a system halt or power failure has occurred. No new print jobs can be loaded before this is done. Jobs running when the interruption occurred are resumed when the queue is reloaded. The printer buffer may have been cleared, however, and part of the currently printing file may have been lost.

Table 7-17 presents the options for PRINT.

Table 7-17. PRINT Options

Option	Description
<code>-c</code>	Cancels single print job
<code>-d</code>	Device identifier
<code>-h</code>	Place job at head of queue
<code>-l</code>	Initialize print queue
<code>-n</code>	Node identifier (Network operations only)
<code>-s</code>	Displays printer status
<code>-t</code>	Terminate all printer operations from this logon

Examples

A>PRINT FIRST.DAT SECOND.TXT

Sends the files FIRST.DAT and SECOND.TXT to the local default printer. FlexOS shows the following status report on your print jobs, including the user name and the name of the default device:

Status	Job_id	User	Filespec	Node	Device
PRINTING	1	tim	FIRST.DAT		prn:
QUEUED	2	tim	SECOND.TXT		prn:

A>PRINT THIRD.DOC -d=lpt1:

Status	Job_id	User	Filespec	Node	Device
PRINTING	6	bob	THIRD.DOC		lpt1:

Specifies a printer other than the default. In this example, the file THIRD.DOC is being sent to device LPT1.

A>PRINT THIRD.DOC -n=AJAX -d=prn:

Status	Job_id	User	Filespec	Node	Device
QUEUED	4	bob	THIRD.DOC	AJAX::	prn:

Sends the file THIRD.DOC to a device on a remote server node named Ajax.

A>**PRINT -s -n=ZEUS -d=lpt2:**

*DEVICE: lpt2: on NODE ZEUS:: STATUS: inuse JOBS: 3

Status	Job_id	User	Filespec
--------	--------	------	----------

PRINTING	7	jack	LETTER.TXT
QUEUED	8	jill	MEMO.DAT
QUEUED	9	john	REPORT.LST

Shows the status of all print jobs on device LPT2 on node Zeus. (To get the status of all print jobs on the default device on the local node, enter the **-s** option alone.)

A>**PRINT -c=job10**

Cancels job 10 on the default printer on the local node. The command **PRINT -t** stops all of your jobs (jobs entered from that login) on all printers.

A>**PRINT RUSH.LTR -h -n=JOVE -d=PRN:**

Status	Job_id	User	Filespec	Node	Device
--------	--------	------	----------	------	--------

QUEUED	9	tad	RUSH.LTR	JOVE::	prn:
--------	---	-----	----------	--------	------

Puts the priority print job RUSH.LTR at the top of the printer queue for device PRN: on node Jove. The status report shows job 9 at the head of the queue, ready to print when the current job is finished.

A>**PRINT -l -n=MARS -d=LPT1:**

Reloads the print queue for device LPT1 on node MARS. To reload all devices, enter the **-l** option alone.

PROCESS

Forms PROCESS
 PROCESS VIEW
 PROCESS VIEW -option
 PROCESS CANCEL ID=#####

Explanation

The PROCESS command lets you view or delete processes currently running on the system. A process is defined as any command, program, or application running under FlexOS.

The PROCESS VIEW options, described in Table 7-18, can be used in any combination.

Table 7-18. PROCESS VIEW Options

Option	Description
-E	Provides extra information.
-F	Provides a Family filter, which displays only those processes that match this user's Family ID. Otherwise, all processes on the (local) system are displayed.
-M	Provides memory statistics.
-P	Provides paged output, as opposed to streamed output.
-U	Provides a User filter, which displays only those processes that match this user's Group and User IDs. Otherwise, all processes on the (local) system are displayed.

Examples**A>PROCESS**

Describes the PROCESS syntax and options.

A>PROCESS VIEW

Displays information on each process running on the system. Information is displayed with the following headings:

name	Process name: a unique 8-character identifier.
pid	Process ID number: a unique decimal value up to 10 digits long.

wndw	Window number in which the process is running. A decimal value up to 3 digits long.
pcon	Physical console number on which the process is running.
fmly	Family ID number: a decimal value up to 5 digits long. Related processes (such as commands within a batch file, or commands chained together on the same command line) have the same family ID.
grp	Group ID of the person running the process: a decimal number 0-255.
user	User ID of the person running the process: a decimal number 0-255.
stat	Status of the process: running, waiting, terminating, or unknown. This heading also tells you if a process is using FlexOS's shared code facilities.

A typical PROCESS VIEW display looks like this:

PROCESS VIEW MODE

```

-----
name: PROCESS  pid : 4   wndw: 0   pcon: 0
fmly: 3        grp : 5   user: 2   stat: running  *
-----
name: COMMAND  pid : 2   wndw: 0   pcon: 0
fmly: 0        grp : 0   user: 0   stat: terminating
-----
name: CONFIG   pid : 1   wndw: 0   pcon: 0
fmly: 0        grp : 0   user: 0   stat: terminating
-----

```


The asterisk following "stat: running" indicates the PROCESS command itself is currently running.

A>PROCESS VIEW -E

Displays the following extra PROCESS VIEW information:

prty	Process priority. The priority for most processes is 200.
ppid	Parent process ID number: a decimal value up to 10 digits long. A parent process is a program or command that causes other processes to begin running automatically. For example, when you run the CONFIG program to start FlexOS, CONFIG automatically starts the COMMAND program. CONFIG is the parent process of COMMAND.
process type	If the process was started by a user, PROCESS displays the heading "user process"; if the process was started by the system, PROCESS displays the heading "system process."
context	If the process was started by a user, PROCESS displays the heading "main context"; if the process was started by the system, PROCESS displays the heading "system context."
evnt	Event flags: a 32-bit binary number. Event flags keep track of operations a process has not yet completed. If a process has completed all its operations, "evnt:" equals zero.

system manager PROCESS displays this heading only if the process was started by the system manager.

The PROCESS VIEW -E display looks like this:

PROCESS VIEW MODE

```

-----
name: PROCESS   pid : 4   wndw: 0       pcon: 0
fmly: 3         grp : 5   user: 2       stat: running   *
prty: 200      ppid: 2   (user process) (main context)
evnt: 0000 0000 0000 0000 0000 0000 0000 0000
-----
name: COMMAND   pid : 2   wndw: 0       pcon: 0
fmly: 0         grp : 0   user: 0       stat: terminating
prty: 200      ppid: 1   (user process) (main context)
evnt: 0000 0000 0000 0000 0000 0000 0000 0000
-----
name: CONFIG    pid : 1   wndw: 0       pcon: 0
fmly: 0         grp : 0   user: 0       stat: terminating
prty: 200      ppid: 0   (user process) (main context)
evnt: 0000 0000 0000 0000 0000 0000 0000 0000
-----

```

A>PROCESS VIEW -F

Displays only those processes whose family IDs match your own. The information displayed is the same as for PROCESS VIEW.

A>PROCESS VIEW -M

Displays memory statistics with the following headings:

maxm Maximum memory: a hexadecimal number up to eight digits long. This is the maximum amount of memory allocated to the process.

code	Code start: a hexadecimal number up to eight digits long. This is the location in memory where the code of the process begins. The size of the code is displayed in the "size:" heading below "code:."
data	Data start: a hexadecimal number up to eight digits long. This is the location in memory where the data associated with the process. The size of the data is displayed in the "size:" heading below "data:."
heap	Heap start: a hexadecimal number up to eight digits long. A heap is a storage area for data. "heap:" is the location in memory where this storage area begins. The size of the heap is displayed in the "size:" heading below "heap:."
memory status	If the memory used by a process can be transferred to and from secondary storage, PROCESS displays the heading "swap." If memory cannot be transferred, PROCESS displays the heading "locked."

The PROCESS VIEW -M display looks like this:

PROCESS VIEW MODE

```

-----
name: PROCESS   pid : 4           wndw: 0           pcon: 0
fmly: 3         grp  : 5           user: 2           stat: running    *
maxm: 0H       code: 41DC0000H  data: 44040000H  heap: 0H
(swap)         size: 2280H        size: FE0H       size: 0H
-----
name: COMMAND   pid : 2           wndw: 0           pcon: 0
fmly: 0         grp  : 0           user: 0           stat: terminating
maxm: 0H       code: 2CD90000H  data: 37850000H  heap: 0H
(swap)         size: AAC0H        size: 4140H       size: 0H
-----

```

name: CONFIG	pid : 1	wndw: 0	pcon: 0
fmly: 0	grp : 0	user: 0	stat: terminating
maxm: 0H	code: 0H	data: 0H	heap: 0H
(swap)	size: 0H	size: 0H	size: 0H

A>PROCESS VIEW -E-M

Displays the information generated by both options.

A>PROCESS VIEW -E-M-P

The -P option displays the information one screenful at a time. At the end of each screen, you are prompted to press any key to continue. If you do not specify -P, process information scrolls on your screen until it reaches the end of the information.

A>PROCESS VIEW -U

Displays only those processes whose User and Group IDs match your own.

A>PROCESS CANCEL ID=4

Cancels the program whose process ID is 4. You can only cancel processes you have created, and you must know its process ID number. After the process is canceled, PROCESS displays "Process deleted." and you are returned to the system prompt.

PROMPT

Forms PROMPT
 PROMPT text

Explanation

PROMPT sets a new system prompt or restores the default setting. The system prompt may not exceed 128 characters.

The first non-blank character after the word PROMPT is the first character of the prompt. FlexOS considers all the text on the PROMPT command line to be the new system prompt.

If you enter the PROMPT command by itself, FlexOS restores the default prompt. You can include special characters in the text in the form \$c, where c is one of the following:

t	time	b	character
d	date	q	= character
g	> character	\$	dollar sign
l	< character	e	ESCape character
p	current directory path	n	current drive (a-p)
h	backspace and erasure of previous character		
_	go to beginning of new line on screen		
v	FlexOS version number		

Any other character is treated as a null character. The PROMPT command takes no action on it.

Examples

CAM!PROMPT

A>_

Restores the current prompt CAM! for drive A: to its default form A>.

```
A>PROMPT XYZ
```

```
XYZ_
```

Changes your current prompt (default or otherwise) to the specified form, in this case "XYZ".

```
XYZPROMPT $n$g
```

```
A>_
```

Sets the system prompt to its default value by explicitly stating its form.

```
A>PROMPT Time=$t$_$p$g
```

```
Time=11:17:33.76
```

```
A:/COMMANDS/>_
```

Sets a two-line prompt that displays the current time and the current directory path of the default drive.

RECDIR

Form RECDIR -d:[pathname]

Explanation

RECDIR is used to analyze and repair disk directories with bad sectors. If you receive repeated read error messages when trying to access a directory, the problem is most likely a defective sector. You can usually recover the directory that contains the bad sector, minus the file entries in the bad sector.

RECDIR analyzes the media in the drive for suitability and then analyzes the disk. RECDIR copies any directory cluster with defective sectors to a new cluster and fills the bad sector with null directory entries. You can recover the affected files by using the -F option of the CHKDSK command and responding Yes when prompted. The lost data is then written to files in the format FILEnnnn.CHK, where nnnn is a sequential four-digit number.

Note: A cluster is the minimum allocation unit of a directory file. This varies with the media and is a multiple of the sector size.

RECDIR requires that you be able to read from and write to the root directory of the disk you are attempting to access. Any I/O directory access or memory allocation errors cause RECDIR to terminate prematurely and an error message appears.

Example

A>RECDIR -B:

Analyzes the media in drive B for suitability before attempting repair. RECDIR locates bad sectors on the entire disk, fills them with nulls, and copies each repaired directory to a new cluster. RECDIR must be entered with a disk drive designator followed by a colon.

A>RECDIR -B:/PROPERTY/TAXES

Restricts the analysis to the directories at the lowest directory level specified by the path.

RECFILE

Forms RECFILE filespec_1 filespec_2 ...filespec_last
 RECFILE -option(s) filespec_1 filespec_2 ...filespec_last

Explanation

RECFILE analyzes and repairs files with hard read errors; it recovers files from a disk which has defective sectors. It also reports the number of clusters and extents for each file. If you receive repeated read or write error messages when trying to access a file, the problem is most likely a defective sector. You can usually recover the file that contains the bad sector, minus the data in the bad sector.

Note: A "cluster" is the minimum allocation unit of a file. This varies with the media and is a multiple of the sector size. An "extent" is a set of contiguous sectors for a file. The more extents that exist for a file, the more fragmented the file is.

Table 7-19 describes the options you can use with RECFILE.

Table 7-19. RECFILE Options

Option	Description
-t	Disables the File Allocation Table (FAT) trace. Displays only the total number of clusters for a file. If not specified, a list of cluster numbers that each file occupies is generated for each file specified.
-i	Translates all filespecs to uppercase before comparing. Normally, RECFILE only converts filespecs to uppercase before comparing them if the media default does not support upper- and lowercase letters. The -i option ensures that translation is performed regardless of the media default.
-f	Causes RECFILE to attempt to repair the file by rereading the bad clusters sector by sector, filling in unreadable sectors with fill data (0x2e), and moving the cluster data to the next free cluster. Bad clusters are then marked as flawed in the File Allocation Table (FAT) so they will not be allocated to other files.

RECFILE displays the name of each file being processed. When it encounters a directory file, as can happen when you use wildcards, RECFILE displays a message that it cannot repair directory files, but will continue to process any other files specified. (Use RECDIR to repair directory files, including the root directory.)

If you do not specify a drive, the current or "default" drive is used. If you do not specify a path, the default path is used. You may use wildcards with RECFILE.

Text files that have been repaired usually require editing before they can be used for normal processing.

Examples

```
A>RECFILE EXPENSES.DAT TRAVEL.DAT INCOME.REC
```

Reads the specified files, cluster by cluster, and displays a list of cluster numbers that the files occupy along with the total number of clusters and number of extents. If any bad clusters are found, their numbers are displayed with an error message.

```
A>RECFILE -f -t -i EXPENSES.DAT INCOME.REC
```

Analyzes both files specified and then does the following:

- Displays the total number of clusters for each file.
- Translates all filespecs to uppercase.
- Attempts to repair the files.

```
A>B:RECFILE -f *.*
```

Attempts to repair all files in the current directory of drive A.

RENAME

Form **RENAME** old_filespec new_filespec

Abbreviation **REN**

Explanation

The **RENAME** command lets you change the name and/or extension of a file. **RENAME** does not change the information in the file.

You can specify a directory path with **RENAME**. If you specify a path with `new_filespec` that is different from the path with `old_filespec`, the file is renamed and moved to the second directory path. (The second directory path must already exist.) If you don't name a directory path or if both paths are the same, the file stays in the same directory after it is renamed.

You cannot rename a directory, and you cannot rename a file to a different drive.

You can use wildcards in both filespecs.

Examples

A>RENAME IN.TXT OUT.TXT

Renames `IN.TXT` to `OUT.TXT`. `OUT.TXT` remains in the current directory.

A>REN B:REVIEW.Q1 B:* .Q2

Renames the file `REVIEW.Q1` on drive `B` to `REVIEW.Q2`, also on drive `B`. The wildcard tells FlexOS not to change the filename.

A>REN B:/WORKFILE/LIST.TXT B:/LEISURE/LISTLESS.TXT

Renames the file LIST.TXT in subdirectory /WORKFILE on drive B to LISTLESS.TXT and moves it to subdirectory /LEISURE.

RESTORE

Forms RESTORE backup:
 RESTORE backup: source:
 RESTORE backup: source: filespec
 RESTORE backup: source: filespec -option

Explanation

The RESTORE command lets you restore to the source disk one or more files that have been backed up to diskettes. For instructions on how to back up files, see the BACKUP command in this section.

In the forms shown above, "backup:" identifies the drive containing the backup disks; "source:" identifies the drive to which you are restoring your files.

If you do not specify a path for the source drive, the files are restored to the current directory.

Table 7-20 summarizes the options you can use with RESTORE. You can use both options on the same command line.

Table 7-20. RESTORE Options

Option	Description
-P	Prompts you before restoring Read-Only files and files that have changed since the last backup. FlexOS system files are marked Read-Only when they are created by the FORMAT and SYS commands. If you want to restore a diskette which has these files on it, use -P.
-S	Restores backed up files to their original directories on the source disk. If a directory no longer exists on the source disk, restoring files to it recreates the directory.

RESTORE reads the backup disks in the order they were created. For that reason, when RESTORE prompts you to insert the backup disk, insert the first disk that might contain the file you want to restore. (If you are not sure, insert backup disk number 1.) If the file is not on the disk you inserted, RESTORE prompts you to insert the next backup disk.

If you specify a wildcard character in the filename, all files matching that filename are restored. RESTORE then prompts you to insert the next backup disk.

Examples

C>RESTORE A:

Restores all files on the diskette in drive A to the current directory of the disk in drive C (in this case, the root directory). Because the source disk is not specified, FlexOS assumes it to be the current drive.

D>RESTORE B: C:

Restores all files in the current directory of the backup disk in drive B to the current directory of the source disk (drive C).

C>RESTORE A: /CUSTOMER/LETTER.TXT

Restores the file LETTER.TXT to the subdirectory CUSTOMER on drive C. Note that it is not necessary to specify drive C because drive C is the current drive.

D>RESTORE A: C:* .MEM

Restores each file with an extension of MEM from the backup disk to the current directory on drive C.

RMDIR

Form RMDIR directory_path
 RMDIR drive: directory_path

Abbreviation RD

Explanation

RMDIR removes a subdirectory from the current or specified disk. You cannot remove the root or current directories. RMDIR removes the last directory name in the specified path.

You must delete all files from a directory before you remove it. If you try to remove a directory containing files, you receive an error message.

Example

A>RD /FOREST/TREES

Removes the directory TREES from the directory path root/FOREST/TREES.

C>RD A:/FOREST/TREES

From drive C, removes the directory TREES from the directory path root/FOREST/TREES on drive A.

SECURITY

Forms View your current file access privileges:

```
SECURITY
```

Change your current file access privileges:

```
SECURITY -O=[R][W][E][D]  
SECURITY -G=[R][W][E][D]  
SECURITY -W=[R][W][E][D]
```

Explanation

SECURITY lets you view or change current file access privileges. These privileges are Read (R), Write (W), Execute (E), and Delete (D). You can change the level for all three user classes: Owner (-O), Group (-G), and World (-W). See the FSET command for an explanation of file access privileges and user classes.

Privileges set with SECURITY are in effect for the duration of a terminal session. We suggest that you include a SECURITY command in your AUTOEXEC.BAT file to set file access privileges automatically each time you start your computer. You can also invoke SECURITY from the command line.

Example

```
A>SECURITY -G,-W [R]
```

Sets the file access privileges for Group and World users to Read access. Note that you can combine user classes by separating them with commas.

SORT

Forms Sort the input from a file:

```
SORT < filespec
SORT -option < filespec
```

Sort the output from a command:

```
COMMAND | SORT
COMMAND | SORT -option
```

Explanation

The SORT filter rearranges the input from a file or the output from a command. SORT does not change the contents of the file; SORT merely changes the way the information is displayed.

Data is sorted using the ASCII collating sequence. See the ASCII table in Appendix C to refer to the sequence.

Table 7-21 summarizes the options you can use with SORT.

Table 7-21. SORT Options

Option	Description
-R	Sorts lines in the reverse order of the ASCII collating sequence. Z comes before Y which comes before X, and so on.
++N	Sorts lines into the ASCII collating sequence based on the character in column N. N must be a whole number.

Examples

The following examples sort a file called GUEST.LST. Note that the heading "Guest List," the line under the heading, and the blank line before the names are also sorted.

```

      Guest List
      -----

Larry   Irwin
Peter   Glass
Becky   Savage
Meryle  Edwards
Terry   Gibson
Sheila  Escovedo
Tom      Andrews

```

1 sorts the input from a text file into the ASCII collating sequence and displays it on your screen. Lines are sorted beginning with the character in column 1.

A>SORT < GUEST.LST

Sorts the list into alphabetical order by the first name of each guest:

```

      -----
      Guest List

Becky   Savage
Larry   Irwin
Meryle  Edwards
Peter   Glass
Sheila  Escovedo
Terry   Gibson
Tom      Andrews

```

A>SORT --+10 < GUEST.LST

Uses the +N option to sort the list alphabetically by last name. (The guest list was prepared so that the last names began in column 10.)

```

-----
Tom      Andrews
Meryle   Edwards
Sheila   Escovedo
Terry    Gibson
Peter    Glass
Larry    Irwin
Becky    Savage
          Guest List

```

The following examples use the output of the DIR command to sort the root directory of drive A. The first example shows the unsorted output of the command.

```

FOG      TXT      64    2-01-1986   2:53a
SAND     TXT     3768   6-18-1986  12:17p
BEACH    LST      81    3-01-1986   3:08p

```

```
Directory of  A:
```

```
Volume in drive A: has no label
```

```
3 Files      241947 bytes free
```

Note that the lines "Directory of...", "Volume in drive...", and "3 Files..." are part of the ASCII collating sequence.

A>DIR | SORT

Sorts the directory alphabetically by filename:

```

          5 Files      241947 bytes free
Directory of  A:
Volume in drive A: has no label
%PIPE1   $$$          0    6-10-1986   4:30p
%PIPE2   $$$          0    6-10-1986   4:30p
BEACH    LST      81    3-01-1986   3:08p
FG       TXT      64    2-01-1986   2:53a
SAND     TXT     3768   6-18-1986  12:17p

```

FlexOS creates the \$\$\$ temporary files for piping data between commands. The next time you use DIR, they will be gone from the directory. When piping data between

commands, do not run programs that erase or modify these files. This can cause your data to be piped incorrectly.

Note: FlexOS does not create these temporary files if you pipe data between exclusively external commands. See "Built-In and External Commands" in Section 1.

A>DIR | SORT-R

Sorts the directory in reverse alphabetical order:

SAND	TXT	3768	6-18-1986	12:17p
FOG	TXT	64	2-01-1986	2:53a
BEACH	LST	81	3-01-1986	3:08p
%PIPE2	\$\$\$	0	6-10-1986	4:30p
%PIPE1	\$\$\$	0	6-10-1986	4:30p

Directory of A:

Volume in drive A: has no label

5 Files 241947 bytes free

A>DIR | SORT-+10

Sorts by the character at column 10 (the column the file extension starts in), and sends the output to the screen. The directory is now sorted alphabetically by file extension:

5 Files 241947 bytes free				
%PIPE1	\$\$\$	0	6-10-1986	4:30p
%PIPE2	\$\$\$	0	6-10-1986	4:30p
BEACH	LST	81	3-01-1986	3:08p
FOG	TXT	64	2-01-1986	2:53a
SAND	TXT	3768	6-18-1986	12:17p

Volume in drive A: has no label

Directory of A:

A>DIR | SORT -R-+10

Combines the -R and -+N options to do a reverse listing of the characters in the specified column number, in this case sorting the directory by file extension in reverse order:

```
Directory of A:
Volume in drive A: has no label
SAND    TXT      3768   6-18-1986  12:17p
FOG     TXT       64    2-01-1986   2:53a
BEACH   LST       81    3-01-1986   3:08p
%PIPE2  $$$       0     6-10-1986   4:30p
%PIPE1  $$$       0     6-10-1986   4:30p
      5 Files      241947 bytes free
```

A>SORT < PARTTIME.LST >> FULLTIME.LST > PRN:

This example combines redirection symbols and a filter to perform a more sophisticated task. PARTTIME.LST is a list of part-time employees' names; FULLTIME.LST is a list of full-time employees' names. The command sorts PARTTIME.LST alphabetically and adds it to the end of FULLTIME.LST. FULLTIME.LST now contains a list of all the employees' names. This file is redirected to the printer device PRN:.

SYS

Forms SYS destination_drive:
 SYS destination_drive: source_filespec

Explanation

The SYS command lets you copy FlexOS to an application program diskette designed to use FlexOS, but sold without it. In this case, the space required for FlexOS files has been purposely left free. SYS transfers the files to this free space. You can also use SYS to transfer FlexOS to a hard disk.

The directory of the disk in the destination drive must have enough contiguous space starting on the first data sector.

SYS copies the operating system loader image and the operating system image file and any file that has the system, hidden, and Read-Only attributes to the specified drive. The disk in the specified drive must be bootable, that is, it must be formatted with the FORMAT utility using the -B or -S options and be capable of storing the system image.

If you do not specify a filespec for the system image file, SYS looks for the default filespec, FLEXLDR.IMG. If it is not found, it then looks for a file called FLEXLDR.SYS. When it finds one of these files, it copies it to the destination drive in the first data sector under the name FLEXLDR.IMG.

The operating system loader image file must contain the proper header information, such as code start and length, and data start and length. This information is used to modify the boot record, but is not copied to the destination drive.

If SYS cannot find the file you specified, it looks for a file called FLEXOS.SYS. If SYS finds FLEXOS.SYS, it uses this file and the boot record to copy an operating system image to the destination drive. The data area is aligned on a sector boundary.

If SYS cannot find the filespec you specify or FLEXOS.SYS, you receive an error message.

Examples

A>SYS C:

Transfers the operating system image file and all files with the system, hidden, and Read-Only attributes from the FlexOS system disk in drive A to drive C.

A>SYS C: SYSTEM:/OSFILES/FLEXOS.SYS

Identifies the operating system file with a filespec. This example copies the operating system file from the filespec A:SYSTEM:/OSFILES/FLEXOS.SYS to drive C. Files with the system, hidden, and Read-Only attributes are also copied.

TIME

Forms Set the time:

TIME
TIME hh:mm:ss

Display the time:

TIME -D

Explanation

TIME displays, sets, or changes the time known to the system. When you create or modify a file, FlexOS records the current time in the file's directory entry.

Note: If you are on a multi-user system, only the system manager can set the time. If you are on a single-user system, you can set the time, as well as display it.

Time is expressed in terms of a 24-hour clock where:

- hh is a number 0–23 for the hour.
- mm is a number 0–59 for the minute.
- ss is a number 0–59 for the second.
- xx is a number 0–99 for the hundredths of a second.

Use colons (:) to separate the hours, minutes, and seconds. Use a period (.) to separate the seconds and the hundredths of a second. If you do not use the proper separators, TIME displays an error message.

As long as the digits are within the defined ranges, any time is acceptable to FlexOS. If you enter a valid time, FlexOS accepts it and displays the system prompt. If you enter an invalid time, TIME displays an error message.

Examples

A>TIME

```
Current time is 08:30:00.00.00
Enter new time : 9:53:26
```

Displays the time known to the system. To change the time, enter a new time at the prompt and press Enter. If you do not want to change the time, just press Enter. FlexOS records the time as 09:53:26.00. You do not have to include a leading zero for A.M. settings.

A>TIME 13:25

Specifies the time from the command line. If you omit any of the TIME components (hours, minutes, and so on), FlexOS fills them in with zeros.

A>TIME -D

```
Current time is 13:25:05.11
```

The -D option only displays the current time; it does not prompt for a new time.

TREE

Forms TREE
 TREE drive:
 TREE -F
 TREE drive: -F

Explanation

The TREE command displays all the directory paths on the current or specified drive. TREE also gives you the option of listing the files in each subdirectory.

Examples

A>>**TREE**

Displays the full path of each directory name on the current drive, along with the names of any directories defined within it. The display takes this form:

Directory Path Listing. The volume is INVEST

Path: A:/

Sub-directories: STOCKS
 BONDS
 COMMODITIES

Path: A:/STOCKS/

Sub-directories: BLUECHIP

Path: A:/STOCKS/BLUECHIP/

Sub-directories: None

```
Path: A:/BONDS/
```

```
Sub-directories: None
```

```
Path: A:/COMMODITIES/
```

```
Sub-directories: None
```

A>TREE C:

Specifies the drive from which to read the tree.

A>TREE -F

Lists all the subdirectories on the current drive and lists the files in each subdirectory. In the following display, there is one subdirectory called MYTEST on drive A. MYTEST contains the files GEORGE, THOMAS, and ABRAHAM.

```
Directory Path Listing. The volume is NAMES
```

```
Path: A:/
```

```
Sub-directories: MYTEST
```

```
Files           : NAMES  
                  DATES  
                  SCORES
```

```
Path: A:/MYTEST/
```

```
Sub-directories: None
```

```
Files           : GEORGE  
                  THOMAS  
                  ABRAHAM
```

TYPE

Form TYPE filespec

Explanation

TYPE displays the contents of a specified text file on the screen. You must enter the filename and the file extension, if there is one. You cannot specify wildcards with TYPE. If you do, you receive an error message.

Typing non-ASCII files, such as object program files, may produce an unreadable display because they contain special characters.

Example

A>TYPE B:FUNCTION.LST

Displays the contents of FUNCTION.LST on drive B on your screen.

VER

Form VER

Explanation

VER displays the version number of your operating system and the hardware environment you are working with. This information is set by your computer manufacturer.

Note: Do not confuse VER with the VERIFY command.

VOL

Forms VOL
 VOL drive:

Explanation

VOL displays the volume label of a disk in the current or specified drive. You can write a volume label on your disk using either the DISKSET or FORMAT commands.

Examples

A>VOL

Displays the volume label of the disk in the current drive (drive A). If the disk has a volume label, VOL displays it like this:

Volume in drive A: is RECORDS

If there is no volume label on the disk, VOL displays:

Volume in drive A: has no label

A>VOL B:

Displays the volume label of the disk in drive B.

End of Section 7

BATCH PROCESSING

A batch file is a text file which contains one or more FlexOS commands. When you enter the name of a batch file on the command line, FlexOS executes each command in the file from top to bottom.

All batch files have the file extension BAT. When you run a batch file, you do not include the file extension.

Your batch file runs until all the commands within it have been run, or you cancel the batch file by pressing Ctrl-C. As long as you do not use any FlexOS commands in your batch file that change the current directory, FlexOS returns you to the directory from which you started your batch file.

FlexOS allows you to use I/O redirection with batch files. You can redirect I/O both from the command line and from within a batch file. Redirection from within a batch file overrides redirection invoked from the command line. See Section 3 for a description of the I/O redirection facilities in FlexOS.

To run a batch file, enter one of the following forms on your command line:

```
batch_filename  
drive: batch_filename  
batch_filename parameters  
drive: batch_filename parameters
```

You can run a batch file from any drive. Parameters are values the batch file uses to get some work done.

Suppose you created a batch file called CLEANUP.BAT that consisted of the following commands:

```
CHDIR /  
DIR *.BAK  
ERASE *.BAK
```

If CLEANUP.BAT is in your current directory of drive A, you can run it by entering this command:

```
A>>CLEANUP
```

After you press Enter, FlexOS runs the commands in order:

1. CHDIR makes the root directory the current directory.
2. DIR displays all backup (BAK) files in the root directory.
3. ERASE deletes all backup (BAK) files from the root directory.

When processing is finished, you are in the root directory of drive A. FlexOS displays the system prompt:

```
A_
```

If you are on drive B and want to run CLEANUP.BAT on drive A, enter this command line:

```
B>A:CLEANUP
```

CLEANUP.BAT runs its commands against the files on drive A. When it is finished, it returns to drive B in the directory where you started.

USING PARAMETERS IN BATCH FILES

Batch files are very useful; however, it is impractical to create a specific batch file for every situation you may encounter. The solution is to create general purpose batch files that can be used in many situations.

To create a general purpose batch file, replace specific information in the batch file, such as drive letters and file names, with "placeholders." Then, when you run the batch file, supply the missing values on your command line. These command line values are called "parameters."

You can have up to 10 different placeholders in a batch file. A placeholder is a percent sign followed by a digit 0-9.

FlexOS takes the first parameter on your command line and substitutes it for each %1 in your batch file. FlexOS substitutes the second parameter for each %2, and so on.

The placeholder %0 is a special case. When FlexOS sees %0 in a batch file, it replaces %0 with the drive name (if you specify one) and the batch filename you are currently running.

The following example shows CLEANUP.BAT with placeholders.

```
CHDIR %1 \  
DIR %1*.BAK  
ERASE %1*.BAK
```

If CLEANUP.BAT is on drive A and you type:

```
A>CLEANUP C:
```

CLEANUP.BAT substitutes C: for each %1 it finds. If your system has a drive C, FlexOS runs each command, displaying it and any output it produces on the screen. The following

example runs CLEANUP.BAT with a parameter of C:.

```
B>CLEANUP C:
```

```
B>CHDIR C:
```

```
B: \
```

```
B>DIR C:*.BAK
```

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

```
ONE          BAK          17      5-08-1985 14:04p  
TWO          BAK          17      5-08-1985 14:04p  
  
                2 Files                12345 bytes free
```

```
B>ERASE C:*.BAK
```

```
B>_
```

Notice that the ERASE command does not have output.

You can put several placeholders in your batch file. For example, ERASEOLD.BAT contains these commands:

```
DEL %1  
DIR %1  
DEL %2  
DIR %2
```

When you run the command line:

```
A>ERASEOLD B:*.BAK C:*.BAK
```

FlexOS substitutes B:*.BAK for %1 and C:*.BAK for %2.

Note: If you want to use the percent sign as part of a filename within a batch file, you must specify two percent signs. For example, to specify the file TAX%.EXE, you must enter it as TAX%%.EXE in the batch file.

You can put placeholders in any order. You can invert the order of the commands in ERASEOLD.BAT and the batch file still works:

```
DEL %2
DIR %2
DEL %1
DIR %1
```

CHAINING BATCH FILES

One of the commands you can include in a batch file is the command to run another batch file. This enables you to run two or more batch files consecutively by entering only one command line. To "chain" two batch files together, make the last command in the first batch file the command to run a second batch file.

Figure 8-1 illustrates the principle of chaining batch files.

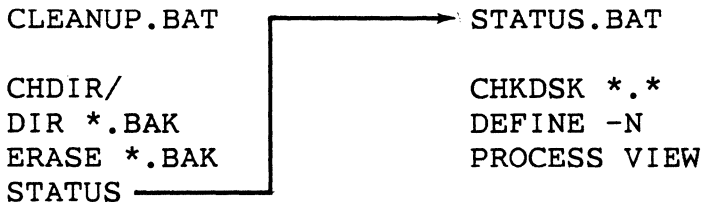


Figure 8-1. Chained Batch Files

To run both batch files, type CLEANUP on your command line

and press Enter:

A>CLEANUP

FlexOS runs the commands in CLEANUP.BAT from top to bottom. When it reaches the command STATUS, control passes to STATUS.BAT. FlexOS runs this file as if you had entered STATUS on the command line. When the commands in STATUS.BAT have finished, FlexOS returns you to your current directory.

If you place the command STATUS ahead of other commands in CLEANUP.BAT, control passes to STATUS.BAT. The remaining commands in CLEANUP.BAT are ignored.

Note: If in Figure 8-3 you include the command CLEANUP as the last command in STATUS.BAT, you create an "infinite loop." CLEANUP.BAT runs its commands, control passes to STATUS.BAT which runs its commands, and then CLEANUP.BAT starts running again. The loop continues until you break out of it with Ctrl-C.

If you press Ctrl-C while running a batch file, FlexOS asks you:

Terminate batch job (Y/N/A)?_

If you press Y, FlexOS ignores the remaining commands in the batch file. If you press N, only the currently running command ends; batch processing continues with the next command in the batch file. If you press A (for "All"), FlexOS stops the current batch file as well as any nested batch files.

NESTING BATCH FILES

Another way to run two or more batch files using one

command line is to use a "nested" batch file. A nested batch file contains the command BATCH followed on the same line by the name of another batch file. Unlike a chained batch file, when the second batch file is finished running, control returns to the first batch file.

Figure illustrates the principle of nested batch files.

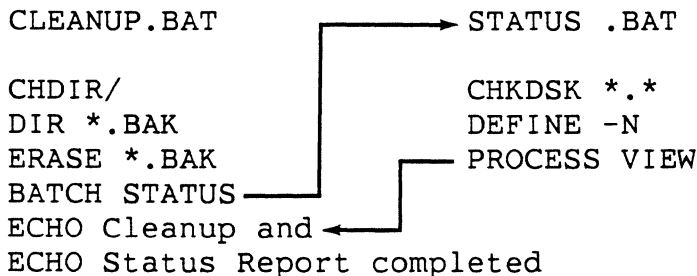


Figure 8-2. Nested Batch Files

In Figure 8-2 FlexOS runs the commands in CLEANUP.BAT from top to bottom. When FlexOS reaches the command BATCH STATUS, control passes to STATUS.BAT. FlexOS runs STATUS.BAT as if you had entered STATUS on the command line. When STATUS.BAT is finished, FlexOS returns to CLEANUP.BAT and continues to run the commands there.

A batch file can contain more than one nested operation. The number of batch files you can nest is limited by the amount of memory available on your computer. If you run out of memory while running a batch file, processing stops and FlexOS displays an error message.

COMMANDS FOR BATCH FILES

Besides the FlexOS commands listed in Section 7, a batch file can contain nine additional commands which are especially

appropriate for batch processing. They are BATCH, CLS, ECHO, FOR, GOTO, IF, PAUSE, REM, and SHIFT. BATCH, GOTO, and SHIFT can only be used in a batch file. The other batch commands can also be entered on the command line, although they are more suited for batch use. The following pages explain these commands in alphabetical order.

BATCH

Form BATCH filespec

Explanation

BATCH directs FlexOS to perform a nested operation. Program control flows from the batch file containing the BATCH filespec command to the named filespec.

"Filespec" is the name of a batch file. If the batch file is not in the current directory, you must specify a directory path with filespec.

When the commands in the second filespec are finished, control returns to the first batch file. Any additional commands in the first batch file are then run.

For examples of nested batch files, see "Nesting Batch Files" above.

CLS

Form CLS

Explanation

CLS clears the screen. Output returns to the screen when you run your next command, application, or batch file.

ECHO

Forms ECHO
 ECHO ON
 ECHO OFF
 ECHO message

Explanation

A batch file is a collection of command lines. The ECHO command determines whether these commands lines are displayed as the batch file runs. Any output and messages produced by a command are always displayed, whether ECHO is on or off.

Examples

```
A>ECHO
```

```
ECHO is ON
```

Displays whether ECHO is on or off.

The command ECHO ON displays the command line of each command as it runs.

When you use ECHO OFF, the command lines in the batch file are not displayed as the file runs. In addition, any remarks in the file made with the REM command are not displayed. Instead, the cursor sits suspended below the command line and no system prompt appears. FlexOS waits for you to enter another command.

The last form consists of the ECHO command followed by a message. This message is always displayed when you run your batch file, whether ECHO is on or off. The message is a string of ASCII characters up to 127 characters long. (For a list of the ASCII characters, see Appendix C.) The following

batch file, saved as ECHOTEST.BAT, illustrates this form:

```
ECHO OFF
TIME -D
ECHO Although ECHO is OFF, this message
ECHO is displayed.
ECHO ON
REM ECHO is ON. Now remarks are displayed
REM as well.
```

When you run ECHOTEST, the following output is displayed:

```
A>ECHOTEST
```

```
A>ECHO OFF
```

```
Current time is 09:41:22.15
```

```
Although ECHO is OFF, this message
is displayed.
```

```
REM ECHO is ON. Now remarks are displayed
REM as well.
```

FOR

Form FOR %c IN (item_1 ... item_n) DO command %c

Explanation

The FOR command lets you repeat another command a number of times for different files, commands, or applications. FOR can be used on the command line or in a batch file, but it is usually used in a batch file. Only one FOR command is allowed on a command line or on one line in a batch file.

The words FOR, IN, and DO are required. The "c" in the "%c" component can be any single character. You can specify two percent signs with the %c component if you wish.

The items in the list can be the names of files (including other batch files), commands, applications, or text. Separate items with one or more blank spaces. There is no limit to the length of your item list; however, the entire FOR command line must be 128 characters or fewer. You can use wildcards with the FOR command, but not path names.

The command at the end of the FOR command can be the name of a FlexOS command, an application, or a batch file. This command is run against each of the items in your list in left-to-right order.

Examples

```
FOR %v IN (FILE1 FILE2 FILE3) DO TYPE %v
```

Runs the TYPE command against FILE1, FILE2, and FILE3, in that order. This is the same as entering TYPE FILE1, TYPE FILE2, and TYPE FILE3 on successive command lines or on successive lines in your batch file.

Items in a FOR command line must be filenames.

GOTO

Form GOTO label . . .

 :label

Explanation

The GOTO command transfers control to the line which has the label matching the one used on the GOTO line. GOTO lets you override the normal top-to-bottom running of commands. You can only use GOTO in a batch file.

Labels in a GOTO command appear in at least two places: after the word GOTO and again on a separate line anywhere else in your batch file. The label name next to GOTO is separated from GOTO by a blank space.

Either label name can be in uppercase or lowercase, or a combination of the two cases.

FlexOS treats the first eight characters in a label name as significant. FlexOS ignores any subsequent characters.

The second label name (the point to which control is passed) must meet three requirements:

- It must be on a line by itself. Any information after the label is treated as a comment.
- It must begin with a colon (:).
- The colon must be in column 1 of the new line.

If the label referenced by GOTO does not meet these conditions, the batch file ends with the message "Label not found."

If you use a label without a corresponding GOTO, then FlexOS treats the labeled line as a comment line.

Example

In the following batch file, GOTO2.BAT, the control passes from GOTO THREE to :THREE. FlexOS ignores the statements between the GOTO statement and the label.

```
REM The name of this file is GOTO2.BAT.
REM GOTO2.BAT exits normally but skips
REM label :TWO.
REM
:ONE
REM I'm in label ONE.
DIR A:
GOTO THREE
:TWO
REM I'm in label TWO.
DIR B:
REM
A:CHKDSK
CLS
:THREE
REM I'm in label THREE.
VER
```

IF

Forms IF EXIST filespec command
 IF NOT EXIST filespec command
 IF string1==string2 command
 IF NOT string1==string2 command
 IF ERRORLEVEL decimal_number command
 IF NOT ERRORLEVEL decimal_number command

Explanation

The IF command lets you test a condition to see if it is true or false. If the condition is true, FlexOS runs the command specified at the end of the IF statement. If the condition is false, FlexOS does not run the command.

The command on the IF/IF NOT line must be a FlexOS command or batch command.

A practical application of this command is to find out if a file is present or absent on disk before you take a certain action.

Examples

The form IF EXIST tests for the presence of a file in the current directory of the current or specified drive. In this case, the filespec is limited to a drive name, a filename, and a file extension; directory paths are prohibited. You can use wildcards in the filespec.

Here is how an IF EXIST statement might be used in a batch file:


```
IF EXIST %1.bak GOTO exit
REM Else filename.BAK does not exist.
REM We need to copy %1 to drive B
REM as filename.BAK.
REM
DIR B:*.BAK
REM
REM Observe the lack of filename.BAK
REM on drive B.
REM
PAUSE
COPY %1 B:%1.BAK
DIR B:*.BAK
REM
REM Confirm filename.BAK has been copied
REM to drive B.
REM
PAUSE
GOTO BYPASS
:EXIT
ECHO filename.BAK already exists
DIR %1.BAK
REM See? I told you so.
:BYPASS
```

The form IF NOT EXIST inverts the logic of IF EXIST. Here is how the same backup task 1 might look using IF NOT EXIST:

```
IF NOT EXIST %1.BAK GOTO BAKUP
REM
:SHOWME
REM
REM filename.BAK exists; prove it
REM
DIR A:*.BAK
DIR B:*.BAK
PAUSE
CLS
GOTO EXIT
REM
:BAKUP
REM Copy %1 to drive B as
REM filename.BAK
REM
COPY %1 B:%1.BAK
GOTO SHOWME
REM
:EXIT
ECHO filename.BAK already exists
```

The preceding examples show that using IF or IF NOT is often a matter of style.

Forms 3 and 4 make sure two strings are identical before running the command. The strings can be any alphanumeric characters. If the two strings are identical, even to the matter of uppercase and lowercase letters, FlexOS runs the command at the end of the line.

For example, if a batch file called DISRAELI.BAT contains the following lines:

```
ECHO
IF %1==Gladstone ECHO %1 is a self-made man.
VOL
```

and if your command line looks like this:

```
DISRAELI Gladstone
```

FlexOS substitutes "Gladstone" for %1 in both places and tests to see if the parameter "Gladstone" matches the string on the right side of the double equal signs. If the test string in the batch file does match the string on the command line, FlexOS runs the ECHO command.

```
A>DISRAELI Gladstone
```

```
A>ECHO
ECHO is ON
```

```
A>ECHO Gladstone is a self-made man.
Gladstone is a self-made man.
```

```
A>VOL
Volume in drive A: is BATCHDISK
```

```
A>_
```

If the two strings are different, FlexOS does not run the ECHO command. FlexOS runs the VOL command whether the strings match or not.

Forms 5 and 6 test exit codes in a program that you create and run against a batch file. If the exit code decimal number in your program is equal to or greater than the decimal number in your batch file, FlexOS runs the specified command. Otherwise, that command is not run and control in your batch file passes to the line following IF ERRORLEVEL.

The ECHO command is often used with the IF and IF NOT statements to inform you of the exit conditions FlexOS encounters while running a program. Consider the following batch file, TESTPROG.BAT:

```
TESTHELP
REM
REM TESTHELP is a program you are running
REM
:CHKERRS
    IF ERRORLEVEL 2 GOTO WRITERR
    IF ERRORLEVEL 1 GOTO READERR
    IF ERRORLEVEL 0 GOTO NORMAL
:WRITERR
    ECHO TESTHELP had a WRITE failure.
    GOTO REALEXIT
:READERR
    ECHO TESTHELP had a READ failure.
    GOTO REALEXIT
:NORMAL
    ECHO TESTHELP finished normally.
    REM It wasn't condition 2 or 1.
    GOTO REALEXIT
:REALEXIT
    REM Th-Th-That's all, folks!
```

TESTPROG.BAT checks each IF ERRORLEVEL number value for a matching value in your running program. Of course, the messages you use with each ERRORLEVEL statement should reflect what is happening in the program.

PAUSE

Forms PAUSE
 PAUSE comment

Explanation

PAUSE suspends batch processing so that you can evaluate the current situation and possibly take an appropriate action. For example, you might want to change diskettes between commands.

Insert PAUSE commands in your batch file where you want to end or continue batch processing. Each PAUSE command stops FlexOS and gives you time to decide whether to end processing. If you decide to end processing, press Ctrl-C. To continue processing, press any other key.

Examples

Form 1 of the PAUSE command suspends processing and displays the message, "Strike a key when ready...." If you press any key (except Ctrl-C), the batch file continues to run with the line following PAUSE. If you press Ctrl-C, batch processing ends. For example, suppose the file MAYBE1.BAT contains the following commands:

```
VER
DIR A:*.BAK
PAUSE
DIR A:*.TXT
PAUSE
COPY A:*. * B:
```

The two PAUSE commands let you examine your screen's contents before you proceed or stop.

Form 2 includes a comment with the PAUSE command. The

comment is displayed when PAUSE suspends processing. The comment is a string of characters up to 121 characters long. For example, suppose MAYBE2.BAT contains the following commands:

```
VER
DIR A:*.BAK
PAUSE The next command erases *.BAK on A!
DEL A:*.BAK
DIR A:*.TXT
PAUSE The next command erases *.TXT on A!
DEL A:*.TXT
```

When you run MAYBE2.BAT, the comments on the PAUSE lines inform you of the consequences of continuing with the batch session. If you want all files with a BAK file extension erased when the first PAUSE occurs, press any key except Ctrl-C. If you decide to continue batch processing, you reach another decision point at the second PAUSE command. Press Ctrl-C to stop or any other key to continue, depending on whether you want to delete all the files on A with a file extension of TXT.

REM

Forms REM
 REM remark

Explanation

The REM command lets you put remarks in your batch file. These remarks are displayed on the screen when FlexOS encounters them in running batch files. Processing is not interrupted by a REM command; the remark is merely displayed and processing continues on the line following the REM command.

REM lines are not displayed when ECHO is OFF.

Examples

Form 1 of the REM command inserts a blank line in your batch file. Careful use of blank lines makes batch files easy to read because of the visual (and hopefully logical) separation of one block of text from another. The following example uses the REM command to separate the COPY commands from the ERASE commands.

```
REM
COPY A:*.BAK C:
COPY B:*.BAK C:
REM
REM
ERASE A:*.BAK
ERASE B:*.BAK
REM
DIR C:*.BAK
```

Form 2 lets you put remarks in your batch file. A remark is a string up to 123 characters. Adding text to your REM line helps keep you informed of actions your batch file took or is

about to take. For example, suppose ENDOFDAY.BAT contains the following commands:

```
REM      copy *.BAK files to drive C
COPY A:*.BAK C:
COPY B:*.BAK C:
REM
REM      Erase *.BAK files on A and B
REM
ERASE A:*.BAK
ERASE B:*.BAK
REM      Show current set of *.BAK files
REM      on drive C
DIR C:*.BAK
```

The meaning of each portion of text becomes clear. Using remarks in your batch files makes the files easier to read and to work with.

SHIFT

Form SHIFT

Explanation

The SHIFT command lets you use more than 10 parameters on one command line. SHIFT does this by shifting all the words on a command line one word to the left. SHIFT can only be used with batch files.

Remember that you can only put 10 placeholders in a batch file (%0 through %9). However, SHIFT lets you use different parameters for the same placeholders in a given batch file.

Because placeholders in a batch file depend on a parameter's position on a command line for placeholder substitution, SHIFT changes the relationship of the parameters to the batch file placeholders in the following way:

- The parameter leftmost on the line before SHIFT was run is no longer available to the running batch file after a SHIFT is run.
- The remaining command line parameters decrement their ordinal values by one, so that the 11th parameter becomes the 10th, the 10th becomes the 9th, and so on.

Compare the relationship of the command line parameters to the batch file placeholders in the following example:

```
A>SHIFT1 FIRST SECOND THIRD
```

```
          SHIFT1.BAT:
```

```
ECHO %0 %1 %2  
SHIFT  
ECHO %0 %1 %2
```

SHIFT**ECHO %0 %1 %2**

FIRST, SECOND, and THIRD are the first, second, and third parameters to be used with **SHIFT1.BAT**.

The batch file has three placeholders: **%0, %1, and %2**. Consequently, **SHIFT1** substitutes for **%0**, **FIRST** for **%1**, **SECOND** for **%2**, and **THIRD** for **%3** before any shifting occurs. When you run **SHIFT1** using the following command line, this output is displayed on the screen:

A>SHIFT1 FIRST SECOND THIRD**A>ECHO SHIFT1 FIRST SECOND
SHIFT1 FIRST SECOND****A>ECHO FIRST SECOND THIRD
FIRST SECOND THIRD****A>ECHO SECOND THIRD
SECOND THIRD****A>ECHO THIRD
THIRD**

Each leftmost parameter from the original command line disappears after each **SHIFT** execution. Notice also how the values for **%0, %1, and %2** change after each **SHIFT**: **FIRST** substitutes for **%0**, **SECOND** for **%1**, and **THIRD** for **%2** after the first **SHIFT**. Finally, notice that the parameter **THIRD** was not available to the running batch file until the parameter list had shrunk by one word -- that is, after the first **SHIFT**.

By structuring your parameters and synchronizing them with the placeholders in your batch file, SHIFT lets you use as many parameters as you can fit on a command line.

VERIFY

Forms **VERIFY**
 VERIFY ON
 VERIFY OFF

Explanation

The VERIFY command makes sure that data is correctly written to the disk and that the data can be read without error.

VERIFY does this whether it is on or off.

End of Section 8

SYSTEM MANAGER

In a multi-user system, the system manager is in charge of maintaining and updating the system, as well as helping other users with problems they encounter. If you are on a single-user system, you are responsible for these tasks; you are your own system manager.

WHAT IS A SYSTEM MANAGER?

The system manager should be an experienced computer user, but not necessarily a programmer. The system manager's duties include:

- Setting up the system.
- Defining the system directory and protecting it.
- Modifying and maintaining the system.
- Serving as a focal point for user problems.

System Manager's Tools

As system manager, you will be working with two important files: USER.TAB and CONFIG.BAT. Both of these files are included on your FlexOS distribution diskettes.

Use the USER.TAB file to set up and modify user accounts. Use CONFIG.BAT to set up the LOGON/LOGOFF procedure, and to set up and modify hardware and software on the system.

SETTING UP LOGON/LOGOFF

FlexOS offers a LOGON/LOGOFF procedure. This procedure is designed to protect user accounts from unauthorized access. Depending on your needs, you may want to implement this procedure on your system.

When the LOGON/LOGOFF procedure is in effect, each user must enter a username and password to gain access to the system. To activate the LOGON/LOGOFF procedure, you must define the logical name PROTECT equal to ON in the CONFIG.BAT file. If you do not include the logical name PROTECT in CONFIG.BAT, or if you define PROTECT equal to OFF, users gain access to the system when they start FlexOS.

The following example shows part of the CONFIG.BAT file and the line of text that sets up the LOGON/LOGOFF procedure for the entire system. The word REM in column 1 of a line makes that line a comment. It is a good practice to document your commands in CONFIG.BAT with comments.

```
REM
REM Set up the LOGON/LOGOFF procedure by
REM defining PROTECT=ON at the system
REM level. Users are prompted for
REM usernames and passwords.
REM
DEFINE -S PROTECT=ON
REM
```

CREATING USER ACCOUNTS

USER.TAB is the file you use to create and modify user accounts. USER.TAB is found on the distribution diskette.

USER.TAB is a text file that can be edited with a text editor. Each FlexOS user is described in eight consecutive fields. The eight fields can occupy more than one line. The first seven fields are separated from each other by commas. If a field has no information in it, FlexOS uses a default value for that field. This is the format of a user account in USER.TAB:

username,password,userid,groupid,home,wmanager,shell,access

Terminate each user account with a carriage return/line feed.

username	LOGON name of the user. Only the first eight characters of this field are significant. You can specify a longer username, but FlexOS only recognizes the first eight characters. This field may be left blank.
password	Encrypted form of the user's password. The password can be up to eight characters long. If you specify eight asterisks (*) in this field, a carriage return will be accepted as input.
userid	User's User ID. This can be a decimal number 0-255. This field must not be empty. The userid of the system manager must be 0.
groupid	User's Group ID. This can be a decimal number 0-255. This field cannot be empty. The groupid of the system manager must be 0.

home	Home directory of the user. This directory is also the initial "default" directory.
wmanager	User's preferred window manager. The window manager lets a user create windows.
shell	Program that displays the system prompt and controls command entry. (See the COMMAND description for an explanation of a shell program.) The shell program runs on each window the user creates.
access	Value that determines how the user can access the computer. The values are defined as follows:

- 1 User can access the computer locally.
- 2 User can access the computer remotely (over the network) only.
- 3 User can access computer both locally and remotely.

4-255 Reserved for future use.

See the DR Net User's Guide for the complete description of the access field. Unless the computer is attached to a network, always set the access value to 1.

The following is a sample user account:

```
TOM,***** ,1,100,A: , , ,1
```


Notice that the `wmanager` and `shell` fields are left blank. FlexOS supplies default values for these fields.

Creating Passwords

All the fields in the `USER.TAB` file can be modified with a text editor. To insert a password in the `USER.TAB` file, you must first initialize the password with eight asterisks, then use the `PASSWORD` command to modify the password field.

See the description of the `PASSWORD` command in Section 10 for a detailed explanation of the command syntax.

```
A>PASSWORD TOM
```

```
New password:
```

```
Confirmation:
```

```
A>_
```

When you enter a user's password, it is encrypted and placed in the password field of `USER.TAB`. When you look at the password in `USER.TAB` you cannot tell what it is. Once a user's password is encrypted, you cannot decipher it.

After a user has been assigned a password, he can change it with the `PASSWORD` command. You should encourage users to keep their passwords secret, and to change them periodically.

CREATING THE SYSTEM DIRECTORY

The "system directory" is a collection of files that can be accessed by any user on the system. The user does not have to have any of these files in his or her own directory to use them. The files in the system directory are called "system files" and can be used, but not updated by the users.

To make a directory the system directory, go in the CONFIG.BAT file and use the DEFINE command with the -S option to define a logical name called SYSTEM:

In the following example, the subdirectory COMMANDS of hard disk drive 1 is defined as the system directory:

```
REM Define the system directory
DEFINE -S SYSTEM:=HD1:/COMMANDS
REM
```

When you define the system directory, FlexOS gives it a default level of protection. As the system manager, you are the owner of the system files: you can read, write, execute, or delete any of the files in the system directory. Group and world users can only execute files in the system directory. You can modify the protection on the system directory with the FSET command.

Note that when you copy a file into the system directory, it does not automatically receive the level of protection assigned to the rest of the system directory. You must modify the file's protection with FSET.

INSTALLING DEVICES

To install a device such as a hard disk, printer, or console, you have to do two basic tasks:

1. Physically attach the device to your system using the documentation that comes with the device.
2. Modify the CONFIG.BAT file so that FlexOS recognizes the device you are installing. This appendix covers only the second task.

Regardless of the type of device you want to add to your system, you use one or more of these commands in CONFIG.BAT:

DVRLOAD
DVRUNIT
DVRLINK
DVRUNLK

Table A-1 summarizes when to use each one of these commands.

Table A-1. Driver Commands

Command	Use It To
DVRLOAD	Install a device for the first time. For example, if you do not have a serial printer on your system and you want to add one, you must include a DVRLOAD command in CONFIG.BAT.
DVRUNIT	Add another unit to an existing device. For example, if you have a serial printer on your system and you want to add another one, use the DVRUNIT command.
DVRLINK	Associate one driver with another. For example, you can attach a printer driver to a port driver. This enables the printer to communicate with your computer through a specified port.
DVRUNLK	Remove a driver from the system that has previously been installed with DVRLOAD.

The driver commands have the following return codes:

- 0 Operation was successful (all driver commands).
- 1 Incorrect parameter (DVRLOAD, DVRUNIT, DVRLINK).
- 2 Operation failed (all driver commands).
- 3 A sub-driver is needed for this operation (DVRLOAD, DVRUNIT, DVRLINK).

Driver command parameters are described in the following descriptions of each command. The term "sub-drivers" refers to drivers that are used by other drivers, as in the case where a printer driver uses a port driver to establish a connection to a port.

DVRLOAD

Form:

DVRLOAD device_name file_name access_level

Explanation:

The DVRLOAD command loads a new driver from a disk. The instructions for the driver are contained in the specified file. The DVRLOAD command is typed in as a line of text in the CONFIG.BAT file. DVRLOAD is not entered from the command line.

Use DVRLOAD if the type of device you are installing has never been installed in your system before.

device_name Logical name of the first unit you are installing. The colon (:) is optional in this logical device name.

device_name Filespec of the device driver.

access_level Determines how the device driver may be used. You set the access level by including one or more options with the DVRLOAD command. The options are presented in Table A-2.

Table A-2. Access Level Options

Option	What It does
P	Permanent Driver. Once the driver is installed, it cannot be removed. You might not want some drivers to be permanent. For instance, you might want to install your printer drivers so that they can be removed at a later date. Other drivers, like your hard disk driver, should be installed so that no one can remove them.
R	Raw Read Access Allowed. You can read the data as it appears on the media. You do not see filenames or the relationships of files within directories.
W	Raw Write Access Allowed. You can write to the data as it appears on the media.
S	Raw Set Access Allowed. You can change the attributes about the driver itself. For example, you can change the country code or baud rate of a console.
E	No exclusive access. Only shared access will be granted, assuming the N bit is set also. Otherwise no access will be allowed.
L	Lockable. This parameter gives you the option of exclusive control of a given device. With the L parameter you can set up a given device for system-wide use. It also allows that device to be locked at a later time for use only by a single user.
M	Multiple Partitions Allowed. When you create a partition on the hard disk with FDISK, you must include the partitioned parameter M with DVRLOAD.

- N** **Shared Access.** This parameter enables two or more processes to access the same device simultaneously. If the N bit is not set, no access is allowed.
- V** **Verify Writes Allowed.** This parameter causes all writes to a device to be verified by FlexOS.
-

You can specify more than one of the access level options when you use DVRLOAD. If you do not specify any options for access level, FlexOS assumes the options PRWSL.

Example:

```
REM This command installs a serial port driver:
REM
DVRLOAD PORT1: BOOT:SERIAL.DVR PRWSL
REM
REM The file specification, BOOT:SERIAL.DVR
REM tells FlexOS that the serial port
REM driver is contained in the file SERIAL.DVR
REM on the boot disk.
REM
```


DVRUNIT**Form:**

DVRUNIT old_device device_name access_level

Explanation:

The DVRUNIT command adds another unit of an existing device to the system. For example, if your system has one hard disk drive and you want to install another, you must use DVRUNIT.

old_device Device name of a driver that has already been installed.

device_name Logical name of the driver's new unit. The colon (:) is optional in this logical device name.

access level Same options for DVRUNIT as for DVRLOAD. If you do not specify any access level options for DVRUNIT, the options RWSL are assumed.

Example:

The following example installs another serial port on the system.

```
REM This command installs a serial port driver:
REM
DVRLOAD PORT1:BOOT:SERIAL.HDVR PRWSEL
REM
REM The file specification, BOOT:SERIAL.DVR tells
REM FlexOS that the serial port driver
REM is contained in the file SERIAL.DVR on the boot
REM disk. PORT1: is the logical device name for
REM the serial port.
REM
REM The DVRUNIT command shown below relates a new unit
REM to a driver that is already installed. PORT2: is
REM the logical device name of the second serial port.
REM
DVRUNIT PORT1: PORT2: PRWSEL
REM
REM
```

DVRLINK

Form:

DVRLINK device_name sub_driver

Explanation:

The DVRLINK command lets you to associate one driver with another.

device_name Name of a previously installed device.

sub_driver Name of the previously installed device you want to associate with device_name.

Example:

The following example associates the serial port, PORT1: with the console, CON1:.

```
REM This is an example of the DVRLINK command.  
REM  
DVRLINK CON1: PORT1:  
REM
```

DVRUNLK**Form:**

DVRUNLK device_name:

Explanation:

The DVRUNLK command lets you remove a driver from your system that you have installed with DVRLOAD.

device_name: Name of a previously installed device.

Example:

This example removes a diskette driver, FD1:, from the system.

```
REM This is an example of the DVRUNLK comma  
REM  
DVRUNLK FD1:  
REM
```

INSTALLING A DISKETTE DRIVE

Your system already has at least one diskette drive installed. The examples below illustrate how to install a diskette drive and a hard disk drive on your system.

The following system has a single diskette drive installed.

```
REM  Load the diskette driver and give
REM  the device the name FLPY0:
REM
DVRLOAD FLPY0:BOOT:FLOPPY.DVR PRWSL
REM
REM  Give the FLPY0: the logical name A:
REM  To access FLPY0:, users will type A:
REM
DEFINE -S A:=FLPY0:
REM
```

To add another diskette drive to the system, use the DVRUNIT command to install a second unit, and define a drive letter for that unit:

```
REM  Install FLPY1:
REM
DVRUNIT FLPY0: FLPY1: PRSWL
REM
REM  Give FLPY1: the logical name B:
REM
DEFINE -S B:=FLPY1:
REM
```

INSTALLING A HARD DISK DRIVE

A hard disk drive uses a different driver, but the procedure for installing one is the same as installing a diskette driver. When you install a hard disk, make sure that you install it with the "unpartitioned" parameter.

```
REM  Install hard disk 1 (HD1:)  
REM  
DVRLOAD HD1:BOOT:HDISK.DVR PRWSEL  
REM  
REM  Give HD1: the logical name C:  
REM  
DEFINE -S C:=HD1:  
REM
```

PREPARING A HARD DISK WITH FDISK

Before you can store FlexOS files on the hard disk, you must prepare it for use. If you try to use the hard disk before you prepare it properly, you receive an error message.

Use the following steps to prepare the hard disk for FlexOS.

1. Make sure that the hard disk driver is installed with the unpartitioned parameter.
2. If the hard disk has never been formatted, format it using the command line shown below. This is called a "physical" format.

Caution: Do not physically format the hard disk if it already contains another operating system or data files. A physical format destroys all the data on the disk. Instead, skip ahead to the next step.

To format the hard disk, enter this command line:

```
A>>FORMAT C:
```

Respond to all FlexOS prompts.

3. Use the FDISK command to invoke the FDISK Options Menu.
4. Use Option 1 to reserve space for FlexOS on the hard disk. This reserved space is called a partition. This section also describes the other FDISK options.
5. Edit CONFIG.BAT to change the parameter of the hard disk driver from unpartitioned to partitioned, and assign drive letters to the partition.
6. Restart FlexOS.
7. Use the FORMAT command to format only the partition (or partitions) you have created. This is called a "logical" format.

Warning: When you create or delete partitions with FDISK, or format a partition with FORMAT, all the data in the partition is destroyed. Before you use these commands, use the BACKUP command to back up any valuable files from the hard disk to diskette. After you have created and formatted your partition, restore your files to the hard disk using the RESTORE command. Both BACKUP and RESTORE are discussed in detail in Section COMMANDS "FlexOS Commands."

Starting FDISK

To display the FDISK Options Menu, type the following command line and press Enter:

```
A>FDISK
```

FDISK displays the FDISK Options Menu shown below.

```
FDISK Options
```

```
Current Hard Disk Drive: C:
```

Choose one of the following:

1. Create FlexOS Partition
2. Change Active Partition
3. Delete FlexOS Partition
4. Display Partition Data
5. Select Next Hard Disk Drive

Enter your choice:[1]_

Notice that in the examples that follow, drive C is the hard disk drive. On your menus, your hard disk drive name appears in place of C. Also note that the line "Current Hard Disk Drive:" and Option 5, "Select Next Hard Disk Drive" only appear if you have more than one hard disk drive.

When you select one of the FDISK options, additional menus and screens are displayed to help you prepare your disk. When FDISK prompts you to enter a value, it often displays a default value in brackets. For example, on the FDISK Options Menu, the default value is 1, "Create FlexOS Partition."

To select the default value on a particular menu, press Enter. If you do not want to use the default value, type your own value at the cursor and press Enter.

If you change your mind about using a particular value or option, you can use the Esc key to cancel your request provided you have not already started the operation by pressing Enter.

Press Esc once to return to the FDISK Options Menu; press Esc twice to return to the system prompt. If you are already at the FDISK Options Menu, press Esc once to return to the system prompt.

CREATING A PARTITION

Option 1 creates a FlexOS partition on your hard disk. You can use this option to reserve the entire hard disk for FlexOS, or to create a partition with a particular size and location.

After you choose Option 1, what you see depend on whether or not partitions already exist on your hard disk.

If No Partitions Exist

If you choose Option 1 and there are no partitions on your hard disk, FDISK displays the following:

```
Create FlexOS Partition
```

```
Current Hard Disk Drive: C:
```

```
Press 'Esc' to return to FDISK Options
```

```
Do you wish to use the entire hard disk  
for FlexOS (Y or N):[Y]
```

The simplest way to partition your hard disk is to reserve the entire disk for FlexOS. If you do not intend to reserve space for other operating systems on your disk, simply press Enter. Your hard disk now has one very large FlexOS partition. FlexOS then displays this message:

```
Partition 1 was created.
```

The FDISK program ends, and you are returned to the system prompt. FlexOS adds this message to your screen:

```
Warning! Be sure to reboot the system  
to incorporate partition changes.
```

You have now created a single FlexOS partition on your hard disk. You are now ready to install the hard disk driver as

partitioned and logically format the hard disk. Skip ahead to "Editing CONFIG.BAT."

If you want to reserve partitions for other operating systems besides FlexOS, type N at the prompt and press Enter. FDISK displays the following:

```
Total hard disk space is xxxx cylinders.
```

```
Maximum available space is xxxx cylinders  
at cylinder xxxx.
```

```
Enter partition size .....[xxxx]
```

In the example above, xxxx is replaced by the appropriate numbers pertaining to your hard disk.

The term "cylinder" refers to an area of your hard disk that stores data. For example, on a hard disk capable of storing 10 megabytes (approximately 10 million characters), there are 305 cylinders each containing 34,816 bytes of data.

If you want the FlexOS partition to use the largest available space on your hard disk, do not enter a value; press Enter. The partition size automatically defaults to the largest value. If you do not want the largest available partition size, enter the partition size you want.

You are then prompted to enter the starting cylinder number:

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

The default value for the starting cylinder number depends on the partition size you specified above. FDISK finds the largest amount of space that can accommodate your partition, and makes the first cylinder of that space the default starting cylinder number. If you want your FlexOS partition to begin

there, press Enter.

If you do not want the default value, enter the value you prefer and press Enter. If you entered a valid partition size, FDISK responds:

Partition 1 was created.

Your screen now shows that you have one partition. The prompt "Enter partition size" returns in case you want to create another partition. You can have up to four partitions on your hard disk.

Create FlexOS Partition

Current Hard Disk Drive: C:

Press 'Esc' to return to FDISK Options

Partition	Status	Type	Start	End	Size
1	N	FlexOS	x	xx	xx

Total hard disk space is xxxx cylinders.

Maximum available space is xxxx cylinders at cylinder xxxx.

Enter partition size[xxxx]

FDISK provides one line of information for each partition.

- The Partition column lists the order in which the partitions appear on the hard disk.
- The Status column shows whether the partition's

operating system is active (A) or not active (N). The active partition contains or will contain the operating system that gains control when the system is started.

- The Type column shows whether or not partitions are FlexOS.
- The remaining columns show the starting cylinder and ending cylinder number of each partition, and the size of the partition in cylinders.

If you do not want to create another partition, press Esc to return to the FDISK Options Menu.

When you are exit FDISK, you receive the message:

```
Warning! Be sure to reboot the system
to incorporate partition changes.
```

You are now ready to install the hard disk driver as partitioned and logically format the hard disk. Skip ahead to "Editing CONFIG.BAT" later in this section.

If Partitions Exist

If partitions already exist on the hard disk, FDISK displays something similar to the following when you choose Option 1:

```
Create FlexOS Partition
```

```
Current Hard Disk Drive: C:
```

```
Press 'Esc' to return to FDISK options
```

Partition	Status	Type	Start	End	Size
1	N	non-	0	37	38
2	N	non-	50	99	50

Total hard disk space is 305 cylinders.

Maximum available space is 205 cylinders
at cylinder 100.

Note: To create a non- partition on your hard disk, you must use the partitioning command of a non- operating system.

The two lines at the bottom of the screen tell you the total amount of space on the hard disk, the size of the largest available space you could use for a partition, and at what cylinder the largest space begins.

Now that you know about the partitions that already exist on your hard disk, FDISK prompts you as before:

Enter partition size[xxxx]

Enter starting cylinder number ...[xxxx]

After you respond to the two prompts, your partition is created. FDISK displays the following message, substituting the number of your new partition for x:

Partition x was created.

If there is still unused space on your hard disk, and there are less than four partitions on it, the status of your partitions is updated, and the prompt "Enter partition size" is redisplayed. At this point you can create another partition or press Esc to return to the FDISK Options Menu.

If there is no unused space on your hard disk, or if there are four partitions on it, you automatically return to the FDISK Options Menu.

When you exit FDISK, you receive the message:

Warning! Be sure to reboot the system to incorporate partition changes.

You are now ready to install the hard disk driver as partitioned and logically format the hard disk. Skip ahead to "Editing CONFIG.BAT" later in this appendix.

Note: If a FlexOS partition already exists on your hard disk, you can use the prompts shown above to change the size of the partition. However, if you decide to alter the size of your FlexOS partition, keep the following points in mind:

- Back up all the files on your FlexOS partition before you manipulate the partition in any way.
- To expand your FlexOS partition, cylinder space must be open before the beginning or after the end of the partition. For example, if your FlexOS partition ends on cylinder 205 and another partition begins on cylinder 206, you cannot expand in that direction. Existing partitions will not "move over" to make room for the expansion.
- When you change the size of a FlexOS partition, it must be logically reformatted.

OTHER FDISK OPTIONS

Changing the Active Partition

Option 2 changes the active partition on your hard disk. When you start your system from a hard disk, it is the operating system in the active partition that gains control of your computer.

When you choose Option 2, FDISK displays a screen similar to the following:

Change Active Partition

Current Hard Disk Drive: C:

Press 'Esc' to return to FDISK Options

Partition	Status	Type	Start	End	Size
1	N	non-	0	37	38
2	N	non-	50	99	50
3	N	FlexOS	100	208	109

Total hard disk space is 305 cylinders.

Enter the number of the partition you want to activate: []_

Enter the number of the partition you want to make active and press Enter. If you want to activate the FlexOS partition, enter 3. FDISK responds:

Partition 3 was activated.

The status of the new active partition is changed from N to A. FDISK prompts you again in case you want to activate another partition.

Enter the number of the partition you want to activate: []_

Press Esc once if you want to return to the FDISK Options Menu; press Esc twice to return to the FlexOS system prompt.

When you exit FDISK, you receive the message:

Warning! Be sure to reboot the system to incorporate partition changes.

If you want to start FlexOS in the new active partition, you must now:

1. Exit the FDISK program.
2. Edit CONFIG.BAT to include the partitioned parameter on the line that installs the hard disk driver.
3. Logically format the hard disk with `FORMAT -S`.
4. Restart the system.

The operating system in the active partition should then start.

Deleting a FlexOS Partition

Warning: Option 3 destroys any information residing in the specified FlexOS partition. Be sure to back up the files in the partition before you continue.

If you want to start an operating system in another partition, make sure the partition is active before you delete the FlexOS partition.

When you choose Option 3, FDISK displays a screen similar to the following:

Delete FlexOS Partition

Current Hard Disk Drive: C:

Press 'Esc' to return to FDISK Options

Partition	Status	Type	Start	End	Size
1	N	non-	0	37	38
2	N	non-	50	99	50
3	A	FlexOS	100	208	109

Total hard disk space is 305 cylinders.

Warning! Partition 3 is the active boot partition!

Warning! Data in partition 3 could be lost.
Do you wish to continue (Y or N)? [N]_

If you have more than one FlexOS partition on your hard disk, FDISK prompts you for the number of the partition you wish to delete.

If you have backed up all the files in your FlexOS partition and you are ready to continue, type Y and press Enter. If you decide to cancel the operation, you can press either Enter to obtain the default N or Esc to return to the FDISK Options Menu.

If you enter Y, FDISK deletes the partition, and displays:

```
Partition x was deleted.
```

The partition information on your screen is then updated. If you have other FlexOS partitions on your hard disk, the Delete FlexOS Partition Menu reappears on your screen in case you want to delete other partitions. The menu stays on your screen until you have no more FlexOS partitions to delete, or you press Esc to return to the FDISK Options Menu.

When you exit FDISK, you receive the message:

```
Warning! Be sure to reboot the system  
to incorporate partition changes.
```

If you did not make another partition active before you deleted FlexOS, the next time you begin a session with FlexOS, you must start FlexOS from another disk.

Displaying Partition Data

Option 4 displays hard disk status information. Your screen looks similar to the following:

```
Display Partition Information
```

```
Current Hard Disk Drive: C:
```

```
Press 'Esc' to return to FDISK Options
```

Partition	Status	Type	Start	End	Size
1	N	non-	0	37	38
2	N	non-	50	99	50
3	A	FlexOS	100	208	109

Total hard disk space is 305 cylinders.

FDISK provides one line of information for each assigned partition. The last line on the screen displays the total amount of space on the hard disk.

Press the Esc key when you are ready to return to the FDISK Options Menu.

Selecting the Next Hard Disk Drive

Option 5 selects the next disk. If you only have one hard disk on your system, this option and the information line "Current Hard Disk Drive:" do not appear on your FDISK Options Menu. If your system has one hard disk, you can skip the discussion of Option 5.

If you have more than one hard disk, choosing Option 5 makes your next hard disk drive the current drive. For example, if your current hard drive is drive C:, this option makes drive D your current hard drive. You can then prepare

the hard disk in drive D using the other FDISK options.

Because your hard disk driver is installed as unpartitioned, drive letters such as C: and D: refer to devices. FlexOS recognizes your partitions as separate logical devices until you reinstall your hard disk driver as partitioned.

After you select the next hard disk drive, FlexOS automatically returns you to the FDISK Options Menu. The information line "Current Hard Disk Drive" displays the hard disk drive you selected. You can now use the FDISK options to work with the current hard disk.

EDITING CONFIG.BAT

Now that you have created one or more FlexOS partitions on your hard disk, you must reinstall your hard disk driver as partitioned and create a driver unit for each partition.

To do this, go in the CONFIG.BAT file and find the section of code that installs your hard disk driver. It looks something like this:

```
REM  Install hard disk 1 (HD1:)  
REM  
DVRLOAD HD1:BOOT:HDISK.DVR P  
REM  
DEFINE -S C:=HD1:  
REM
```

Enter the partitioned parameter, M, at the end of the line that starts with the DVRLOAD command.

Now assign a driver unit to each partition, and give each partition a drive letter with the DEFINE command. Each FlexOS partition must have its own driver unit and drive letter.

FlexOS considers each partition a separate hard disk. You must include the partitioned parameter with the DVRUNIT command as well.

The following example defines the partition as drive D:

```
REM  Install hard disk 1 (HD1:)
REM
DVRLOAD HD1:BOOT:HDISK.DVR PM
REM
DEFINE -S C:=HD1:
REM
REM  Use the DVRUNIT command to associate
REM  the partition with the hard disk
REM  driver that is already installed.
REM  FlexOS considers the
REM  partition as a "logical" hard disk.
REM
DVRUNIT HD2: HD1: M
REM
REM  Define drive letters for each
REM  FlexOS partition:
REM
DEFINE -S C:=HD1:
DEFINE -S D:=HD2:
REM  The FlexOS partition
REM  is identified by D:
```

Restart FlexOS to incorporate your partition changes.

LOGICALLY FORMATTING THE PARTITION

After you install your hard disk driver as partitioned and restart FlexOS, the next step in preparing the hard disk is logically formatting the partition.

The way you format your partition depends on how you want to use it. You have two choices:

1. You can format your partition so that you can store both data files and a copy of FlexOS in your partition. When you format your partition this way, you have the option of starting FlexOS from the hard disk rather than from a diskette. This alternative is described under "Formatting With System Tracks," below.
2. You can format your partition so that it only stores data files. When you format your partition this way, you must start FlexOS from a diskette. This alternative is described under "Formatting Without System Tracks," below.

We recommend that you format your partition using the first alternative, because it gives you more flexibility. If you choose the second alternative and change your mind later, it will take a lot of time to back up your files and reformat the partition.

Formatting With System Tracks

To reserve space for FlexOS on your partition, enter the command line shown below. This form of the FORMAT command formats your partition, reserves space for the FlexOS system tracks and prompts you for a volume label. In this example, the hard disk drive letter is C.

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

Be sure to respond to all FlexOS prompts.

After the partition is formatted, copy your FlexOS distribution diskettes onto the hard disk. In the example below, the FlexOS diskette is inserted in drive A, and the hard disk is drive C. Although the drive letters may be different on your computer, the form of the command is the same. The files

are listed as they are copied.

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

Be sure to copy all your FlexOS distribution diskettes onto the hard disk.

Formatting Without System Tracks

If you prefer to start FlexOS from diskette, you can format your partition without system tracks. To do this, enter the command line shown below. After your partition is formatted you are prompted for a volume label, but space is not reserved for system tracks. In this example, the hard disk drive letter is C.

```
A>FORMAT C:-V
```

Your hard disk is now ready for use.

INSTALLING PARALLEL PRINTERS

The following example installs a parallel printer and gives it the logical name PRN:.

```
REM Install a parallel printer.
REM
DVRLOAD BOOT:PARALLEL.DVR PORT5:
REM
DEFINE -S PRN:=PORT5:
REM
```

Your system must have at least one printer defined as the PRN: device.

DEFINING CONSOLES

To add a console to the system, you must install the appropriate driver file, and give the console a logical name. When you define a console, you also define the default window manager that will run on it.

The following example loads the console driver VT52.DVR, gives the console the logical name CONSOLE:, and runs the user's shell and window manager on the console. The shell and window manager are specified in the USER.TAB file.

To run each user's shell and window manager, you must specify BACK LOGON on the line after each console definition.

```
REM  Install console driver.
REM
DVRLOAD BOOT:VT52.DVR CON1: P
REM
REM  Give CON1: the logical name
REM  CONSOLE:, and run the
REM  user's shell and window
REM  manager on the console.
REM
DEFINE CONSOLE:=CON1:
BACK LOGON
REM
```

ADDING AN APPLICATION TO THE SYSTEM

To add an application to the system, copy the application into the system directory, and use FSET to give the file the system attribute. Files do not automatically receive the system attribute when they are copied into the system directory. The following example adds the word processing application WPROC.EXE to the system directory. In this example, the

system directory is the root directory of the hard disk, drive C.

```
A>COPY WPROC.EXE C:
```

```
A>FSET C:WPROC.EXE -S=ON
```

```
A>_
```

INSTALLING THE PRINT SPOOLER

Before installing the Print Spooler driver, make sure that you have the following files on the boot disk:

PRINTER.DRV	Parallel printer driver
SPLDRV.DRV	Spooler driver
SPOOL.286	Spooler load module
DESPOOL.286	Despooler load module

The spooler command line interface file, PRINT.286, needs to be available but need not be on the boot disk. You can place it in the utilities directory.

The Print Spooler is normally installed by adding the following lines to the boot script:

```
dvrload lpt0: boot:printer.drv lnrws
define -s bgprn: = lpt0:
dvrload spldrv: boot:spldrv.drv lnrws
define -s prn = prn:
define -s prn: = spldrv:
define -s tempdir: = hd0:/temp/
```

The above boot script lines define LPT0: as the name of the background print device (BGPRN). You can use any device designation. (Typically, this would be the printer most often used.) Since PRN: must also be defined at the system level

for the PrtSc function, it's a good idea to define PRN: here. Then it will be defined for the COPY and TYPE utilities where PRN: is also typically used. If you need more flexibility, you can also define PRN: at the process level. Process level definitions have priority over system level definitions.

Note that the background printer (BGPRN) is defined before the spooler driver (SPLDRV) is installed. This is so that the spooler driver knows which printer needs a permanent despooler installed. Other despoolers in the system are terminated when their queues are empty, but the default printer (BGPRN) needs to have a permanent despooler. If you do not define BGPRN, you will only be able to print with the PRINT utility and you will have to specify a device for each entry you make.

A temporary directory (named TEMP above) is used by the spooler driver to store the temporary files that it creates. Thus, the directory definition must be made before you can use a printer. (Use any name other than TEMPDIR. TEMPDIR is reserved for system use.) Be sure to include the trailing slash or the temporary files will not be stored in that directory.

You can also install the Print Spooler driver by modifying the CONFIG.C module, but this is a more difficult method.

How to use the Print Spooler

You can use the Print Spooler

1. with the PRINT command
2. with the COPY or TYPE command, for example:

```
A>COPY filespec prn:
```

```
A>TYPE filespec > prn:
```

3. with the PrtSc (Print Screen) key on your keyboard.

USING THE PRINT SPOOLER WITH DR NET

Note: DR NET must be installed on the system as described in the DR NET User's Guide.

This section uses the following terminology: The "server" refers to the remote node with the printer you want to access. The "requester" is the node that originates the print jobs. A node can be both server and requester for different purposes depending on the system configuration. The files

```
SPLDRV.DRV  
SPOOL.286  
DESPOOL.286
```

must be installed on each server node. The implications of printing jobs within a network using each of the methods previously defined are described below.

Method 1 -- PRINT

To use PRINT without entering a device name each time, you must define the background (BGPRN) printer (in the system define table) to be the printer most often used. This is also the case if the printer is on a remote server node. If you replace the line:

```
define -s prn: = spldrv:
```

in the boot script example shown above with

```
define -s prn: = server::spldrv:
```

a node or device will not have to be defined when a job is entered for this printer. Replace the word "server" with the name of the node desired.

From a server node you can use the PRINT command to send a print job to another node by entering

```
A>PRINT server::dev:filespec
```

where "server" is the destination node name and "dev" is the device name on that node.

Method 2 -- COPY or TYPE

If you use COPY or TYPE for printing, define PRN: as follows:

```
define -s prn: = server::spldrv:
```

Print jobs entered with COPY or TYPE are then queued on the default printer on the node specified as "server". If no remote server is defined, you can still direct print jobs to remote nodes by entering the node designation with each command line. For example:

```
A>COPY filespec server::prn:
```

```
A>TYPE filespec > server::prn:
```

where the actual node name replaces "server." Print jobs entered this way will be printed on the default printer of the node specified.

Method 3 -- PrtSc

To use the PrtSc (Print Screen) function, PRN: must be defined in the system define table as it is for COPY and TYPE. You must also have the Window Manager installed.

End of Appendix A

ERROR HANDLING

If you are a beginning user, you should set your error message help level to 4. If an error occurs, the level 4 message provides you with the most information about the problem. As you become more familiar with FlexOS, you will not need such detailed information to correct your errors and you can set your help level to a lower level. To set the error message help level, see the DEFINE command in Section 7.

This appendix discusses several classes of errors, their causes, and some recovery procedures. It has the following sections:

- Troubleshooting Checklist
- Disk Handling
- Corrupted Files
- Disk Error Messages
- Insufficient Memory
- Unresolvable Errors

TROUBLESHOOTING CHECKLIST

This checklist covers common error conditions whose origin might be unknown to you:

- Are all electrical connections to and from all the peripheral devices connected properly to your computer and to electrical outlets?
- Is your hard-disk unit turned on? If you have a separate hard-disk unit, it must be turned on before you can access it.

- Have you used **FORMAT** and **FDISK** to prepare a FlexOS partition? If you have a hard disk as part of your system, you cannot access your hard-disk drive until you have prepared the hard disk with these commands. **FORMAT** is described in Section 7, "FlexOS Commands." **FDISK** is described in Appendix A, "System Manager."
- If a file is set to Read-Only, you can read the file but you cannot write to it. If you are the owner of the file, you can use the **FSET** command to give yourself the write privilege. If you are not the owner of the file, ask the owner to give you a copy of the file or have him modify the protection on the file so you can write to it.
- If the drive is set to Read-Only, you cannot write to files on that drive. This can occur when you exchange disks when files are still open on the disk you removed.
- If you press **Ctrl-S** or **Scroll Lock**, your keyboard is locked for that window until you press **Ctrl-Q** to unlock it.
- Did you specify a drive when entering the command line? If you did not specify a drive, FlexOS looks for a file only in the current directory of the current drive.
- Did you specify a valid drive? FlexOS drives can be identified by the letters **A-P**.
- Do you have sufficient privileges to work with the file or disk? FlexOS has three classes of user: owner, group, and world. Each of these classes can have different privileges. For example, the owner of the file may be able to delete the file, but group and world users may not. If you are unfamiliar with the FlexOS file and disk security, see the **DISKSET** and **FSET** commands in Section 7.
- Are you running a program under the correct operating system version? Most FlexOS commands run only under the operating system with which they are shipped.

DISK HANDLING

Many errors arise from the handling of disks. Either the disk is incorrectly inserted or the disk itself is damaged. Use the following checklist to locate the error:

- Is the disk correctly inserted?
 - If there is no disk in the drive, you may receive an error. Make sure that you have a disk in the drive before you continue.
 - If the disk is not securely in place, or if the drive door is open, you may also receive an error. Make sure that the disk is correctly inserted and the drive door securely closed before you continue.
 - This class of errors can also occur when the disk is not in the correct drive. The FlexOS system prompt usually identifies the current drive.
- Is the disk the correct number of sides for the drive? Double-sided drives can read single- and double-sided disks. Single-sided drives can access only single-sided disks.
- Is the disk physically write-protected? 5 1/4-inch disks are write-protected in the absence of a notch on the upper-right disk edge. Check the documentation provided by the disk manufacturer for instructions on removing the write-protection on the disk.
- If the error is not caused by one of the above problems, the disk might be damaged. Use the COPY command to copy as much of the information on the disk as possible to a fresh disk. Replace any files you cannot copy with the files from your backup disk. If you have not maintained a backup disk, you must recreate the missing files.

You can use the CHKDSK command to verify the integrity of your disks. CHKDSK is described in Section 7, "FlexOS Commands."

If these directions fail to correct an error condition, you probably are faced with a hardware error. As far as you are able, reconstruct the exact situation that produced the error message, then consult the dealer from whom you purchased your system.

CORRUPTED FILES

Files can be corrupted by unforeseen events. Power outages are the most common cause. Replace the damaged file from your backup copy. If you do not have a backup copy, you will have to recreate the file.

If you are writing your own programs, it is possible to create one that writes over other files on the disk, or which corrupts the directory or other FlexOS system tracks. If this class of error occurs after you have run a program of your own, debug your program carefully before attempting to run it again.

DISK ERROR MESSAGES

If you receive an error related to disks, such as "Disk not ready", go through "Troubleshooting Checklist" and "Disk Handling," above. Check for the obvious: open drive doors, hard-disk units not turned on, upside-down disks, write-protect tabs on disks. If, after running through the two checklists, you still cannot discover the error condition, you probably have a problem with the disk medium. Copy as many files as you can to a formatted and verified diskette.

INSUFFICIENT MEMORY

You might receive a message indicating that you have insufficient memory to load a program. Such an error might also occur when there is space in memory, but the available memory is fragmented.

If you encounter an insufficient memory message, you can stop programs running in different windows to create more free space in memory. The CANCEL command lets you terminate programs running in any window.

UNRESOLVABLE ERRORS

If you encounter an error condition that you are unable to correct, contact the dealer from whom you purchased your personal computer. Be prepared to provide the following information:

- The serial number of the operating system you are using. This number appears in the sign-on message when you start FlexOS.
- Your personal computer's configuration: the number of drives, quantity of memory, and additional equipment such as printers, modems, and so forth.
- How the error occurred. Indicate which programs were running at the time the error occurred. If possible, provide a disk with a copy of the program.
- Provide the text of the error message you received.

End of Appendix B

ASCII CONVERSION TABLES

This appendix contains decimal and hexadecimal conversion tables for the ASCII character set. ASCII is an acronym for American Standard Code for Information Interchange.

Refer to these tables if you need the decimal or hexadecimal equivalent of ASCII characters.

Most ASCII characters are the alphanumeric and special characters you find on your keyboard. These are the "printable" characters. The special ASCII Ctrl symbols are listed in Table C-1. Table C-2 gives the decimal and hexadecimal equivalents of the complete ASCII character set.

Table C-1. ASCII Symbols

Symbol	Meaning	Symbol	Meaning
ACK	acknowledge	FS	file separator
BEL	bell	GS	group separator
BS	backspace	HT	horizontal tabulation
CAN	cancel	LF	line feed
CR	carriage return	NAK	negative acknowledge
DC	device control	NUL	null
DEL	delete	RS	record separator
DLE	data link escape	SI	shift in
EM	end of medium	SO	shift out
ENQ	enquiry	SOH	start of heading
EOT	end of transmission	SP	space
ESC	escape	STX	start of text
ETB	end of transmission	SUB	substitute
ETX	end of text	SYN	synchronous idle
FF	form feed	US	unit separator
		VT	vertical tabulation

Table C-2. ASCII Conversion Table

ASCII	Decimal	Hex
NUL	000	00
SOH (CTRL-A)	001	01
STX (CTRL-B)	002	02
ETX (CTRL-C)	003	03
EOT (CTRL-D)	004	04
ENQ (CTRL-E)	005	05
ACK (CTRL-F)	006	06
BEL (CTRL-G)	007	07
BS (CTRL-H)	008	08
HT (CTRL-I)	009	09
LF (CTRL-J)	010	0A
VT (CTRL-K)	011	0B
FF (CTRL-L)	012	0C
CR (CTRL-M)	013	0D
SO (CTRL-N)	014	0E
SI (CTRL-O)	015	0F
DLE (CTRL-P)	016	10
DC1 (CTRL-Q)	017	11
DC2 (CTRL-R)	018	12
DC3 (CTRL-S)	019	13
DC4 (CTRL-T)	020	14
NAK (CTRL-U)	021	15
SYN (CTRL-V)	022	16
ETB (CTRL-W)	023	17
CAN (CTRL-X)	024	18
EM (CTRL-Y)	025	19
ESC (CTRL-[)	027	1B
FS (CTRL-\)	028	1C
GS (CTRL-])	029	1D

Table C-2. Continued

ASCII	Decimal	Hex
RS (CTRL-^)	030	1E
US (CTRL-_)	031	1F
(SPACE)	032	20
!	033	21
"	034	22
#	035	23
\$	036	24
%	037	25
&	038	26
'	039	27
(040	28
)	041	29
*	042	2A
+	043	2B
,	044	2C
-	045	2D
.	046	2E
/	047	2F
0	048	30
1	049	31
2	050	32
3	051	33
4	052	34
5	053	35
6	054	36
7	055	37

Table C-2. Continued

ASCII	Decimal	Hex
8	056	38
9	057	39
:	058	3A
;	059	3B
<	060	3C
=	061	3D
>	062	3E
?	063	3F
@	064	40
A	065	41
B	066	42
C	067	43
D	068	44
E	069	45
F	070	46
G	071	47
H	072	48
I	073	49
J	074	4A
K	075	4B
L	076	4C
M	077	4D
N	078	4E
O	079	4F
P	080	50
Q	081	51

Table C-2. Continued

ASCII	Decimal	Hex
R	082	52
S	083	53
T	084	54
U	085	55
V	086	56
W	087	57
X	088	58
Y	089	59
Z	090	5A
[091	5B
\	092	5C
]	093	5D
^	094	5E
<	095	5F
'	096	60
a	097	61
b	098	62
c	099	63
d	100	64
e	101	65
f	102	66
g	103	67
h	104	68
i	105	69
j	106	6A
k	107	6B

Table C-2. Continued

ASCII	Decimal	Hex
l	108	6C
m	109	6D
n	110	6E
o	111	6F
p	112	70
q	113	71
r	114	72
s	115	73
t	116	74
u	117	75
v	118	76
w	119	77
x	120	78
y	121	79
z	122	7A
{	123	7B
	124	7C
}	125	7D
~	126	7E
DEL	127	7F

End of Appendix C

INDEX

- * (star) wildcard, 1-11
- <NEXT> key, 2-1
- <number> command, 2-5
- <PREV> key, 2-1
- <WINDOW> key, 2-1, 2-2
- ? (question mark) wildcard, 1-12

DR EDIX

- * wildcard, 5-9
- ? wildcard, 5-9
- ^ A command, 6-1
- ^ E command, 5-9
- ^ F command, 6-1
- ^ KC command, 5-5, 6-4
- ^ KV command, 5-5
- ^ KY command, 5-5, 6-4
- ^ QE command, 6-1
- ^ QX command, 6-1
- ^ X command, 5-9
- buffers, 5-1, 5-4
- circular search, 5-8
- continuing searches, 5-8
- global replacement, 5-11
- local replacement, 5-11
- marking single line, 5-6
- null strings, 5-11
- patterns, 5-8
- restoring single window, 5-2

- searching text strings, 5-8
- wildcards, 5-9
- windows, 5-4

- ^ (caret) wildcard, 1-13

A

- Access field in USER.TAB, A-4
- Adding a device to the system, A-7, A-13
- Adding an application to the system, A-35
- Application
 - adding to the system, A-35
- Archive attribute, 7-73, 7-76
- Archive files, 7-73
- Archiving, 7-73
- ASCII collating sequence, 7-117
- ASCII Symbols, C-1
- ASCII Tables, C-1
- ASSIGN command, 7-5
- Associating one device driver with another, A-15

B

- BACK command, 7-7
- Backing up files, 7-9

BACKUP command, 7-9
Bad cluster, 7-105
Bad sector, 7-105, 7-107
BATCH command, 8-9
Batch files
 basic forms, 8-1
 chaining, 8-5
 commands, 8-7
 including remarks, 8-23
 nesting, 8-6
 suspending processing, 8-21
 testing conditions in, 8-16
 using more than 10
 parameters, 8-25
 using parameters, 8-3
 using placeholders, 8-3
Batch processing
 commands, 8-7
 concepts, 8-1
 files, 8-1
 I/O redirection, 8-1
 running a batch file, 8-1
 suspending, 8-21
 using more than 10
 parameters, 8-25
 using parameters, 8-3
 using placeholders, 8-3
Baudrate, 7-29
Blinking cursor, 7-79
Boot script, A-37
BREAK command, 7-12
Buffers, 4-1
Built-in commands, 1-3

C

C (Create Window) Command,
 2-4
CANCEL command, 7-13
Canceling a process, 7-102
Canceling a program, 7-13
Chaining commands, 3-4
Changing file attributes, 7-73
Changing the active partition,
 A-26
Changing the current directory,
 7-16
Changing the date, 7-40
Changing the time, 7-124,
 7-125
CHDIR (Change Current
 Directory) Command,
 7-16
CHDIR command, 7-16
CHKDSK Command), 7-18
Classes of user
 group, 7-74, 7-76
 owner, 7-74, 7-76
 world, 7-74, 7-77
Clear screen command, 8-10
CLS command, 8-10
Code, 7-100
Code Size, 7-100
Code Start, 7-100
Combining files, 7-36
COMMAND Command, 7-23
Command line, 1-2
Command line
 cancelling, 1-5
 continuing, 1-5
 deleting, 1-3
 editing keys, 1-3
 moving cursor on, 1-3

- options, 1-3
- recalling, 1-3
- resuming, 1-5
- stopping, 1-5
- suspending, 1-5
- Command prompt, 2-5
- Commands
 - built-in, 1-3
 - COPY, 1-8
 - DEFINE, 1-14
 - ERASE, 1-8
 - external, 1-3
 - FIND, 3-5
 - FORMAT, 1-5
 - MKDIR, 1-9
 - MORE, 3-5
 - RENAME, 1-8
 - SORT, 3-5
 - TYPE, 1-8
- COMP Command, 7-26
- Comparing files, 7-26
- Comparing two diskettes, 7-50
- CONFIG Command, 7-29
- CONFIG.BAT file
 - creating the system directory, A-5
 - hard disk drive, A-18
 - installing a diskette drive, A-17
 - installing hard disk as partitioned, A-31
 - purpose, A-1
 - setting up the LOGON/LOGOFF procedure, A-2
- Configuring serial ports, 7-29
- COPY, 7-32
- COPY command, 1-8, A-37
- Copying ASCII files, 7-33
- Copying binary files, 7-33
- Copying diskettes, 7-53
- Copying files, 1-8, 7-32
- Copying hidden and system files, 7-33
- Copying to devices, 7-38
- Creating logical names, 7-42
- Creating passwords, A-5
- Creating the system directory, A-5
- Creating user accounts, A-3
- Creating windows, 2-4
- Ctrl key (DR EDIX), 4-1
- Ctrl-Q, B-2
- Ctrl-S, B-2
- CTTY Command, 7-39
- Current window, 2-1
- Cursor, 1-2
- Cursor movement DR EDIX, 4-6
- Cylinder, A-22

D

- D (Delete window) Command, 2-4
- Data, 7-100
- Data Size, 7-100
- Data Start, 7-100
- Date
 - changing, 7-40
 - displaying, 7-40
- DATE Command, 7-40
- Decimal conversion, C-1
- Default drive, 1-2

- Defective cluster, 7-105
- Defective sector, 7-105, 7-107
- DEFINE command, 1-14, 7-42
- Defining a search order for
 - directory paths, 7-89
- Defining a search order for file
 - extensions, 7-85
- Defining consoles, A-35
- Defining the standard input and
 - output device, 7-39
- DEL (Delete) Command, 7-59
- Delete privilege, 7-75
- Deleting a partition, A-28
- Deleting logical names, 7-42,
 - 7-45
- Deleting windows, 2-4
- Devname, 7-29
- DIR Command, 7-46
- Directories, 1-9
- Directories
 - changing, 1-10
 - recovering from a defective
 - disk, 7-105
- Directory
 - changing current, 7-16
 - listing files in, 7-46
- Directory path
 - displaying, 7-126
 - maximum length, 1-10, 7-81
- Disabling disk protection,
 - 7-56, 7-57
- Disk drives
 - displaying assignments,
 - 7-44, 7-45
 - reassigning, 7-5
 - resetting, 7-5
- Disk Handling, B-3
- Disk protection
 - disabling, 7-56, 7-57
 - displaying, 7-57
 - enabling, 7-56, 7-57
 - modifying, 7-57
 - setting, 7-56
 - writing a volume label, 7-56,
 - 7-58
- DISKCOMP Command, 7-50
- DISKCOPY Command, 7-53
- Diskette
 - formatting, 7-68
- Diskettes
 - statistics, 7-18
- Disks
 - statistics, 7-18
- DISKSET Command, 7-56
- Displaying a volume label,
 - 7-130
- Displaying directory paths,
 - 7-126
- Displaying disk protection,
 - 7-57
- Displaying drive assignments,
 - 7-44, 7-45
- Displaying logical names,
 - 7-42, 7-44, 7-45
- Displaying partition status,
 - A-30
- Displaying text files, 7-128
- Displaying the date, 7-40
- Displaying the time, 7-124,
 - 7-125
- DR EDIX, 4-1
- DR EDIX
 - ^G, 4-7
 - ^J, 4-11

^J command, 4-1, 4-5, 6-5
 ^KB command, 5-5, 6-4
 ^KC command, 5-7
 ^KE, 4-9
 ^KE command, 6-3
 ^KF command, 5-3, 6-3
 ^KQ, 4-12
 ^KQ command, 4-5, 6-3
 ^KR command, 5-12, 6-3
 ^KS, 4-9
 ^KS command, 6-3
 ^KV command, 5-7, 6-4
 ^KY command, 5-7
 ^L command, 5-8, 6-5
 ^QA command, 5-8, 5-10, 6-5
 ^QF command, 5-8, 6-5
 ^QY, 4-7
 ^V command, 4-6
 ^Y command, 4-7, 6-1
 ^Z1 command, 5-2, 5-3, 6-4
 ^Z2 command, 5-2, 5-3, 6-4
 ^ZA, 4-7
 ^ZA command, 6-1
 ^ZB command, 6-3
 ^ZG command, 5-12, 6-1
 ^ZI, 4-7
 ^ZI command, 6-1
 ^ZO command, 5-2, 5-3, 6-4
 ^ZS, 4-12
 ^ZS command, 5-4, 5-6, 6-5
 ^ZU, 4-7
 assigning buffers to windows, 5-3
 assigning filename to buffer, 5-3
 Backspace key, 4-7
 backup files, 4-11
 block commands, 5-5
 buffer assignments, 5-3
 buffers, 4-1
 changing filename, 5-13
 changing window buffer assignments, 5-3
 command summary, 6-1
 commands, 6-1
 copying blocks, 5-7
 cursor control keys, 4-6
 Del key, 4-7
 deleting blocks, 5-7
 deleting lines, 4-7
 deleting text, 4-7
 DREDIX.286, 4-2
 editing text, 4-7
 Enter, 4-7
 Enter key, 4-5, 4-6
 entering text, 4-6
 file-naming conventions, 4-11
 files, 4-2, 5-4
 help facility, 4-1, 6-5
 help text, 4-2
 HELP.EDX, 4-2
 insert mode, 4-6
 inserting blank lines, 4-7
 joining lines, 4-9
 line numbers, 5-12
 LSN.EDX, 4-2
 marking another block, 5-6
 marking blocks, 5-5
 maximum line length, 5-1
 maximum number of lines,

- 5-1
- merging files, 5-12
- message area, 4-3, 4-5
- messages, 6-6
- moving blocks, 5-7
- moving the cursor, 4-6
- opening files, 4-9
- overtyping mode, 4-6
- prompt area, 4-3, 4-4
- quitting, 4-12
- recovering patterns, 5-9
- replace command, 5-10
- replacing, 6-5
- replacing text strings, 5-10
- saving work, 4-9
- searching, 6-5
- searching and replacing text, 5-8
- splitting lines, 4-7
- starting, 4-3
- status area, 4-3, 4-5
- status command, 5-4
- status display, 4-12
- status screen, 6-5
- status table, 5-4
- STUDENT.BAT, 4-2
- text area, 4-3, 4-4
- TEXT file, 4-2
- tutorial, 4-2, 4-3
- tutorial, exiting, 4-3
- tutorial, starting, 4-3
- undeleting lines, 4-7
- unmarking blocks, 5-6, 5-12
- up-arrow key, 5-11
- window commands, 5-3
- windows, 5-2
- write errors, 4-11

- DREDIX.286, 4-2, 4-3
- Driver access levels
 - exclusive access only, A-10
 - lockable, A-10
 - permanent driver, A-10
 - raw read, A-10
 - raw set, A-10
 - raw write, A-10
- DVRLINK command, A-7, A-15
- DVRLOAD command, A-7, A-10
- DVRUNIT command, A-7, A-13
- DVRUNLK command, A-7, A-16

E

- ECHO command, 8-11
- Editing Keys
 - BACKSPACE, 1-3
 - Ctrl-C, 1-5
 - Ctrl-Q, 1-5
 - Ctrl-R, 1-3
 - Ctrl-S, 1-5
 - Ctrl-X, 1-3
 - DELETE, 1-3
 - Enter, 1-3
 - left arrow, 1-3
 - right arrow, 1-3
- Embedded backslashes, 7-81
- Enabling disk protection, 7-56, 7-57
- ERASE command, 1-8, 7-59
- Erasing files, 1-8
- Error Handling Procedures, B-1

- Error message help level, 7-42
- Error messages
 - help level, 7-42
- Errors Unresolvable, B-5
- Esc (escape) key, A-20
- ESC key, A-29
- Event Flags, 7-99
- Evnt, 7-99
- Exclusive driver access level,
 - A-10
- Execute privilege, 7-75
- EXIT Command, 7-61
- Exiting secondary shell, 7-61
- Exiting secondary user
 - environment, 7-61
- External commands, 1-3

F

- Family ID, 7-13, 7-97
- FDISK command, 7-62, A-18
- FDISK Options Menu, A-19
- File access privileges, 7-116
- File specification, 1-2
- Files
 - archive, 7-73
 - changing attributes, 7-73
 - changing filespecs, 7-110
 - combining, 7-36
 - comparing, 7-26
 - copying, 7-32
 - copying ASCII, 7-33
 - copying binary, 7-33
 - copying hidden and system,
 - 7-33
 - deleting, 7-59

- displaying attributes, 7-75
- displaying text, 7-128
- erasing, 7-59
- fragmented, 7-20
- hidden, 7-73
- protecting, 7-72
- Read-Only, 7-74
- recovering from a defective
 - disk, 7-107
- renaming, 7-110
- restoring, 7-112
- setting protection, 7-74
- statistics, 7-18
- system, 7-73
- with noncontiguous areas,
 - 7-20

- Files Corrupted, B-4
- Filespec, 1-2
- Filter commands
 - FIND, 7-63
 - MORE, 7-83
 - SORT, 7-117
- Filters, 3-5
- FIND command, 3-5, 7-63
- FlexOS image file, 7-122
- FlexOS window information,
 - 7-97
- Fmly, 7-97
- FOR command, 8-13
- FORMAT, 1-5
- FORMAT Command, 7-66,
 - A-18, A-19
- Formatting a diskette, 7-68
- Formatting disks, 1-5
- Formatting with system tracks,
 - A-33
- Formatting without system

tracks, A-34
FSET command, 7-72, B-2

G

GOTO command, 8-14
Group, 7-74
Group ID, 7-97
Groupid field in USER.TAB,
A-3
Grp, 7-97

H

Hard disk
changing partition size, A-26
creating a partition, A-21
deleting a partition, A-28
displaying partition status,
A-30
formatting, 7-68
formatting with system tracks,
A-33
formatting without system
tracks, A-34
installing as partitioned,
A-31
logically formatting, A-32
maximum number of
partitions, A-23
partition ending cylinder
number, A-24
partition size, A-22, A-24
partition starting cylinder
number, A-24

partition status, A-23
partition type, A-24
preparing for use, 7-62,
A-18
selecting, A-30
Heap, 7-100
Heap Size, 7-100
Heap Start, 7-100
Help level
for error messages, 7-42
HELP.EDX, 4-2, 4-3
Hexadecimal conversion, C-1
Hidden attribute, 7-73, 7-75
Hidden files, 7-73
Home field in USER.TAB, A-3

I

I/O redirection, 3-1
I/O redirection
adding output to device, 3-2
adding output to file, 3-2
chaining commands, 3-4
input from device, 3-3
input from file, 3-3
output to device, 3-2
output to file, 3-2
piping commands, 3-3
symbols, 3-1
techniques, 3-1
IF command, 8-16
Input/output redirection
See also I/O redirection
Installing a device for the first
time, A-7, A-10
Installing devices

defining consoles, A-35
diskette drive, A-17
DVRLINK command, A-15,
A-7
DVRLOAD command, A-7,
A-10
DVRUNIT command, A-7,
A-13
DVRUNLK command, A-16,
A-7
hard disk drive, A-18
overview, A-7
parallel printers, A-34
Insufficient Memory, B-5

J

Joining lines
with DR EDIX, 4-9

K

L

LIST command, 7-78
Listing built-in commands,
7-78
Listing files in a directory,
7-46
Lockable driver access level,
A-10
Logical devices, 1-14
Logical filenames, 1-15
Logical format, 7-69, A-19,

A-32
Logical information, 1-16
Logical names, 1-13
Logical names
creating, 1-14, 7-42
deleting, 7-42, 7-45
displaying, 7-42, 7-44, 7-45
displaying equivalent names,
7-45
file names, 1-15
process level, 7-43, 7-44
requirements, 1-14
reserved, 1-16
system level, 7-43, 7-44
LOGOFF command, 7-79
Logoff message, 7-79
LOGON command, 7-80
LOGON message, 7-80
LOGON prompt, 7-80
LSN.EDX, 4-2

M

Maximum Memory, 7-100
Maxm, 7-100
Memory
statistics, 7-18
Memory lock status, 7-100
Message Window, 2-2, 2-5
MKDIR command, 1-9, 7-81
Modifying disk protection,
7-57
MORE command, 3-5, 7-83

N

O

Operating system
 version number, 7-129
ORDER command, 7-85
Owner, 7-74

P

Parent Process ID, 7-99
Parity, 7-29
Partition
 changing active, A-26
 changing size, A-26
 creating, A-21
 definition, A-19
 deleting, A-28
 displaying status, A-30
 ending cylinder number,
 A-24
 formatting, A-19
 logically formatting, A-32
 maximum number of, A-23
 size, A-22, A-24
 starting cylinder number,
 A-22, A-24
 status, A-23
 type, A-24
Partitioned parameter, A-10
PASSWORD command, 7-87,
 A-5
Password field in USER.TAB,
 A-3
Path
 displaying, 7-126
PATH command, 7-89

PAUSE command, 8-21
Pcon, 7-97
Permanent driver access level,
 A-10
Physical format, 7-68, A-18
Pid, 7-97
Piping commands, 3-3
Ppid, 7-99
PRINT command, 7-91
Print queue, loading, 7-91
Print spooler, 7-91, A-36
Print Spooler
 installing, A-36
PROCESS command, 7-96
Process ID, 7-13, 7-97
Process information
 displaying, 7-97
Process level logical names,
 7-43, 7-44
Process Name, 7-13, 7-97
Process Priority, 7-99
Process Status, 7-97
PROCESS VIEW
 user filter option, 7-102
Program
 canceling, 7-13
 stopping, 7-13
PROMPT command, 7-103
Protecting files, 7-72
PrtSc key, A-37
Prty, 7-99

Q

R

- Raw read access level, A-10
- Raw set access level, A-10
- Raw write access level, A-10
- Read privilege, 7-75
- Read-only, B-2
- Read-Only attribute, 7-74, 7-76
- Read-Only file, 7-74
- Reassigning disk drives, 7-5
- RECDIR command, 7-105
- RECFILE command, 7-107
- REM command, 8-23
- RENAME command, 1-8, 7-110
- Renaming files, 1-8
- Resetting disk drive assignments, 7-5
- Restarting the system, A-27
- RESTORE command, 7-112
- Restoring backed up files, 7-112
- RMDIR command, 7-115
- Root directory, 1-9

S

- Sample user account, A-4
- Screen (physical console) information, 7-97
- Search order for directory paths, 7-89
- Search order for file extensions, 7-85
- Sectors, 1-5
- SECURITY command, 7-116
- Selective backup, 7-73
- Serial ports

- configuring, 7-29
- Setting file protection, 7-77
- Setting up the LOGON/LOGOFF procedure, A-2
- Shared access, A-11
- Shell
 - definition, 7-23
 - exiting secondary, 7-23, 7-61
 - invoking, 7-23
 - permanent, 7-23
 - secondary, 7-23
- Shell field in USER.TAB, A-4
- SHIFT command, 8-25
- SORT command, 3-5, 7-117
- Starting FlexOS
 - from a diskette, A-34
 - from the hard disk, A-33
- Stat, 7-97
- Status lines, 2-3
- Status table, 5-4
- Status Window, 2-2
- Status Window
 - command menu, 2-4
- Stopbits, 7-29
- Stopping a program, 7-13
- STUDENT.BAT (DR EDIX file), 4-2
- Subdirectories, 1-9
- Subdirectories
 - how to create, 7-81
 - listing, 7-127
- Subdirectory
 - removing, 7-115
- Suspending processing of batch files, 8-21
- SYS Command, 7-122

- System attribute, 7-73, 7-76
- System directory
 - creating, A-5
- System files, 7-73
- System level logical names,
 - 7-43, 7-44
- System manager
 - creating user accounts, A-3
 - definition, A-1
 - duties, A-1
 - setting up the LOGON/LOGOFF procedure, A-2
- System prompt, 1-2
- System prompt setting, 7-103

T

- Testing conditions in a batch file, 8-16
- TEXT (DR EDIX file), 4-2
- Time
 - changing, 7-124, 7-125
 - displaying, 7-124, 7-125
- TIME command, 7-124
- Transferring FlexOS to another disk, 7-122
- TREE command, 7-126
- Troubleshooting checklist, B-1
- TYPE command, 1-8, 7-128, A-37
- Typing files, 1-8

U

- User, 7-97
- User classes, 7-116
- User environment
 - definition, 7-23
 - exiting secondary, 7-61
 - invoking, 7-23
 - secondary, 7-23
- User ID, 7-97
- USER.TAB file
 - access field, A-4
 - creating user accounts, A-3
 - groupid field, A-3
 - home field, A-3
 - password field, A-3
 - purpose, A-1
 - sample user account, A-4
 - shell field, A-4
 - userid field, A-3
 - username field, A-3
 - wmanager field, A-4
- UserId field in USER.TAB, A-3
- Username field in USER.TAB, A-3

V

- VER command, 7-129
- VERIFY command, 8-28
- Verify writes parameter, A-11
- VOL command, 7-130
- Volume label, 7-56
- Volume label displaying, 7-130

W

- Wildcards, 1-11
- Wildcards
 - caret (^), 1-13
 - question mark (?), 1-12
 - star (*), 1-11
- Window number, 7-13
- Windows, 2-1
- Windows
 - creating, 2-4
 - deleting, 2-4
 - HELP command, 2-5
 - Message Window, 2-2, 2-5
 - sample use of, 2-8
 - status lines, 2-3
 - Status Window, 2-2
- Wmanager field in USER.TAB,
 - A-4
- Wndw, 7-97
- Wordlength, 7-29
- World, 7-74
- Write privilege, 7-75, B-2
- Write-protected disk, B-3
- Writing a volume label, 7-56,
 - 7-58

X

Y

Z

