

**PERKIN-ELMER**

# **OS/32 COPY**

**User Guide**

48-101 F00 R00

The information in this document is subject to change without notice and should not be construed as a commitment by the Perkin-Elmer Corporation. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place, Oceanport, New Jersey 07757

© 1983 by The Perkin-Elmer Corporation

Printed in the United States of America

## TABLE OF CONTENTS

PREFACE	v
CHAPTERS	
1 OS/32 COPY OVERVIEW	
1.1 GENERAL DESCRIPTION OF OS/32 COPY	1-1
2 GETTING STARTED	
2.1 INTRODUCTION	2-1
2.2 LOADING OS/32 COPY	2-1
2.3 STARTING OS/32 COPY	2-3
2.3.1 Command Device: COMMAND=fd1	2-4
2.3.2 Log Device: LOG=fd2	2-4
2.3.3 List Device: LIST=fd3	2-4
2.3.4 START Examples	2-4
2.4 COMMUNICATION WITH OS/32 COPY	2-5
2.4.1 Mode of Program Operation	2-5
2.4.1.1 Interactive Mode	2-5
2.4.1.2 Batch Mode	2-6
2.5 STOPPING OS/32 COPY	2-6
2.5.1 Terminating the Program	2-6
2.5.2 Terminating a Command Operation	2-6
2.6 PAUSING THE PROGRAM	2-7
3 GUIDE TO OS/32 COPY FEATURES	
3.1 INTRODUCTION	3-1
3.1.1 Overview of OS/32 COPY Commands	3-1
3.2 FUNDAMENTAL COPY OPERATIONS	3-2
3.2.1 Copying a Disk File to Another File or Device	3-3
3.2.2 Copying Unlabelled Magnetic Tapes	3-3
3.2.2.1 Copying to Unlabelled Magnetic Tapes	3-3
3.2.2.2 Copying from Unlabelled Magnetic Tapes	3-4

## CHAPTERS (Continued)

3.2.2.3	Copying Between Unlabelled Tapes	3-5
3.2.2.4	Default Values	3-5
3.2.2.5	More Examples	3-5
3.2.3	Copying Labelled Magnetic Tape	3-6
3.2.3.1	Copying from Labelled Tapes	3-6
3.2.3.2	Output Labelled Tapes	3-9
3.2.4	Copying To and From Other Devices	3-12
3.2.5	User-Specified Terminator	3-12
3.2.6	Handling of End of File (EOF), End of Volume (EOV), and End of Medium (EOM) Conditions	3-13
3.2.6.1	End of File (EOF)	3-13
3.2.6.2	End of Volume (EOV)	3-14
3.2.6.3	End of Medium (EOM)	3-15
3.3	MORE ELABORATE COPY OPERATIONS	3-16
3.3.1	Implicit Specification of Input and Output Files	3-16
3.3.2	Copying Multiple Files with a Single Command	3-17
3.3.3	Copying Part of a File	3-18
3.3.4	Automatic Allocation of Output Files	3-20
3.3.5	Copying Disk Files with Protection Keys	3-21
3.3.6	Copying Variable Length Records	3-22
3.4	VERIFYING	3-23
3.4.1	Verification Following a Copy Operation	3-23
3.4.2	Verifying Without a COPY Command	3-25
3.4.3	Use of the CONTINUE/NOCONT Option	3-26
3.5	DISPLAYING FILES	3-26
3.6	FILE POSITIONING	3-28
3.6.1	Forward File (FFILE) and Backward File (BFILE)	3-30
3.6.2	Record Positioning	3-30
3.6.3	REWIND Command	3-31
3.6.4	Writing Filemarks	3-31
3.7	OPTIONS, LIST, NOLOG, AND FILES COMMANDS	3-31
3.7.1	OPTIONS Command	3-32
3.7.1.1	Filemark Recognition	3-32
3.7.1.2	ASCII/BINARY Mode	3-32
3.7.1.3	Pseudo Filemark Recognition	3-33
3.7.1.4	Terminator Sequence	3-33
3.7.1.5	Use of CONTINUE/NOCONT	3-34
3.7.1.6	FILL Character Designation	3-34
3.7.2	LIST Command	3-35
3.7.3	LOG and NOLOG Commands	3-35
3.7.4	FILES Command	3-35
3.8	BATCH OPERATIONS	3-35
3.8.1	Default Assignments of List and Log Devices	3-36
3.8.2	Error and End of Volume (EOV) Handling	3-36

## CHAPTERS (Continued)

3.8.3	Handling of Nonempty Output Files in the COPY Command	3-36
3.8.4	SEND STOP Processing	3-36
4	OS/32 COPY REFERENCE GUIDE	
4.1	INTRODUCTION	4-1
4.2	FILE HANDLING	4-1
4.2.1	Current File Maintenance	4-1
4.2.2	Determination of File Attributes	4-1
4.2.3	Labelled Magnetic Tape	4-2
4.3	COMMAND SYNTAX	4-3
4.3.1	Continuation of Commands	4-3
4.3.2	Command Parameters	4-4
4.3.3	Implicit Specification of a File Descriptor (fd) Parameter	4-4
4.3.4	Comment Statements	4-5
4.4	OS/32 COPY COMMANDS	4-5
4.4.1	ALLOCATE Command	4-6
4.4.2	BFILE Command	4-9
4.4.3	BRECORD Command	4-10
4.4.4	CONTINUE Command	4-11
4.4.5	COPY Command	4-12
4.4.6	DELETE Command	4-16
4.4.7	DISPLAY Command	4-17
4.4.8	END Command	4-19
4.4.9	FFILE Command	4-20
4.4.10	FILES Command	4-22
4.4.11	FRECORD Command	4-25
4.4.12	HELP Command	4-26
4.4.13	INPUT Command	4-28
4.4.14	LIST Command	4-31
4.4.15	LOG Command	4-32
4.4.16	NOLOG Command	4-33
4.4.17	OPTIONS Command	4-34
4.4.18	OUTPUT Command	4-37
4.4.19	PAUSE Command	4-40
4.4.20	RENAME Command	4-41
4.4.21	REWIND Command	4-42
4.4.22	SEND STOP Command	4-43
4.4.23	VERIFY Command	4-44
4.4.24	WFILE Command	4-47
5	ERROR HANDLING AND MESSAGES	
5.1	ERROR HANDLING	5-1
5.1.1	Interactive Mode	5-1
5.1.2	Batch Mode	5-1
5.1.3	Return Code	5-1

## CHAPTERS (Continued)

5.2	MESSAGE SUMMARY	5-2
-----	-----------------	-----

## APPENDIXES

	A	COMMAND SUMMARY	A-1
	B	TASK ESTABLISHMENT AND SIZE INFORMATION	B-1
	C	TABLE OF DEVICE CHARACTERISTICS	C-1
	D	LABELLED TAPE FILE FORMAT SUMMARY	D-1
	E	THE TAPE LABELLER PROGRAM	E-1

## TABLES

2-1	START COMMAND DEFAULTS	2-4
4-1	INPUT FILE DEFAULTS	4-2

INDEX	Ind-1
-------	-------

## PREFACE

This manual describes the functions and features of the Perkin-Elmer OS/32 COPY program, part number 03-215, and replaces publication number 29-676. OS/32 COPY allows users to copy, verify, and display data from disk files or devices.

Users of OS/32 COPY should be familiar with the operation of Perkin-Elmer OS/32 and MTM. Users who are working with labelled tapes must also be familiar with the TAPE LABELLER program, part number 03-149. OS/32 COPY executes as a user task under OS/32 MT4.1 or higher.

Chapter 1 provides a general overview of the program's capabilities. Chapter 2 explains loading and starting OS/32 COPY, and mode of operation. Chapter 3 is intended for the new user and explains the fundamentals of copy, verify, and display operations. Chapter 4 provides complete information about the commands that are introduced in Chapter 3. Chapter 5 lists and explains the messages produced by OS/32 COPY. The appendixes provide: a summary of commands; information required for establishing OS/32 COPY as a task under OS/32; a table of device characteristics; tape label format information; TAPE LABELLER program information.

Revision 00 includes the following: addition of a HELP command; a list of recommended blocking factors for transferring data to magnetic tape; addition of comment statement support; addition of two new file types, extendable contiguous and nonbuffered indexed, to the ALLOCATE command; and addition of the VERR=Count option to the OPTION command.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.





## CHAPTER 1 OS/32 COPY OVERVIEW

### 1.1 GENERAL DESCRIPTION OF OS/32 COPY

The OS/32 COPY program provides OS/32 users with a comprehensive copy utility to use in an OS/32 or OS/32 Multi-Terminal Monitor (MTM) environment.

OS/32 COPY supports:

- media independent copy operations, providing data transfer between two files or devices of like or unlike characteristics,
- a verify operation providing the user with the ability to guarantee the integrity of data copied,
- standard IBM/ANSI magnetic tape labels, allowing either unblocked or blocked, labelled input or output tapes,
- blocked or unblocked magnetic tapes without tape labels,
- a basic set of commands which give the user the capability to specify simple directives; e.g., COPY FILEA,FILEB; where the program automatically allocates the output file with the appropriate characteristics, and
- an optional set of commands which provide added flexibility that allows the user to uniquely determine the characteristics of the file being allocated; e.g., OUTPUT FILEB, file characteristics...COPY FILEA,FILEB.

OS/32 COPY also provides commands which allow:

- portions of files to be copied,
- explicit allocation, renaming, and deletion of files,
- display of file data in hexadecimal format, and
- explicit file positioning.



## CHAPTER 2 GETTING STARTED

### 2.1 INTRODUCTION

This chapter discusses the steps required to make OS/32 COPY operational in your system: loading and starting the program are explained; interactive and batch operations are discussed.

If you are using a multi-terminal monitor (MTM) system, it is assumed that you are familiar with the SIGNON/SIGNOFF procedures required to use the system.

### 2.2 LOADING OS/32 COPY

To make OS/32 COPY operational, it must be established as a task prior to loading. See Appendix B. Once established, it must be loaded from an external storage device (e.g., disk, tape) into memory and can be subsequently started through the START command. In this discussion, assume that OS/32 COPY has been established as a task and is stored on disk under the filename COPY32.TSK.

Following are examples showing how to load OS/32 COPY in an OS/32 and an MTM environment. In the examples, the "\*", which is output by the system, indicates that the system is ready to accept a command from the user.

OS/32 Environment:

- In the OS environment you enter the LOAD command followed by the TASK command. The TASK command is required to set COPY32 as the currently selected task. For example:

```
*LOAD COPY32,,seg-size increment
*TASK COPY32
```

- In this example, both commas are required. See the OS/32 Operator Reference Manual for a detailed description of the LOAD command.

MTM Environment:

Only the LOAD command is required.

\*LOAD COPY32,seg-size increment

"seg-size increment" in the above example is an optional argument used to provide additional memory for COPY32 working space, thus generally improving response time of the program. Seg-size increment is a decimal integer argument which represents the amount of memory in kbytes to be used for the program's working space.

The record length and block size of the files being copied should be considered when determining the seg-size increment field.

The seg-size increment should be greater than or equal to:

$$\frac{2 * \text{recmax} + 1}{1024}$$

Where:

recmax is the largest input record, output record, or magnetic tape block size.

Example:

Suppose you intend to copy a magnetic tape containing 126 byte records blocked 10 records to a disk file. In this case:

input record size            1260 (i.e., block of 10 records, each 126 bytes long)

output record size         126

then recmax = input record size = 1260

$$\frac{2 * \text{recmax} + 1}{1024} = \frac{2 * 1260 + 1}{1024} = 3$$

The seg-size increment is 3, and the LOAD command would be:

```
OS/32:  *LOAD COPY32,,3
```

or

```
MTM:    *LOAD COPY32,3
```

When copying contiguous files, the minimum input or output buffer size is 256. However, copying proceeds more quickly if a large value for seg-size increment is specified. A value of 10 or 12 is recommended for optional performance.

### 2.3 STARTING OS/32 COPY

After OS/32 COPY has been loaded, the START command is used to begin execution of the program. The format of this command is the same for both the OS/32 and MTM environments.

The START command for OS/32 COPY has three optional arguments. The minimum acceptable abbreviations for each argument are indicated by underlining. The format, in the simplest case, is:

```
START
```

where defaults are used for the command, log and list devices are explained below.

Alternatively, the user may override any of the defaults by explicitly specifying the associated command, log, and/or list device:

```
START,COMMAND=fd1,LOG=fd2,LIST=fd3
```

Any or all of the arguments may be specified in any order.

When the simple form of the START command is used, OS/32 COPY automatically uses the device CON: (e.g., Model 1100, Carousel) as both the command and the list device.

The console device (often called the terminal) is the device at which you enter commands to OS/32, MTM, or OS/32 COPY.

If the use of the console device for both command input and output is not appropriate for your needs, one of the alternate START forms can be used to specify a different device.

### 2.3.1 Command Device: COMMAND=fd1

The parameter fd1 (file descriptor 1) specifies the command input device. You enter commands to OS/32 COPY from the command input device. If a command input device is not specified, commands are read from the console device, CON:.

### 2.3.2 Log Device: LOG=fd2

The parameter fd2 specifies the device for recording commands and responses. This device is called the log device or sometimes the command output device. If the log device is not specified, a default is used, as explained below.

The log device also receives all messages and error indications. The log device may be altered during program execution by using the LOG command. Logging may be suspended by entering the NOLOG command. If a disk file is specified as the log device, the file must already exist.

### 2.3.3 List Device: LIST=fd3

The parameter fd3 specifies the output device to be used by the VERIFY and DISPLAY commands. If the LIST device is not specified, a default is used, as explained below. The list device may be changed during program execution using the LIST command. If a disk file is specified as the list device, the file must already exist.

### 2.3.4 START Examples

Table 2-1 illustrates the START command defaults associated with the various START commands:

TABLE 2-1 START COMMAND DEFAULTS

COMMAND	COMMAND DEVICE	LIST DEVICE	LOG DEVICE
ST	CON:	CON:	no logging enabled
ST,LIST=PR:	CON:	PR:	no logging enabled
ST,C=TTY:,LO=PR1:	TTY:	TTY:	no logging enabled
ST,C=FILE.CMD	FILE.CMD	PR:	PR:

When the log and the list devices are the same, they share one logical unit, ensuring that displays are intermingled with the commands and messages in the order in which they occurred.

The default device assignments are explained below in the section describing mode of program operation.

## 2.4 COMMUNICATION WITH OS/32 COPY

Once OS/32 COPY has been started with the START command, OS/32 COPY displays its program identification, followed by the greater than (>) symbol. OS/32 COPY outputs the symbol whenever it expects you to enter a command:

```
OS/32 COPY Rn  
>
```

### Where:

Rn is the revision number of the OS/COPY program that you are using.

> is the prompt character indication that OS/32 COPY is ready to accept a command from the user.

When you enter a command in response to the prompt character, COPY interprets the command and performs the specified operation. When execution of the operation is completed, COPY outputs the prompt character again.

### 2.4.1 Mode of Program Operation

Commands to OS/32 COPY may be entered interactively or in batch mode. The mode is determined by the command device specified in the START command. If the command device is interactive, the program executes interactively; if the command device is non-interactive, or is a disk file, the program executes in batch mode.

The information in this manual is directed toward the user who is operating the program interactively. When program operation differs between batch and interactive mode, these differences are cited within the specific descriptions.

#### 2.4.1.1 Interactive Mode

In interactive mode, the user enters commands from an interactive device. Each command is terminated by depressing the RETURN key.

In this case, the list device defaults to the command device, allowing all command responses and error messages to be output to the command input device.

#### 2.4.1.2 Batch Mode

In batch mode, commands are input from a disk file or a non-interactive device. In this case, both LIST and LOG default to the device named PR:. All command input and all program messages are output to the log device. The list device is used for output produced by the DISPLAY and VERIFY commands.

### 2.5 STOPPING OS/32 COPY

This section describes terminating the OS/32 COPY program and OS/32 COPY commands.

#### 2.5.1 Terminating the Program

OS/32 COPY can be normally terminated by entering the command:

END

An abnormal termination of the program occurs only if invalid START arguments are encountered.

#### 2.5.2 Terminating a Command Operation

Certain OS/32 COPY commands can be terminated before normal completion using the OS/32 SEND STOP command. When running as a task under OS/32, two commands are necessary: TASK taskid and SEND STOP.

When running under MTM, only the SEND STOP command is required.

In interactive mode, the next command is read. If the next command entered is:

CONTINUE

the interrupted operation is resumed. If any other command is entered, the new command is processed.

In batch mode, the SEND STOP has no effect.



## 2.6 PAUSING THE PROGRAM

You may cause program execution to pause by entering the command:

**PAUSE**

You may now enter OS/32 or MTM commands. Execution of COPY may be resumed by entering:

**\*CONTINUE**



CHAPTER 3  
GUIDE TO OS/32 COPY FEATURES

3.1 INTRODUCTION

This chapter illustrates the features available to you as a user of OS/32 COPY and gives examples of the most commonly used operations. The examples assume you have loaded and started OS/32 COPY as described in Chapter 2.

3.1.1 Overview of OS/32 COPY Commands

- The basic data transfer operations are accomplished by:

COPY	Copies files and/or records
DISPLAY	Displays data
VERIFY	Compares files and/or records

- The basic COPY or VERIFY operations can be modified using:

OPTIONS	operands control writing filemark, binary format, etc.
---------	--

- Files can be explicitly described for use by COPY, DISPLAY, or VERIFY using:

INPUT	Operands give blocking factor, record size, tape label information, etc.
OUTPUT	

- Program flow is controlled by:

CONTINUE
END
PAUSE
SEND STOP

- The operational environment for OS/32 COPY is controlled by:

LIST	Designates a list device
LOG	Designates log device
NOLOG	Disables logging
START	Arguments determine command input device for batch or interactive operation, list and log devices.

- Information about current file assignments can be obtained by:

FILES

- OS/32 file management functions are provided by the inclusion of:

ALLOCATE  
DELETE  
RENAME

File Positioning Commands:

BRECORD  
FRECORD  
BFILE  
FFILE  
REWIND  
WFILE

- | ● Information about the use of all COPY commands can be obtained  
| by:

| HELP

### 3.2 FUNDAMENTAL COPY OPERATIONS

This section describes the basic compatibilities provided by the COPY command. Copying to and from disk files, labelled and unlabelled tapes, and other devices is described. End of file, end of volume, end of medium, and user specified terminators are discussed. Examples of several different COPY sequences are given.

### 3.2.1 Copying a Disk File to Another File or Device

To copy the contents of a disk file (fd1) to another disk file or device (fd2) you need only enter the command:

```
COPY fd1,fd2
```

For example, if the input file is FILE.CAL and the output device is CP:, you would enter:

```
COPY FILE.CAL,CP:
```

The file is then copied. When copying is completed, a message:

```
EOF ENCOUNTERED - n RECORD(s) COPIED  
COPY COMPLETE - m FILE(s) COPIED
```

is output, where n is the number of records and m is the number of files copied.

If you wish to copy to a disk file, you may explicitly allocate a new file or allow OS/32 COPY to automatically allocate the file.

Automatic allocation of output disk files and automatic deletion and reallocation of disk files are discussed in later sections.

### 3.2.2 Copying Unlabelled Magnetic Tapes

This section describes OS/32 COPY operations on unlabelled magnetic tapes. For a description of operations on labelled magnetic tapes, see Section 3.2.3.

#### 3.2.2.1 Copying to Unlabelled Magnetic Tapes

To copy the contents of a disk file to an unlabelled magnetic tape, you may enter a single command:

```
COPY fd1,fd2
```

Here fd1 is the file descriptor (fd) of the input disk file and fd2 is the fd of the magnetic tape drive containing the unlabelled magnetic tape. For example, you might enter:

```
COPY A.CAL,MAG1:
```

The data copied to the tape has the same record length as the input data, and is not blocked.

If you want the data on the output tape to be blocked, you can specify the record length and block size using the OUTPUT command. In this case, the appropriate OUTPUT command would be in the form:

```
OUTPUT fd,lrec/blk
```

Here fd is the fd of the magnetic tape drive, lrec is the record length in bytes of the data, and blk is the blocking factor, i.e., the number of records in a block.

| To decrease the possibility of errors in a data transfer to  
| magnetic tape, the recommended blocking factors are:

TAPE   RECORDING   DENSITY	RECOMMENDED BLOCKING FACTOR
800 bpi	12.5k
1600 bpi	25.0k
6250 bpi	100.0k

| Using larger blocking factors than those recommended gains little  
| additional storage space and results in an insignificant  
| reduction in processing time. However, it does increase the  
| probability of data transfer errors that result in verify errors.

| It is possible to avoid entering the fd of the magnetic tape  
| drive twice. This is explained in Section 4.3.3.

### 3.2.2.2 Copying from Unlabelled Magnetic Tapes

When the input data resides on an unlabelled magnetic tape, you may use the INPUT command to enter the record length and blocking factor of the data if the default values are not suited to your needs. The defaults are 80 byte records and a blocking factor of 1.

The INPUT command has the same form as the OUTPUT command:

```
INPUT fd,lrec/blk
```

Here fd is the fd of the magnetic tape drive, lrec is the record length, and blk is the blocking factor. The default for blk is 1.

Suppose you are copying data from an unlabelled tape mounted on MAG2: to a printer named PRT:. If the tape file consists of 120 byte records blocked 5 records per block, you can copy the data to the printer by entering:

```
INPUT MAG2:,120/5
COPY MAG2:,PRT:
```

### 3.2.2.3 Copying Between Unlabelled Tapes

If you want to copy between unlabelled tapes, you can use both the INPUT and the OUTPUT commands. To copy the data on MAG2: consisting of 120 byte records, blocked 5 records per block, to a tape mounted on MAG1: where the output data is unblocked, enter:

```
INPUT MAG2:,120/5
OUTPUT MAG1:,120
COPY MAG2:,MAG1:
```

### 3.2.2.4 Default Values

You can take advantage of default values for record length and block size to avoid entering the INPUT or OUTPUT commands. If you omit the INPUT command, OS/32 COPY assumes that data on an input tape is in the form of unblocked 80 byte records. If you omit the OUTPUT command, the record length of output data defaults to that of the input data; the blocksize is 1 unless the input is blocked. In this case, the blocking factor on output is the same as on input.

### 3.2.2.5 More Examples

#### Example:

Suppose you want to copy the contents of file A.PRT to magnetic tape on MAG1:. The input file consists of 120 byte records. The output data is to be unblocked 120 byte records; the command is:

```
COPY A.PRT:,MAG1:
```

In this case, no INPUT command is required because the record size associated with the disk file A.PRT is used. No OUTPUT command is required because the defaults are desired (i.e., a blocking factor of 1, a record size equal to the input data).

Suppose you are copying data from an unlabelled tape mounted on MAG2: to a printer named PRT:. If the tape file consists of 120 byte records blocked 5 records per block, you can copy the data to the printer by entering:

```
INPUT MAG2:,120/5
COPY MAG2:,PRT:
```

### 3.2.2.3 Copying Between Unlabelled Tapes

If you want to copy between unlabelled tapes, you can use both the INPUT and the OUTPUT commands. To copy the data on MAG2: consisting of 120 byte records, blocked 5 records per block, to a tape mounted on MAG1: where the output data is unblocked, enter:

```
INPUT MAG2:,120/5
OUTPUT MAG1:,120
COPY MAG2:,MAG1:
```

### 3.2.2.4 Default Values

You can take advantage of default values for record length and block size to avoid entering the INPUT or OUTPUT commands. If you omit the INPUT command, OS/32 COPY assumes that data on an input tape is in the form of unblocked 80 byte records. If you omit the OUTPUT command, the record length of output data defaults to that of the input data; the blocksize is 1 unless the input is blocked. In this case, the blocking factor on output is the same as on input.

### 3.2.2.5 More Examples

#### Example:

Suppose you want to copy the contents of file A.PRT to magnetic tape on MAG1:. The input file consists of 120 byte records. The output data is to be unblocked 120 byte records; the command is:

```
COPY A.PRT:,MAG1:
```

In this case, no INPUT command is required because the record size associated with the disk file A.PRT is used. No OUTPUT command is required because the defaults are desired (i.e., a blocking factor of 1, a record size equal to the input data).



### Example:

Here you want to copy the input file on MAG1: to the tape on MAG2:. The data is blocked eight 126 byte records to a block and the output data is to be blocked the same way. The command is:

```
INPUT MAG1:,126/8
COPY MAG1:,MAG2:
```

In this case, an INPUT command is required to specify the desired record and block sizes. No OUTPUT command is required because the defaults are desired (i.e., the record length and block size of the input data).

### 3.2.3 Copying Labelled Magnetic Tape

Using OS/32 COPY you can copy to or from labelled tapes. Tape labels may be in either IBM or ANSI format. OS/32 COPY copies single file multi-volume tapes. It does not support multi-file labelled tapes. For a description of labelled tape formats, see Appendix D.

To use labelled tapes, use the INPUT or OUTPUT command to enter necessary label and tape handling information followed by a COPY, VERIFY, or DISPLAY command.

Output labelled tapes must first be prepared by using the Tape Labeller program. See Appendix E for the operation of the Tape Labeller program.

#### 3.2.3.1 Copying from Labelled Tapes

In order to copy data from a labelled tape, use the INPUT command to enter information about the tape file.

You must enter the fd of the tape drive on which the tape is mounted and the keyword LABEL.

If you enter a record length and blocking factor, these parameters are compared to those written on the tape label. A mismatch causes the COPY, VERIFY, or DISPLAY operation to be rejected.

Similarly, you can enter a file identifier and a set of volume identifiers (ids). If you do, they are checked against the label and a mismatch causes the COPY, VERIFY, or DISPLAY operation to be rejected. If you omit one or both of these parameters, the corresponding information on the tape label is ignored.

**Format:**

```
INPUT fd [lrec [blk]] [LABEL] [FNAME=fn]
         [VOLID=(n1,...)] [PAD=xx]
```

**Where:**

fd is the fd of the tape drive.

lrec is the record length.

blk is the blocking factor.

LABEL signifies that the input tape is labelled.

fn is the file identifier of the tape file, from 1- to 17-character alphanumeric string.

n1 is from 1- to 6-alphanumeric characters. If more than one volume id is listed, the ids are separated by commas and enclosed in parentheses. The parentheses may be omitted for a single tape volume. The input tape volumes must be mounted in the order indicated by the list of volume ids.

xx are two hexadecimal digits used to specify the pad character. A record of all pad characters will be ignored. The default is FF.

The fd of the tape drive must be the first parameter entered. The other parameters may be entered in any order. If you must use more than one line to enter all the data, you may do so. A line whose last nonblank character is a comma is assumed to be continued on the next line. You can enter as many as 20 volume ids in addition to other parameters by using continuation lines.

**Example:**

Suppose you want to copy a labelled tape file residing on a single tape volume to a disk file. The record length of the input file id is 256 and the file id LABDATA. You do not want to check the volume id. The following sequence of commands can be used:

```
ALLOCATE LAB.DTA, IN, 256
INPUT MAG1:, LABEL, FNAME=LABDATA
COPY MAG1:, LAB.DTA
REWIND OUTPUT
VERIFY
```

Note that the labelled tape is automatically rewound.

The last two commands are only required if you wish to verify that the data was copied correctly. The ALLOCATE command could have been omitted and COPY would have automatically allocated an indexed file with the proper record length. See below for details on automatic allocation of output files.

**Example:**

Suppose you want to copy a labelled tape file to cards using the card punch CP:. The tape file has no file name and you do not know the volume id, but you do want to check that the input data consists of 80 byte records blocked 15 to a block.

Enter:

```
INPUT MAG3: ,80/15,LABEL
COPY MAG3: ,CP:
VERIFY MAG3: ,CR:
```

Here, again, the VERIFY is optional. The device CR: is a card reader which is loaded with the cards punched during the COPY operation.

**Example:**

Data file BIGJOB resides on three tape volumes whose volume ids are 606100, 606101, 606102. You want to copy the data to an unlabelled tape. The record length of the data is 120 bytes; you want to block the output data 50 records per block. Enter the following commands:

```
INPUT MAG1: ,LABEL,FNAME=BIGJOB,
VOLID=(606100,606101,606102)
OUTPUT MAG2: ,120/50
COPY
```

Note that no arguments are used with the COPY command since both the input and output files are previously established. When the first volume has been copied, a message is output:

```
MOUNT NEXT VOLUME 606102 ON INPUT FOR MAG1:
```

The input tape is automatically rewound. You should now mount the next volume and enter the CONTINUE command to resume copying.

If OS/32 COPY is executing in batch mode, it pauses after the above messages are output. When the operator enters CONTINUE, the copy operation is resumed.

The same procedure is followed when the second volume has been copied.

### 3.2.3.2 Output Labelled Tapes

The Tape Labeller program must be used to initially create a labelled tape before it can be used for output. The operation of this program is described in Appendix E.

Writing labelled tapes is similar to reading them; but there are several more parameters to consider.

The format of the OUTPUT command used with labelled tapes is as follows:

Format:

```
OUTPUT fd [ ,lrec [ /blk ] ] [ ,LABEL ] [ ,FNAME=fn ]  
          [ ,EXPIRY=yyddd ] [ ,VOLID=(n1, . . . ) ] [ ,PAD=xx ]
```

Where:

- |       |  |
|-------|--|
| fd    | is the fd of the tape drive on which the tape is mounted.  |
| LABEL | signifies that the output tape is labelled.  |
| lrec  | is the record length; if omitted the default is the same as input record length.   |
| blk   | is the blocking factor; if omitted, the default is 1, unless the input is a tape, in which case, it defaults to the input blocking factor. |
| fn    | is the file id, a 1- to 17-character alphanumeric sequence. If omitted, the filename field in the tape label is filled with blanks.        |

yyddd is the expiration date expressed as a two-digit year and a three-digit day. If the expiry date is supplied, it is compared to the date in the expiry date field in the label. If the specified date is less than the tapes expiry date, a message is logged and the command is terminated. Otherwise, the new expiry date is written on the label, and the current date is inserted in the creation date field.

xx are two hexadecimal digits used to pad the last block of the output file if the number of records in the file is not a multiple of the blocking factor. The default is FF.

nl is from 1- to 6-alphanumeric characters specifying the volume id. If there is a single volume id, the parentheses may be omitted. If more than one volume id is given, the list should be separated by commas and enclosed in parentheses. More than one command line can be used to enter the list of volids; if the last nonblank character on a line is a comma, the next line is treated as a continuation of the previous one.

**Example:**

Suppose you want to copy a disk file, S.FTN, to a labelled tape. The input record length is 80. The output tape has volume id 789789.

Enter:

```
OUTPUT MAG2:,80/10,LABEL,FNAME=STAPE.FTN,  
VOLID=789789  
COPY S.FTN,MAG2:
```

The output data is written with a blocking factor of 10. The expiry date is not checked, and the pad character is hexadecimal FF.

**Example:**

Suppose you want to copy a card file from card reader CR: to a labelled tape. This new data should remain valid until January 1, 1989. Suppose you want to block the records 10 records per block and pad the last block with zeros. No filename is to be given to this tape file.

Enter:

```
OUTPUT MAG1:,80/10,LABEL,EXPIRY=89001,PAD=00
COPY CR:,MAG1:
(cards are repositioned in card reader)
VERIFY
```

The last command is only required if verification is desired. If the date in the expiry date field of the label is later than the specified date in the OUTPUT command, a message is output and the command terminates.

Example:

Suppose you want to copy the data on a 4 volume tape file to another tape file.

Enter:

```
INPUT MAG1:,256/10,LABEL,FNAME=BIGDATA,
VOLID=(111111,111112,111113,111114)
OUTPUT MAG2:,LABEL,FNAME=BIGDATA2,
VOLID=(FRED1,FRED2,FRED3,FRED4)
COPY
```

Here the output tape uses the record length and blocking factor of the input data. The expiry date of the output tape is ignored and the pad character is hexadecimal FF. When it is necessary to change either an input or an output volume, a message is output:

```
MOUNT NEXT VOLUME xxxxxx ON { INPUT } FOR fd
                             { OUTPUT }
```

The tape is automatically rewound. When you have mounted a new tape (volume xxxxxx) on device fd, enter:

CONTINUE

### 3.2.4 Copying To and From Other Devices

If you are copying to or from devices other than magnetic tape, you need only enter the COPY command with input and output file descriptors as arguments:

```
COPY fd1,fd2
```

Here fd1 is the input fd and fd2 is the output fd. For example, to copy from card reader CR: to printer PRT:, you would enter:

```
COPY CR:,PRT:
```

OS/32 COPY determines the record lengths of the input and output devices.

### 3.2.5 User-Specified Terminator

In order to terminate the COPY operation entered above, you would end the card deck with a card containing a terminator sequence. By default, this sequence is /\* in columns 1 and 2 followed by all blanks on the rest of the card. This card is not copied. You can choose your own terminator sequence by using the TERMINATOR parameter of the OPTIONS command. For example, to specify the terminator characters EOF, you enter:

```
OPTIONS TERMINATOR=EOF
```

The sequence of terminator characters is 1- to 4-characters long with no imbedded blanks or commas. The rest of the record must consist of blanks. You can choose to have no terminator sequence at all by entering:

```
OPTIONS NOTERM
```

This option remains in effect until you enter another OPTIONS command with the TERMINATOR parameter. You might choose the NOTERM option if you were copying data from tape or disk and you wanted all records to be copied until the physical end of file was reached.

The TERMINATOR sequence is only checked when copying in ASCII mode. The mode can be set in the OPTIONS command. The default is ASCII mode.

To copy a binary file such as a task image, which should only be terminated by a filemark, you should copy in binary mode. To set binary mode, enter:

#### OPTIONS BINARY

### 3.2.6 Handling of End of File (EOF), End of Volume (EOV), and End of Medium (EOM) Conditions

In most cases, execution of the COPY command continues until an end of file (EOF) is encountered on the input file. At that time a message is logged indicating the number of records that have been copied. Then the next command is read from the command device. However, conditions such as end of volume (EOV) and end of medium (EOM) can interrupt or terminate the COPY command.

#### 3.2.6.1 End of File (EOF)

The EOF condition can arise as follows, depending upon the file or device characteristics:

DEVICE/FILE TYPE	EOF CONDITION
ASCII devices (e.g., card reader, printer, terminal)	ASCII terminator sequence
Unlabelled Magnetic Tape	Filemark
Labelled Magnetic Tape	EOF label
Indexed, non-buffered indexed, and extendable contiguous files	after processing of last logical record
Contiguous file (PSEUDO FILEMARK OPTION disabled)	after processing of last physical record
Contiguous file (PSEUDO FILEMARK OPTION enabled)	after processing of pseudo filemark or last physical record.

An EOM condition on contiguous files is treated as an EOF. When encountering an EOF condition while copying a file, OS/32 COPY issues the message:

```
EOF ENCOUNTERED  -n RECORDS COPIED
END OF MEDIUM
```



The pseudo filemark option (PSFM) determines whether filemarks are recognized on contiguous files. The PSFM option is enabled or disabled using the OPTIONS command:

```
OPTIONS PSFM          (to enable the pseudo filemark option)
```

```
OPTIONS NOPSFM       (to disable the pseudo filemark option)
```

The default is PSFM.

You can take advantage of the filemark option (FM/NOFM) to specify that an EOF indicator is to be copied. When the FM option is set, which is by default, an EOF indicator is written on the output file whenever EOF is reached on input (only if the output file is a disk file that supports filemarks or magnetic tape). If the input file is a contiguous file, end of medium on input is treated as EOF.

The NOFM option is set by entering:

```
OPTIONS NOFM
```

When NOFM is set, no filemark is written on the output file when EOF is encountered on input. However, you may enter the WFILE command to write a filemark.

The NOFM option is useful when you want to concatenate two data files. For example, to concatenate A1.DTA and A2.DTA onto a magnetic tape mounted on MAG2:, you would enter the following commands:

```
OPTIONS NOFM
COPY A1.DTA,MAG2:
OPTIONS FM
COPY A2.DTA,MAG2:
```

The purpose of the second OPTIONS command is to cause a filemark to be written at the end of the data copied as a result of the second COPY command. The WFILE command may also be used to accomplish this.

### 3.2.6.2 End of Volume (EOV)

On input devices other than labelled tapes, end of volume occurs when two successive end of file conditions occur on the input file. The COPY, VERIFY, or DISPLAY operation in progress is terminated.

On labelled tapes, EOVS is defined on input by an EOVS label.

When EOVS occurs on input from a labelled tape, a message is logged:

```
MOUNT NEXT VOLUME xxxxxx ON INPUT FOR fd
```

The volume id xxxxxx names the next volume to be read and fd is the fd of the input device. The tape is automatically rewound.

The COPY, VERIFY or DISPLAY operation in progress can be continued by entering the CONTINUE command or terminated by entering another command. If OS/32 COPY is in batch mode, it pauses after issuing the EOVS message and continues the COPY, VERIFY or DISPLAY operation in progress when CONTINUE is entered.

On output, EOVS occurs only on magnetic tapes. When the end of a reel of tape is detected, a message is logged. For labelled tapes, the message is:

```
MOUNT NEXT VOLUME xxxxxx ON OUTPUT FOR fd
```

The volume id xxxxxx names the next volume to be written and fd is the fd of the output device. An EOVS label is written on the tape and the tape is rewound.

For unlabelled tapes, one filemark is written followed by an EOVS id and one filemark, and the following messages are logged:

```
END OF VOLUME  -n RECORDS  { COPIED  
                           { VERIFIED  
                           { DISPLAYED  
MOUNT NEXT VOLUME FOR fd
```

where n is the number of records processed up to the point of EOVS.

The COPY, VERIFY or DISPLAY operation in progress may be continued as described above.

### 3.2.6.3 End of Medium (EOM)

EOM can occur on input from a contiguous file or on output to either a contiguous or an indexed file. When reading from a magnetic tape, the end of tape marker is ignored and OS/32 COPY continues to read until it encounters an EOVS condition. When writing to a magnetic tape, the end of tape marker is interpreted as EOM.

When OS/32 COPY encounters an EOM condition while copying from a contiguous file, it issues a message:

```
END OF MEDIUM - n RECORD(S) { COPIED
                             VERIFIED
                             DISPLAYED }
```

where n is the number of records processed to the point of EOM. OS/32 COPY then treats the EOM condition as an EOF. The number of records is output only if n is greater than 0.

When OS/32 COPY encounters an EOM condition on output to a disk file, it treats the condition as an input/output (I/O) error.

### 3.3 MORE ELABORATE COPY OPERATIONS

This section explains: implicit specification of input and output files; copying multiple files or part of a file; automatic allocation of output files; using protection keys for disk files; and variable length record processing.

#### 3.3.1 Implicit Specification of Input and Output Files

When you enter an fd by using the INPUT command or by specifying the fd as the first parameter of the COPY, DISPLAY, or VERIFY commands, you establish this file as the current input file. Until another fd is entered as the first parameter in one of these commands, this file remains the input file.

Similarly, by entering an fd using the OUTPUT command, or as second parameter of the COPY or VERIFY command, you establish a file as the current output file.

Once a file has been established as the input file or output file, you can enter it implicitly in the commands INPUT, OUTPUT, COPY, VERIFY, and DISPLAY by entering an asterisk (\*).

For example, suppose you want to copy a tape mounted on MAG1: to printer PRT:. The data on the tape is in the form of 120 byte records blocked 15 records per block. You would enter:

```
INPUT MAG1:,120/15
COPY *,PRT:
```

The first command establishes MAG1: as the input file as well as describing the input data. The asterisk in the COPY command implicitly refers to MAG1:. By entering PRT: in the COPY command, you have established PRT: as the output file. You may use an asterisk to implicitly designate this output file in subsequent commands.

For example, suppose there is a second data file on the tape mounted on MAG1: which consists of 80 byte records blocked 10 to a block. Since both the input and output files have already been established, you can copy the data to the printer by entering:

```
INPUT *,80/10
COPY *,*
```

In case both input and output files are designated by asterisks and no other arguments are to be entered, you can omit all arguments. In this case, the above sequence could be:

```
INPUT *,80/10
COPY
```

The use of asterisks to implicitly designate files is optional. You may enter fds whenever they are appropriate. OS/32 COPY determines whether a new input or output file is established and does not reassign the input or output logical unit (lu) if the current file is being referenced.

If you want to use the current output file as an input file, or the current input file as the output file, it may be necessary to precede the COPY, VERIFY, or DISPLAY command with an INPUT or OUTPUT command, whichever is appropriate. The INPUT or OUTPUT command is necessary if the specified file has attributes other than the default attributes. For example:

```
OUTPUT DATA.R1,256
COPY MAG1:,DATA.R1
INPUT DATA.R1,256
DISPLAY DATA.R1
```

allows you to display the contents DATA.R1.

### 3.3.2 Copying Multiple Files with a Single Command

An unlabelled magnetic tape may contain any number of data files separated by single filemarks. The end of a tape volume is signified by a filemark, an EOVI indicator, and a filemark.

By using the NFILES parameter of the COPY file or VERIFY command, you can copy or verify several files on a tape, or all files on a tape volume. One restriction on copying multiple files is that they must all have the same record length and blocking factor.

The NFILES parameter may also be used to copy several files contained in a contiguous file and separated by pseudo filemarks, or to copy a number of ASCII files on any medium where a terminator sequence is specified as a filemark.

The formats of the COPY and VERIFY commands with the NFILES parameter are:

```
COPY fd1,fd2,NFILES=n
VERIFY fd1,fd2,NFILES=n
```

n can be a decimal integer or the keyword ALL.

When n is an integer, copying or verifying continues until the specified number of files has been processed, or until an EOF is encountered, whichever comes first. If ALL is entered, copying or verifying continues until EOF is encountered.

Example:

All files on a tape mounted on MAG1: are to be copied to a tape mounted on MAG2:. All the files consist of 256 byte records blocked 5 records to a block. The output data is to be unblocked.

Enter:

```
INPUT MAG1:,256/5
OUTPUT MAG2:,256
COPY *,*,NFILES=ALL
```

The asterisks are required because of the presence of the NFILES parameter. The use of the asterisks is explained above.

In order to verify the files copied above, enter:

```
REWIND INPUT
REWIND OUTPUT
VERIFY *,*,NFILES=ALL
```

### 3.3.3 Copying Part of a File

By using the NRECS parameter along with the file positioning commands, you may copy or verify a portion of a file. The NRECS parameter may not be used with labelled tapes.

The formats of the COPY and VERIFY commands with the NRECS parameter are:

```
COPY fd1,fd2,NRECS=n
VERIFY fd1,fd2,NRECS=n
```

n is a decimal integer which indicates the number of records to be copied. Copying or verifying continues until the specified number of records has been processed or until an EOF is encountered, whichever comes first.

**Example:**

To copy the first 20 records of file B.CAL to PR:, enter:

```
COPY B.CAL,PR:,NRECS=20
```

**Example:**

Suppose you want to copy and verify records 101 through 200 from a tape mounted on MAG1: to a tape mounted on MAG2:. The record length of the data is 80 and the blocking factor is 5. You enter:

```
INPUT MAG1:,80/5
OUTPUT MAG2:,80/5
FRECORD INPUT,20
COPY*,*,NRECS=100
BRECORD INPUT,20
REWIND OUTPUT
VERIFY *,*,NRECS=100
```

The FRECORD command skips the first 20 blocks or 100 records thus positioning the input tape at record 101. Similarly, the BRECORD command positions the input tape at record 101.

Note that the arguments of the FRECORD and BRECORD commands denote physical records or blocks, whereas the NRECS parameter denotes logical records.

### 3.3.4 Automatic Allocation of Output Files

The COPY command can be used to allocate output files on a disk. If when using the COPY command a nonexistent output file is entered, OS/32 COPY will automatically perform the allocation and copy for you. An output file that has been automatically allocated by COPY will have the same attributes as the input file if copying is done from disk to disk. However, if copying from magnetic tape to disk, and the input file is a contiguous file, you must allocate the output file and specify contiguous. If you do not, OS/32 COPY will assume an indexed file.

Indexed files allocated by OS/32 COPY have index and data block sizes of 1.

#### Examples:

##### Disk-to-Disk:

If you want to copy a file named FILEA.TSK to an output file of the same name on volume M301, enter:

```
COPY M300:FILEA.TSK,M301:FILEA.TSK
```

In the above example, the output file will be the same file type (indexed or contiguous) as the input file.

##### Tape-to-Disk:

For automatic allocation of an indexed output file with a record size of 256, enter:

```
INPUT MAG1:,256
COPY *,FILEA
```

In the above example, COPY32 will first allocate FILEA as an indexed file on disk and then copy the contents of MAG1: to FILEA.

For allocation of a contiguous output file with a record size of 256, enter:

```
INPUT MAG1:,256
ALLOCATE FILEA, CO,100
COPY *,FILEA
```

In the above example, the ALLOCATE command specifies that a contiguous (CO) file was input. If the ALLOCATE command had been omitted, OS/32 COPY would allocate an indexed file with the filename FILEA.

You may also cause an existing output disk file to be deleted and reallocated. You do this by entering the DELETE keyword after the output fd. The current output file is deleted and a new output file is allocated following the same rules as if the file had not previously existed.

#### Example:

Delete the file A.OBJ and reallocate before copying 126 byte records from tape.

Enter:

```
INPUT MAG1:,126
COPY *,A.OBJ,DELETE
```

The file is reallocated as an indexed file with index and data block size of 1 and 126 byte records.

If DELETE is not entered, and you name as the output file an existing contiguous file or an existing indexed file that contains one or more logical records, a message is logged:

```
NON-EMPTY OUTPUT FILE
```

In interactive mode, you must enter CONTINUE to have the COPY instruction executed. In batch mode the COPY operation is executed without requiring CONTINUE to be entered.

### 3.3.5 Copying Disk Files with Protection Keys

If you want to COPY, VERIFY, or DISPLAY disk files that have nonzero protection keys, you must use the INPUT and OUTPUT commands to enter the keys so that the files can be assigned. The format of the INPUT and OUTPUT commands in this case is:

```
INPUT fd,KEYS=xxxx
OUTPUT fd,KEYS=xxxx
```

Here xxxx is the four-hexadecimal digit protection key of the file, fd. The file must exist when these commands are entered.



**Example:**

Copy and verify the file IMPORT.DTA to the file IMPORT2.DTA. The keys of both files are E34A.

Enter:

```
INPUT IMPORT.DTA,KEYS=E34A
OUTPUT IMPORT2.DTA,KEYS=E34A
COPY
```

### 3.3.6 Copying Variable Length Records

It is possible to copy magnetic or paper tapes having variable length records. To do this, you enter the INPUT command with the fd of the tape drive, the keyword VARIABLE, and the record size of the largest record on the tapes.

For example, suppose you want to copy a tape mounted on MAG1:, where the largest record may be 256 bytes, to a disk file, EXTM.F00, you enter:

```
INPUT  MAG1:,256,VARIABLE
COPY   MAG1:,EXTM.F00
```

The records are read from MAG1: and written to EXTM.F00; each record on EXTM.F00 is 256 bytes long. Each output record is padded with the current "fill" character. The fill character is initially X'20' and can be changed at any time by entering:

```
OPTIONS  FILL=xx
```

where xx is the hexadecimal representation of the desired fill character.

You may also copy variable length records to a device which supports variable length records. Suppose you want to copy records from a magnetic tape, where the largest record is 80 bytes, to paper tape, you enter:

```
INPUT  MAG1:,80,VARIABLE
OUTPUT PTRP:,80,VARIABLE
COPY
```

Each output record is the same length as the corresponding input record.

The VARIABLE keyword may also be used when copying to or from interactive devices. To copy from the interactive device CRT: to a magnetic tape mounted on MG3: and have each record padded with asterisks, you enter:

```
OPTIONS FILL= 2A
INPUT  CRT: ,VARIABLE
OUTPUT MG3: ,72
COPY
```

You would enter lines of varying length; each line is copied to a 72 byte record on MG3:. The magnetic tape records are padded out with the fill character, \* , as specified in the OPTIONS command. If you had not entered the OUTPUT command, each record output to MG3: would be the same length as the corresponding input record. Had you entered the OUTPUT command without a specified record length or with the VARIABLE keyword, each output record would be the same length as the corresponding input record.

### 3.4 VERIFYING

This section explains the use of the VERIFY command, its use with or without a previously issued COPY command, and also the effect of the CONTINUE option.

#### 3.4.1 Verification Following a Copy Operation

After copying a file, you may want to verify that the data was copied correctly. The VERIFY command allows you to do this. It compares the contents of two files record by record and halts if a pair of records is found that do not match. By using the CONTINUE option you may also list all pairs of records which differ in the two files.

Before you can verify that the data you have copied is correct, you must reposition the input and output files.

For disk files, use the REWIND command to position the files to the beginning. You may also use REWIND to position to the first file of data on the magnetic tape. For data on magnetic tape which begins just after a filemark, you must use both the BFILE and FFILE commands. Labelled tapes are automatically rewound. For multi-volume verification, the volumes must be remounted.

The following examples show how to verify files following a COPY. Notice that the fds, once entered in the COPY command, do not have to be reentered.

**Example:**

Copy and verify disk file A.CAL with B.CAL.

```
COPY A.CAL,B.CAL
REWIND INPUT
REWIND OUTPUT
VERIFY
```

Note that the output file is used as a second input file for the VERIFY command.

**Example:**

Copy and verify disk file A.CAL with a tape mounted on MAG2:. The tape is at the load point at the time the first command is entered.

```
COPY A.CAL,MAG2:
REWIND INPUT
REWIND OUTPUT
VERIFY
```

**Example:**

Copy and verify file C.CAL with the same tape used above. The tape is positioned just beyond the filemark ending the first file copied. After the COPY, the tape is backspaced two filemarks, followed by forwarding one filemark in order to position the tape to perform the verify operation.

```
COPY C.CAL,*
REWIND INPUT
BFILE OUTPUT,2
FFILE OUTPUT
VERIFY
```

If a pair of records does not match when a VERIFY operation is in progress, the first byte found which does not match is listed along with the associated records formatted in hexadecimal and ASCII. This data is output to the list device. An EOF on one file that does not match the other file is also reported.

### 3.4.2 Verifying Without a COPY Command

To verify two files which have not just been copied, you must explicitly name the files in the VERIFY command or in the INPUT and OUTPUT commands. In the case of disk files, it is only necessary to issue a VERIFY command:

```
VERIFY A.FTN,A1.FTN
```

If the input file is on magnetic tape, use the INPUT command preceding the VERIFY command unless the default of unblocked 80 byte records is applicable.

#### Example:

The input data on MAG1: is in the form of 126 byte records blocked 10 to a block. The output file is a disk file:

```
INPUT MAG1:,126/10
VERIFY *,FILR.OBJ
```

If the output file is on magnetic tape, you may need to use the OUTPUT command unless: the input data is also on tape and the record length and blocking factor of input and output data are the same; or, the output data has the same record length as the input and is unblocked.

#### Example:

The input data is in file FILR.OBJ, while the output data on MAG2: consists of 126 byte records blocked 10 per block:

```
OUTPUT MAG2:,126/10
VERIFY FILR.OBJ,*
```

#### Example:

The output data is on MAG1: and consists of unblocked 126 byte records, the same as the input file.

```
VERIFY FILR.OBJ,MAG1:
```

### 3.4.3 Use of the CONTINUE/NOCONT Option

The CONTINUE option in the OPTIONS command controls the VERIFY command operation. You can use the VERIFY command with the CONTINUE option in effect to list all pairs of input and output records which differ.

When program execution begins, the NOCONT option is in effect. Under this option, execution of the VERIFY command terminates if a pair of records is found that do not match. You may, however, enter the command CONTINUE if you wish verification to continue with the next pair of records. If you want verification to continue until at least one of the files reaches EOF, you should set the CONTINUE option.

Enter:

```
OPTIONS CONTINUE
```

Since verification continues until EOF, it is possible that a large number of records will not match and a sizable amount of data is written to the list file.

### 3.5 DISPLAYING FILES

The DISPLAY command may be used to produce a hexadecimal display of a file. The display is output to the list device. By default, the list device is the command device if you are entering commands interactively, or PR: if commands are being entered from a noninteractive device or file.

You can change the list device by entering the LIST command:

```
LIST fd
```

If you want to produce a hexadecimal display of a file fd, you would enter:

```
DISPLAY fd
```

**Example:**

To display the contents of a disk file A.OBJ, enter:

```
DISPLAY A.OBJ
```

If you want to DISPLAY a file residing on magnetic tape or a disk file which has nonzero protection keys, you must use the INPUT command.

**Example:**

Suppose you want to display the contents of a file of data on magnetic tape. The tape volume is on MAG1; the data is in the form of unblocked 120 byte records.

Enter:

```
INPUT MAG1:,120/1
DISPLAY MAG1:
```

**Example:**

To display the contents of a file on a labelled tape with file id DATA.DMP, enter:

```
INPUT MAG1:,LABEL,FNAME=DATA.DMP
DISPLAY MAG1:
```

If you are entering the DISPLAY command after having entered a COPY, VERIFY, or another DISPLAY command, and you want to display the file which was the input file in the previous command, you can enter:

```
REWIND INPUT
DISPLAY *
```

Furthermore, if you enter DISPLAY with a file descriptor as argument, that file becomes the new input file. Subsequent COPY, VERIFY, and DISPLAY commands can use an asterisk to implicitly designate that file.

The INPUT command must be used in order to display a file or device which has other than the default keys, record length, or blocking factor, and is not the current input file.

The NFILES or NRECS parameter may be used with the DISPLAY command to display multiple files or a particular number of records in a file. Only one of these parameters should be entered in a command, and neither parameter can be used with labelled tapes. The format of the DISPLAY command used with these parameters is:

Format:

$$\text{DISPLAY fd} \left[ \begin{array}{l} \{ \text{NFILES}=\text{n} \} \\ \{ \text{NRECS}=\text{n} \} \end{array} \right]$$

Example:

Suppose an unlabelled tape mounted on MAG1: contains 3 files separated by filemarks. The data in each file consists of 80 byte records blocked 5 records to a block. To produce a formatted dump of the files, enter:

```
INPUT MAG1:,80/5
DISPLAY *,NFILES=3
```

### 3.6 FILE POSITIONING

A set of commands is provided which are similar to the OS/32 commands for positioning files and writing filemarks. These commands cannot be used with labelled tapes, except REWIND.

The file positioning commands can only be executed for files that support the operation indicated by the command.

A table of device attributes is in Appendix C.

The commands and their formats are:

<code>BFILE { INPUT } [n]</code> <code>          { OUTPUT }</code>	Backspace over a filemark
<code>BRECORD { INPUT } [n]</code> <code>          { OUTPUT }</code>	Backspace a record
<code>FFILE { INPUT } [ , { n } ]</code> <code>      { OUTPUT } [ { ALL } ]</code>	Forward space over a filemark
<code>FRECORD { INPUT } [n]</code> <code>          { OUTPUT }</code>	Forward space a record
<code>REWIND { INPUT }</code> <code>       { OUTPUT }</code>	Position to beginning of file/device
<code>WFILE { LIST } [n]</code> <code>      { OUTPUT }</code>	Write a filemark

These commands can only be applied to the current input, output, and list files. The first file named in a COPY, VERIFY, DISPLAY, or INPUT command is the current input file. The second file named in a COPY or VERIFY command or the file named in an OUTPUT command is the current output file.

If you wish to position a file before using it in a COPY, DISPLAY, or VERIFY command, you must use the INPUT or OUTPUT command before issuing one of the file positioning commands.

**Example:**

Suppose you want to write a filemark on a tape mounted on MAG1: before copying to the tape; you would enter:

```
OUTPUT MAG1:
WFILE OUTPUT
COPY X.Y,MAG1:
```



### 3.6.1 Forward File (FFILE) and Backward File (BFILE)

These commands skip over one or more filemarks. FFILE skips forward while BFILE skips backward. Filemarks on magnetic tape or pseudo filemarks on contiguous files are considered to be filemarks by these commands. However, pseudo filemarks are ignored unless the PSFM option is set. A terminator string, if in effect, is ignored.

The parameter *n*, a decimal integer, specifies the number of filemarks to be skipped. If omitted, the default is 1.

#### Example:

To skip over 3 filemarks on the input file and skip back over one filemark on the output file, enter:

```
FFILE INPUT,3  
BFILE OUTPUT
```

Command execution of FFILE terminates when the indicated number of filemarks have been skipped or when two consecutive filemarks, an EOF, or an EOVS condition is encountered. In order to skip forward to the EOVS, enter ALL instead of a decimal integer.

BFILE terminates when the indicated number of filemarks has been skipped or when beginning of medium is encountered.

After the execution of FFILE, the file is positioned just beyond the last filemark skipped. After the execution of BFILE, the file is positioned just before the last filemark skipped.

This means that FFILE positions to the end of a file if no filemarks are encountered; BFILE positions to the beginning of a file if no filemarks are encountered.

### 3.6.2 Record Positioning

The FRECORD and BRECORD commands skip forward or backward over one or more records of the input or output files. The number of records skipped is determined by the parameter *n*, which is entered as a decimal integer. If *n* is omitted, the default is 1.

The execution of a FRECORD or BRECORD command is terminated when the *n* records have been skipped or when EOF is encountered, whichever comes first. Terminator strings are not considered filemarks when executing these commands.

**Example:**

To copy record 51 through and including record 70 of file A.CAL to the tape mounted on MAG2:, enter:

```
INPUT A.CAL
FRECORD INPUT,50
COPY *,MAG2:,NRECS=20
```

### 3.6.3 REWIND Command

Using the REWIND command, you can rewind the current input or output file. This command positions the file at its beginning.

**Example:**

To rewind a tape mounted on MAG2:, enter:

```
OUTPUT MAG2:
REWIND OUTPUT
```

### 3.6.4 Writing Filemarks

With the WFILE command, you can write one or more filemarks on either the list or output device. The parameter n specifies the number of filemarks to be written. Filemarks are written only to devices which support filemarks. Terminator strings are not written as filemarks by the WFILE command. Although the user may specify writing a filemark to an indexed file, a filemark is not actually written. This is consistent with OS/32 processing.

**Example:**

To write two EOF marks on MAG2:, enter:

```
OUTPUT MAG2:
WFILE OUTPUT,2
```

## 3.7 OPTIONS, LIST, NOLOG, AND FILES COMMANDS

This section describes the OPTIONS, LIST, NOLOG, and FILES commands.

### 3.7.1 OPTIONS Command

By using the OPTIONS command, you can set one or more options which govern OS/32 COPY command execution. Once set, a particular option remains in effect until it is changed by a subsequent OPTIONS command. Also, default options are in effect when program execution begins.

The form of the OPTIONS command is:

```
OPTIONS opt1,opt2,...
```

Here opt1,opt2,... are options; these can be entered in any order. Only those options which are entered are changed. The others remain in effect.

Entering the OPTIONS command with no parameters causes the current setting of all options to be output.

#### 3.7.1.1 Filemark Recognition

The NOFM/FM option suppresses or causes the writing of a filemark on the output file when EOF on input is reached. The default is FM.

#### Example:

Concatenate two input files onto a magnetic tape. The files are A1.STR and A2.STR. The record length is 80 bytes and the tape is to be unblocked. You would enter:

```
OPTIONS NOFM
COPY A1.STR,MAG1:
OPTIONS FM
COPY A2.STR,*
```

Resetting the option to FM causes a filemark to be written on the tape when EOF is reached on A2.STR.

#### 3.7.1.2 ASCII/BINARY Mode

The ASCII/BINARY option sets the I/O mode. Some devices support only ASCII I/O while others support both BINARY and ASCII. Appendix C contains a table of device attributes which shows the modes each device supports. The default mode is ASCII.

### 3.7.1.3 Pseudo Filemark Recognition

The PSFM/NOPFSM option determines whether or not a pseudo filemark (X'1313' at the beginning of a record) is recognized as an EOF on contiguous disk files.

The NOPFSM option is useful when copying binary data, such as task images, from a contiguous file. The default is PSFM.

#### Example:

Suppose you have just established the task USEFUL.TSK, using LINK. To copy it to magnetic tape:

```
OPTIONS NOPFSM,BINARY
COPY USEFUL.TSK,*
```

The BINARY option disables recognition of a possible embedded ASCII terminator sequence which might be in effect.

### 3.7.1.4 Terminator Sequence

You can use the TERMINATOR option to establish a sequence of 1- to 4-characters as an EOF indicator for the COPY, VERIFY, and DISPLAY commands. The NOTERM option disables all terminator sequences. The ASCII option must be set in order for a terminator string to be recognized. The default terminator is /\*. A record must not only begin with the terminator sequence but must also consist entirely of blanks beyond the terminator in order to be recognized as EOF. The form of the command is:

```
OPTIONS TERMINATOR=xxxx
```

where xxxx is a 1- to 4-character sequence without embedded blanks or commas that establishes this sequence as the current terminator.

Enter:

```
OPTIONS NOTERM
```

suppresses the current ASCII terminator string. Filemarks or pseudo-filemarks are still recognized.

Enter:

```
OPTIONS TERMINATOR
```

causes the current terminator string to be displayed.

**Example:**

Suppose you wish to copy a deck of cards to tape. The end of the card deck is signalled by a card with /@ in columns 1 and 2 and all blanks in columns 3 through 80. You would enter:

```
OPTIONS TERMINATOR=@  
COPY CR:,MAG1:
```

#### 3.7.1.5 Use of CONTINUE/NOCONT

The CONTINUE/NOCONT option determines the action of OS/32 COPY when, during a VERIFY operation, a mismatch occurs. Under the option NOCONT, which is the default, verifying halts. Under the option CONTINUE, verify continues until an EOF is reached.

#### 3.7.1.6 FILL Character Designation

The FILL option is used when the output record is longer than the input record in a COPY command. In this case, the longer output record is padded with the FILL character.

The default FILL character is X'20', which is the ASCII representation for a blank space.

You can set the FILL character to any 2 digit hexadecimal value by entering:

```
OPTIONS FILL=xx
```

Enter:

```
OPTIONS FILL
```

displays the current fill character in hexadecimal format.

### 3.7.2 LIST Command

If you want to change the device used by the DISPLAY and VERIFY commands for output, you use the LIST command.

#### Example:

```
LIST      PR1:      Change the list device to PR1:
DISPLAY   FILE1     Display FILE1 on PR1:
REWIND    INPUT     Rewind the file
LIST      PR2:      Change the list device to PR2:
DISPLAY   FILE1     Display FILE1 on PR2:
```

### 3.7.3 LOG and NOLOG Commands

It is possible to change the device to which commands and error messages are logged. You may also disable logging using the NOLOG command. If the command device is interactive, error messages are still written to the command device whether or not logging is enabled.

#### Example:

```
LOG PR1:      Change the logging device to PR1:
NOLOG        Disable logging
```

### 3.7.4 FILES Command

The FILES command allows you to display the current input, output, list, and log devices. The FILES command has no arguments. The current attributes of the input and output files as established by the INPUT and OUTPUT commands are also displayed. The format of the display is given in Chapter 4.

## 3.8 BATCH OPERATIONS

OS/32 COPY operates in interactive mode when the command device is interactive and in batch mode when the command device is not interactive.

For the most part, program operation is identical in the two modes. The differences occur in these areas:

1. default assignments of the list and log devices,
2. error and EOVS handling,
3. handling of nonempty output files in the COPY command, and
4. SEND STOP recognition.

### 3.8.1 Default Assignments of List and Log Devices

The defaults for list and log devices are given in Chapter 2. You may issue the NOLOG command in batch mode to suppress logging. This results in no record of your batch run. Therefore, it is advisable to use the default device which is also the list device or to specify a file or device for logging.

### 3.8.2 Error and End of Volume (EOV) Handling

Error handling is described in detail in Chapter 5. In batch mode, any syntax error or unsuccessful program operation other than I/O errors to the input, output, list, or log devices causes termination. In case of I/O errors or EOVS conditions requiring your attention, a message is logged to the console device and the program is paused. If you continue OS/32 COPY, the operation which was in progress when the I/O error or EOVS occurred is resumed.

### 3.8.3 Handling of Nonempty Output Files in the COPY Command

An output file is considered nonempty if it is a contiguous file or if it is an indexed file and has at least one logical record.

If you designate a nonempty file as the output file in interactive mode, you are warned that you may be destroying data in your copy operation. You may enter CONTINUE to perform the copy anyway. In batch mode, the copy is done without the need for a CONTINUE command; however, a message is logged informing you that you may have destroyed data.

### 3.8.4 SEND STOP Processing

SEND STOP has no effect in batch mode.

## CHAPTER 4 OS/32 COPY REFERENCE GUIDE

### 4.1 INTRODUCTION

This chapter discusses OS/32 COPY file handling command syntax, followed by detailed descriptions of each OS/32 COPY command.

### 4.2 FILE HANDLING

Throughout this chapter, the words file and device are used interchangeably to mean any direct access file or nondirect access device which can be designated by an OS/32 file descriptor (fd). Occasionally, a particular kind of device or direct access file is indicated. This should be clear from the context and from designations such as disk file or magnetic tape.

#### 4.2.1 Current File Maintenance

OS/32 COPY keeps track of the current input, output, list, log, and command files. When you enter an fd as an input or output file in a command, you establish that file as the current input or output file until you designate a new input or output file. Similarly, specifying a list or log file in the START parameters or in a LIST or LOG command establishes that file as the current list or log file until you enter a new LIST or LOG command.

A file is established as the current input file by entering its fd as the first parameter of the INPUT, COPY, VERIFY, or DISPLAY commands. A file is established as the current output file by entering its fd as the first parameter of the OUTPUT command, or as the second parameter of the COPY or VERIFY command.

#### 4.2.2 Determination of File Attributes

Whenever possible, OS/32 COPY determines the file attributes of the input, output, list, log, and command files. These attributes are: record length, ASCII/BINARY input/output (I/O), interactive/noninteractive, capable of being rewound, support for filemarks, support for backspace and forward space for records and for filemarks.



Attributes which cannot be determined are ASCII/BINARY mode for files supporting both, record length and blocking factor for magnetic tape, block sizes and keys for disk files, and information pertinent to labelled tapes. There are default values for these attributes. You can specify other values using the INPUT, OUTPUT, and OPTIONS commands.

Table 4-1 lists defaults for input files.

TABLE 4-1 INPUT FILE DEFAULTS

ATTRIBUTE	DEFAULT
ASCII/BINARY	ASCII
Record length	80 bytes or physical record length
Blocking factor	1
Keys	0000
Mag tape	unlabelled
Labelled tapes	
File name	blank
Volid	blank
Expiry date	blank
Pad character	X'FF'

The defaults for the output file are taken from the input file attributes. However, a magnetic tape must be explicitly declared to be labelled if labelled tape handling is desired. Defaults for labelled tape attributes are the same on input and output.

#### 4.2.3 Labelled Magnetic Tape

OS/32 COPY copies tapes with IBM or ANSI labels. Labelled tape information is entered using the INPUT and OUTPUT commands. See the descriptions of the INPUT and OUTPUT commands.

Single file/single volume and single file/multi-volume tapes may be copied and written. See Appendix D for a complete description of tape and label formats.

For input tapes, OS/32 COPY verifies the filename, record length, blocking factor, and volume identifier (volid), if you enter the required values in the INPUT command. For output tapes, the expiration date and volid are validated if you enter the required values in the OUTPUT command. COPY also writes the filename, record length, blocking factor, and creation date on the tape.

Only entire files may be copied from labelled tapes. End of file (EOF) labels are written on an output labelled tape at the conclusion of each COPY command.

File positioning commands cannot be used with labelled tapes.

OS/32 COPY rewinds a labelled tape at the end of each operation. When the end of tape volume is reached, OS/32 COPY logs a message:

```
MOUNT NEXT VOLUME [valid] ON { INPUT } FOR fd
                               { OUTPUT }
```

and rewinds the tape. Valid is output if valids have been specified.

In batch mode, the program pauses so that you can mount the next tape volume. Continuing the program causes the operation in progress to be resumed.

In interactive mode, you can mount another tape, and then enter the CONTINUE command. If a command other than CONTINUE or PAUSE is entered, the current operation is terminated, and the new command is executed.

#### 4.3 COMMAND SYNTAX

Commands consist of a mnemonic, followed by one or more blanks, followed by a list of parameters.

Command mnemonics may be abbreviated. The minimum abbreviation of each command is indicated by underlining in each command description.

Commands may be entered in upper or lower case.

A command is terminated by depressing the RETURN key in interactive mode or by end of record in batch mode.

##### 4.3.1 Continuation of Commands

Commands may extend over more than one line. A line terminated by a comma is considered to be continued on the next line. The commands that might not fit on a single line are the INPUT and OUTPUT commands. These commands can require additional lines when several valids are entered in a single INPUT or OUTPUT command.

### 4.3.2 Command Parameters

The list of command arguments consists of positional parameters optionally followed by keyword parameters.

The positional parameters are:

- the input and output fd in COPY and VERIFY
- the input fd in DISPLAY and INPUT
- the output fd in OUTPUT, and
- all parameters in ALLOCATE, RENAME, FFILE, BFILE, FRECORD, BRECORD, and WFILE.

Keyword parameters may appear in any order after the positional parameters. In most cases, there are default values for keyword parameters. When there are default values, the keyword parameters may be omitted. Keyword parameter abbreviations are indicated by underlining.

The following applies to the parameters:

- [ ] brackets indicate optional parameters.
- { } braces indicate a choice of parameters.

### 4.3.3 Implicit Specification of a File Descriptor (fd) Parameter

If a command requires an input or output fd to be entered, and you wish to specify the current input or output file, you may enter an asterisk (\*) in place of the fd. If both the current input and output files are to be specified, and no keyword parameters are to be entered, you may omit both asterisks and the comma. However, if you enter a keyword parameter, you must enter the asterisks.

The following commands are acceptable and specify that the current input or output file is to be used in the commands.

```
COPY *,*  
COPY  
INPUT *,80/10  
COPY *,*,NRECS=20  
COPY *,MAG3:  
COPY ABC.001,*
```

The following commands are unacceptable. The presence of the keyword parameter, or of the fd means that an asterisk or fd must be entered:

```
COPY ,,NRECS=20
COPY NRECS=20
COPY MAG3:
```

#### 4.3.4 Comment Statements

Comment statements can appear anywhere in the command input. They allow the user to include easy to read information in a command file.

Comment statements must always start with an asterisk (\*) in the first character position. It is possible to have a comment as the only data on a line.

#### Example:

```
* THIS IS A COMMENT LINE
* IT MAY APPEAR ANYWHERE IN THE INPUT
```

#### 4.4 OS/32 COPY COMMANDS

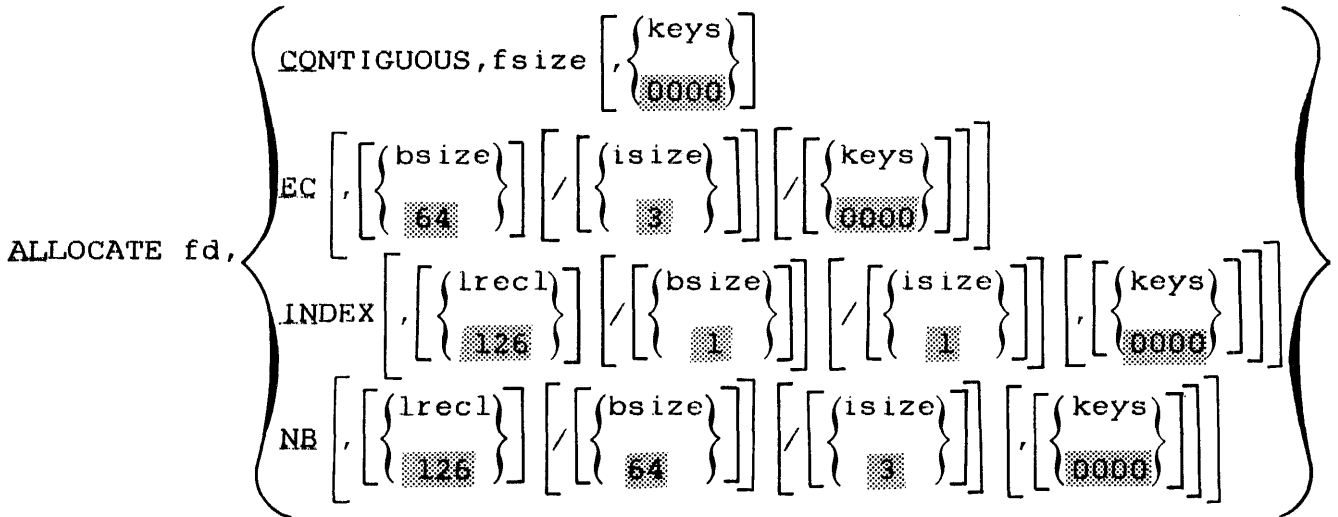
The following sections describe each OS/32 COPY command.

-----  
ALLOCATE

4.4.1 ALLOCATE Command

This command allocates a contiguous, extendable contiguous (EC), indexed, or non-buffered indexed (NB) file on a disk. The command format can be one of the following:

Format:



Parameters:

- fd is the file descriptor of the device or file to be allocated.
- CONTIGUOUS specifies that the file type to be allocated is contiguous.
- fsize is a decimal number indicating file size, which is required for contiguous files. It specifies the total allocation size in 256-byte sectors. This size may be any value up to the number of contiguous free sectors existing on the specified volume at the time the command is entered.

keys	specifies the write and read protection keys for the file. These keys are in the form of a hexadecimal halfword, the left byte of which signifies the write key and the right byte, the read key. If this parameter is omitted, both keys default to 0.
EC	specifies that the file type to be allocated is extendable contiguous.
bsize	is a decimal number specifying the physical block size to be used for buffering and debuffering operations on the index file, and for the data blocks used for indexed, non-buffered indexed, and extendable contiguous files. When INDEX, EC, or NB is specified, bsize represents the block size in sectors of the physical data blocks containing the file. For INDEX files, this parameter cannot exceed the maximum block size established by the system generation (sysgen) procedure. For EC and NB files, this parameter may be any value between 1 and 255 inclusive. If bsize is omitted, the default value for INDEX files is 256 bytes (one sector). For EC and NB files, the default is 64 sectors.
isize	is a decimal number specifying the index block size. For INDEX files, the default value is one sector (256 bytes). For EC and NB files, the default value is three sectors (768 bytes). The index block size cannot exceed the maximum diskblock size established by the sysgen procedure. Neither bsize nor isize may exceed 255.
INDEX	specifies the file type to be allocated is indexed.
lrecl	is a decimal number specifying the logical record length of an indexed or nonbuffered indexed file, or a data communications device. It cannot exceed 65,535 bytes. Its default is 126 bytes. It may optionally be followed by a slash (/), which delimits lrecl from bsize.
NB	specifies that the file type to be allocated is nonbuffered indexed.

## Functional Details:

This command might be entered when the user wants to use as an output file a file other than the default output file generated by the COPY command.

## Examples:

```
AL THISFILE,CO,20
```

Allocate a contiguous file consisting of 20 sectors.

```
AL THATONE,IN,80/5
```

Allocate an indexed file consisting of 80 byte records blocked in 5 sector blocks.

```
AL THEOTH.ER1,IN,80/5/2,AA03
```

Allocate an indexed file with 80 byte records, blocked 5 sectors to a data block, with 2 sectors to an index block, write key AA and read key 03.

```
; AL ANOTHER.FLE,NB,250/1
```

```
; Allocate a non-buffered indexed file with 250-byte  
; logical records with a data block size of one sector and  
; an index block size of three sectors.
```

#### 4.4.2 BFILE Command

This command skips backwards over one or more filemarks.

#### Format:

$$\text{BFILE} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \right\} \left[ \begin{array}{l} n \\ \text{FILE} \end{array} \right]$$

#### Parameters:

- INPUT specifies that the command applies to the current input file.
- OUTPUT specifies that the command applies to the current output file.
- n is a decimal integer specifying the number of filemarks to be skipped. The default is 1.

#### Functional Details:

On contiguous files and magnetic tapes, BFILE skips backwards over filemarks. When the command has been executed, magnetic tapes are positioned just before the last filemark skipped, while contiguous files are positioned on the last filemark skipped. In either case, the next read operation returns an EOF status. When the file specified is an indexed file, the file is positioned at the beginning.

This command is only valid for devices that support filemarks. In order for filemarks to be recognized in contiguous files, the pseudo filemark (PSFM) options must be set. See the OPTIONS command for information on the PSFM option.

This command cannot be issued for a labelled magnetic tape.

#### Example:

```
COPY A.CAL,MAG1: Copy a file to tape
COPY B.CAL,MAG1: Copy another file
BFILE OUTPUT,2 Position before filemark
                  ending first file.
```



-----  
BRECORD

#### 4.4.3 BRECORD Command

This command skips backwards over one or more records.

Format:

$$\text{BRECORD} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \right\} \left[ \begin{array}{l} n \\ 1 \end{array} \right]$$

Parameters:

INPUT	specifies that the command applies to the current input file.
OUTPUT	specifies that the command applies to the current output file.
n	is a decimal integer specifying the number of records to be skipped. The default is 1. When the file to be positioned is a blocked, unlabelled magnetic tape, n signifies the number of blocks to be skipped beginning at the current position of the tape.

Functional Details:

BRECORD skips records until n records have been skipped or until a filemark, end of medium (EOM), or EOF is reached, whichever comes first.

This command cannot be issued for a labelled magnetic tape.

Example:

COPY A,CAL,PR:	Copy file to printer
BRECORD INPUT 5	Go back 5 records
COPY *,PNCH:	Punch last 5 records

#### 4.4.4 CONTINUE Command

This command resumes the execution of a command that was suspended due to an I/O error, a nonverify condition under option NOCONT, the SEND STOP command, or to an end of volume (EOV) condition on magnetic tape. CONTINUE is only effective in interactive mode. If entered in batch mode, the command has no effect.

#### Format:

CONTINUE

-----  
COPY

#### 4.4.5 COPY Command

The COPY command copies data from the input file to the output file. All or part of a single file may be copied, or more than one file can be copied with a single COPY command. The command format is:

Format:

$$\text{COPY } \left\{ \begin{array}{c} * \\ \text{fd1} \end{array} \right\}, \left\{ \begin{array}{c} * \\ \text{fd2} \end{array} \right\} \left[ \left( \begin{array}{l} \text{NRECS} = \left\{ \begin{array}{c} m \\ \text{ALL} \end{array} \right\} \\ \text{NF ILES} = \left\{ \begin{array}{c} n \\ 1 \end{array} \right\} \end{array} \right) \right] \left[ \text{DELETE} \right]$$

Parameters:

- \* denotes the current input or output file.
- m is a decimal number up to 6 digits specifying the number of records to be copied. The default is all records until an EOF.
- fd1 is the file descriptor of the input file.
- fd2 is the file descriptor of the output file.
- n is a decimal number up to 6 digits in length specifying the number of files to be copied or the word ALL indicating that all files up to EOF are to be copied. The default is 1 file.
- DELETE if entered, indicates that if fd2 is an existing disk file, it is to be deleted and reallocated according to the attributes of fd1.

Functional Details:

When the input file is a disk file or fixed record length device, its record length is determined by the program. When the input file is a magnetic tape, its record length and block size may be specified using the INPUT command. If the INPUT command is not entered, unlabelled tapes are assumed to be unblocked with a record length of 80 bytes. In the case of labelled tapes, this information is taken from the label if not specified in the INPUT command.

Output file attributes are also obtained by the program. If the output file is a disk file which does not exist, a file of the same type is allocated with the same attributes as the input file. For indexed files, the index and data block sizes are one.

For magnetic tape, the record length and blocking factor may be specified using the OUTPUT command or the input record length is assumed. If the input file is blocked magnetic tape, the output data is blocked the same way.

Blocked, unlabelled tapes are written with all blocks the same size. However, the last block of a tape file may be smaller than the others. This depends on the blocking factor and the number of records in the file.

OS/32 COPY determines if the output file specified in an OUTPUT or COPY command is a contiguous file or if it is a nonempty indexed file. If it is either type, the following message is logged:

#### NONEMPTY OUTPUT FILE

If the program is in batch mode, the COPY command is executed. If in interactive mode, you must enter CONTINUE to cause the COPY command to be executed.

The above action only occurs the first time a file is established as the current output file in a COPY command that does not specify DELETE.

An existing file may be deleted by entering the DELETE keyword in a COPY command. The specified output disk file is deleted and reallocated with the same file type and record length (and size for a contiguous file) of the input file. If the input file is a device, an indexed file is allocated. If the volume name is not specified, the default volume is used.

The parameter NRECS specifies the number of records to be copied. The default is the number of records in the file. Copying stops if an EOF is encountered on input.

The parameter NFILES specifies the number of files to be copied. The default value is one. A file is terminated by a filemark or by an ASCII terminator. See the OPTIONS command for information on ASCII terminators.

Copying terminates if an EOV is encountered on input. EOV is defined by two consecutive filemarks.

After each file has been copied, a message indicating the number of records which have been copied in that file is logged.

If NFILES=ALL is entered, all files are copied until EOV occurs on input.

EOM processing for magnetic tapes is intended to allow you to conveniently copy an input file to more than one tape volume.

When physical EOM is detected on an output tape and there are remaining input records to be copied, a filemark and an EOV indicator followed by another filemark are written on the tape, and the tape is rewound. If it is an unlabelled tape, a message is logged:

```
END OF VOLUME -n RECORDS COPIED
```

where n is the number of records copied so far.

If the tape is labelled, an EOV label is written and the tape is rewound. A message is logged:

```
MOUNT NEXT VOLUME volid ON OUTPUT FOR fd
```

where volid is the volume identifier of the next tape. It is output only if volids are specified in the OUTPUT command. fd is the file descriptor of the magnetic tape drive.

In batch mode, the program pauses. When it is continued, copying resumes. While the program is paused, you can mount the next volume.

In interactive mode, you can mount another tape and then enter CONTINUE to cause copying to resume. If you enter a command other than CONTINUE or PAUSE, the previous COPY is terminated.

OS/32 COPY assumes that the last file on an unlabelled input magnetic tape is followed by two filemarks. When EOM is encountered on an input tape, COPY continues to read until an EOV indicator is read. EOM on a contiguous file is treated like an EOF on input, and as an error on output.

When the input record length exceeds the output record length, the input record is truncated before it is copied or verified.

When the input record length is less than the output record length, the output record is padded with a fill character before it is copied. The fill character can be set by the FILL options in the OPTIONS command. The default is a hexadecimal 20 (ASCII blank).

When the COPY command is successfully completed, a message is logged:

```
COPY COMPLETE -n FILES COPIED
```

where n is the number of files copied.

**Examples:**

```
COPY A.CAL,B.CAL
```

Copy one disk file to another.

```
COPY A.CAL,B.CAL,DELETE
```

Delete and reallocate output file; copy file A.CAL to B.CAL.

```
COPY A.CAL,PR:,NRECS=100
```

Copy first 100 records to printer.

```
INPUT MAG1:,80/5
```

Input file is a blocked unlabelled tape with 80 byte records, 5 records per block.

```
OUTPUT MAG2:,80/1
```

Output file is an unblocked unlabelled tape.

```
COPY
```

Copy one file.

```
COPY*,*,NRECS=15
```

Copy first fifteen records of second file from the input tape to the output tape.

-----  
DELETE

#### 4.4.6 DELETE Command

This command is used to delete a disk file.

##### Format:

```
DELETE fd [,fd...]
```

##### Parameters:

fd identifies the file to be deleted.

##### Functional Details:

A file may be deleted only if its read and write protection keys are zero and it is not currently assigned to any logical unit (lu) of any task. The only exception is when the current input, output, list, or log file is to be deleted. In this case, OS/32 COPY closes its own lu assignment and deletes the file.

##### Examples:

```
DELETE ART.XXX
```

```
DELETE ART1.D,ART2.D,ART3.D
```

#### 4.4.7 DISPLAY Command

This command displays a file in hexadecimal and ASCII format.

**Format:**

$$\text{DISPLAY } \left\{ \begin{array}{c} * \\ \text{fd} \end{array} \right\} \left[ \left( \begin{array}{l} \text{NFILES} = \left\{ \begin{array}{c} m \\ 1 \end{array} \right\} \\ \text{NRECS} = \left\{ \begin{array}{c} n \\ \text{ALL} \end{array} \right\} \end{array} \right) \right]$$

**Parameters:**

- fd            is the file descriptor of the file to be displayed. If \* is entered, the current input file is displayed. If an fd is entered, that file is established as the new input file.
- m            is a decimal number up to 6 digits in length specifying the number of files to be displayed or the keyword ALL indicating that all files are to be displayed up to EOF. The default is 1 file.
- n            is a decimal number up to 6 digits specifying the number of records to be displayed. If n is not entered, all records are displayed until EOF.

**Functional Details:**

The file display is output to the list device.

No filemarks are written when input EOF is reached. However, you can write a filemark on the list device using the WFILE command.

If the file to be displayed has other than the default attributes, the INPUT command must be used before the DISPLAY command.



**Examples:**

DISPLAY M.DTA

Display the file M.DTA on the list device.

DISPLAY \*

Display the current input file on the list device.

DISPLAY \*,NRECS=1

Display one record of the current input file.

INPUT MAG1:,120/1 DISPLAY \*,NFILES=2

Display two files from MAG1:

-----  
END

#### 4.4.8 END Command

This command terminates OS/32 COPY.

Format:

END

-----  
FFILE

#### 4.4.9 FFILE Command

This command skips forward over one or more filemarks.

##### Format:

$$\text{FFILE} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \right\} \left[ \left\{ \begin{array}{l} n \\ \text{ALL} \\ \text{I} \end{array} \right\} \right]$$

##### Parameters:

INPUT	specifies that the command applies to the input file.
OUTPUT	specifies that the command applies to the output file.
n	is a decimal integer specifying the number of filemarks to be skipped. The default is 1.
ALL	is a keyword, which, when entered, specifies that all filemarks are to be skipped until EOV is reached.

##### Functional Details:

On contiguous files and magnetic tapes, NFILE skips forward over filemarks. When the command has been executed, magnetic tapes are positioned just after the last filemark skipped, while contiguous files are positioned at the record after the last filemark skipped. When the file specified is an indexed file, the file is positioned at the end of the file.

The INPUT or OUTPUT file must support filemarks. In order for pseudo filemarks to be recognized in contiguous files, the PSFM option must be set. See the OPTIONS command for information on the PSFM option.

This command cannot be applied to a labelled magnetic tape.

When an integer, n, is entered, command execution ends when n filemarks are skipped or when EOM is encountered.

When the keyword ALL is entered, command execution ends when EOVS or two consecutive filemarks are read.

**Examples:**

FFILE INPUT

skip one filemark on the input device.

FFILE OUTPUT,3

skip three filemarks on the output device.

FFILE OUTPUT,ALL

Skip to EOVS on the output device.

-----  
FILES

**4.4.10 FILES Command**

The FILES command produces a display of current input, output, list, command, and log files, including various file attributes established by the INPUT and OUTPUT commands.

**Format:**

FILES

The display has the following form:

INPUT	fd	fname VARIABLE	lrecl /blk	record no	pad	
	VOLID	xxxxxx	xxxxxx	xxxxxx	xxxxxx	xxxxxx
OUTPUT	fd	fname VARIABLE	lrecl /blk	record no	pad	expiry date
	VOLID	xxxxxx	xxxxxx	xxxxxx	xxxxxx	xxxxxx
LIST	fd					
COMMAND	fd					
LOG	fd					

**Parameters:**

- fd            is the file descriptor of the indicated file.
- fname        is the filename on labelled tapes; omitted if not labelled tape.
- VARIABLE     is output if variable length record processing is in effect.
- lrecl/blk    is the record length and blocking factor; blk is omitted for devices other than tapes.

record no            is the current record number last processed in the file; a zero value indicates positioning in front of a filemark on a contiguous file.

expiry date        is expiration date for output labelled tapes.

xxxxxx            is the volids for labelled tapes.

It is not possible to determine the correct record position as a result of a FFILE command issued for contiguous files, or FFILE, BFILE, FRECORD, or BRECORD for magnetic tapes. The record position in the file is output as a relative position.

The labelled tape information is omitted if it is not known or is blank.

**Examples:**

FILES

Typical output produced on the log device is:

INPUT	MAG1:	80/5	1
OUTPUT	M300:A.CAL/P	80	1
LIST	CON:		
COMMAND	CON:		
LOG	CON:		

The input tape is unlabelled. If it were labelled, a pad character would be displayed.

**Example:**

Labelled tape on output:

INPUT	M300:	A.DATE/P	120	1		
OUTPUT	MAG1:	FNAME.L	120/5	1	20	80001
	VOLID	100000	100001	100002	100003	
LIST	CON:					
COMMAND	CON:					
LOG	CON:					

The output tape is labelled, blocked five 120 byte records to a block, positioned at record 1, has a pad character of X'20', an expiry data of January 1, 1980, and volids of 100000, 100001, 100002, and 100003.

```
INPUT      MAG4:          80/2          *
LIST       PR:
LOG        CON:
COMMAND    CON:
```

The input tape is blocked two 80 byte records per block. The current record position is undefined.

```
INPUT      M301;FILE.CON/P    256          -2
LIST       PR:
LOG        CON:
COMMAND    CON:
```

The input file is contiguous and due to a previous file positioning command, the file is positioned two records in front of a filemark.

**Example:**

```
FF IN      (EOM message is output)
BR IN,2
FILES
```

The record number output is -2.

Output file is a magnetic tape.

```
FF OUT
FILES
```

Record no EOF

```
BR OUT
FILES
```

Since one magnetic tape record may contain any number of physical records, an asterisk is output for record position.

#### 4.4.11 FRECORD Command

This command skips forward over one or more records.

##### Format:

$$\text{FRECORD} \left\{ \begin{array}{l} \text{INPUT} \\ \text{OUTPUT} \end{array} \right\} \left[ \left\{ \begin{array}{l} (n) \\ \text{E} \end{array} \right\} \right]$$

##### Parameters:

INPUT	specifies that the command applies to the current input file.
OUTPUT	specifies that the command applies to the current output file.
n	is a decimal integer specifying the number of records to be skipped. The default is 1. When the file to be positioned is a blocked unlabelled magnetic tape, n signifies the number of blocks to be skipped beginning at the current position of the tape.

##### Functional Details:

FRECORD skips records until n records are skipped or until a filemark of EOM is reached, whichever comes first.

This command cannot be applied to a labelled magnetic tape.

##### Examples:

FRECORD INPUT

Skip one record forward on the input device.

FRECORD OUTPUT,20

Skip 20 records forward on the output device.



-----  
HELP

| 4.4.12 HELP Command

| The HELP command displays information on OS/32 COPY program commands.

| Format:

| HELP [ { mnemonic } ]  
|           \*           

| Parameters:

- | mnemonic           is any valid COPY command mnemonic.
- | \*                   causes a list of all COPY commands to  
|                   be displayed to the list device.

| Functional Details:

| When the HELP command is entered with a command mnemonic,  
| information on how to use that particular command is displayed to  
| the list device. When the HELP command is entered with the  
| asterisk (\*), a list of all OS/32 COPY commands is displayed to  
| the list device. The list also shows the minimum abbreviation  
| required for each command. If no parameters are entered,  
| information on how to use the HELP command is displayed to the  
| list device.

| Examples:

- | 1. HELP command is entered with a mnemonic:

| HELP ALLOCATE  
|   ALLOCATE  
|       ALLOCATE fd,CONTIGUOUS,fsize [,keys]  
|       or  
|       ALLOCATE fd,INDEX [,lrecl] [/bsize] [/isize] [,keys]

where:

fd is the file descriptor of the file to be allocated.

fsize is the size of the contiguous file in sectors.

keys are the write-read keys of the file. Default: zero for both keys.

lrecl is the logical record length of the indexed file. Default: 126.

bsize is the data block size in sectors of the indexed file. Default: 1.

isize is the index block size in sectors. Default: 1.

2. HELP command is entered with an asterisk.

```
HELP *
AL(LOCATE)      DE(LETE)      REN(AME)      COP(Y)
PA(USE)         REW(IND)      FF(ILE)      BF(ILE)
FR(ECORD)      BR(ECORD)     WF(ILEMARK)  OP(TIONS)
DI(SPLAY)      VE(RIFY)      FI(LES)      CON(TINUE)
IN(PUT)        OU(TPUT)     LI(ST)       LO(G)
NOL(OG)        EN(D)        H(ELP)
FOR HELP ON ANY OF THE ABOVE COMMAND MNEMONICS, TYPE HELP
MNEMONIC
```

3. HELP command is entered without a parameter.

```
HELP
FOR A LIST OF COMMANDS TYPE HELP *
FOR HELP ON ANY COMMAND MNEMONIC, TYPE HELP MNEMONIC
```

#### 4.4.13 INPUT Command

This command allows you to specify information about the input file. Its use is required when labelled or blocked unlabelled tapes, disk files with nonzero keys, or records of variable length are to be copied, verified, or displayed. Its use is also necessary when a file or device is to be positioned (using the positioning commands, FRECORD, FFILE, etc.) if the file to be positioned is not the current input file.

#### Format:

$$\text{INPUT } \left\{ \begin{array}{c} * \\ \text{fd} \end{array} \right\} \left[ \left[ \left\{ \begin{array}{c} \text{lrecl} \\ 80 \end{array} \right\} \right] \left[ \left\{ \begin{array}{c} \text{blk} \\ 1 \end{array} \right\} \right] \left[ \text{VARIABLE} \right] \left[ \left\{ \begin{array}{c} \text{keys} \\ 0000 \end{array} \right\} \right] \left[ \text{LABEL} \right] \right. \\ \left. \left[ \text{FNAME=fn} \right] \left[ \text{VOLID=id} \right] \left[ \text{PAD=pp} \right] \right]$$

#### Parameters:

- \* is used to designate the current input file.
- fd is the file descriptor of the input file. This file becomes the current input file. If fd is a disk file, it must already exist.
- lrecl is the record length for tapes. The default is 80 bytes.
- blk is the blocking factor for tapes. The default is 1 record per block. The product of lrecl and blk must not exceed  $2^{16} - 1$  or 32,760 for labelled tapes.
- VARIABLE specifies variable length record processing. Default is fixed record length. Can only be used with unlabelled, unblocked magnetic tape, paper tape, and interactive devices. lrecl should specify maximum expected record length.
- keys are the write-read keys, two hexadecimal bytes. The default is 0000.
- LABEL must be entered if the tape is labelled. If it is not, FNAME, PAD, and VOLID are ignored.

fn is a 1- to 17-character file id for a labelled tape. If not entered, the file id is ignored.

id is a list of volids. Each id may be up to 6 alphanumeric characters. If more than one id is entered, the list of volume ids must be enclosed in parentheses. If not entered, volids are not validated. If entered, input tapes must be mounted in the order specified by the volids.

pp is a two-digit hexadecimal pad character. The default is X'FF'. Records of labelled tapes which consist of all pad characters are ignored.

**Examples:**

INPUT A.CAL,KEYS=AA03

Make A.CAL the new input file. Set AA03 as the write-read key.

INPUT MAG1:,80/5

The tape on MAG1: becomes the current input file. The records length is 80 and the blocking factor is 5.

INPUT \*,120

The current input file is to be read with a record length of 120 and a blocking factor of 1.

IN MAG2:,LABEL,FNAME=ALPHA,VOLID=(100000,100001)

The tape on MAG2: is labelled. Its file id is ALPHA; two tape volumes are to be mounted for input; the volids should be mounted in the order 100000, 100001. The record length and block size are taken from the label.

INPUT MAG1:,LABEL,80/10,VOL=999888

The labelled tape on MAG1: must have valid 999888; it must have 80 byte records blocked 10 records per block or the tape will not be read. The file id in the label will be ignored.

INPUT PTRP:,80,VARIABLE

The paper tape in the paper tape reader contains records of variable length. The maximum expected record length is 80 bytes.

#### 4.4.14 LIST Command

This command changes the list device.

##### Format:

LIST fd

##### Parameters:

fd is the file descriptor of the new device.

##### Functional Details:

The list and log devices may be the same. In this case, the list and log outputs are intermixed since the same lu is used for both.

##### Example:

LIST PR1: use PR1: as the list device.

-----  
LOG

#### 4.4.15 LOG Command

This command changes the log device.

##### Format:

LOG fd

##### Parameters:

fd is the file descriptor of the new log device.

##### Functional Details:

The log and list devices may be the same. In this case, the log and list outputs are intermixed since the same lu is used for both.

##### Example:

LOG PR1:

Use PR1: as the log device.

#### 4.4.16 NOLOG Command

This command suspends logging until a log command is entered.

#### Format:

NOLOG



-----  
OPTIONS

#### 4.4.17 OPTIONS Command

This command sets or displays options which are in effect during the execution of the program.

**Format:**

OPTIONS [ { NOFM } ] [ { NOPSFM } ] [ { BINARY } ] [ , FILL=xx ]  
          [ { FM } ] [ { PSFM } ] [ { ASCII } ]  
          [ { TERMINATOR=xxxx } ] [ { CONTINUE } ] [ , VERR=count ]  
          [ { NOTERM } ] [ { NOCONTINUE } ] [ , 0 ]

**Parameters:**

NOFM FM	These options control the writing of filemarks and positioning of the output file after a copy to a magnetic tape or contiguous file. See below for details.
NOPSFM PSFM	These options control recognition of the pseudo filemark (X'1313') associated with contiguous files. If the NOPSFM option is set, pseudo filemarks are ignored when reading a contiguous file or when a WFILE, BFILE, FRECORD, or BRECORD command is issued for a contiguous file.
BINARY ASCII	These options control the I/O mode.
FILL=xx	This option sets the fill character for output records which are longer than input records. The fill character is specified as xx, two hexadecimal digits.
TERMINATOR =xxxx NOTERM	The TERMINATOR= option sets a sequence of characters to be recognized as an EOF indicator. xxxx consists of 1- to 4-ASCII characters with no imbedded blanks. For an input record to be recognized as EOF, the program must be in ASCII mode, the first characters of the record must match the terminator sequence, and the rest of the record must consist of blanks. NOTERM suppresses recognition of the indicator as set by TERMINATOR.

CONTINUE            These options control OS/32 COPY actions  
NOCONTINUE        following a mismatch during a VERIFY  
                    operation. When CONTINUE is in effect, OS/32  
                    COPY continues verifying the files. When  
                    NOCONTINUE is in effect, verifying ends if a  
                    mismatch occurs.

VERR=              Used to designate the number of errors that  
                    can occur when verifying data, before OS/32  
                    COPY either pauses or goes to end of task. If  
                    the CONTINUE option is used and the number of  
                    errors encountered exceeds the number  
                    specified in the VERR=count option, the  
                    program terminates with end of task code 2 if  
                    in batch mode. If in interactive mode, the  
                    program pauses. If NOCONTINUE is used, both  
                    VERR=count and CONTINUE are overridden.

If the VERR=count option is omitted, the  
default is 0.

count              is a 1- to 5-digit decimal number specifying  
                    the number of records in a file (maximum 31K,  
                    minus 1) that must fail to match before the  
                    program either pauses or goes to end of task.

The default values are in effect when OS/32 COPY begins  
execution. Each time an OPTIONS command is entered, only those  
options that are entered are changed. The others remain as they  
were.

If OPTIONS is entered with no parameters, the current options are  
displayed.

When an EOF is encountered in the input file or an EOM if the  
input file is a contiguous file, the following occurs:

- if NOFM is set and the output file is contiguous, no filemark  
is written;
- if NOFM is set and the output file is unlabelled magnetic  
tape, two filemarks are written; and
- the tape is backspaced two filemarks.

If FM is set and the output file is contiguous, a filemark is  
written. Failure to write this filemark because of EOM status is  
not considered to be an error. If the output file is unlabelled  
magnetic tape, two filemarks are written and the tape is  
backspaced one filemark.

As a result of this positioning, the next COPY to the current output device is separated from the previous data by a filemark if FM is set, and is not separated if NOFM is set. Furthermore, the last file copied to an unlabelled magnetic tape is followed by two filemarks indicating EOV.

The FM/NOFM option does not apply to labelled tapes.

**Examples:**

OPTIONS

Obtain a list of the current options.

OPTIONS NOTERM,ASCII

Set a mode to ASCII; no terminator sequence is to be recognized.

OPTIONS CONTINUE

Continue verifying an entire file even when mismatches occur.

OPT TERM=ENDF,FILL=00

ENDF is the terminator string, larger output records are to be filled with zeros.

OPT NOFM

Suppress filemarks on output.

#### 4.4.18 OUTPUT Command

This command allows you to specify information about the output file. Its use is required when labelled or blocked unlabelled tapes, disk files with nonzero keys, or variable length records are to be copied, verified, or displayed. Its use is also necessary when a file or device is to be positioned (using the positioning command: FRECORD, FFILE, etc.) if the file to be positioned is not the current output file.

#### Format:

```
OUTPUT { * } [ , [lrecl] [/blk] [VARIABLE] [ , { keys } ] [ , LABEL ]
          [ , FNAME=fn ] [ , VOLID=id ] [ , PAD=pp ] [ , EXPIRY=yyddd ] ]
```

#### Parameters:

- \* is used to designate the current output file.
- fd is the fd of the output file which becomes the current output file. If fd is a disk file, it must already exist.
- lrecl is the logical record length for tapes. The default is the record length of the input file.
- blk is the blocking factor for tapes. Default is 1 record per block or that of the input tape. The product of lrecl and blk must not exceed  $2^{16}-1$ , or 32,760 for labelled tapes.
- VARIABLE specifies variable length record processing. Default is fixed record length. Can only be used with unlabelled, unblocked magnetic tape, paper tape, and interactive devices. lrecl can specify a maximum output record length, otherwise the output record length is obtained from the input file.
- keys are the write-read keys, two hexadecimal bytes. The default is 0000.

**LABEL** must be entered if the tape is labelled. If it is not, **FNAME**, **PAD**, **EXPIRY** and **VOLID** are ignored.

**fn** is a 1- to 17-character file id for a labelled tape. If it is not entered, the file id is written as blanks.

**id** is a list of volids. Each id is 6 alphanumeric characters. If more than one id is entered, the list volids must be enclosed in parentheses. If entered, tape volumes must be mounted in the order specified by the volids.

**pp** is a two-digit hexadecimal pad character. The default is **X'FF'**. This character pads the last block of a labelled tape if the number of records written does not fill the block.

**yyddd** is the expiration date expressed as **yyddd**, where **yy** is the year (0-99) and **ddd** is the day within the year (1-365). If the **EXPIRY** parameter is specified and the expiration date on the tape has not yet been reached, the output operation is not executed. If that expiration date has occurred, the value entered in the **EXPIRY** parameter is written on the tape label and the current date is written in the creation date field.

#### Functional Details:

The **fd** or asterisk must be the first parameter entered. The other parameters may be entered in any order.

#### Examples:

**OUTPUT A.CAL,KEYS=AA03**

Make **A.CAL** the new output file and set the write-read keys to **X'AA03'**.

**OUTPUT MAG1:,80/5**

The tape on **MAG1:** becomes the output file. The record length is 80 bytes and the blocking factor is 5 records per block.

OUTPUT \*,120

The current output tape is to be written with a record length of 120 bytes and a blocking factor of 1 record per block.

OUT MAG2:,LABEL,FNAME=BETA, VOLID=(100000,100001)

The tape on MAG2: is labelled. Its file id is BETA, and two tape volumes are to be mounted for output; the volids are mounted in the order 100000, 100001. The expiration date on the output tape is not checked.

OUTPUT MAG1:,LABEL,80/10,VOL=999888

The labelled tape on MAG1: must have volid 999888; it will be written with 80 byte records blocked 10 records per block. No file id is written. The expiration date is not checked.

OU MAG3:,LABEL,FNAME=ATOZ,120/4,PAD=31 EXPIRY=79365

The labelled tape on MAG3: is to be written with file id ATOZ. It will have 120 byte records, blocked 4 records per block. The pad character is hexadecimal 31. The new file expires on the 365th day of 1979.

OUTPUT MAG1:,VARIABLE,132

The tape on MAG1: will contain records of variable length where the largest record is 132 bytes long.

-----  
PAUSE

**4.4.19 PAUSE Command**

This command causes OS/32 COPY to pause.

**Format:**

PAUSE

**Functional Details:**

Multi-terminal monitor (MTM) or operating system (OS) commands may then be entered. A CONTINUE command causes OS/32 COPY processing to resume.

#### 4.4.20 RENAME Command

This command is used to change the name of a disk file.

##### Format:

```
RENAME oldfd,newfd
```

##### Parameters:

```
oldfd          is the old filename
newfd          is the new filename
```

##### Functional Details:

A file may be renamed only if its read and write protection keys are zero and it is not currently assigned to any lu of any task. If the file specified is the current input, log, or list file for OS/32 COPY, the corresponding lu is closed and the file is then renamed.

The renamed file is not available for OS/32 COPY processing until it is explicitly selected again.

Account numbers may not be specified as part of the fds. However, under MTM, the new filename can have a group file or system file indicator (/G or /S).

##### Example:

```
RENAME A1.CAL,A1.MAC
```

Change the name of A1.CAL to A1.MAC



-----  
REWIND

4.4.21 REWIND Command

This command repositions a file or device at its beginning record if the device is rewindable.

Format:

REWIND { INPUT }  
          { OUTPUT }

Parameters:

- INPUT               specifies that the command applies to the current input file.
- OUTPUT             specifies that the command applies to the current output file.

Functional Details:

If the file or device specified is not rewindable, an attribute warning occurs.

Examples:

- REWIND OUTPUT  
    Rewind the current output file.
  
- REW IN  
    Rewind the current input file.

#### 4.4.22 SEND STOP Command

This command allows you to stop the execution of a COPY, DISPLAY, VERIFY, FRECORD, BRECORD, FFILE, or BFILE command.

##### Format:

SEND STOP

##### Functional Details:

The SEND STOP command is issued in response to an OS or MTM prompt. A prompt is obtained by depressing the BREAK key, possibly several times. When OS/32 COPY is executing in the OS environment, it is necessary to select the OS/32 COPY task as the current task using the TASK command before issuing the SEND STOP command.

Issuing the SEND STOP command causes OS/32 COPY to terminate the command which is currently being executed. After processing the SEND STOP command, OS/32 COPY is ready to accept the next command. In interactive mode, if the next command is CONTINUE, COPY resumes executing the previously terminated command. SEND STOP is ignored in batch mode.

##### Examples:

###### Under MTM:

SEND STOP

Stop execution of the current command. OS/32 COPY accepts the next command.

###### Under OS/32:

TASK COPY

Select copy as the current task.

SEND STOP

Stop execution of the current command.

```

-----
|  VERIFY  |
-----

```

#### 4.4.23 VERIFY Command

The VERIFY command compares the current input file to the current output file record by record.

If a pair of records is found which does not match, the two records are displayed on the list device in the hexadecimal dump format produced by the DISPLAY command. The number of the first byte which does not match is also written.

All or part of a single file may be verified, or more than one file can be verified with a single VERIFY command.

Format:

$$\text{VERIFY } \left\{ \begin{array}{c} * \\ \text{fd1} \end{array} \right\}, \left\{ \begin{array}{c} * \\ \text{fd2} \end{array} \right\} \left[ \left( \begin{array}{l} \text{NRECS} = \left\{ \begin{array}{c} m \\ \text{ALL} \end{array} \right\} \\ \text{NF FILES} = \left\{ \begin{array}{c} n \\ \text{ALL} \\ 1 \end{array} \right\} \end{array} \right) \right]$$

Parameters:

- \* indicates the current input or output file.
- fd1 is the file descriptor of the input file.
- fd2 is the file descriptor of the output file.
- m is a decimal number up to 6 digits in length specifying the number of records to be verified. The default is all records until EOF is read.
- n is a decimal number up to 6 digits in length specifying the numbers of files to be verified or the word ALL indicating that all files up to EOF are to be verified. The default is 1 file.

## Functional Details:

In case a mismatch occurs, the next action is determined by the CONTINUE/NOCONT option as previously set using the OPTIONS command or by default. The default is NOCONT.

If NOCONT is in effect and the program is in interactive mode, a prompt is issued. You may continue verifying by entering CONTINUE, or you may enter a different command. If the program is in batch mode, a message is logged and the next command is read.

If CONTINUE is set, verifying continues until all files or records specified have been compared, or until an EOF on one file is not matched by an EOF on the other. All records which do not match are written on the list device.

The parameter NRECS specifies the number of records which are to be verified. The default is the number of records in the file. Verifying stops if an EOF is encountered on input.

The parameter NFILES specifies the number of files which are to be verified. The default value is one. A file is terminated by a filemark or by an ASCII terminator. See the OPTIONS command for information on ASCII terminators. Verifying terminates if an EOF is encountered on input. EOF is defined by two consecutive filemarks on magnetic tape, or EOM on contiguous files. After each file has been verified, a message is logged indicating the number of records which have been verified in that file.

If NFILES=ALL is entered, all files are verified until EOF occurs on input.

When the output record length exceeds the input record length, the output record is assumed to be padded with a fill character. The fill character can be set by the FILL option. See the OPTIONS command for information on the FILL option. The default is a hexadecimal 20 (ASCII blank).

Data in both records is verified for the length of the input record. All remaining bytes in the output record must contain the fill character or a nonverify message is output.

When the input record is larger than the output record, only the number of bytes contained in the output record are verified.

When the VERIFY command is successfully completed, a message is logged:

```
VERIFY COMPLETE - n FILES VERIFIED
```

where n is the number of file verified.

If one file or a portion of a file is verified (using NRECS), the message logged is:

VERIFY COMPLETE - n RECORD VERIFIED

**Examples:**

VER A.CAL,CR:

Verify a disk file with data read from cards.

VERIFY

Verify the current input and output files.

VER \*,\*,NRECS=50

Verify 50 records of the current input and output files.

VERIFY MAG1:,\* ,NFILES=2

Verify two files on MAG1: with two files on the current output file.

#### 4.4.24 WFILE Command

This command writes one or more filemarks on the current output or the current list file.

Format:

$$\text{WFILE } \left\{ \begin{array}{l} \text{LIST} \\ \text{OUTPUT} \end{array} \right\} \left[ \left\{ \begin{array}{l} n \\ \text{E} \end{array} \right\} \right]$$

Parameters:

- LIST                    specifies that the command applies to the current list file.
- OUTPUT                specifies that the command applies to the current output file.
- n                      is a decimal integer specifying the number of filemarks to be written. The default is 1.

Functional Details:

The output or list device on which the filemark is to be written must support the write filemark operation. See Appendix C for a table of device attributes.

This command cannot be applied to a labelled magnetic tape and has no effect on indexed files.

**Examples:**

WFILES OUTPUT

Write one filemark on the output file.

WF LI,2

Write two filemarks on the list file.

WF OUT,3

Write three filemarks on the output device.

## CHAPTER 5 ERROR HANDLING AND MESSAGES

### 5.1 ERROR HANDLING

OS/32 COPY error processing depends on the nature of the error and the mode of execution, interactive or batch. In either case, a message is written to the log device. If the command device is interactive, the error message is output to both the log device and the command device. In batch mode, if there is no log device or if the error applies to the log device, the message is logged to the system console.

#### 5.1.1 Interactive Mode

In interactive mode, OS/32 COPY accepts another command following an error.

The sole exception occurs when the error is an input/output (I/O) error involving the command device. In this case, the message is logged to the console and the program terminates.

In all other cases, OS/32 COPY prompts you for another command. It may be necessary to reenter the command if the error is a syntax, ALLOCATE, attribute, labelled tape, ASSIGN, or DELETE/RENAME error. If the error is an I/O error, you may be able to correct the condition causing the error, and then enter CONTINUE to cause the operation to continue. The only other command you may enter before CONTINUE is PAUSE. If you enter any other command, the previous command is considered to be terminated.

#### 5.1.2 Batch Mode

In batch mode, a syntax error, ALLOCATE, ASSIGN, attribute labelled tape, or DELETE/RENAME error causes OS/32 COPY to terminate. An I/O error causes OS/32 COPY to pause. If OS/32 COPY is continued, it retries the I/O operation which caused the error to arise.

#### 5.1.3 Return Code

If OS/32 COPY terminates normally due to an END command, it issues an end of task message with a return code of 0. If an error occurs when the START parameters are being processed, COPY terminates with an error code of 1. Any other errors causing abnormal termination result in an error code of 2.



## 5.2 MESSAGE SUMMARY

\*\*\* ALLOCATE ERROR sslu FOR fd  
explanatory message

### Where:

- ss is the status
- lu is the logical unit
- fd identifies the device or file
- explanatory message is one of the following:

DUPLICATE FILE NAME  
FILE DESCRIPTOR ERROR  
INSUFFICIENT DISK SPACE  
PRIVILEGE CANNOT BE GRANTED  
VOLUME NOT DIRECT ACCESS  
VOLUME ERROR

### Program Action:

Terminates in batch mode; returns prompt in interactive mode.

### Required User or Operator Action:

The user should correct the problem, then reissue the command if OS/32 COPY is in interactive mode. The program must be restarted if it was in batch mode.

\*\*\* ASSIGN ERROR sslu ON xxxx FOR fd  
explanatory message

**Where:**

- ss is the status
- lu is the logical unit
- xxxx is one of the following:

COMMAND  
INPUT  
LIST  
LOG  
OUTPUT

- fd identifies the device or file
- explanatory message is any of the following:

FILE DESCRIPTOR ERROR  
FILE DOES NOT EXIST  
INSUFFICIENT DISK SPACE  
INSUFFICIENT SYSTEM SPACE  
MISMATCH ON READ-WRITE KEYS  
PRIVILEGE CANNOT BE GRANTED  
VOLUME ERROR

**Program Action:**

Terminates in batch mode, returns prompt in interactive mode. If an assign error is encountered on the work file (lu4) during a RENAME, then 'ON xxx' is not output with the message.

**Required User or Operator Action:**

The user should correct the problem, then reissue the command if OS/32 COPY is in interactive mode. The program must be restarted if it is in batch mode.

\*\*\* ATTRIBUTE ERROR ON xxxx FOR fd  
explanatory message

**Where:**

- xxxx is one of the following:

COMMAND  
INPUT  
LIST  
LOG  
OUTPUT

- fd identifies the device or file
- explanatory message is one of the following:

DEVICE DOES NOT SUPPORT READ  
DEVICE DOES NOT SUPPORT WRITE

**Program Action:**

The file identified by fd is closed by the program. If the error is detected on the log device, logging is disabled and the program continues to execute without the logging feature. For attribute errors on devices other than the log device, the program terminates if it is in batch mode. It outputs a program prompt if it is in interactive mode. If the command device does not support read, the program terminates with end of task code 1.

**Required User or Operator Action:**

If the program was terminated while executing in batch mode, it must be restarted. In interactive mode, the logging feature may be reenabled via the LOG command. In all other instances the command can be reentered, specifying the correct device mnemonic. Alternatively, a new command may be entered.

\*\*\* ATTRIBUTE WARNING ON xxxx FOR fd  
explanatory message

**Where:**

- xxxx is one of the following:

COMMAND  
INPUT  
LIST  
LOG  
OUTPUT

- fd identifies the device or file
- explanatory message is one of the following:

DEVICE DOES NOT SUPPORT BINARY  
DEVICE DOES NOT SUPPORT FORWARD RECORD  
DEVICE DOES NOT SUPPORT BACKSPACE RECORD  
DEVICE DOES NOT SUPPORT FORWARD FILE  
DEVICE DOES NOT SUPPORT BACKSPACE FILE  
DEVICE DOES NOT SUPPORT WRITE FILEMARK  
DEVICE DOES NOT SUPPORT REWIND

**Program Action:**

If the error is detected on the log device, logging is disabled, and the program continues to execute without the logging feature. For attribute warnings on devices other than the log device, the program terminates if it is in batch mode, or outputs a program prompt if it is in interactive mode.

**Required User or Operator Action:**

If the program was terminated in batch mode, it has to be restarted after the device mnemonic has been corrected. In interactive mode, the logging feature may be reenabled via the LOG command. In all other instances the command can be reentered. Alternatively, a new command may be entered.

\*\*\* DELETE ERROR sslu FOR fd  
explanatory message

**Where:**

- ss is the status
- lu is the logical unit
- fd identifies the file
- explanatory message is one of the following:

FILE CURRENTLY ASSIGNED  
FILE DOES NOT EXIST  
INSUFFICIENT SYSTEM SPACE  
PRIVILEGE CANNOT BE GRANTED  
VOLUME NOT DIRECT ACCESS  
VOLUME ERROR

**Program Action:**

Terminates in batch mode; returns prompt in interactive mode.

**Required User or Operator Action:**

The user should correct the problem, then reissue the command if the program is in interactive mode. The program must be restarted if it is in batch mode.

\*\*\* Error - fd ASSIGNED TO INPUT

**Where:**

- fd identifies the file

**Program Action:**

File specified for output is also specified for input in a COPY or VERIFY command. The present command is aborted. The program terminates in batch mode; it outputs a prompt in interactive mode.

**Required User or Operator Action:**

In interactive mode, the user should reissue the command with the appropriate output fd. The program has to be restarted in batch mode.

\*\*\* INPUT FILE MUST BE ENTERED

**Program Action:**

Terminates if it is in batch mode; outputs a new prompt if it is in interactive mode.

**Required User or Operator Action:**

The user can correct the problem by specifying an input file in response to the program prompt. In batch mode, the program must be restarted with the correct command sequence.

\*\*\* INVALID BLOCK/RECORD SIZE

**Where:**

- the value used for blk in the INPUT or OUTPUT command is zero, or the product of lrec and blk exceeds  $2^{16} - 1$  or 32,760 for labelled tapes

**Program Action:**

The program terminates if it is in batch mode. A new prompt is output in interactive mode.

**Required User or Operator Action:**

Restart the program in batch mode with the correct parameters. In interactive mode, reenter the INPUT or OUTPUT command using the correct parameters.



\*\*\* INSUFFICIENT MEMORY -nKB EXTRA REQUIRED

**Where:**

- the value of n reflects the number of additional kb required by the program

**Program Action:**

The program terminates if it is in batch mode. If the program is in interactive mode, a new prompt is output.

**Required User or Operator Action:**

The program should be reloaded with a larger segment size increment.

\*\*\* I/O ERROR lussss ON xxxx FOR fd  
explanatory message

**Where:**

- lu is the logical unit
- ssss is the status halfword
- xxxx is one of the following:

COMMAND  
INPUT  
LIST  
LOG  
OUTPUT

- fd identifies the device or file
- explanatory message is any of the following:

DEVICE UNASSIGNED  
DEVICE UNAVAILABLE  
END OF FILE  
END OF MEDIUM  
ILLEGAL FUNCTION  
PARITY OR RECOVERABLE ERROR  
UNRECOVERABLE ERROR

**Program Action:**

Pauses if error is on the command device. Pauses in batch mode; returns program prompt in interactive mode.

**Required User or Operator Action:**

Appropriate corrective action should be taken if possible. If the program is in batch mode, or if error is on the command device, it may then be continued with the OS CONTINUE command. The OS CANCEL command should be entered if the error condition cannot be corrected. In interactive mode, a prompt will be issued. The OS/32 command CONTINUE may be used to continue execution of the previous command. If it is not possible to correct the error, a new command may be entered in response to the prompt. Any command other than PAUSE or CONTINUE will terminate the previous command.

\*\*\* LABELLED TAPE ERROR ON xxxx  
explanatory message

**Where:**

- xxxx is either INPUT or OUTPUT
- the explanatory message is one of the following:

BLOCK COUNT CHECK FAILS  
BLOCK/RECORD SIZE DIFFERS FROM HEADER LABEL  
EOM ON WRITE TAPE  
FILE IDENTIFIER DOES NOT MATCH  
FILE NOT OPENED  
FILE NOT YET EXPIRED  
INVALID BLOCK/RECORD SIZE  
INVALID COMMUN. AREA FORMAT  
INVALID LABEL FORMAT  
VOLUME IDENTIFIER DOES NOT MATCH

**Program Action:**

Terminates if it is in batch mode, returns a prompt if it is in interactive mode.

**Required User or Operator Action:**

After the error has been corrected, the program has to be restarted if it is in batch mode. The command can be reissued if the program is in interactive mode. Alternatively, a new command may be entered.

\*\*\* LIST FILE MUST BE ENTERED

**Program Action:**

Terminates if it is in batch mode and outputs a new prompt if it is in interactive mode.

**Required User or Operator Action:**

In interactive mode, the user can correct the problem by entering a LIST command with a list fd. If the program is in batch mode, it must be restarted after the command has been corrected.

\*\*\* OUTPUT FILE MUST BE ENTERED

**Program Action:**

Terminates if it is in batch mode; returns a prompt if it is in interactive mode.

**Required User or Operator Action:**

The user can correct the problem by specifying an output file in response to the program prompt. In batch mode the program must be restarted with the correct command sequence.

\*\*\* RECORD LENGTH ERROR ON xxxx  
n BYTES READ, m BYTES EXPECTED

**Where:**

- xxxx is either INPUT or OUTPUT
- n is the number of bytes read
- m is the logical record length

**Program Action:**

Terminates if it is in batch mode; returns a prompt if it is in interactive mode.

**Required User or Operator Action:**

The user should reissue the INPUT or OUTPUT command specifying the correct logical record length if fixed length record processing is desired, or the VARIABLE keyword if variable length records are expected.

\*\*\* RENAME ERROR sslu FOR fd  
explanatory message

Where:

- ss is the status
- lu is the logical unit
- fd identifies the file
- explanatory message is one of the following:

FILE DESCRIPTOR ERROR  
FILE DOES NOT EXIST  
NAME ALREADY EXISTS  
PRIVILEGE CANNOT BE GRANTED

Program Action:

Terminates if it is in batch mode; returns a prompt if it is in interactive mode.

Required User or Operator Action:

The user should correct the problem, then reissue the command if OS/32 COPY is in interactive mode. The program must be restarted if it is in batch mode.

\*\*\* SYNTAX ERROR - xxxx

Where:

- xxxx is one of the following:

COMMAND LINE TOO LONG  
INVALID CHARACTER C' char '  
INVALID CHARACTER X' hex.value '  
INVALID DECIMAL PARAMETER yyyy  
INVALID FILE DESCRIPTOR yyyy  
INVALID FILL yyyy  
INVALID FNAME yyyy  
INVALID HEX PARAMETER yyyy  
INVALID JULIAN DATE yyyy  
INVALID KEY yyyy  
INVALID KEYWORD yyyy  
INVALID PAD yyyy  
INVALID TERMINATOR yyyy  
INVALID VOLID yyyy  
MISSING PARAMETERS(S)  
TOO MANY PARAMETERS  
UNBALANCED PARENTHESIS

- yyyy identifies the invalid parameter
- char is an ASCII character
- hex value is the hexadecimal value of a nonprintable character

Program Action:

The program terminates if it is in batch mode; returns a prompt if it is in interactive mode.

Required User or Operator Action:

After the problem has been corrected, the program must be restarted if it is in batch mode. The command can be reissued if the program is in interactive mode.

END OF TASK CODES:

- 0 - Normal Termination
- 1 - Start Option Error
- 2 - Errors in Batch Mode





APPENDIX A  
COMMAND SUMMARY

ALLOCATE fd, { CONTIGUOUS, fsize [ { keys } ]  
 EC [ [ { bsize } ] [ / [ { isize } ] ] [ / [ { keys } ] ] ]  
 INDEX [ [ { lrecl } ] [ [ { bsize } ] [ / [ { isize } ] ] [ / [ { keys } ] ] ] ]  
 NB [ [ { lrecl } ] [ [ { bsize } ] [ / [ { isize } ] ] [ / [ { keys } ] ] ] ] }

BFILE { INPUT } [ n ]  
 { OUTPUT } [ 1 ]

BRECORD { INPUT } [ n ]  
 { OUTPUT } [ 1 ]

CONTINUE

COPY { \* } { \* } [ { NRECS = { ALL } } ] [ DELETE ]  
 { fd1 } { fd2 } [ { NFILES = { 1 } } ]

DELETE fd [ fd... ]

DISPLAY { \* } [ { NFILES = { 1 } } ]  
 { fd } [ { NRECS = { ALL } } ]

END

FFILE { INPUT } [ { n } ]  
          { OUTPUT } [ , { ALL } ]  
                          [ 1 ]

FILES

FRECORD { INPUT } [ { n } ]  
          { OUTPUT } [ 1 ]

HELP [ { mnemonic } ]  
      [ \* ]

INPUT { \* } [ fd ] [ , [ { lrecl } ] [ { blk } ] [ , VARIABLE ] [ { keys } ] [ LABEL ]  
                  [ , FNAME=fn ] [ , VOLID=id ] [ , PAD=pp ] ]  
  [ 0000 ] [ 1 ]

LIST fd

LOG fd

NOLOG

OPTIONS [ { NOFM } ] [ { NOESFM } ] [ { BINARY } ] [ FILL=xx ]  
          { FM } [ , { PSFM } ] [ { ASCII } ]  
          [ { TERMINATOR=xxxx } ] [ { CONTINUE } ] [ VERR=count ]  
          { NOTERM } [ , { NOCONTINUE } ] [ 0 ]

OUTPUT { \* } [ fd ] [ , [ lrecl ] [ /blk ] [ , VARIABLE ] [ { keys } ] [ LABEL ]  
                  [ , FNAME=fn ] [ , VOLID=id ] [ , PAD=pp ] [ , EXPIRY=yyddd ] ]  
  [ 0000 ]

PAUSE

RENAME oldfd,newfd

REWIND { INPUT }  
          { OUTPUT }

SEND STOP

VERIFY { \* } , { \* } [ ( NRECS = { m } )  
          { fd1 } , { fd2 } ( NFILES = { n } )  
                                  { ALL } ) ]

WFILE { LIST } [ ( n )  
       { OUTPUT } ( ) ]

APPENDIX B  
TASK ESTABLISHMENT AND SIZE INFORMATION

OS/32 COPY executes as a user task (u-task). The approximate size of the program, exclusive of buffer space provided when the task is loaded, is as follows:

PURE segment	31 Kbytes
IMPURE segment	5 Kbytes

The following sequence can be used to build OS/32 COPY as a task:

```
LOG CON:
TITLE COPY32
ESTABLISH TASK
INCLUDE COPY32
MAP CON:
BUILD COPY32
END
```

COPY32 contains its LINK options imbedded within the distributed object file. Those options are:

```
OPT SEG,WORK=X3200,SYSSPACE=XFFFFFF,ACP
```

The OPTION WORK command specifies the number of bytes to be reserved beyond the end of task. The particular value n for "WORK=" can be calculated by:

$$n = 2 * \text{recmax} + 1$$

where:

recmax equals the largest expected record or magnetic tape block size.

Should the user wish to override these options, he need only insert the new option command and parameter(s) after the INCLUDE command and before the BUILD command.



APPENDIX C  
TABLE OF DEVICE CHARACTERISTICS

	CONT. FILES	INDEXED FILES	TTY/KP	TTY/RP	LOCAL CRT	HIGH SP PTRP	LINE PRINTER
Read	X	X	X	X	X	X	
Write	X	X	X	X*	X	X	X
Binary	X	X		X		X	
Wait	X	X	X	X	X	X	X
Random	X	X					
Uncond. Proceed	X	X	X	X	X	X	X
Image	X	X	X	X	X	X	X
Rewind	X	X					
Backsp Record	X	X					
Forwsp Record	X	X					
Write Filemark	X						
Backsp Filemark	X	X					
Inter- active			X		X		
Halt I/O			X	X	X	X	X
Device Codes	0(0)	2(2)	16(10) 17(11) 18(12) 21(15) 22(16) 23(17)	81(51) 82(52) 83(53)	34(22) 36(24) 37(25) 38(26) 39(27)	80(50)	112(70) 113(71) 114(72)

\* Except for Carousel Reader

	CARD READER	CD RDR PUNCH	MV HEAD DISK	FLOPPY DISK	CASS. TAPE	9 TRK MAGTAPE
Read	X	X	X	X	X	X
Write		X	X	X	X	X
Binary	X	X	X	X	X	X
Wait	X	X	X	X	X	X
Random			X	X		
Uncond. Proceed	X	X	X	X	X	X
Image	X	X	X		X	X
Rewind					X	X
Backsp Record					X	X
Forwsp Record					X	X
Write Filemark					X	X
Forwsp Filemark					X	X
Backsp Filemark					X	X
Inter- active						
Halt I/O	X	X			X	X
Device Codes	96(60) 97(61)	105(69)	49(31) 50(32) 51(33) 52(34) 53(35) 54(36)	55(37)	66(42)	64(40) 65(41)



APPENDIX D  
LABELLED TAPE FILE FORMAT SUMMARY

Tape Layouts

A diagrammatic summary of the IBM standard format for files occupying one or more tape volumes is as follows (where each kind of label, e.g. VOL1, HDR1, is an 80 byte record):

Single File Single Volume	Single File Multiple Volumes	Single File Multiple Volumes	
VOL1	VOL1	VOL1	Volume Label
HDR1	HDR1	HDR1	Header Label
HDR2*	HDR2*	HDR2*	
UHL1*	UHL1*	UHL1*	User Header Labels
UHLn*	UHLn*	UHLn*	
FM	FM	FM	
File	First Part of File	Last Part of File	
FM	FM	FM	
EOF1	EOV1	EOF1	File Trailer Labels
EOF2	EOV2	EOF2	
UTL1*	UTL1*	UTL1*	User Trailer Labels
UTLn*	UTLn*	UTLn*	
FM	FM	FM	End of File
FM	FM	FM	

\*Optional labels

A diagrammatic representation of the ANSI standard format for files occupying one or more tape volumes is as follows:

Single File  
Single Volume

VOL1
HDR1
HDR2*
UHL*
UHL*
FM
FILE
FM
EOF1
EOF2*
UTL*
UTL*
FM
FM

Single File  
Multiple Volumes

VOL1
HDR1
HDR2*
UHL*
UHL*
FM
First Part of File
FM
EOV1
EOV2*
UTL*
UTL*
FM
FM

VOL1
HDR1
HDR2*
UHL*
UHL*
FM
Last Part of File
FM
EOF1
EOF2*
UTL*
UTL*
FM
FM

Volume Label

File Header  
Labels

User Header  
Labels

File Trailer  
Labels

User Trailer  
Labels

End of  
File

\* Optional labels

For Labelled Files OS/32 COPY:

1. Requires HDR2, EOF2 and EOVS labels to be present on all input files, and always writes such labels to output files.
2. Skips over (without processing) any number of user labels on input files, and writes none to output files.
3. Checks for only one filemark at the end of a tape volume terminated by one or more end of volume (EOV) labels, but always writes two.
4. Checks, or writes to, only those fields within labels which are a common subset of the IBM and ANSI standards.
5. Assumes all identifiers (ids) within labels to include only characters 0-9, A-Z and -.

### TAPE LABEL PROCESSING

The fields processed within the various labels are as follows:

1. Volume Label (80 bytes)
  - Label Identifier (VOL1)  
(Bytes 1-4)
  - Volume Identifier (6 bytes)  
(Bytes 5-10) (alphanumeric)
2. First Header Label (80 bytes)
  - Label Identifier (HDR1)  
(Bytes 1-4)
  - File Identifier (17 bytes)  
(Bytes 5-21) (left justified,  
space filled)
  - Creation Date (6 bytes)  
(bytes 42-47) ( yyddd)  
(taken from the  
system clock)
  - Expiry Date (6 bytes)  
(Bytes 48-53) ( yyddd)
3. Second Header Label (80 bytes)
  - Label Identifier (HDR2)  
(Bytes 1-4)

- Record Format (F)  
(Byte 5)
  - Block Length (5 bytes, maximum  
(Bytes 6-10) decimal 32,760)
  - Record Length (5 bytes, maximum  
(Bytes 11-15) decimal 32,760)
  - Blocked Record Indicator (B if blocked  
(Byte 39) records,  
blank otherwise)
4. First Trailer Label (80 bytes)
- Label Identifier (EOF1)  
(Bytes 1-4) (EOV1)
  - File Identifier (17 bytes)  
(Bytes 5-21) (as in HDR1)
  - Block Count (6 bytes decimal)  
(Bytes 55-60)
5. Second Trailer Label
- Label Identifier (EOF2)  
(Bytes 1-4) (EOV2)

Extra labels are bypassed but not processed or checked.

APPENDIX E  
THE TAPE LABELLER PROGRAM

To label unlabelled tapes on Perkin-Elmer 32-bit processors, a free-standing tape labeller program is supplied, to be run under OS/32 or MTM. It is a non-overlaid program, supplied in task format.

The program:

1. is supplied with a volume identifier and magnetic tape device name as start parameters;
2. checks that the volume identifier is a 6-character name;
3. rewinds the tape to be labelled, if necessary;
4. writes a Volume Label (VOL1);
5. writes a Header Label (HDR1) which is blank or zero other than the Label Identifier field;
6. writes a tape mark.

The START command is of the form:

$$ST , \left. \begin{matrix} (A) \\ (E) \end{matrix} \right\} 'Volume-Identifier', Device-Name$$

where A and E indicate ASCII and EBCDIC encoding, respectively.

**Example:**

```
ST ,A'600445',MAG1:  
ST ,E'999402',MAG2:
```

The device with the tape to be labelled is assigned to lul automatically by the program.

End of task return codes are:

- 0 No error
- 1 Error

The program's size is 2.0Kb.

Error messages generated by the Tape Labeller Program:

1. Start Parameter Error Messages

Message:

DEVICE NAME IS TOO LONG  
or  
VOLUME IDENTIFIER IS TOO LONG  
or  
VOLUME IDENTIFIER CHARACTER IS NOT 0-9,A-Z OR -  
or  
QUOTES OR COMMA MISSING  
or  
A (ASCII) OR E (EBCDIC) MISSING

Send to:

System Console or MTM Terminal.

Program Action:

Ends with end of task code 1.

Required User or Operator Action:

Rerun with corrected start parameters.

## 2. File Access Error Messages

### Message:

\*\*\* ASSIGN ERROR

explanatory message

where the explanatory message is one of:

DEVICE NAME NOT FOUND

DEVICE DOES NOT SUPPORT WRITE

DEVICE IS UNAVAILABLE

DEVICE IS NOT A TAPE DRIVE

### Send to:

System Console or MTM Terminal.

### Program Action:

Ends with end of task code 1.

### Required User or Operator Action:

Correct and rerun.

## 3. I/O Error Messages

### Message:

\*\*\* I/O ERROR

explanatory message

where the explanatory message is one of:

DEVICE IS UNAVAILABLE

END OF MEDIUM ENCOUNTERED

PARITY ERROR

UNRECOVERABLE WRITE ERROR

**Send to:**

System Console or MTM Terminal.

**Program Action:**

Ends with end of task code 1.

**Required User or Operator Action:**

Correct and rerun.



INDEX

A			
ALLOCATE command	3-2	COPY command	3-1
	4-6		4-12
ALLOCATE error	5-2	COPY operations	
ASCII/BINARY mode	3-32	copying disk files with	
ASSIGN error	5-3	protection keys	3-21
ATTRIBUTE error	5-4	copying multiple files	3-17
ATTRIBUTE warning	5-5	copying part of a file	3-18
		copying variable length	
		records	3-22
		implicit file	
		specification	3-16
B		Copying	
Backward file	3-30	between unlabelled tapes	3-5
Batch operations	3-35	disk file to another	
BFILE command	3-2	file or device	3-3
	4-9	disk files with	
Blocking factors	3-4	protection keys	3-21
BRECORD command	3-2	from labelled tapes	3-6
	4-10	from unlabelled magnetic	
		tapes	3-4
		labelled magnetic tape	3-6
		multiple files with a	
		single command	3-17
		part of a file	3-18
		to and from other devices	3-12
		to unlabelled magnetic	
		tapes	3-3
		variable length records	3-22
		Current file maintenance	4-1
C		D	
Command summary	A-1	Default assignments of list	
Command syntax		and log devices	2-4
comment statements	4-5		3-36
continuation of	4-3	Default values	
fd parameter	4-4	for block size	3-5
mnemonics	4-3	for record length	3-5
parameters	4-4	DELETE command	3-2
Commands			4-16
ALLOCATE	4-6	DELETE error	5-6
BFILE	4-9	Determination of file	
BRECORD	4-10	attributes	4-1
CONTINUE	4-11	Device characteristics	C-1
COPY	4-12	DISPLAY command	3-1
DELETE	4-16		4-17
DISPLAY	4-17	Displaying files	3-26
END	4-19		
FFILE	4-20	E	
FILES	4-22	END command	3-1
FRECORD	4-25		4-19
HELP	4-26	End of file	3-13
INPUT	4-28	End of medium	3-15
LIST	4-31	End of volume	3-14
LOG	4-32	End of volume handling	3-36
NOLOG	4-33	Error handling	3-36
OPTIONS	4-34	batch mode	5-1
OUTPUT	4-37	interactive mode	5-1
PAUSE	4-40		
RENAME	4-41		
REWIND	4-42		
SEND STOP	4-43		
VERIFY	4-44		
WFILE	4-47		
Communicating with OS/32 COPY	2-5		
Concatenating files	3-14		
CONTINUE command	3-1		
	4-11		
CONTINUE/NOCONT option	3-26		

Error handling (Continued)			L	
return code	5-1		Labelled magnetic tape	4-2
			LABELLED TAPE error	5-12
		F,G	Labelled tape file format summary	D-1
Features available			List and log devices default assignments of	3-36
accomplishing data			LIST command	3-2
transfer operations	3-1			3-35
controlling operational environment	3-2			4-31
controlling program flow	3-1		LIST FILE error	5-13
describing files for use	3-1		Loading OS/32 COPY	2-1
modifying COPY or VERIFY operations	3-1		in MTM environment	2-2
obtaining file assignment information	3-2		in OS/32 environment	2-1
obtaining help on command use	3-2		LOG command	3-2
positioning files	3-2			3-35
providing file management functions	3-2			4-32
FFILE command	3-2			
	4-20		M	
File attributes			Message summary	5-2
determination of	4-1		Modes of program operation	2-5
File handling	4-1		batch	2-6
File maintenance			interactive	2-5
current	4-1			
File positioning	3-28		N	
File positioning commands	3-2		NOLOG command	3-2
	4-3			3-35
Filemark recognition	3-32			4-33
Filemarks	3-30		Nonempty output files	
Files			handling of	3-36
displaying	3-26			
FILES command	3-2			
	3-35			
	4-22		O	
FILL character designation	3-34		OPTIONS command	3-1
Forward file	3-30			4-34
FRECORD command	3-2		OS/32 COPY	
	4-25		communicating with	2-5
			features available	3-1
			features supported by	1-1
			general description	1-1
			loading	2-1
			pausing	2-7
			resuming	2-7
			starting	2-3
			stopping	2-6
			OUTPUT command	3-1
				4-37
			Output files	
I/O error	5-11		automatic allocation of	3-20
Implicit file specification	3-16		disk-to-disk	3-20
INPUT command	3-1		tape-to-disk	3-20
	4-28		Output labelled tapes	
Input file defaults	4-2		use of tape labeller	
INPUT FILE error	5-8		program	3-9
INSUFFICIENT MEMORY error	5-10			
INVALID BLOCK/RECORD SIZE error	5-9			
			P,Q	
			Parameters	4-4





PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From \_\_\_\_\_ Date \_\_\_\_\_

Title \_\_\_\_\_ Publication Title \_\_\_\_\_

Company \_\_\_\_\_ Publication Number \_\_\_\_\_

Address \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

FOLD

FOLD

Check the appropriate item.

Error Page No. \_\_\_\_\_ Drawing No. \_\_\_\_\_

Addition Page No. \_\_\_\_\_ Drawing No. \_\_\_\_\_

Other Page No. \_\_\_\_\_ Drawing No. \_\_\_\_\_

Explanation:

FOLD

FOLD

CUT ALONG LINE

Fold and Staple  
No postage necessary if mailed in U.S.A.

STAPLE

STAPLE

FOLD

FOLD



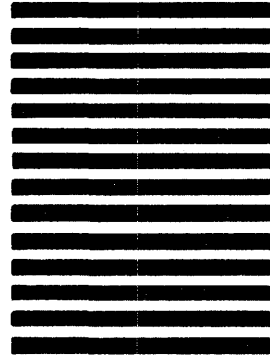
NO POSTAGE  
NECESSARY  
IF MAILED  
IN THE  
UNITED STATES

**BUSINESS REPLY MAIL**  
FIRST CLASS      PERMIT NO. 22      OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

**PERKIN-ELMER**

Computer Systems Division  
2 Crescent Place  
Oceanport, NJ 07757



TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE