Publication Number S29-676

OS/32 COPY

USER GUIDE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport. N.J 07757

PAGE REVISION STATUS SHEET

PUBLICATION NUMBER S29-676

TITLE OS/32 COPY USER GUIDE

REVISION R00 DATE March 1979

PAGE	REV.	DATE	PAGE	REV.	DATE	PAGE	REV.	DATE
i/ii	R00	3/79						
iii thru vi	R00	3/79						
1-1	R00	3/79						
2-1 thru 2-6	R00	3/79						
3-1 thru 3-36	R00	3/79						
4-1 thru 4-39	R00	3/79						
5-1 thru 5-18	R00	3/79						
A-1	R00	3/79						
A-2	R00	3/79						
B-1	R00	3/79						
C-1	R00	3/79						
C-2	R00	3/79						
D-1 thru D-5	R00	3/79						
E-1 thru E-3	R00	3/79						
Index -1 thru Index -4	R00	3/79						

PREFACE

This manual describes the functions and features of the OS/32 COPY program, Part Number 03-215. OS/32 COPY allows users to copy, verify, and display data from disc files or devices.

User Prerequisites

Users of OS/32 COPY should be familiar with the operation of OS/32 and MTM. Users who are working with labelled tapes must also be familiar with the TAPE LABELLER program, Part Number 03-149.

Synopsis of Chapters

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.

Appendices 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.

Related Publications

OS/32 Operator Reference Manual, Publication Number S29-574
OS/32 Multi-Terminal-Monitor (MTM) Reference Manual, Publication Number S29-591
Perkin-Elmer 32-bit Sort/Merge Level II Reference Manual, Publication Number 29-615

System Requirements

OS/32 COPY executes as a user task under OS/32 MT4.1 or higher.

TABLE OF CONTENTS

APPENDICES

CHAPTER 1
OS/32 COPY OVERVIEW

1.1 GENERAL DESCRIPTION OF OS/32 COPY

The OS/32 COPY program provides OS/32 users a comprehensive copy utility to use in an OS/32 or OS/32 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.

- 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

- explicit file positioning

CHAPTER 2
GETTING STARTED

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 operation are discussed.

If you are using an MTM system, it is assumed that you are familiar with the SIGNON/SIGNOFF procedures required to use the system.

2.1 LOADING OS/32 COPY

To make OS/32 COPY operational, it must be established as a task prior to loading. Refer to Appendix B. Once established, it must be loaded from an external storage device (ex. disc, 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 disc 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 you, 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.

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

In this example, both commas are required. The OS/32 Operator's Reference Manual describes the LOAD command in detail.

MTM Environment:

Only the LOAD command is required.

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

"seg-size increment" in the above examples 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 k-bytes 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:

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

where

recmax = largest input record or 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 disc 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

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

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

```
          OS/32 MT:   *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 optimal performance.

2.2 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 as 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 for output is not appropriate for your needs, one of the alternate START forms can be used to specify a different device.

2.2.1 Command Device: COMMAND=fd1

The file descriptor fd1 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:.

2.2.2 Log Device: LOG=fd2

The file descriptor 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 disc file is specified as the log device, the file must already exist.

2.2.3 List Device: LIST=fd3

The file descriptor 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 disc file is specified as the list device, the file must already exist.

2.2.4 START Examples

The following examples illustrate the START command defaults associated with the various START commands:

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.

2.3 COMMUNICATING 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/32 COPY program that you are using;

> is the prompt character indicating 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.

2.3.1 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 disc 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.

2.3.1.1 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.

2.3.1.2 Batch Mode

In batch mode, commands are input from a disc 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.

2.4 STOPPING OS/32 COPY

2.4.1 Terminating the Program

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

```
END
```

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

2.4.2 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
*SEND STOP
```

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

```
*SEND STOP
```

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.

2.5 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	To display 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 OUTPUT	Operands give blocking factor, record size, tape label information, etc.

- 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
  ```

3.2 FUNDAMENTAL COPY OPERATIONS

This section describes the basic capabilities provided by the COPY command. Copying to and from disc 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 Disc File to Another File or Device

To copy the contents of a disc file (fd1) to another disc 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 disc file, you may explicitly allocate a new file or allow OS/32 COPY to automatically allocate the file.

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

3.2.2 Copying Unlabelled Magnetic Tapes

This section describes OS/32 COPY operations on unlabelled magnetic tapes. Section 3.2.3 describes labelled magnetic tapes.

3.2.2.1 Copying to Unlabelled Magnetic Tape

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

```
COPY fd1,fd2
```

Here fd1 is the file descriptor of the input disc file and fd2 is the file descriptor 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 file descriptor 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.

For example, if you want to write blocks of 80 byte records with 10 records per block when copying A.CAL, you would enter:

```
OUTPUT MAG1:,80/10
COPY A.CAL,MAG1:
```

It is possible to avoid entering the file descriptor of the magnetic tape drive twice in the above command sequence. This is explained in the section describing implicit specification of file descriptors.

3.2.2.2 Copying from Unlabelled Magnetic Tape

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 file descriptor 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 disc 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 the Appendix.

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 the Appendix 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 file descriptor 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. 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.

The format of the INPUT command is:

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

where:

fd — is the file descriptor 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.

nl — is from 1 to 6 alphanumeric characters. If more than one volume id is listed, the id's 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 id's.

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 file descriptor 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 non-blank character is a comma is assumed to be continued on the next line. You can enter as many as 20 volume id's 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 disc file. The record length of the input file is 256 and the file identifier is 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 filename 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 id's 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 the Appendix.

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:

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

where:

fd — is the file descriptor 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 identifier, 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 tape's 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 non-blank 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 disc 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 chance 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 file descriptor and fd2 is the output file descriptor. 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 colums 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 one to four 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 disc 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 file mark, you should copy in binary mode. To set binary mode, enter:

```
OPTIONS BINARY
```

3.2.6 Handling of End-of-File, End-of-Volume and End-of-Medium Conditions

In most cases, execution of the COPY command continues until an end-of-file 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 and end-of-medium can interrupt or terminate the COPY command.

3.2.6.1 End-of-File (EOF)

The end-of-file 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 file	after processing of last logical record
Contiguous file (PSEUDO FILE MARK OPTION disabled)	after processing of last physical record
Contiguous file (PSEUDO FILE MARK OPTION enabled)	after processing of pseudo file mark or last physical record.

An end-of-medium condition on contiguous files is treated as an end-of-file. When encountering an end-of-file 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 file marks are recognized on contiguous files. The PSFM option is enabled or disabled using the OPTIONS command:

```
OPTIONS PSFM      (to enable the pseudo file mark option)

OPTIONS NOPSFM    (to disable the pseudo file mark option)
```

The default is PSFM.

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

The NOFM option is set by entering:

```
OPTIONS NOFM
```

When NOFM is set, no filemark is written on the output file when end-of-file 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, end-of-volume is defined on input by an end-of-volume label.

When end-of-volume occurs on input from a labelled tape, a message is logged:

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

The volume identifier xxxxxx names the next volume to be read and fd is the file descriptor 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 end-of-volume message and continues the COPY, VERIFY or DISPLAY operation in progress when CONTINUE is entered.

On output, end-of-volume only occurs 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 idendifier xxxxxx names the next volume to be written and fd is the file descriptor of the output device. An end-of-volume label is written on the tape and the tape is rewound.

For unlabelled tapes, one filemark is written followed by an end-of-volume identifier 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 end-of-volume.

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

3.2.6.3 End-of-Medium (EOM)

End-of-medium 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 end-of-volume condition. When writing to a magnetic tape, the end-of-tape marker is interpreted as end-of-medium.

When OS/32 COPY encounters an end-of-medium condition while copying from a contiguous file, it issues a message:

$$\text{END OF MEDIUM - n RECORD(S)} \left\{\begin{array}{l}\text{COPIED}\\ \text{VERIFIED}\\ \text{DISPLAYED}\end{array}\right\}$$

where n is the number of records processed to the point of end-of-medium. OS/32 COPY then treats the end-of-medium condition as an end-of-file. The number of records is only output if n is greater than Ø.

When OS/32 COPY encounters an end-of-medium condition on output to a disc file, it treats the condition as an 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 disc files; and, variable length record processing.

3.3.1 Implicit Specification of Input and Output Files

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

Similarly, by entering a file descriptor 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 the 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 * 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 may 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 command sequence cound be:

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

The use of asterisks to implicitly designate files is optional. You may enter file descriptors 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 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, which ever 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 of 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 end-of-volume 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 end-of-volume is encountered, whichever comes first. If ALL is entered, copying or verifying continues until end-of-volume 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 end-of file 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

When copying to a disc file you may want to allocate a file and then immediately copy to the file. OS/32 COPY performs the allocation and the copy for you if you enter as the output file the file descriptor of a non-existent file. The file allocated has the same attributes as the input file. An indexed file has an index block size and data block size of 1.

Example: If you want to copy the file GOOD.TSK to a file of the same name on volume M301, you enter:

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

If the input file is a contiguous file, the output file is allocated as a contiguous file of the same size.

You may also cause an existing output disc file to be deleted and reallocated. You do this by entering the DELETE keyword after the output file descriptor. 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 Disc Files with Protection Keys

If you want to COPY, VERIFY, or DISPLAY disc files that have non-zero 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 tapes or paper tapes having variable length records. To do this, you enter the INPUT command with the file descriptor of the tape drive, the keyword VARIABLE, and the record size of the largest record on the tape.

For example, suppose you want to copy a tape mounted on MAG1;, where the largest record may be 256 bytes, to a disc 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 hexidecimal 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: is 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 are found which 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 disc 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 a 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 multivolume verification, the volumes must be remounted.

The following examples show how to verify files following a COPY. Notice that the file descriptors, once entered in the COPY command, do not have to be re-entered.

Example: Copy and verify disc 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 disc 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 file marks, followed by forwarding one file mark 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 end-of-file 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 disc 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 disc 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 which does 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 end-of-file, you should set the CONTINUE option. Enter:

```
OPTIONS CONTINUE
```

Since verification continues until end-of-file, 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 non-interactive 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 disc file A.OBJ, enter:

```
DISPLAY A.OBJ
```

If you want to DISPLAY a file residing on magnetic tape or a disc file which has non-zero 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 identifier 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 * 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:

```
DISPLAY[fd] ⎧[,NFILES=n]⎫
            ⎩[,NRECS=n] ⎭
```

Example: Suppose an unlabelled tape mounted on MAG1: contains 3 files separated by file marks. 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 the Appendix.

The commands and their formats are:

```
BFILE   ⎧INPUT ⎫[,n]           Backspace over a
        ⎩OUTPUT⎭               filemark

BRECORD ⎧INPUT ⎫ [,n]          Backspace a record
        ⎩OUTPUT⎭

FFILE   ⎧INPUT ⎫ [,n]          Forward space over a
        ⎩OUTPUT⎭ [,ALL]        filemark

FRECORD ⎧INPUT ⎫ [,n]          Forward space a record
        ⎩OUTPUT⎭

REWIND  ⎧INPUT ⎫               Position to beginning
        ⎩OUTPUT⎭               of file/device

WFILE   ⎧LIST  ⎫ [,n]          Write a filemark
        ⎩OUTPUT⎭
```

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 and Backward File

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 or an end-of-file, or an end-of-volume condition is encountered. In order to skip forward to the end-of-volume, 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 beyound 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 file marks 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 end-of-file 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 end-of-file marks on MAG2:, enter:

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

3.7 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 end-of-file 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 end-of-file 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/) while others support both BINARY and ASCII. The Appendix 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/NOPSFM option determines whether or not a pseudo filemark (X'1313' at the beginning of a record), is recognized as an end-of-file on contiguous disc files.

Thr NOPSFM 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 TET/32. To copy it to magnetic tape:

```
OPTIONS NOPSFM,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 one to four characters as an end-of-file 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 end-of-file. The form of the command is:

```
OPTIONS TERMINATOR=xxxx
```

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

```
OPTIONS NOTERM
```

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

```
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 3-80. 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, verifying continues until an end-of-file 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
```

Entering:

```
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              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 end-of-volume handling, (3) Handling of non-empty output files in the COPY command, (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 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 end-of-volume 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 end-of-volume occurred is resumed.

3.8.3 Handling of Non-empty Output Files in the COPY Command

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

If you designate a non-empty 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 non-direct access device which can be designated by an OS/32 file descriptor. Occasionally, a particular kind of device or direct access file is indicated. This should be clear from the context and from designations such as "disc 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 a file descriptor 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 file descriptor as the first parameter of the INPUT, COPY, VERIFY, or DISPLAY commands. A file is established as the current output file by entering its file descriptor 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 I/O, interactive/non-interactive, capable of being rewound, support for filemarks, support for backspace and forwardspace 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 disc 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.

The defaults for input files are as follows:

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 id, if you enter the required values in the INPUT command. For output tapes, the expiration date and volume id 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 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 [ volid] ON (INPUT ) FOR fd
                                  (OUTPUT)
```

and rewinds the tape. Volid is output if volid's have been specified.

In batch mode, the program pauses so that you may 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 volume ids 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 file descriptor in COPY and VERIFY; the input file descriptor in DISPLAY and INPUT; the output file descriptor in OUTPUT; and all parameters in ALLOCATE, and 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 Parameter

If a command requires an input or output file descriptor to be entered, and you wish to specify the current input or output file, you may enter an asterisk in place of the file descriptor. 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 file descriptor mean that an asterisk or file descriptor must be entered:

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

4.4 OS/32 COPY COMMANDS

The following sections describe each OS/32 COPY command.

4.4.1 ALLOCATE Command

This command allocates a contiguous or indexed file on a disc. The command format can be one of the following:

```
ALLOCATE fd, CONTIGUOUS,fsize [,keys]

ALLOCATE fd, INDEX [,[lrec][/[bsize2[/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. The keys are specified as a hexadecimal halfword; the first byte is the write key, the second byte is the read key. The default is zero for both keys.

lrecl   is the logical record length of the indexed file. The default is 126 bytes.

bsize   is the data block size in sectors of the indexed file. The default is 1.

isize   is the size of an indexed block in sectors. The default is 1.

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.
```

4.4.2 BFILE Command

This command skips backwards over one or more filemarks.

The command format is

$$\underline{\text{BF}}\text{ILE} \left\{ \begin{array}{l} \underline{\text{I}}\text{NPUT} \\ \underline{\text{O}}\text{UTPUT} \end{array} \right\} [,n]$$

where:

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.

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 end-of-file 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 PSFM option 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.
```

4.4.3 BRECORD Command

This command skips backwards over one or more records.

The command format is:

```
BRECORD  (INPUT )  [,n]
         (OUTPUT)
```

where:

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.

BRECORD skips records until n records have been skipped or until a filemark, end-of-medium, or end-of-file 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 non-verify condition under option NOCONT, the SEND STOP command, or to an end-of-volume condition on magnetic tape. CONTINUE is only effective in interactive mode. If entered in batch mode, the command has no effect.

The command format is:

```
                    CONTINUE
```

4.4.5 COPY Command

4.4.5.1 Command Format

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:

$$\underline{\text{COP}}\text{Y}\begin{Bmatrix}*\\ \text{fd1}\end{Bmatrix},\begin{Bmatrix}*\\ \text{fd2}\end{Bmatrix}\quad\begin{Bmatrix}{[},\underline{\text{NR}}\text{ECS=m}]\\ {[},\underline{\text{NF}}\text{ILES=n}]\end{Bmatrix}\qquad{[},\underline{\text{DE}}\text{LETE}]$$

where:

fd1 is the file descriptor of the input file

fd2 is the file descriptor of the output file

* 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 end-of-file.

n is a decimal number up to 6 digits in length specifying the number of files to be copied or the word <u>ALL</u> indicating that all files up to end-of-volume are to be copied. The default is 1 file.

DELETE if entered, indicates that if fd2 is an existing disc file, it is to be deleted and reallocated according to the attributes of fd1.

4.4.5.2 Default Allocation and Record Lengths

When the input file is a disc 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 disc 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.

4.4.5.3 Existing Output Disc Files

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

```
NON-EMPTY 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 disc 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.

4.4.5.4 NFILES and NRECS

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 end-of-file 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 end-of-volume is encountered on input. End-of-volume 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 end-of-volume occurs on input.

4.4.5.5 End-of-Medium

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

When physical end-of-medium is detected on an output tape and there are remaining input records to be copied, a filemark, and an end-of-volume 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 end-of-volume 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 only output 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 file marks. When end-of-medium is encountered on an input tape, COPY continues to read until an end-of-volume indicator is read. End-of-medium on a contiguous file is treated like an end-of-file on input, and as an error on output.

4.4.5.6 Copying Records With Different Record Lengths

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 option in the OPTIONS command. The default is a hexadecimal 20 (ASCII blank).

4.4.5.7 Completion of COPY command

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

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

where n is the number of files copied.

4.4.5.8 Examples.

`COPY A.CAL ,B.CAL`	Copy one disc file to another.
`COPY A.CAL,B.CAL,DELETE`	Delete and reallocate output file; copy file A.CAL to B.CAL.
`COPY MAG1:,MAG2:,NFILES=ALL`	Copy all files from one tape to another.
`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.

4.4.6 DELETE Command

This command is used to delete a disc file. The command format is

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

where:

fd identifies the file to be deleted.

A file may only be deleted if its read and write protection keys are zero and it is not currently assigned to any Logical Unit 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 Logical Unit 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.

The command format is:

$$\underline{\text{DI}}\text{SPLAY}\begin{Bmatrix}*\\ \text{fd}\end{Bmatrix}\begin{Bmatrix}[,\underline{\text{NF}}\text{ILES=m}]\\ [,\underline{\text{NR}}\text{ECS=n}]\end{Bmatrix}$$

where:

fd is the file descriptor of the file to be displayed. If * is entered the current input file is displayed. If a file descriptor 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 end-of-volume. 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 end-of-file.

The display is output to the list device.

No filemarks are written when input end-of-file 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:

4.4.8 END Command

This command terminates OS/32 COPY. The command format is:

```
                    END
```

4.4.9 FFILE COMMAND

This command skips forward over one or more filemarks.

The command format is

```
FFILE  ⎧INPUT ⎫  ⎧[,n]  ⎫
       ⎩OUTPUT⎭  ⎩[,ALL]⎭
```

where:

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 end-of-volume is reached.

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 end-of-medium is encountered.

When the keyword ALL is entered, command execution ends when end-of-volume 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 end of volume on the output device.

4.4.10 FILES Command

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

```
                              FILES
```

The display has the following form:

```
INPUT    fd        fname       lrecl    record no     pad
                   VARIABLE    /blk

         VOLID:    xxxxxx      xxxxxx   xxxxxx    xxxxxx    xxxxxx

OUTPUT   fd        fname       lrecl    record no     pad  expiry
                   VARIABLE    /blk                         -date

         VOLID:    xxxxxx      xxxxxx   xxxxxx    xxxxxx    xxxxxx

LIST     fd

COMMAND  fd

LOG      fd
```

where:

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 file mark on a contiguous file.

expiry-date — is expiration date for output labelled tapes

xxxxxx — the volume ids 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 date of Jan. 1, 1980, and volume identifiers 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   (End-of-medium 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.

The command format is

$$\underline{\text{FR}}\text{ECORD} \left\{\begin{matrix}\underline{\text{I}}\text{NPUT} \\ \underline{\text{O}}\text{UTPUT}\end{matrix}\right\}[,n]$$

Where:

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 is 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.

FRECORD skips records until n records are skipped or until a filemark of end-of-medium 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.

4.4.12 INPUT Command

This command allows you to specify information about the input file. Its use is required when labelled or blocked unlabelled tapes, or disc files with non-zero 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.

The command format is:

```
INPUT  { *  } [,lrecl[/blk]] [,VARIABLE] [,KEYS=xxxx] [,LABEL]
       { fd }

       [,FNAME=fn] [,VOLID=id] [,PAD=pp]
```

where:

fd — is the file descriptor of the input file. This file becomes the current input file. If fd is a disc file, it must already exist.

* — is used to designate the current input file.

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 32760 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.

xxxx — 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 identifier for a labelled tape. If not entered, the file identifier is ignored.

id — is a list of volume ids. 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, volume ids are not validated. If entered, input tapes must be mounted in the order specified by the volume ids.

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 keys.

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

The tape on MAG1: becomes the current input file. The record 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 identifier is ALPHA; two tape volumes are to be mounted for input; the volume ids 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 volume id 999888; it must have 80 byte records blocked 10 records per block or the tape will not be read. The file identifier 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.

4.4.13 LIST Command

This command changes the list device. The command format is:

```
LIST fd
```

where:

fd is the file descriptor of the new list device.

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

Example:

```
LIST PR1:  use PR1: as the list device.
```

4.4.14 LOG Command

This command changes the log device. The command format is:

```
LOG fd
```

where:

fd   is the file descriptor of the new log device.

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

Example:

```
LOG PR1:  use PR1: as the log device.
```

4.4.15 NOLOG Command

This command suspends logging until a LOG command is entered. The command format is:

```
                    NOLOG
```

4.4.16 OPTIONS Command

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

```
OPTIONS  { [ NOFM ] } { [, NOPSFM ] } { [, BINARY ] }
         {   [FM]   } {  [,PSFM]    } {  [,ASCII]   }

[,FILL=xx]  { [, TERMINATOR=xxxx ] } { [, CONTINUE ] }
            {  [,NOTERM]           } {  [,NOCONT]    }
```

where:

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
This option sets a sequence of characters to be recognized as an end-of-file indicator. xxxx consists of one to four ASCII characters with no imbedded blanks. For an input record to be recognized as end-of-file, 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
NOCONT
These options control OS/32 COPY actions following a mismatch during a VERIFY operation. When CONTINUE is in effect, OS/32 COPY continues verifying the files. When NOCONT is in effect, verifying ends if a mismatch occurs.

The default values are:

FM
PSFM
ASCII
FILL=20
TERMINATOR=/*
NOCONT

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 end-of-file is encountered in the input file or an end-of-medium 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 end-of-medium 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 end-of-volume.

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.

4.4.17 OUTPUT Command

This command allows you to specify information about the output file. Its use is required when labelled or blocked unlabelled tapes, disc files with non-zero 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.

The command format is:

```
     OUTPUT { * } [,lrecl[/blk]][,VARIABLE][,KEYS=xxxx][,LABEL]
            {fd }

            [,FNAME=fn][,VOLID=id][,PAD=pp][,EXPIRY=yyddd]
```

where:

fd — is the file descriptor of the output file which becomes the current output file. If fd is a disc file, it must already exist.

* — is used to designate the current output file.

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 32760 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.

xxxx — are the write-read keys, two hexadecimal bytes. The default is 0000.

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

fn — is a 1 to 17 character file identifier (fdid) for a labelled tape. If not entered, the file identifier is written as blanks.

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

pp — is a digit two 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.

The file descriptor 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 identifier is BETA, and two tape volumes are to be mounted for output; the volume ids 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 volume
          id 999888; it will be written with 80 byte
          records blocked 10 records per block.  No
          file identifier 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 identifier 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.
```

4.4.18 PAUSE Command

This command causes OS/32 COPY to pause. The command format is:

```
                    PAUSE
```

MTM or OS commands may then be entered. A CONTINUE command causes OS/32 COPY processing to resume.

4.4.19 RENAME Command

This command is used to change the name of a disc file. The command format is:

```
RENAME oldfd,newfd
```

where:

oldfd   is the old file name

newfd   is the new file name

A file may only be renamed if its read and write protection keys are zero and it is not currently assigned to any Logical Unit of any task. If the file specified is the current input, log, or list file for OS/32 COPY, the corresponding Logical Unit 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 file descriptors. However, under MTM, the new file name can have a group file or system file indicator (/G or /S).

Example:

```
RENAME Al.CAL,Al.MAC     Change the name of Al.CAL
                         to Al.MAC
```

4.4.20 REWIND Command

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

The command format is:

$$\underline{\text{REW}}\text{IND} \begin{Bmatrix} \underline{\text{I}}\text{NPUT} \\ \underline{\text{O}}\text{UTPUT} \end{Bmatrix}$$

where:

INPUT specifies that the command applies to the current input file.

OUTPUT specifies that the command applies to the current output file.

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

Example:

REWIND OUTPUT	rewind the current output file
REW IN	rewind the current input file

4.4.21 SEND STOP Command

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

The command format is:

<u>SEND</u> <u>STOP</u>

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/32MT:

TASK COPY	Select copy as the current task.
SEND STOP	Stop execution of the current command.

4.4.22 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.

4.4.22.1 Command Format

The command format is:

$$\text{VERIFY} \left\{\begin{matrix} * \\ \text{fd1} \end{matrix}\right\} , \left\{\begin{matrix} * \\ \text{fd2} \end{matrix}\right\} \left\{\begin{matrix} [,\underline{\text{NR}}\text{ECS=m}] \\ [,\underline{\text{NF}}\text{ILES=n}] \end{matrix}\right\}$$

where:

fdl is the file descriptor of the input file

fd2 is the file descriptor of the output file

* indicates the current input or output file

m is a decimal number up to 6 digits specifying the number of records to be verified. The default is all records until end of file 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 end-of-volume are to be verified. The default is 1 file.

4.4.22.2 Effect of CONTINUE/NOCONT Option

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 end-of-file on one file is not matched by an end-of-file on the other. All records which do not match are written on the list device.

4.4.22.3 NFILES and NRECS

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 end-of-file 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 end-of-volume is encountered on input. End-of-volume is defined by two consecutive filemarks on magnetic tape, or end-of-medium on contiguous files. After each file has been verified, a message indicating the number of records which have been verified in that file is logged.

If NFILES=ALL is entered, all files are verified until end-of-volume occurs on input.

4.4.22.4 Verifying Records Having Different Record Lengths

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 non-verify 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.

4.4.22.5 Completion of VERIFY Command

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

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

where n is the number of files verified.

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

```
VERIFY COMPLETE - n RECORD VERIFIED
```

4.4.22.6 Examples

VER A.CAL,CR:	Verify a disc 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.

4.4.23 WFILE Command

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

The command format is

$$\underline{\text{WF}}\text{ILE} \begin{Bmatrix} \underline{\text{LI}}\text{ST} \\ \underline{\text{O}}\text{UTPUT} \end{Bmatrix} \text{ [,n]}$$

where:

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.

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 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 DISC 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 DISC 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 (LU 4) during a RENAME then 'ON xxxx' 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 and 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 file descriptor. 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 32760 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 Operation Action:

Restart the program in batch mode with the correct parameters in interactive mode, re-enter 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 COPY 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 file descriptor. 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
UNBLANACED 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]

ALLOCATE  fd,INDEX[,[lrecl][/[bsize][/isize]][,keys]]

        ( INPUT  )
BFILE   {        }  [,n]
        ( OUTPUT )

          ( INPUT  )
BRECORD   {        }    [,n]
          ( OUTPUT )

CONTINUE

       (  *  )   (  *  )   ( [,NRECS=m]  )
COPY   {     } , {     }   {             }    [,DELETE]
       ( fd1 )   ( fd2 )   ( [,NFILES=n] )

DELETE  fd[,...]

          ( *  )   ( [,NFILES=m] )
DISPLAY   {    }   {             }
          ( fd )   ( [,NRECS=n]  )

END

         ( INPUT  )   ( [,n]  )
FFILE    {        }   {       }
         ( OUTPUT )   ( [ALL] )

FILES

          ( INPUT  )
FRECORD   {        }    [,n]
          ( OUTPUT )
        ┌    ┐
INPUT  {  *   } [,lrecl[/blk]] [,KEYS=xxxx] [,LABEL]
       {  fd  }
        └    ┘
      [,VARIABLE] [,FNAME=]  [,VOLID=]  [,PAD=]

LIST fd

LOG fd

NOLOG

          ┌       ┐  ┌           ┐ ┌            ┐
OPTIONS  { [NOFM]  } { , [NOPSFM] } { , [BINARY] }
         {  [FM]   } {   [PSFM]   } {   [ASCII]  }
          └       ┘  └           ┘ └            ┘
                      ┌                    ┐ ┌              ┐
    [,FILL=xx]       { , [TERMINATOR=xxxx] } { , [CONTINUE] }
                     {   [NOTERM]          } {   [NOCONT]   }
                      └                    ┘ └              ┘

         ┌    ┐
OUTPUT  {  *   } [,lrecl[/blk]] [,KEYS=xxxx] [,LABEL]
        {  fd  }
         └    ┘
      [VARIABLE] [,FNAME=] [,VOLID=] [,PAD=] [,EXPIRY=]

PAUSE

RENAME oldfd,newfd

         ┌        ┐
REWIND  {  INPUT   }
        {  OUTPUT  }
         └        ┘

SEND STOP

          ┌     ┐   ┌     ┐  ┌              ┐
VERIFY   {  *    } , {  *  } { [,NRECS=m]    }
         {  fd1  }   { fd2 } { [,NFILES=n]   }
          └     ┘   └     ┘  └              ┘
         ┌        ┐
WFILE   {  LIST    } [,n]
        {  OUTPUT  }
         └        ┘
```

APPENDIX B

TASK ESTABLISHMENT AND SIZE INFORMATION

OS/32 COPY executes as a user 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 establish OS/32 COPY as a task from an object version named COPY32.OBJ:

```
*LOAD TET32
*ASSIGN 5,CON:
*ASSIGN 7,CON:
*START
 ESTABLISH TASK
 INCLUDE COPY32
 EXPAND n
 BUILD TASK,COPY32
 MAP PR:
 END
```

OS/32 COPY is segmented into pure and impure code for shared use with Operating Systems which support segmentation. If the program is to be established as a sharable task the ESTABLISH TASK command should be modified to ESTABLISH TASK,PURE.

The argument n of the EXPAND command specifies the number of 256-byte blocks to be reserved beyond the end of the task. The value of n can be calculated by:

$$n = \frac{2 * recmax}{256} + 1$$

where

recmax = largest expected record
or magnetic tape block size

The size of workspace thus created may be overridden at load time.

APPENDIX C

TABLE OF DEVICE CHARACTERISTICS

	CONT.	INDEXED	TTY/KP	TTY/RP	LOCAL	HIGH SP	LINE
	FILES	FILES			CRT	PTRP	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	X					
Backsp Filemark	X	X					
Inter-active			X		X		
HaltI/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 DISC	FLOPPY DISC	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						
HaltI/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

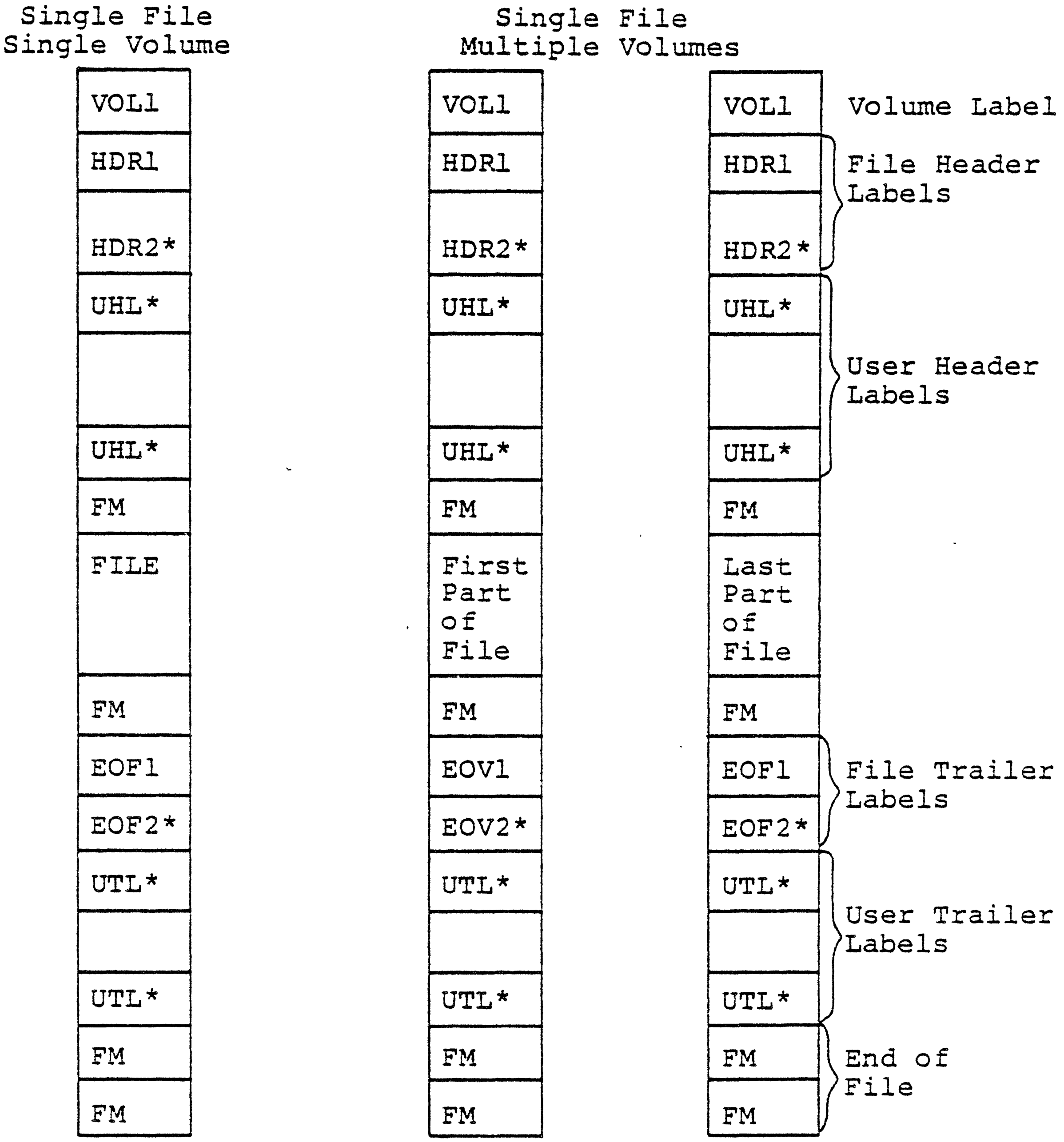
Tape Layouts

A diagramatic 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		
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	

*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	Single File Multiple Volumes		
VOL1	VOL1	VOL1	Volume Label
HDR1	HDR1	HDR1	File Header Labels
HDR2*	HDR2*	HDR2*	
UHL*	UHL*	UHL*	User Header Labels
UHL*	UHL*	UHL*	
FM	FM	FM	
FILE	First Part of File	Last Part of File	
FM	FM	FM	
EOF1	EOV1	EOF1	File Trailer Labels
EOF2*	EOV2*	EOF2*	
UTL*	UTL*	UTL*	User Trailer Labels
UTL*	UTL*	UTL*	
FM	FM	FM	End of File
FM	FM	FM	

* optional labels

For Labelled Files OS/32 COPY:

1. Requires HDR2, EOF2 and EOV2 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 file mark at the end of a tape volume terminated by one or more 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 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 (Bytes 1-4)	(VOL1)
- Volume Identifier (Bytes 5-10)	(6 bytes) (alphanumeric)
2. First Header Label	(80 bytes)
- Label Identifier (Bytes 1-4)	(HDR1)
- File Identifier (Bytes 5-21)	(17 bytes) (left justified, space filled)
- Creation Date (bytes 42-47) (taken from the system clock)	(6 bytes) (yyddd)
- Expiry Date (Bytes 48-53)	(6 bytes) (yyddd)
3. Second Header Label	(80 bytes)
- Label Identifier (Bytes 1-4)	(HDR2)
- Record Format (Byte 5)	(F)
- Block Length (Bytes 6-10)	(5 bytes, maximum decimal 32760)
- Record Length (Bytes 11-15)	(5 bytes, maximum decimal 32760)
- Blocked Record Indicator (Byte 39)	(B if blocked records, blank otherwise)
4. First Trailer Label	(80 bytes)
- Label Identifier (Bytes 1-4)	(EOF1) (EOV1)
- File Identifier (Bytes 5-21)	(17 bytes) (as in HDR1)

- Block Count (Bytes 55-60)	(6 bytes decimal)

5. Second Trailer Label

- Label Identifier (Bytes 1-4)	(EOF2) (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:

```
          A
ST ,         'Volume-Identifier',Device-Name
          E
```

where A and E indicate ASCII and EBCDIC encoding respectively.

For example:

```
ST ,A'600445',MAG1:

ST ,E'999402',MAG2:
```

The device with the tape to be labelled is assigned to LU 1 automatically by the program.

EOT return codes are:

0 No error;

1 Error.

The program's size is 2.0 Kb.

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:

Re-run 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 re-run.

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

INDEX (cont'd)

INDEX (cont'd)

INDEX (cont'd)