

HP 3000 Computer System



FCOPY Reference Manual



5303 STEVENS CREEK BLVD., SANTA CLARA, CALIFORNIA 95050

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

NOTICE

The information contained in this document is subject to change without notice.

HEWLETT-PACKARD MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. Hewlett-Packard shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance or use of this material.

Hewlett-Packard assumes no responsibility for the use or reliability of its software on equipment that is not furnished by Hewlett-Packard.

This document contains proprietary information which is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced or translated to another program language without the prior written consent of Hewlett-Packard Company.

LIST OF EFFECTIVE PAGES

The List of Effective Pages gives the most recent date on which the technical material on any given page was altered. If a page is simply re-arranged due to a technical change on a previous page, it is not listed as a changed page. Within the manual, changes are marked with a vertical bar in the margin.

Fifth Edition Feb 1978

PRINTING HISTORY

New editions incorporate all update material since the previous edition. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The date on the title page and back cover changes only when a new edition is published. If minor corrections and updates are incorporated, the manual is reprinted but neither the date on the title page and back cover nor the edition change.

First Edition	Oct 1973	
Second Edition	Mar 1975	
Third Edition	Jul 1975	
Fourth Edition	Jun 1976	
Update Package.	Dec 1976	
Reprint, Updates Incorporated.	Mar 1977	
Update Package.	Aug 1977	32212A.02
Fifth Edition	Feb 1978	32212A.03
Update Package.	Jun 1978	32212A.03
Update Incorporated	Sept 1978	32212A.03

PREFACE

This manual describes the HP 3000 File Copier, FCOPY/3000, which enables you to copy files or selected portions of files from any supported input device to any supported output device. FCOPY runs under the control of either the Multiprogramming Executive operating system (MPE-C) on an HP 3000 Computer System or the Multiprogramming Executive II operating system (MPE-II) on an HP 3000 Series II Computer System. This manual applies to both operating environments.

In order to use FCOPY you should have a working knowledge of the following:

- How to run jobs and sessions
- The HP 3000 accounting structure (the relationship between accounts, groups, and files)
- File/device relationships
- File security (file access restrictions and lockwords)
- The various file characteristics and how to specify or modify them.

In addition, if you are using a system which is not attended by a console operator you should be familiar with some of the console operator commands and how to enter them through the system console.

You may need to consult the following manuals:

- *Using the HP 3000* (HP part number 03000-90121) provides a beginner's tutorial introduction to the use of the HP 3000.
- *MPE/3000 Operating System Reference Manual* (32000-90002) describes the MPE-C operating system.
- *MPE/3000 Console Operator's Guide* (32000-90004) describes the MPE-C console operator's capabilities and commands.
- *HP 3000 Series II Computer System General Information Manual* (30000-90008) provides an overview of the HP 3000 Series II Computer System and the MPE-II environment.
- *MPE Commands Reference Manual* (30000-90009) describes the MPE-II commands.
- *MPE Intrinsic Reference Manual* (30000-90010) describes the MPE-II system intrinsics.
- *HP 3000 Series II Computer System Console Operator's Guide* (30000-90013) describes the MPE-II console operator's capabilities and commands.
- *KSAM/3000 Reference Manual* (30000-90079) includes a description of the use of FCOPY with KSAM files.

If your HP 3000 runs under the control of the MPE-C operating system, the keywords listed below cannot be used in the FCOPY function list:

EBCDIKIN EBCDIKOUT KEY= NOKSAM

Likewise, the (dfile,kfile) parameter of the TO= keyword is not permitted.

Of course, even if your system operates under the control of the MPE-III, the KEY= and NOKSAM keywords do not function if your system does not have the KSAM utility package.

This manual is arranged in four sections and three appendices. Section I introduces you to FCOPY, summarizes its capabilities, and presents some examples illustrating the use of FCOPY. Section II describes all of the functions which can be specified in an FCOPY command and includes examples of the use of each function. Section III presents nine complete examples illustrating the use of FCOPY in both batch jobs and interactive sessions. Section IV presents all of the FCOPY warning and error messages and includes a brief description of each message's meaning and the recommended corrective action. Appendix A describes how to use FCOPY in conjunction with Interactive Display Terminal peripherals (the magnetic tape cartridges and the optional line printer). Appendix B summarizes the default file definition values that are used when FCOPY opens a specified "from" or "to" file. Appendix C contains three code conversion tables. The first is an ASCII/EBCDIC table, the second is a JIS/EBCDIK table, and the last is an ASCII/BCDIC table.

CONTENTS

Section I	Page	Section III	Page
INTRODUCING FCOPY/3000		EXAMPLES	3-1
Examples	1-4		
Card Reader to Line Printer (Batch Job)	1-6		
Disc to Line Printer (Batch Job)	1-7		
Disc to Disc (Interactive Session)	1-9		
Disc to Terminal (Interactive Session)	1-9		
Section II	Page	Section IV	Page
FCOPY FUNCTIONS		FCOPY WARNING AND ERROR MESSAGES	
File Attribute Requirements	2-1	Two Error Types	4-1
Interrupting FCOPY	2-2	File Information Display	4-2
Syntax	2-3	Warning and Error Messages	4-3
Copy	2-4		
Comments	2-4		
Carriage Control/No Carriage Control	2-6	Appendix A	Page
Examples	2-9	HP 2644A TERMINAL PERIPHERALS	
Character Translate	2-10	Examples	A-2
Comments	2-11		
Examples	2-12		
Compare	2-13	Appendix B	Page
Examples	2-13	DEFAULT FILE DEFINITION VALUES	
Deblock	2-14	FROM=card reader	B-2
Files From Other Systems	2-14	FROM=paper tape reader	B-3
Display Character Symbols	2-17	FROM=magnetic tape	B-4
Examples	2-19	FROM=named disc file	B-5
Display Numerical Codes	2-20	FROM=\$CTUL or \$CTUR	B-6
Examples	2-21	TO=card punch	B-7
Ignore Errors	2-23	TO=paper tape punch	B-8
Multiple Tape File Copying	2-24	TO=magnetic tape	B-9
Examples	2-26	TO=named disc file	B-10
New File	2-28	TO=named disc file;NEW	B-11
Non-KSAM	2-29	TO=line printer	B-12
Examples	2-29	TO=\$CTUL or \$CTUR	B-13
No User Labels	2-30	TO=\$HARD	B-14
Examples	2-30		
Skip End-Of-File	2-31		
examples	2-33	Appendix C	Page
Specify Key	2-34	CODE CONVERSION TABLES	
Examples	2-34	ASCII/EBCDIC Conversion Table	C-2
Subsets	2-35	JIS/EBCDIK Conversion Table	C-4
Comments	2-36	ASCII/BCDIC Conversion Table	C-6
Examples	2-36		
Upshift	2-38		
Verify	2-39		
Examples	2-39		

TABLES

Title	Page
FCOPY Functions	1-2
FCOPY Warning and Error Messages	4-2
Code Conversion Tables	C-2 to C-7

FCOPY/3000 is a utility program within an HP 3000 Multiprogramming Executive operating system environment that makes it possible for you to copy data from one file to another. With FCOPY you can copy the content of an entire file or a selected portion of a file from any supported input device to any supported output device. FCOPY is very convenient for making multiple copies of files, for making account-independent magnetic tape copies of disc files (this makes it easier to move files from one HP 3000 to another), or for transferring programs or data from one medium to another (such as from punched cards or magnetic tape to disc). In addition, FCOPY has several powerful features that extend its usefulness far beyond these basic functions. The various FCOPY functions are summarized in Table 1-1 and some of the more important ones are discussed in the following paragraphs.

The FCOPY subset function allows you to perform operations using a specified portion of a file. You may designate the file subset as being a range of contiguous records (such as records 40 through 75) or as being all records that contain a certain data item (such as all records that contain the character string PASTDUE in columns 60 through 66). This makes it possible for you to copy or display only the desired records of a file or to divide an existing file into several separate files.

Occasionally you may encounter a disc file or a magnetic tape whose contents are unknown. The FCOPY display functions that show both the numeric codes and the character symbols are particularly useful in that they allow you to examine the contents of such files either at a terminal or by a line printer listing.

FCOPY also provides a code translating function that converts data from one computer code system to another as part of a copying operation. This makes it easy for you to prepare magnetic tapes to be used at different computer installations or to incorporate magnetic tapes from other computer installations into your HP 3000 computer system.

Each file on magnetic tape is terminated by an end-of-file (EOF) mark. The FCOPY skip end-of-file function gives you great control over FCOPY operations involving magnetic tape in that it allows you to position the tape at the desired place for either reading or writing. For example, to copy the third file from a magnetic tape to disc you can use the skip end-of-file function to skip over the first two files on the tape. Similarly, if you have a tape that already contains four files and you want to add a fifth file to it you can use the skip end-of-file function to skip over the existing four files on the tape. You can also use the skip end-of-file function for skipping over volume and header labels on a magnetic tape.

A compiled and prepared version of FCOPY/3000 resides in the file FCOPY in the public group of the system account. To use FCOPY, you must first bring it into memory and pass control to it. You do this with the following command:

```
:RUN FCOPY.PUB.SYS
```

NOTE

In a session the colon prompt is automatically issued by MPE. You do *not* type it. In a job you must type the entire command (including the colon) with the colon appearing in column 1.

Table 1-1. FCOPY Functions

Action To Be Performed	FCOPY Function Specified in <i>functionlist</i>	Syntax Example	Text Discussion
Copy data from one file to another without performing any of the other FCOPY functions listed below.	(none)	FROM=A; TO=B	2-4
Change character codes from ASCII to EBCDIC, ASCII to BCDIC, JIS to EBCDIK or vice versa.	$\left\{ \begin{array}{l} \text{EBCDICIN} \\ \text{BCDICIN} \\ \text{EBCDICOUT} \\ \text{BCDICOUT} \end{array} \right\} \left[\begin{array}{l} = \left\{ \begin{array}{l} \text{field} \\ (\text{field}[\text{;field}[\text{;..}]] \end{array} \right\} \\ \text{[,EXCLUDE]} \end{array} \right]$	FROM=A; TO=B; EBCDICIN	2-10
Convert any lower-case characters to upper case.	UPSHIFT	FROM=A; TO=B; UPSHIFT	2-38
Use only a specified portion of a file in an FCOPY operation.	SUBSET="characterstring"[, [column] [,EXCLUDE]] SUBSET=#patternlist#[, [column] [,EXCLUDE]] SUBSET = (range[,range[;..]]) SUBSET	FROM=A; TO=B; SUBSET="MALE",15,EXCLUDE	2-35
Skip end-of-file (EOF) marks or specify files on unlabeled magnetic tapes used for either or both the "from" and "to" files.	SKIPEOF = $\left[\begin{array}{l} \{+/-\} \text{ from-eofs} \\ \text{from-file-number} \end{array} \right]$ $\left[, \left\{ \begin{array}{l} \{+/-\} \text{ to-eofs} \\ \text{to-file-number} \end{array} \right\} \right]$	FROM=*A; TO=*B; SKIPEOF=2,6	2-31
Ignore bad records in the "from" file on magnetic tape and report each bad record so ignored.	IGNERR=number-of-errors	FROM=*A; TO=B; IGNERR=15	2-23
Open a new "to" disc file under the specified name.	NEW	FROM=A; TO=B; NEW	2-28
Verify the accuracy of a copy immediately after creating it, where both files are on disc or magnetic tape.	VERIFY [=number-of-errors]	FROM=A; TO=B; VERIFY=3	2-39
Compare, without copying or changing the contents, the "from" and "to" files.	COMPARE [=number-of-errors]	FROM=A; TO=B; COMPARE=15	2-13
Display numerical code of the records of a file on a word-by-word basis.	$\left\{ \begin{array}{l} \text{HEX} \\ \text{OCTAL} \end{array} \right\} \left[\begin{array}{l} \text{CHAR} \\ \text{CLEAR} \\ \text{KANA} \end{array} \right] \text{[,NORECNUM]} \\ \text{[,TITLE="title"]}$	FROM=A; TO=B; OCTAL; NORECNUM	2-20
Display character images of the records of a file on a word-by-word basis.	$\left\{ \begin{array}{l} \text{CHAR} \\ \text{CLEAR} \\ \text{KANA} \end{array} \right\} \left[\begin{array}{l} \text{HEX} \\ \text{OCTAL} \end{array} \right] \text{[,NORECNUM]} \\ \text{[,TITLE="title"]}$	FROM=A; TO=B; CHAR; TITLE="ABCDE"	2-17
When copying KSAM files, specify the location of a key whose value determines the sequence in which the file is copied.	KEY=character location	FROM=KSAM; TO=B; KEY=1	2-34
When copying KSAM files, copy the data file without regard to any key value and also include deleted records.	NOKSAM	FROM=KSAM; TO=B; NOKSAM	2-29

Table 1-1. FCOPY Functions (Continued)

Action To Be Performed	FCOPY Function Specified in <i>functionlist</i>	Syntax Example	Text Discussion
Do not copy user labels.	NOUSERLABELS	FROM=FILE1;TO=FILE2 NOUSERLABELS	2-30
Specify carriage control or no carriage control for the "to" file.	CCTL NOCCTL	FROM=DISCA;TO=*FILE3 ;CCTL	2-6
Deblock records and align on word boundaries.	DEBLOCK=logical-record-length	FROM=*BYTE;TO=WORD ;DEBLOCK=-77	2-14
Copy more than one file from a magnetic tape.	FILES= $\left\{ \begin{array}{l} \text{number-of-files} \\ \text{ALL} \end{array} \right\}$	FROM=*TAPEA;TO=*TAPEB ;FILES=4	2-24

If you are copying files to or from devices other than disc, you must define the files and their associated devices with MPE :FILE commands prior to calling FCOPY. For example, if a file is to be copied from punched cards to magnetic tape you could define the two files as follows:

```
:FILE FILE1;DEV=CARD
:FILE FILE2;DEV=TAPE;REC=-80,25,F,ASCII
:RUN FCOPY.PUB.SYS
```

where FILE1 and FILE2 are the formal file designators you will be supplying to FCOPY when you actually perform the copying operation, CARD is the device class name for a card reader, and TAPE is the device class name for a magnetic tape unit. Note that device class names are defined by each installation at system configuration time and may vary from one installation to another. The MPE :FILE command is described in Section VI of the *MPE Commands Reference Manual* (HP part number 30000-90009).

Once you have called FCOPY in an interactive session, it prompts you by printing a greater-than symbol (>) in column 1 of the next line. You now enter an FCOPY command that defines an operation to be performed and then press the return key. FCOPY performs the operation and then prompts you for another command. This cycle continues until you enter EXIT, at which time FCOPY terminates and returns control to the MPE Command Interpreter.

If you are using FCOPY in a batch job, you do *not* supply the FCOPY prompt character (>) and the FCOPY commands begin in column 1 of each record.

The FCOPY command has the following format:

$$FROM = \left\{ \begin{array}{l} fromfile \\ * \\ (empty) \end{array} \right\} ; TO = \left\{ \begin{array}{l} tofile \\ (dfile,kfile) \\ * \\ (empty) \end{array} \right\} [; functionlist]$$

where: braces { } indicate one entry must be selected.
brackets [] indicate an optional entry.
fromfile is the name of the input file.
tofile is the name of the output file.
(*dfile,kfile*) are the names of the data file and key file, respectively, of a new KSAM file.
* (an asterisk) specifies "continue using the file specified in the preceding FCOPY command at the next record after the record where the preceding FCOPY operation stopped."
(*empty*) i.e., no entry, specifies the job/session input device \$STDIN for the "from" file or the job/session list device \$STDLIST for the "to" file.
functionlist is one or more entries to specify one or more FCOPY functions (see Table 1-1).

Some examples of FCOPY commands are as follows:

In a Job:

FROM=;TO=	Copy from \$STDIN to \$STDLIST.
FROM=DISCFILE; TO=*ANYFILE;CHAR;OCTAL	List the content of the disc file named DISCFILE in both character and octal formats on the device defined for the formal file designator ANYFILE, as defined on a previous :FILE card.
FROM=*;TO=NEWFILE;NEW	Copy the content of the "from" file specified in the preceding FCOPY command to a newly-created disc file named NEWFILE (FCOPY interacts with the MPE File System to create NEWFILE).

In a Session:

>FROM=DISC1;TO=DISC2	Copy the content of the disc file named DISC1 to the disc file named DISC2.
>FROM=*;TO=DISC3;NEW	Copy the content of the "from" file specified in the preceding FCOPY command to a newly-created disc file named DISC3 (FCOPY interacts with the MPE File System to create DISC3).
>FROM=MYFILE;TO=*MAGTP; SUBSET="OVERDUE",35;SKIPEOF=,+2	Position the magnetic tape defined for the formal file designator MAGTP following the second end-of-file mark on the tape and then copy only those records in the disc file named MYFILE which contain the character string OVERDUE starting in character position 35 to the magnetic tape.

The previous examples illustrate the syntax of the FCOPY command. Complete examples of the use of FCOPY are presented in the remainder of this section and in Section III of this manual.

1-1. EXAMPLES

The remainder of this section presents four annotated examples, two showing how to use FCOPY from a batch job and two showing how to use it from an interactive session. For more examples, see Section IV.

1.2 CARD READER TO LINE PRINTER (Batch Job)

This example shows a batch job in which FCOPY reads a deck of cards and then lists them on a line printer.

The job deck is as follows:

		:JOB USER.ACCOUNT
		:RUN FCOPY.PUB.SYS
		FROM=; TO=
Card deck to be listed	{	THIS IS CARD 1.
		THIS IS CARD 2.
		THIS IS CARD 3.
		THIS IS CARD 4.
		THIS IS CARD 5.
		THIS IS CARD 6.
		:EOD
		:EOJ

The FCOPY command FROM=;TO= specifies (by default) that the "from" file is the standard job input device \$STDIN, in this case a card reader, and the "to" file is the standard job listing device \$STDLST, in this case a line printer. The job produces output on the line printer as shown below.

```
:JOB      USER.ACCOUNT, PUB
PRI= DS;  INPRI= 8 ;  TIME= ?
JOB NUMBER = 8359
TUE, APR 13, 1976,  3:01 PM
HP32002A.00.49
```

```
:RUN FCOPY.PUB.SYS
```

```
PAGE 1  HP32212A.3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977  TUE, JAN 3, 1976, 3:02 PM
FROM=;TO=
```

```
WARNING: FROMFILE RECSIZE IS 80 BYTES, TOFILE RECSIZE IS 133 BYTES.
```

```
THIS IS CARD 1.
THIS IS CARD 2.
THIS IS CARD 3.
THIS IS CARD 4.
THIS IS CARD 5.
THIS IS CARD 6.
EOF FOUND IN FROMFILE AFTER RECORD 5
```

NOTE

FCOPY considers the first record to be record 0.

```
6 RECORDS PROCESSED *** 0 ERRORS
```

```
END OF PROGRAM
:EOJ
```

```
CPU (SEC) = 2
ELAPSED (MIN) = 2
TUE, APR 13, 1976,  3:02 PM
END OF JOB
```

1-3. DISC TO LINE PRINTER (Batch Job)

This example shows a batch job in which FCOPY displays (in octal) the content of several disc files via a line printer.

The job deck is as follows:

```
:JOB USER.ACCOUNT
:RUN FCOPY.PUB.SYS
FROM=DISC1;TO=;OCTAL
```

NOTE

•
•
•

The standard job list output and octal listings are shown only for DISC1 and DISC7.

```
FROM=DISC7;TO=;OCTAL
EXIT
:EOJ
```

The above job deck produces a series of listings on the line printer: the standard job list output followed by the octal file listings.

```
:JOB    USER.ACCOUNT, PUB
PRI= DS;  INPRI= 14;  TIME= ?
JOB NUMBER = #J73
TUE, APR 13, 1976,  4:13 PM
HP32002A.00.49

:FILE LIST;DEV=LP
:RUN FCOPY.PUB.SYS
```

```
PAGE 1  HP32212A.3.00 FILE COPIER (C) HEWLETT-PACKARD CO, 1977  TUE, JAN 3, 1978, 3:02 PM

FROM=DISC1;TO=*LIST;OCTAL
EOF FOUND IN FROMFILE AFTER RECORD 5

6 RECORDS PROCESSED *** 0 ERRORS

FROM=DISC7;TO=*LIST;OCTAL
EOF FOUND IN FROMFILE AFTER RECORD 3

4 RECORDS PROCESSED *** 0 ERRORS

EXIT

END OF PROGRAM
:EOJ

CPU (SEC) = 3
ELAPSED (MIN) = 8
TUE, APR 13, 1976,  4:16 PM
END OF JOB
```

STANDARD JOB LIST OUTPUT

DISC1.PUB.ACCOUNT RECORD 0 (%0)

000000: 052110 044523 020111 051440 046111 047105 020061 027040
 000010: 020124 044105 020070 026504 044507 044524 020116 052515
 000020: 041105 051123 020101 052040 052110 042440 051111 043510
 000030: 052040 040522 042440 052110 042440 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 030460 030060

DISC1.PUB.ACCOUNT RECORD 1 (%1)

000000: 052105 054124 020105 042111 052117 051040 046111 047105
 000010: 020116 052515 041105 051123 027040 020116 047524 042440
 000020: 052110 040524 020124 044105 020122 042503 047522 042040
 000030: 047125 046502 042522 051440 020040 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 031060 030060

DISC1.PUB.ACCOUNT RECORD 2 (%2)

000000: 051525 050120 046111 042504 020102 054440 043103 047520
 000010: 054440 051524 040522 052040 040524 020132 042522 047440
 000020: 053510 044514 042440 052110 042440 052105 054124 020105
 000030: 042111 052117 051040 046111 047105 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 031460 030060

OCTAL LISTING OF DISC1

DISC7.PUB.ACCOUNT RECORD 0 (%0)

000000: 052110 044523 020111 051440 046111 047105 020117 047105
 000010: 027040 020124 044111 051440 042530 040515 050114 042440
 000020: 051510 047527 051440 040440 041510 040522 040503 052105
 000030: 051040 042111 051520 046101 054440 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 030460 030060

DISC7.PUB.ACCOUNT RECORD 1 (%1)

000000: 052523 044516 043440 052110 042440 047117 051105 041516
 000010: 052515 020117 050124 044517 047056 020040 040507 040511
 000020: 047040 052110 042522 042440 044523 020116 047440 052111
 000030: 052114 042456 020040 044506 020040 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 031060 030060

DISC7.PUB.ACCOUNT RECORD 2 (%2)

000000: 052110 042440 042111 051520 046101 054440 041510 040522
 000010: 040503 052105 051123 020106 052516 041524 044517 047040
 000020: 044523 020116 047524 020103 047515 041111 047105 042040
 000030: 053511 052110 020117 041524 040514 020040 020040 020040
 000040: 020040 020040 020040 020040 030060 030060 031460 030060

OCTAL LISTING OF DISC7

1-4. DISC TO DISC (Interactive Session)

This example shows part of an interactive session in which FCOPY is used to copy a file from one account (ACCT1) to another (ACCT2). The file in ACCT2 is created by FCOPY and has the same attributes as the file in ACCT1. The commands entered by the user are underscored. The FCOPY portion of the session is as follows:

```
.
.
:RUN FCOPY.PUB.SYS
HP32212A.3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977

>FROM=MASTER.GRP1.ACCT1; TO=MASTER.GRP1.ACCT2; NEW
EOF FOUND IN FROMFILE AFTER RECORD 679

680 RECORDS PROCESSED *** 0 ERRORS

>EXIT

END OF PROGRAM
```

1-5. DISC TO TERMINAL (Interactive Session)

This example shows part of an interactive session in which FCOPY is used to display five records (records 30 through 34) of a disc file at the user's log-on terminal. In this example the disc file is an ASCII character file called CUSTDATA containing the names and addresses of customers. Since it is desired to display the content of the five records in the simplest and most easy to read format the user has specified the display characters function (CHAR) with the NORECNUM option (this option suppresses the printing of the file name and record numbers). The commands entered by the user are underscored. The FCOPY portion of the session is as follows:

```
.
.
:RUN FCOPY.PUB.SYS
HP32212A.3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977

>FROM=CUSTDATA; TO=; SUBSET=29:33; CHAR; NORECNUM
XYZ MANUFACTURING CO.          131 PINE STREET SF
WIDGET, INC.                   43721 OAK STREET DET
ROBERT ALCOTT & ASSOCIATES     752 SURF AVENUE CHI
WILLIAM LEWIS & SON            3217 W. MADISON AVENUE NY
SUPER GIZMO OF NEVADA, INC.    431 S. JACKPOT LANE RENO
5 RECORDS PROCESSED *** 0 ERRORS

>EXIT

END OF PROGRAM
```

In the FCOPY command, the parameter TO= specifies (by default) that the standard session list device \$STDLIST (the user's log-on terminal) is the "to" file. Note that FCOPY considers the first record to be record 0. The parameter SUBSET=29:33 specifies that only records 30 through 34 are to be displayed.

This section describes all of the functions that you can specify in an FCOPY command. The various functions can be used individually or in combination. Included in the description of each is a set of annotated examples. Since the intent of these examples is to illustrate the proper FCOPY command syntax for using the particular function, the associated MPE :FILE commands (if required) are not shown. Complete examples are presented in Section IV.

With the exception of the primary function, listed under the copy heading, all functions are listed alphabetically according to their respective purposes. For example, the FILES= function is listed under Multiple Tape File Copying between Ignore Errors and New File.

2-1. FILE ATTRIBUTE REQUIREMENTS

FCOPY imposes a few requirements as to the attributes of the "to" and "from" files. The requirements vary according to the FCOPY function being used and are as follows:

Function	Restrictions
Copy, or Character Translate, or Subset, or Upshift	The "from" file must have read access, the "to" file must have write access; or, to append a copy onto a file, the "to" file must have append access.
Skip End-of-File	<i>Either or both</i> the "from" file and the "to" file device must be an unlabeled magnetic tape unit, whichever is to use this function; the "from" file must have read access, the "to" file must have write access.
Ignore Errors	The "from" file device must be a magnetic tape unit with read access, the "to" file must have write access.
New File	The "from" file must have read access.
Verify	Both the "from" file and the "to" file devices must be a disc drive or a magnetic tape unit; the "from" file must have read access, the "to" file must have read/write access.
Compare	The "from" file and the "to" must both have read access; the "to" file record size must equal that of the "from" file or both files must have variable-length records.
Display Character Images, or Display Numerical Codes	The "from" file must have read access, the "to" file must have write access. Further, the "to" file, normally, a list device such as \$STDLIST, can be another device for intermediate storage of the data. The record size for such an intermediate storage device should be that of the list device that ultimately will print the data and must be ≥ 60 bytes (30 words).

Function	Restrictions
Multiple tape files	The "from" file device must be a magnetic tape unit.
Key, or non-KSAM	The "from" file must be a KSAM file.

2-2. INTERRUPTING FCOPY

In a session you can prematurely terminate an FCOPY operation by entering a CONTROL-Y (press and hold the CONTROL key and press Y). FCOPY terminates the operation and then prompts you for another command.

Once you have called FCOPY, you can temporarily return control to the MPE Command Interpreter by pressing the BREAK key. The command interpreter responds by issuing a colon (:) prompt. You may then enter MPE commands (:FILE, :BUILD, :LISTF, and so forth). When you are ready to continue with the interrupted FCOPY processing, enter :RESUME.

SYNTAX

$$> \text{FROM} = \left\{ \begin{array}{c} \text{filename} \\ * \\ \text{<empty>} \end{array} \right\}; \text{TO} = \left\{ \begin{array}{c} (dfile, kfile) \\ \text{filename} \\ * \\ \text{<empty>} \end{array} \right\} \quad [; \text{NOUSERLABELS}] \quad \left[\begin{array}{c} \text{CCTL} \\ ; \\ \text{NOCCTL} \end{array} \right] \quad [; \text{NEW}]$$

$$\left[\begin{array}{c} \text{CHAR} \\ ; \text{CLEAR} \\ \text{KANA} \end{array} \right] \left[\begin{array}{c} \text{HEX} \\ ; \\ \text{OCTAL} \end{array} \right] \quad [; \text{NORECNUM}] \quad [; \text{TITLE} = \text{"title"}]$$

[;DEBLOCK=*logical-record-length*]

$$\left[\begin{array}{c} \text{EBCDICIN} \\ \text{EBCDICOUT} \\ ; \text{BCDICIN} \\ \text{BCDICOUT} \\ \text{EBCDIKIN} \\ \text{EBCDIKOUT} \end{array} \right] \left[\begin{array}{c} = \left\{ \begin{array}{c} \text{field} \\ (\text{field}[, \text{field}[, \dots]]) \end{array} \right\} \quad [; \text{EXCLUDE}] \end{array} \right]$$

$$\left[; \text{FILES} = \left\{ \begin{array}{c} \text{number of files} \\ \text{ALL} \end{array} \right\} \right]$$

[;IGNERR=[*number of errors*]] [;COMPARE=[*number of errors*]]

$$\left[; \text{SKIPEOF} = \left[\begin{array}{c} \left\{ +/ - \right\} \text{ from-eofs} \\ \text{from-file-number} \end{array} \right] \left[\begin{array}{c} \left\{ +/ - \right\} \text{ to-eofs} \\ \text{to-file-number} \end{array} \right] \right]$$

$$\left[; \text{SUBSET} \left[\begin{array}{c} = \text{"character string"} \quad [, [\text{column}] \quad [, \text{EXCLUDE}]] \\ = \# \text{pattern list\#} \quad [, [\text{column}] \quad [, \text{EXCLUDE}]] \\ = (\text{range} \quad [; \text{range} \dots]) \end{array} \right] \right]$$

[;NOKSAM]

[;KEY=*character location*]

[;UPSHIFT]

[;VERIFY=*number-of-errors*]

COPY

FROM = ;TO =

2-3. COPY

To copy the content of one file to another without performing any of the other FCOPY functions, use an FCOPY command with the following format:

FROM=[*fromfile*];TO=[*tofile*]

where: *fromfile* specifies the file to be copied. This can be an actual file designator, a formal file designator, or an asterisk (*). An asterisk specifies the “from” file designated in the immediately preceding FCOPY command. If *fromfile* is omitted, the standard job/session input device \$STDIN is assumed.

tofile specifies the file to receive the data. This can be an actual file designator, a formal file designator, or an asterisk (*). An asterisk specifies the “to” file designated in the immediately preceding FCOPY command. If *tofile* is omitted, the standard job/session list device \$STDLIST is assumed.

For example, to copy the contents of a disc file named DISC1 into another disc file named COPY1 use the following FCOPY command:

```
>FROM=DISC1; TO=COPY1
```

There are many FCOPY functions that can also be specified in an FCOPY command. They are described in the remainder of this section. If additional functions are included in the FCOPY command, they are separated from each other (and from the TO= parameter) by semicolons. For example, consider the following command:

```
>FROM=*TAPE; TO=DISCFILE; SUBSET; EBCDICIN; SKIPEOF=2; NEW
```

The above command specifies that one EBCDIC file (the second file on the tape) is to be copied from the magnetic tape *TAPE to a new permanent disc file (named DISCFILE) created by FCOPY. The data being transferred is converted from EBCDIC to ASCII during the course of the copying operation.

2-4. COMMENTS

The following guidelines apply to the use of FCOPY commands.

1. If you are copying files to or from devices other than disc, you should define the files and their associated devices with MPE :FILE commands prior to issuing the FCOPY command.

FROM = ;TO =

2. The “to” file definition (:FILE command) can specify a different blocking factor than that of the “from” file.
3. The “to” file definition (:FILE command) can specify access type APPEND, to append the “from” file contents onto the end of the “to” file.
4. When FCOPY functions are combined in one command they appear in any order.
5. There are three limitations on combining FCOPY functions:
 - a. COMPARE can be combined only with SKIPEOF, SUBSET, or IGNERR.
 - b. CHAR, CLEAR, KANA, OCTAL, and HEX *cannot* be combined with VERIFY, NEW, EBCDICOUT, EBCDIKOUT, or BCDICOUT.
 - c. KEY cannot be combined with NOKSAM.
6. You may include blanks between the elements of an FCOPY command for clarity:

```
>FROM=*TAPE; TO=MEN; SUBSET="MALE",30; SKIPEOF=2
```

7. Each FCOPY command may be continued on more than one line (record). To do this, use an ampersand (&) as the final character in each line except the last:

```
>FROM=DATAFILE; &  
>TO=WOMEN; &  
>SUBSET="FEMALE",30; &  
>SKIPEOF=2
```

8. More than one FCOPY command can be issued during a given job or session, with one restriction: \$STDIN (the standard job/session input device) can be used as the “from” file only once, for the last or only FCOPY command. Data input through \$STDIN must be terminated by a line or record containing only :EOD. The :EOD command terminates FCOPY and returns control to the MPE Command Interpreter.
9. Numeric parameters in FCOPY commands can be supplied in either octal or decimal form. To signify that a numeric parameter is in octal, merely precede it with a percent sign (%). Otherwise the parameter is assumed to be in decimal.
10. Using the NEW file function, FCOPY can copy an old KSAM file into a new, compacted KSAM file with the same data and key files. KSAM files can also be copied into existing KSAM files using the KEY= or copy functions.
11. If the user’s terminal has a line printer dedicated to it, and has built-in cartridge tape units, these can be used in an FCOPY command as “from” or “to” files. The FCOPY file names associated with these devices are \$CTUL for the left cartridge unit, \$CTUR for the right tape unit and \$HARD for the terminal line printer. Refer to Appendix A for further discussion on the use of these peripherals.



CARRIAGE CONTROL/NO CARRIAGE CONTROL

CCTL
NOCCTL

2-5. CARRIAGE CONTROL/NO CARRIAGE CONTROL

FCOPY allows the user to specify whether or not the first character of each record in the "from" file is to be used as a carriage control character in the "to" file.

In any FCOPY command line, the user can elect to use CCTL or NOCCTL, or neither. The choice is governed by whether or not the carriage control option was specified in the file equations or disc labels for the "from" and "to" files:

NEITHER CCTL NOR NOCCTL SPECIFIED IN FCOPY COMMAND. The following table shows what happens if the user chooses not to use either CCTL or NOCCTL in an FCOPY command.

CCTL specified in :FILE equation or disc label for "from" file? "to" file?		EFFECT
NO	NO	The entire record from the "from" file is copied into the "to" file.
NO	YES	A single spacing character is supplied; the entire "from" record is copied into the "to" file.
YES	NO	The entire "from" record, including the carriage control character is copied into the "to" file as data.
YES	YES	The entire "from" record is copied into the "to" file. The carriage control character from the "from" record becomes the carriage control character for the "to" record.

CARRIAGE CONTROL/NO CARRIAGE CONTROL

CCTL
NOCCTL

CCTL SPECIFIED IN FCOPY COMMAND. The carriage control function of FCOPY has the following format:

CCTL

The carriage control function specifies that the first character of each record in the "from" file is to be used as a carriage control character in the "to" file. The effect of using CCTL in an FCOPY command depends upon whether the carriage control option was specified in the file equations or in a disc label for either or both the "from" and "to" files.

The following table shows what happens if the user chooses to use the CCTL function in an FCOPY command.

When a file equation or disc file label specifies NOCCTL in a "to" file, and CCTL is specified in an FCOPY command, the NOCCTL specification in the file equation or disc label overrides the FCOPY specification. This is represented in the table below by the starred cases.

CCTL specified in :FILE equation or disc label for "from" file? "to" file?		EFFECT
NO	NO ***	Entire record of "from" file copied to "to" file. No carriage control character supplied.
NO	YES	Entire record of "from" file copied to "to" file. First character of "from" file record interpreted as a carriage control character, not as data.
YES	NO ***	Entire record of "from" file copied as data, including the carriage control character.
YES	YES	Entire record of "from" file copied into "to" file, with carriage control character of "from" record used as carriage control character for "to" record.

CARRIAGE CONTROL/NO CARRIAGE CONTROL

CCTL
NOCCTL

NOCCTL SPECIFIED IN FCOPY COMMAND. The no carriage control function has the following format:

NOCCTL

The no carriage control function specifies that the first character of each record in the "from" file is not to be used as a carriage control character in the "to" file. The effect of using NOCCTL depends upon whether the carriage control option was specified in the file equations or disc labels for either or both the "from" and "to" files.

The following table shows what happens if the user chooses to specify the NOCCTL function in an FCOPY command.

CCTL specified in :FILE equation or disc label for "from" file? "to" file?		EFFECT
NO	NO	Entire record of "from" file copied to "to" file. No carriage control character supplied.
NO	YES ***	Entire record of "from" file copied to "to" file. Single spacing carriage control character supplied for each record of "to" file.
YES	NO	Entire record of "from" file copied as data, including the carriage control character.
YES	YES ***	Entire record of "from" file copied into "to" file, including carriage control character of "from" record. A single spacing carriage control character is supplied for each "to" file record.

When a file equation or disc file label specifies CCTL in a "to" file, and NOCCTL is specified in an FCOPY command, the CCTL specification in the file equation or disc label overrides the FCOPY command. This is represented in the above table by the starred cases.

CARRIAGE CONTROL/NO CARRIAGE CONTROL

CCTL
NOCCTL

2-6. EXAMPLES

Assume that ALPHA is a disc file whose label specifies the CCTL option, and BETA is a file whose equation is .FILE BETA,NOCCTL.

The FCOPY command

```
>FROM=ALPHA,TO=*BETA,NOCCTL
```

copies each record from ALPHA to BETA, including the carriage control character as data.

The FCOPY command

```
>FROM=*BETA,TO=ALPHA,CCTL
```

copies each record from BETA to ALPHA, and supplies a single spacing carriage control character for each record.

The FCOPY command

```
>FROM=ALPHA,TO=GAMMA,NEW,CCTL
```

copies the entire record from ALPHA into GAMMA, using each carriage control character from each record of ALPHA as the carriage control character for each copied record of GAMMA.

CHARACTER TRANSLATE

EBCDICIN
EBCDICOUT
BCDICIN
BCDICOUT
EBCDIKIN
EBCDIKOUT

2-7. CHARACTER TRANSLATE

The character translate function converts all or part of a record from the characters of one computer code system to another as part of a copying operation. This makes it easy for you to prepare discs or tapes for use at different computer installations or to incorporate magnetic tapes from another computer installation into your HP 3000 computer system.

The following list shows what each function translates:

EBCDICIN	translates from EBCDIC to ASCII.
EBCDICOUT	translates from ASCII to EBCDIC.
BCDICIN	translates from BCDIC to ASCII.
BCDICOUT	translates from ASCII to BCDIC.
EBCDIKIN	translates from EBCDIK to JIS.
EBCDIKOUT	translates from JIS to EBCDIK.

NOTE: Neither EBCDIKIN nor EBCDIKOUT can be used on an HP 3000 that runs under the control of MPE-C.

The character translate function has the following format:

EBCDICIN	$\left[= \left\{ \begin{array}{l} \text{field} \\ (\text{field} [;\text{field} [;\dots]]) \end{array} \right\} \quad [, \text{EXCLUDE}] \right]$
EBCDICOUT	
BCDICIN	
BCDICOUT	
EBCDIKIN	
EBCDIKOUT	

Where

field is a pair of unsigned integers of the form a,b or a:b.

In the form a,b

a is the starting column

b is the number of columns to be converted.

In the form a:b

a is the starting column

b is the ending column.

EXCLUDE specifies that the referenced fields are not to be converted.

CHARACTER TRANSLATE

EBCDICIN
EBCDICOUT
BCDICIN
BCDICOUT
EBCDIKIN
EBCDIKOUT

2-8. COMMENTS

When the *field* option is used, only data in the referenced field or fields is converted unless it is used in conjunction with the EXCLUDE option; in that case, all except the referenced fields are converted.

If the *field* option is not used, FCOPY converts the entire record.

Note that the first column (*field*) of a record is column (*field*) number 1. The maximum number of fields that can be specified is 255.

When EBCDICIN, BCDICIN, or EBCDIKIN is used in conjunction with the SUBSET= function, conversions are made before the SUBSET= function is executed; likewise, when EBCDICOUT, BCDICOUT, or EBCDIKOUT is used with the SUBSET= function, conversions are made following the execution of the SUBSET= function.

In BCDIC there are five character codes which can be interpreted in either of two ways. FCOPY translates them as follows:

BCDIC	ASCII
□ and)	)
& and +	+
% and (	(
# and =	=
@ and '	'

For a complete listing of conversion codes between ASCII and EBCDIC, JIS and EBCDIK, and ASCII and BCDIC, refer to Appendix C.

Note that when user labels are present in a file that is being translated, the labels are NEVER translated. The character translate function does not apply to them.

CHARACTER TRANSLATE

EBCDICIN
EBCDICOUT
BCDICIN
BCDICOUT
EBCDIKIN
EBCDIKOUT

2-9. EXAMPLES

```
>FROM=*TAPE;TO=*DISC1;EBCDICIN=(3,5;14;27)
```

In the above example, columns 3 through 7 and 14 through 27 of each record in the copied file are converted from EBCDIC to ASCII as they are copied from *TAPE to *DISC1. All other fields are copied to *DISC1 without conversion.

```
>FROM=THIS,TO=THAT,EBCDIKIN=11:80
```

In the example above, if THIS is a file of 80 character records, all except the first ten characters in each record are converted from EBCDIK to JIS as the records are copied into THAT.

```
>FROM=FILE1;TO=FILE2;BCDICIN=(1,5;10;30),EXCLUDE
```

In the above example, all characters except those in columns 1 through 5 and 10 through 30 are converted from BCDIC to ASCII as they are copied from FILE1 to FILE2.

COMPARE

COMPARE =

2-10. COMPARE

The compare function compares the content of the “from” file against the content of the “to” file, record-by-record, without changing either. The two files may reside on any devices; however, the record characteristics of the two must be identical (record size, record format (fixed/variable/undefined), blocking factor, record type (ASCII/binary)). The compare function may be combined only with the subset, skip end-of-file, and ignore errors functions.

The compare function has the following format:

COMPARE=*number-of-errors*

where *number-of-errors* is an integer specifying the maximum number of comparison errors to be tolerated. If this number is reached, the operation terminates. If *number-of-errors* is omitted, the default value is 1.

Each time FCOPY detects an error during a compare operation, it prints a message on the standard listing device (your terminal in a session or a line printer in a job) specifying the number of the “from” file record in which the error occurred.

2-11. EXAMPLES

```
>FROM=FILEA; TO=DUP1; COMPARE
```

In the above example, FCOPY compares the content of the disc file FILEA against the content of the disc file DUP1. If it finds an error, FCOPY prints a message on the standard listing device and terminates the operation.

```
>FROM=*TAPE4; TO=DISC2; SUBSET; COMPARE=5
```

In the above example, FCOPY compares the content of the first file on the magnetic tape *TAPE against the content of the disc file DISC2. Each time it finds an error, FCOPY prints a message on the standard listing device. The fifth error terminates the operation.

DEBLOCK RECORDS

DEBLOCK =

2-12. DEBLOCK RECORDS

The deblock function makes it possible to deblock blocked records, to make smaller records from large ones, and, of particular interest, to copy files from other systems whose records are not aligned on word boundaries.

The deblock function has the following format:

$\text{DEBLOCK} = \text{logical-record-length}$

where *logical-record-length* is a signed integer. If the *logical-record-length* is positive, FCOPY interprets the length as words; if negative, FCOPY interprets it as bytes.

When the DEBLOCK= function is used in an FCOPY command, each group of bytes or words specified by the *logical-record-length* in DEBLOCK= is assumed by FCOPY to be a record. This record is copied into the "to" file, and then aligned on word boundaries.

If the *logical-record-length* is specified as an odd number of bytes, an extra byte is added to each output record in order to facilitate alignment.

The user should be cautious in specifying the *logical-record-length*, since if the input record length is not a multiple of the *logical-record-length*, any fractional remainder of the input record is lost. To illustrate:

Input record: 250 bytes.

45 b	45 b	45 b	45 b	45 b	25 b
------	------	------	------	------	------

Logical-record-length specified as 45 bytes results in:

46 b	46 b	46 b	46 b	46 b	
output rec # 1	output rec # 2	output rec # 3	output rec # 4	output rec # 5	lost 25 bytes

Note that each output record has 46 bytes. The extra byte was added to each record after it was copied to make possible the proper alignment.

2-13. FILES FROM OTHER SYSTEMS

The major reason for DEBLOCK= is to copy files from other systems whose records are aligned on byte boundaries. If the other systems use word boundaries to align records, DEBLOCK= is not generally needed. However, if records from other systems are aligned on byte boundaries, DEBLOCK= should be used to copy the records into files so that they are aligned on word boundaries, thus making them usable on an HP 3000 system.

DEBLOCK RECORDS

DEBLOCK =

As an example of how files from other systems can be copied into a usable file for the HP 3000 system, assume that a magnetic tape file from another system is to be copied. Further, assume that the records are 79 bytes long, blocked in groups of ten, and that the records are aligned on byte boundaries.

Without the deblock function, FCOPY assumes that each record is aligned on word boundaries. The result is that the first record is copied as it should be, but the second record lacks its first byte and contains the first byte of the third record, the third record lacks its first two bytes and contains the first two bytes of the fourth record, and so forth until the tenth record in the block lacks its first nine bytes.

To illustrate:

```
:FILE TAPEBYTE, REC=-79,10,F,ASCH  
:RUN FCOPY.PUB.SYS  
> FROM=*TAPEBYTE, TO=DISC1
```



RECORD NUMBER ONE	
ECORD NUMBER TWO	R
CORD NUMBER THREE	RE
ORD NUMBER FOUR	REC
RD NUMBER FIVE	RECO
D NUMBER SIX	RECOR
NUMBER SEVEN	RECORD
UMBER EIGHT	RECORD N
MBER NINE	RECORD NU
BER TEN	

Note that each record is 80 bytes long; FCOPY added one byte to each copied record to align them on word boundaries.

DEBLOCK RECORDS

DEBLOCK =

To avoid copying the records erroneously, DEBLOCK= is used in the following manner:

First, a new file equation is used to treat each block as a record by defining the record length of the file equation to be the old record length times the old blocking factor (790 in our example), and setting the new blocking factor to 1:

```
:FILE TAPEBYTE, REC=-790,LF,ASCII  
:RUN FCOPY PUB.SYS  
> FROM=*TAPEBYTE, TO=DISC1, DEBLOCK=-79
```

RECORD NUMBER ONE	
RECORD NUMBER TWO	
RECORD NUMBER THREE	
RECORD NUMBER FOUR	
RECORD NUMBER FIVE	
RECORD NUMBER SIX	
RECORD NUMBER SEVEN	
RECORD NUMBER EIGHT	
RECORD NUMBER NINE	
RECORD NUMBER TEN	

FCOPY writes each 79 bytes of TAPEBYTE as a record onto DISC1 and aligns it on word boundaries. Note that -79 was chosen to insure that no records were lost. Also, as the records were specified as an odd number of bytes, FCOPY added one byte to each output record in order to align them on word boundaries.

DISPLAY CHARACTER SYMBOLS

CHAR
CLEAR
KANA

2-14. DISPLAY CHARACTERS

The display character symbols functions allow you to display the content of a file, record-by-record, in the form of the character images. Normally the "to" file is a line printer. However, it can be a keyboard display terminal or an intermediate disc file to be listed at a later time. If the display is being routed to an intermediate disc file, the record size of the intermediate disc file must be at least 60 bytes (30 words) in length.

Note that some devices are only capable of displaying uppercase roman alphabetic characters. For these devices, the device controller automatically converts lowercase roman alphabetic characters to uppercase. In such instances it may be desirable to combine the character and the numerical display functions to obtain the actual (unaltered) character codes in addition to the character images.

The display characters function has the following format:

$$\left\{ \begin{array}{l} \text{CHAR} \\ \text{CLEAR} \\ \text{KANA} \end{array} \right\} \left[\left[\begin{array}{l} ;\text{OCTAL} \\ ;\text{HEX} \end{array} \right] \right] \left[;\text{NORECNUM} \right] \left[;\text{TITLE}=\text{"title"} \right]$$

where: CHAR	specifies the record-by-record display, as character symbols, of all the character codes which are displayable as symbols in ASCII code. Character codes not represented by symbols in ASCII code are displayed as decimal points.
CLEAR	specifies the record-by-record display, as character symbols, of all the codes found in the file. No character codes are replaced by decimal points. The graphic representation used for each code is a function of how the displaying device interprets the code. That is, as a command, as a displayable character, or as a non-displayable character. If a code is interpreted as a command by an output device, nothing is displayed unless a "display functions" mode is invoked, as, for example, on the HP 2645 terminal.
KANA	specifies the record-by-record display, as character symbols, of all the character codes which are displayable as symbols in JIS code. Character codes not represented by symbols in JIS code are displayed as decimal points. This keyword is not available on systems that use MPE-C.

DISPLAY CHARACTER SYMBOLS

CHAR
CLEAR
KANA

where:	OCTAL	specifies that the display octal and display characters functions are to be combined.
	HEX	specifies that the display hexadecimal and display characters functions are to be combined.
	NORECNUM	specifies that the file identification, record numbers, and word offset numbers are to be omitted from the display.
	<i>title</i>	is a character string to be used as the title of the display. If the "to" file is a line printer, the title is printed at the top of each page. Otherwise the title is written only once, at the beginning of the "to" file. The TITLE= parameter may only be used in conjunction with the FCOPY display functions (CHAR, CLEAR, KANA, OCTAL, and/or HEX). <i>title</i> may contain up to 70 characters. The character string that makes up the title may not be broken and continued on the following line or record.

Only one of the character displaying keywords, CHAR, KANA, or CLEAR, can be used at a given time. However, each of them can be combined with the keywords, HEX or OCTAL, to display the hexadecimal or octal value of each character code along with its character symbol. This is especially important if the output device you are using for the dump automatically upshifts lower case characters. Only one of the numerical code displaying keywords, HEX or OCTAL, can be used at a given time.

DISPLAY CHARACTER SYMBOLS

CHAR
CLEAR
KANA

2-15. EXAMPLES

```
>FROM=TEXT1; TO=*LP; CHAR
```

The above example produces the following type of display:

```
TEXT1.PUB.ACCOUNT RECORD 0 (%0)
000000:  THIS IS LINE 1.  THE 8-DIGIT NUMBERS AT THE RIGHT ARE THE      00001000
TEXT1.PUB.ACCOUNT RECORD 1 (%1)
000000:  TEXT EDITOR LINE NUMBERS.  NOTE THAT THE RECORD NUMBERS      00002000
TEXT1.PUB.ACCOUNT RECORD 2 (%2)
000000:  SUPPLIED BY FCOPY START AT ZERO WHILE THE TEXT EDITOR LINE    00003000
TEXT1.PUB.ACCOUNT RECORD 3 (%3)
000000:  NUMBERS START AT ONE.  THIS EXAMPLE SHOWS A CHARACTER        00004000
TEXT1.PUB.ACCOUNT RECORD 4 (%4)
000000:  DISPLAY WHICH INCLUDES THE FILE IDENTIFICATION, RECORD        00005000
TEXT1.PUB.ACCOUNT RECORD 5 (%5)
000000:  NUMBERS, AND WORD NUMBERS.  THERE IS NO TITLE.                00006000
```

The file TEXT1.PUB ACCOUNT is an ASCII file created using the HP 3000 Text Editor. Note that the record numbers supplied by FCOPY start at zero while the text editor line numbers start at one. The octal word number at the start of each line of data specifies the sequential number within the record of the first word of data in the line. In this particular example, the word number is of no assistance to you because it is always 000000.

```
>FROM=TEXT2; TO=*LP; CHAR; NORECNUM
```

The above example produces the following type of display:

```
THIS IS LINE ONE.  THIS EXAMPLE SHOWS A CHARACTER DISPLAY      00001000
USING THE NORECNUM OPTION.  AGAIN THERE IS NO TITLE.  AS      00002000
IN THE PREVIOUS EXAMPLE, THE 8-DIGIT NUMBERS AT THE RIGHT    00003000
ARE THE TEXT EDITOR LINE NUMBERS.  IF THE DISPLAY CHAR-      00004000
ACTERS FUNCTION IS NOT COMBINED WITH OCTAL OR HEX, THIS      00005000
IS THE MOST COMMONLY USED FORMAT.                             00006000
```

TEXT2 is also an ASCII file created using the text editor.

DISPLAY NUMERICAL CODES

HEX
OCTAL

2-16. DISPLAY CODES

The display numerical codes functions allow you to display the content of a file, record-by-record, in the form of the character code numbers. Normally the "to" file is a line printer. However, it can be a keyboard display terminal or an intermediate disc file to be listed at a later time. If the display is being routed to an intermediate disc file, the record size of the intermediate disc file must be at least 60 bytes (30 words) in length.

The display numerical codes function has the following format:

$$\left\{ \begin{array}{l} \text{HEX} \\ \text{OCTAL} \end{array} \right\} \left[\left\{ \begin{array}{l} \text{CHAR} \\ \text{CLEAR} \\ \text{KANA} \end{array} \right\} \right] \left[; \text{NORECNUM} \right] \left[; \text{TITLE} = \text{"title"} \right]$$

where: HEX	specifies the record-by-record display of the character code numbers in Hexadecimal form.
OCTAL	specifies the record-by-record display of the character code numbers in Octal form.
CHAR	specifies that the display ASCII character symbols and the display numeric code functions are to be combined.
CLEAR	specifies that the display all character symbols and the display numerical code functions are to be combined.
KANA	specifies that the display JIS character symbols and the display numerical code functions are to be combined. The MPE-C operating system does not recognize this keyword.
NORECNUM	specifies that the file identification information, record numbers, and word offset numbers are to be omitted from the display.
<i>title</i>	is a character string to be used as the title of the display. If the "to" file is a line printer, the title is printed at the top of each page. Otherwise the title is written only once, at the beginning of the "to" file. The TITLE= parameter may only be used in conjunction with the FCOPY display functions (CHAR, CLEAR, KANA, OCTAL, and/or HEX). <i>title</i> may contain up to 70 characters. The character string that makes up the title cannot be broken and continued on the following record or line.

Only one of the numerical code displaying keywords, HEX or OCTAL, can be used at a given time. However, each of them can be combined with the keywords, CHAR, CLEAR, or KANA, to display the character symbols along with their numerical character codes. Only one of the character displaying keywords, CHAR, CLEAR or KANA can be used at a given time.

DISPLAY NUMERICAL CODES

HEX
OCTAL

2-17. EXAMPLES

```
>FROM=TEXT3; TO=ALP; OCTAL
```

The above example produces the following type of display:

TEXT3.PUB.ACCOUNT RECORD 9 (%11)

```
000000: 052110 044523 020111 051440 046111 047105 020061 027040
000010: 020124 044105 020070 026504 044507 044524 020116 052515
000020: 041105 051123 020101 052040 052110 042440 051111 043510
000030: 052040 040522 042440 052110 042440 020040 020040 020040
000040: 020040 020040 020040 020040 030060 030061 030060 030060
```

TEXT3.PUB.ACCOUNT RECORD 10 (%12)

```
000000: 052105 054124 020105 042111 052117 051040 046111 047105
000010: 020116 052515 041105 051123 027040 020116 047524 042440
000020: 052110 040524 020124 044105 020122 042503 047522 042040
000030: 047125 046502 042522 051440 020040 020040 020040 020040
000040: 020040 020040 020040 020040 030060 030061 030460 030060
```

TEXT3.PUB.ACCOUNT RECORD 11 (%13)

```
000000: 051525 050120 046111 042504 020102 054440 043103 047520
000010: 054440 051524 040522 052040 040524 020132 042522 047440
000020: 053510 044514 042440 052110 042440 052105 054124 020105
000030: 042111 052117 051040 046111 047105 020040 020040 020040
000040: 020040 020040 020040 020040 030060 030061 031060 030060
```

The file TEXT3.PUB.ACCOUNT is an ASCII file created using the HP 3000 Text Editor. The octal word number at the start of each line of data specifies the sequential number within the record of the first word of data in the line. The same file is used in the next example (a combination hexadecimal and character display).

```
>FROM=TEXT3; TO=ALP; HEX; CHAR;
>TITLE="TITLE LINE FOR CHAR/HEX DISPLAY EXAMPLE."
```

The above example produces the following type of display:

TITLE LINE FOR CHAR/HEX DISPLAY EXAMPLE.

TEXT3.PUB.ACCOUNT RECORD 0 (%0)

```
000000: 5448 4953 2049 5320 4C49 4E45 2031 2E20 THIS IS LINE 1.
000010: 2054 4845 2038 2D44 4947 4954 204E 554D THE 8-DIGIT NUM
000020: 4245 5253 2041 5420 5448 4520 5249 4748 BERS AT THE RIGH
000030: 5420 4152 4520 5448 4520 2020 2020 2020 T ARE THE
000040: 2020 2020 2020 2020 3030 3030 3130 3030 00001000
```

DISPLAY NUMERICAL CODES

HEX
OCTAL

TEXT3.PUB.ACCOUNT RECORD 1 (%1)

```
000000: 5445 5854 2045 4449 544F 5220 4C49 4E45 TEXT EDITOR LINE
000010: 204E 554D 4245 5253 2E20 204E 4F54 4520 NUMBERS. NOTE
000020: 5448 4154 2054 4845 2052 4543 4F52 4420 THAT THE RECORD
000030: 4E55 4D42 4552 5320 2020 2020 2020 2020 NUMBERS
000040: 2020 2020 2020 2020 3030 3030 3230 3030 00002000
```

TEXT3.PUB.ACCOUNT RECORD 2 (%2)

```
000000: 5355 5050 4C49 4544 2042 5920 4643 4F50 SUPPLIED BY FCOP
000010: 5920 5354 4152 5420 4154 205A 4552 4F20 Y START AT ZERO
000020: 5748 494C 4520 5448 4520 5445 5854 2045 WHILE THE TEXT E
000030: 4449 544F 5220 4C49 4E45 2020 2020 2020 DITOR LINE
000040: 2020 2020 2020 2020 3030 3030 3330 3030 00003000
```

The file TEXT3.PUB.ACCOUNT is an ASCII file created using the HP 3000 Text Editor. The octal word number at the start of each line of data specifies the sequential number within the record of the first word of data in the line. The same file is used in the next example (a combination octal and character display).

```
>FROM=TEXT3; TO=*LP; OCTAL; CHAR; &
>TITLE="TITLE LINE FOR CHAR/OCTAL DISPLAY EXAMPLE."
```

The above example produces the following type of display:

TITLE LINE FOR CHAR/OCTAL DISPLAY EXAMPLE.

TEXT3.PUB.ACCOUNT RECORD 0 (%0)

```
000000: 052110 044523 020111 051440 046111 047105 020061 027040 THIS IS LINE 1.
000010: 020124 044105 020070 026504 044507 044524 020116 052515 THE 8-DIGIT NUM
000020: 041105 051123 020101 052040 052110 042440 051111 043510 BERS AT THE RIGH
000030: 052040 040522 042440 052110 042440 020040 020040 020040 T ARE THE
000040: 020040 020040 020040 020040 030060 030060 030460 030060 00001000
```

TEXT3.PUB.ACCOUNT RECORD 1 (%1)

```
000000: 052105 054124 020105 042111 052117 051040 046111 047105 TEXT EDITOR LINE
000010: 020116 052515 041105 051123 027040 020116 047524 042440 NUMBERS. NOTE
000020: 052110 040524 020124 044105 020122 042503 047522 042040 THAT THE RECORD
000030: 047125 046502 042522 051440 020040 020040 020040 020040 NUMBERS
000040: 020040 020040 020040 020040 030060 030060 031060 030060 00002000
```

TEXT3.PUB.ACCOUNT RECORD 2 (%2)

```
000000: 051525 050120 046111 042504 020102 054440 043103 047520 SUPPLIED BY FCOP
000010: 054440 051524 040522 052040 040524 020132 042522 047440 Y START AT ZERO
000020: 053510 044514 042440 052110 042440 052105 054124 020105 WHILE THE TEXT E
000030: 042111 052117 051040 046111 047105 020040 020040 020040 DITOR LINE
000040: 020040 020040 020040 020040 030060 030060 031460 030060 00003000
```

IGNORE ERRORS

IGNERR=

2-18. IGNORE ERRORS

The ignore errors function bypasses certain errors, such as bad blocks (physical records in which a parity error is detected), in a magnetic tape “from” file and reports each such error so ignored. It can be used only when the “from” file resides on magnetic tape.

The ignore errors function has the following format:

IGNERR=*number-of-errors*

where *number-of-errors* is an integer specifying the maximum number of errors to be ignored. If the specified number of errors is exceeded, the copying operation terminates.

In the following example, a maximum of five errors will be ignored while copying the first file from the magnetic tape unit *TAPE to the disc file FILE3 (the copying operation terminates if a sixth error is encountered).

```
>FROM=*TAPE; TO=FILE3; SUBSET; IGNEPP=5
```

MULTIPLE TAPE FILE COPYING

FILES =

2-19. MULTIPLE TAPE FILE COPYING

The FILES function of FCOPY allows the user to copy multiple files from both labeled and unlabeled magnetic tapes.

The multiple file copying function has the following format:

$$\text{FILES} = \begin{cases} \text{number-of-files} \\ \text{ALL} \end{cases}$$

Where:

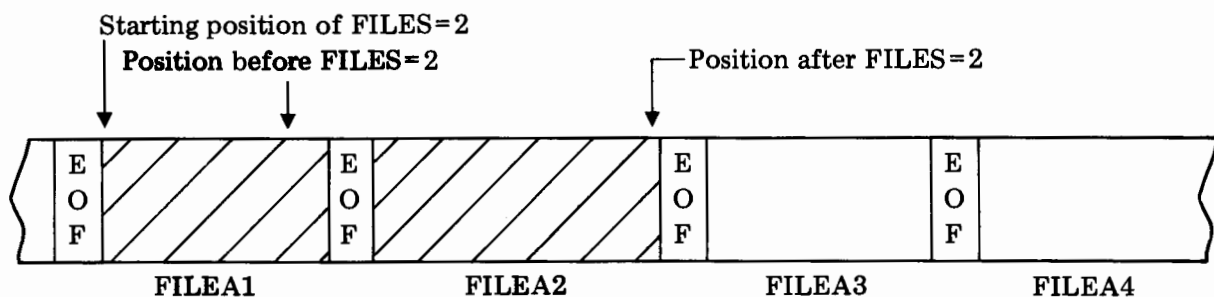
number-of-files is an integer specifying the number of files to be copied.

ALL specifies that all files from the current file to the end of the tape are to be copied.

With a single exception (number 4 below), FILES= acts upon labeled and unlabeled tapes in the same way:

1. FCOPY positions the tape at the beginning of the current file.

To illustrate:



The shaded area represents the data copied.

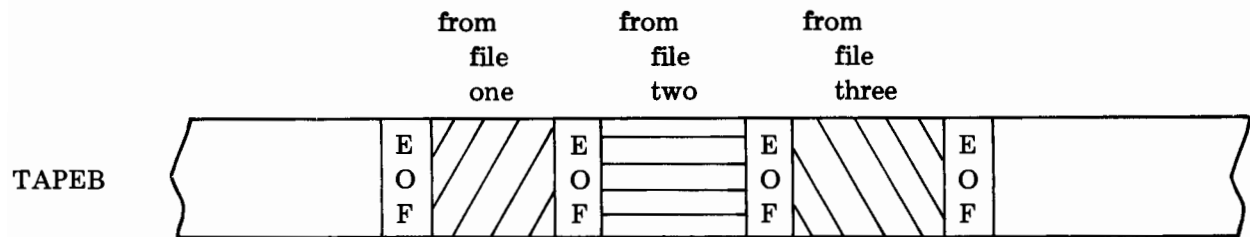
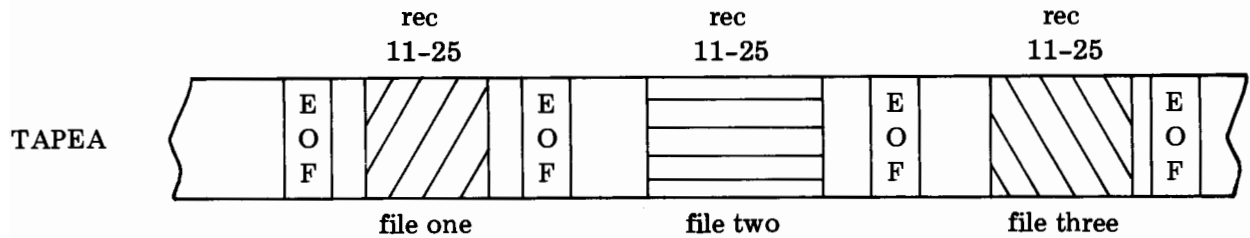
2. When SUBSET= is used in conjunction with the FILES= function, FCOPY assumes that the SUBSET function applies to every file to be copied.

To illustrate:

```
> FROM=*TAPEA, TO=*TAPEB, FILES=3, SUBSET=11:25
```

MULTIPLE TAPE FILE COPYING

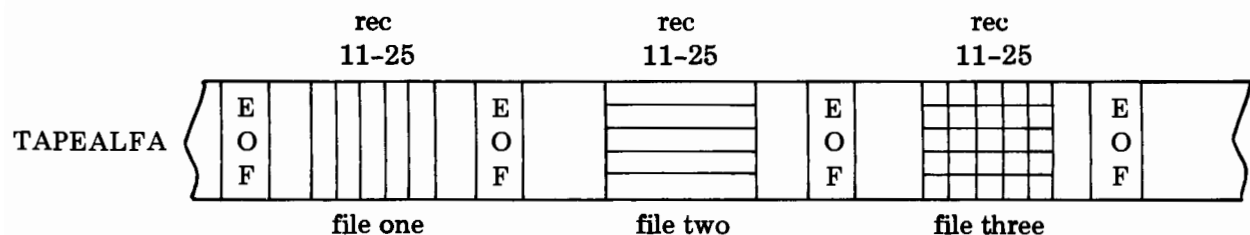
FILES =



- Note that if the FILES= function is used in conjunction with the SKIPEOF= function, the skipping is done first.
- Any time that the FILES= function is used with a LABELED tape, unless the user wants to copy the same file repeatedly, NEXT must be specified as the sequence option of the ;LABEL parameter in the :FILE equation. For more information on the sequence option of the :FILE command, refer to the FILE command in the MPE Commands Reference Manual.

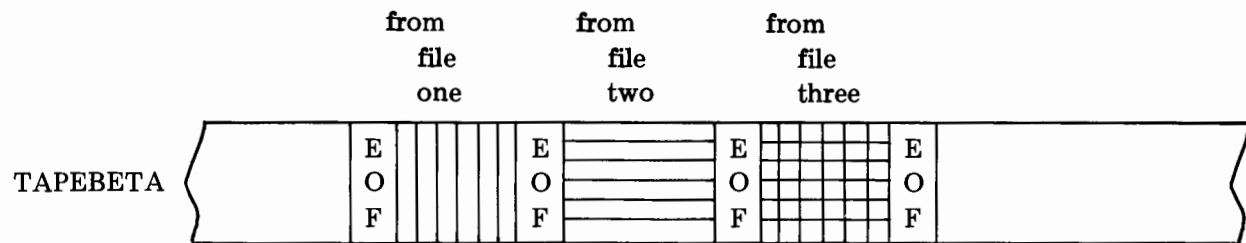
To illustrate:

```
FILE TAPEALFA; LABEL=, , NEXT
FROM=*TAPEALFA, TO=TAPEALFA; FILES=3; SUBSET=11:25
```

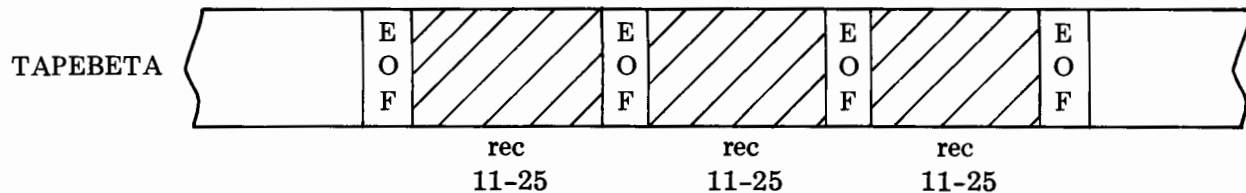
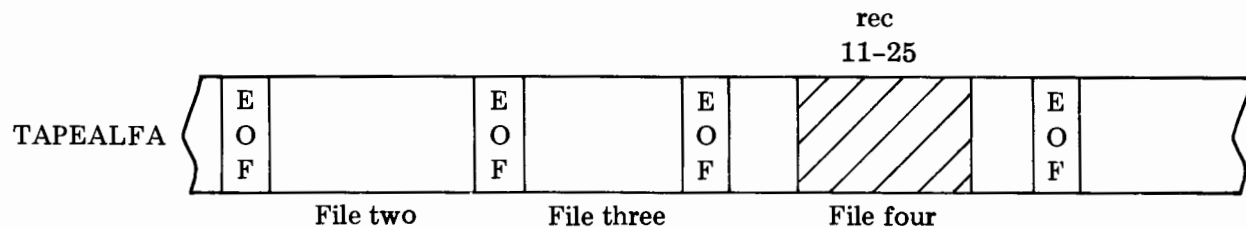


MULTIPLE TAPE FILE COPYING

FILES =



As a counter illustration, assume that the file equation is :FILE TAPEALFA;LABEL=,,,4 and the FCOPY command is the same as in the above illustration.



All from file four of TAPEALFA

EXAMPLES:

> FROM=*T1; TO=*T7; FILES=4

In the above example, FCOPY copies four files, starting in the current file, from magnetic tape *T1 to tape *T7.

MULTIPLE TAPE FILE COPYING

FILES =

```
> FROM=*T5; TO=*T2; SKIPEOF=3,+4; FILES=2
```

In the example above, FCOPY positions *T5 on its third file, positions *T2 four files forward from its starting position, and then copies two files (files 3 and 4) from *T5 to *T2.

```
> FROM=*ME; TO=*YOU; FILES=ALL
```

In the above example, FCOPY copies all of the files of *ME into files of *YOU.

NEW FILE

NEW

2-20. NEW FILE

The new file function creates a new permanent disc file which FCOPY then uses as the “to” file in performing the requested copying operation. The new file exists in the account and group specified at the start of your job or session and has the *filename* specified in the TO= parameter of the FCOPY command.

The new file function has the following format:

NEW

In the following example, FCOPY creates a new permanent disc file named NEWFILE in the user's log-on account and group and then copies the disc file OLDSTUFF into NEWFILE.

```
> FROM=OLDSTUFF, TO=NEWFILE, NEW
```

If an FCOPY operation containing the new file function is terminated by an error or by the use of the CONTROL-Y key, the new file is discarded.

FCOPY interacts directly with the MPE/3000 File System to create the new disc file. If you are going to subsequently manipulate the newly-created file, you should know the characteristics of the file. These characteristics are summarized in Appendix B of this manual.

Note that you may predefine characteristics of the new file by issuing an MPE :FILE command for the file and then backreferencing to that :FILE command in the particular FCOPY command.

KSAM files can be copied into new KSAM files by using the following form of the TO= parameter:

TO=(*dfile*, *kfile*)

where *dfile* is the new data file name, and *kfile* is the new key file name.

NOTE: This syntax is not recognized by a system that does not contain the KSAM utility subsystem.

When using this form of the TO parameter, the “from” file must be a KSAM file, and the “to” file is created by FCOPY. For example:

```
> FROM=KSAM1, TO=(KSAM2,KSAM2KEY)
```

this command copies KSAM1 into a new KSAM file named KSAM2 using the same parameters as KSAM1. Since only active records are copied, KSAM2 is a compact version of KSAM1. The new key file, KSAM2KEY, duplicates the characteristics of the key file for KSAM1.

Note that the option ;NEW was not used in the above example. It was not required because both a new data file and a new key file were specified. The command could have been

```
> FROM=KSAM1, TO=(KSAM2,KSAM2KEY);NEW
```

with the same effect as the previous command.

If the user wishes to copy an old KSAM file, the standard copy function, FROM=*filename*;TO=*filename*, is used.

2-21. NON-KSAM

The non-KSAM function is used to copy the data file of a fixed record length keyed sequential access method (KSAM) file into another file. Thus the "from" file must be a KSAM file; there are no restrictions on the "to" file.

NOTE: As with the KEY= function, and the (dfile,kfile) syntax, this keyword is not recognized by a system that does not contain the KSAM utility subsystem.

The non-KSAM function has the following format:

NOKSAM

2-22. EXAMPLES

```
> FROM=KSAMFILE, TO=FILEX, NOKSAM
```

In the above example, all records in the data file of KSAMFILE are copied, in the same order that they appear in the file, into FILEX.

```
> FROM=KSAMONE, TO=, NOKSAM, SUBSET=#%377,%377#
```

In the above example, all of the deleted records in the KSAM file named KSAMONE are listed on the standard list device.

For a complete description of KSAM files and how they operate with FCOPY, refer to the KSAM/3000 Reference Manual.

NO USER LABELS

NOUSERLABELS

2-23. NO USER LABELS

The NOUSERLABELS function allows the user to omit user labels when copying a tape or disc file into another file. If the NOUSERLABELS function is not specified, and both the "from" and "to" files are labeled, FCOPY copies the label of the "from" file onto the "to" file label.

The NOUSERLABELS keyword can only be used with two disc files if your HP 3000 runs under the control of the MPE-C operating system; it does not function on such a system when a magnetic tape and a disc file are specified.

The NOUSERLABELS function has the following format:

NOUSERLABELS

When copying from a tape to a disc or from a disc to a tape, the user should be aware of what happens to the labels.

A disc user label is 256 bytes long, and a user label on a tape is 80 bytes long. If NOUSERLABELS is omitted when copying from a disc to a tape, only the first 80 bytes of the label are copied. The last 176 are truncated; if the NOUSERLABEL function is omitted when copying from a tape to a disc file, the user label is padded with zeros in order to make it the required 256 bytes long.

A warning message is issued by FCOPY when the user copies a user labeled tape file onto a disc file, or from a disc file to a tape file.

EXAMPLES:

```
> FROM=*TAPEA, TO=DISC1, SUBSET, NOUSERLABELS
```

In the above example, FCOPY copies the first file of *TAPEA onto a file of DISC1, without copying the user label of the tape file.

```
> FROM=*, TO=DISC2, SUBSET
```

In the above example, FCOPY copies the second file of TAPEA, including the user label, onto DISC2. The last 176 bytes of the new disc label are zeros.

```
> FROM=DISCLBLD, TO=*TOTAPE1
```

In the above example, FCOPY copies the first 80 bytes of the user label on DISCLBLD onto the tape file *TOTAPE1, and then copies the file.

SKIP END-OF-FILE

SKIPEOF =

2-24. SKIP END-OF-FILE

The skip end-of-file function positions a magnetic tape at the desired file before any copying is performed. It can only be used in conjunction with magnetic tape units.

THIS FUNCTION CANNOT BE USED WITH LABELED TAPES.

The skip end-of-file function has the following format:

$$\text{SKIPEOF} = \left[\left\{ \begin{array}{l} \{+/-\} \text{ from-eofs} \\ \text{from-file-number} \end{array} \right\} \right] \left[, \left\{ \begin{array}{l} \{+/-\} \text{ to-eofs} \\ \text{to-file-number} \end{array} \right\} \right]$$

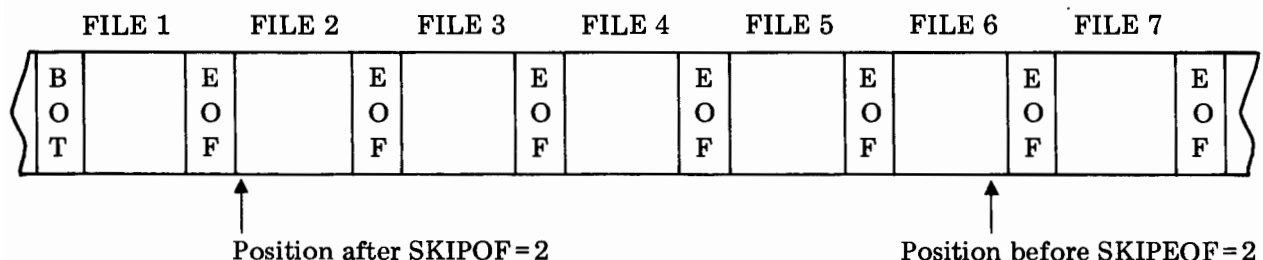
Where:

<i>+/-</i>	specify forward or reverse movement. A plus sign specifies forward; a minus sign specifies reverse.
<i>from-eofs</i>	is an integer specifying how many files are to be skipped on the "from" tape.
<i>from-file-number</i>	is an integer specifying an absolute file number on the "from" tape. Absolute file numbers begin with 1 for the first file on the tape.
<i>to-eofs</i>	is an integer specifying how many files are to be skipped on the "to" tape.
<i>to-file-number</i>	is an integer specifying an absolute file number on the "to" tape.

The following illustrations make clear how SKIPEOF= works under various circumstances.

1. For *from-* or *to-file-numbers*:

- Zero (0) cannot be specified for a from-file-number or a to-file-number.
- When a non-zero from- or to-file-number is specified, it does not matter what the current position of the tape is; SKIPEOF= positions the tape to the beginning of the specified file. For SKIPEOF=2:

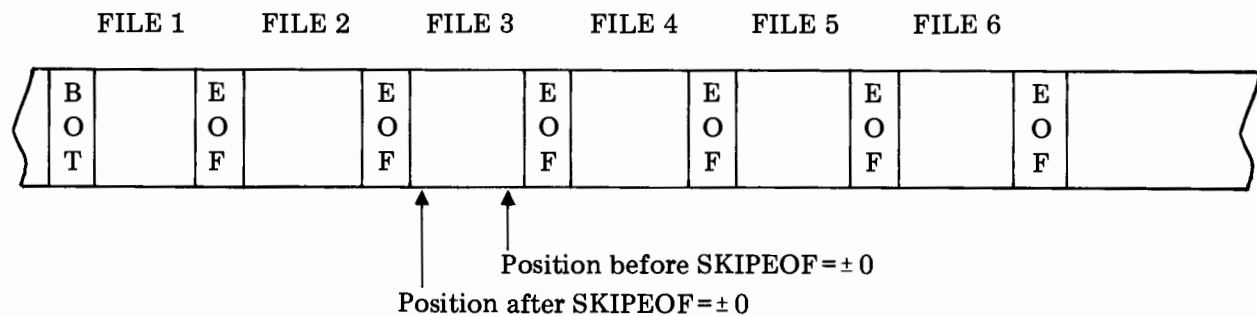


SKIP END-OF-FILE

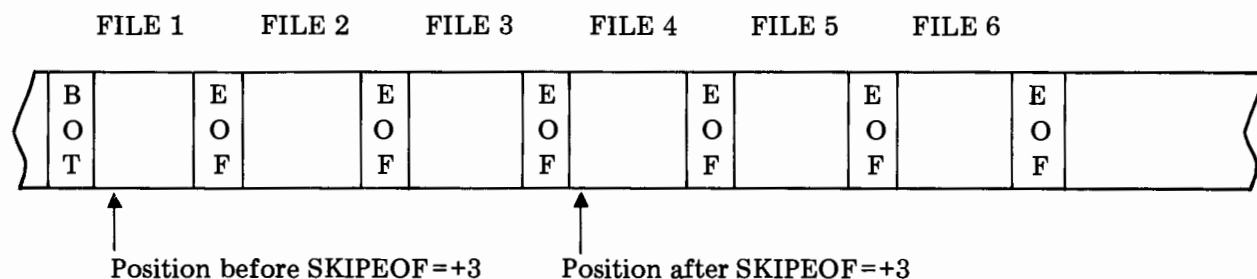
SKIPEOF =

2. For *from-* or *to-eofs*:

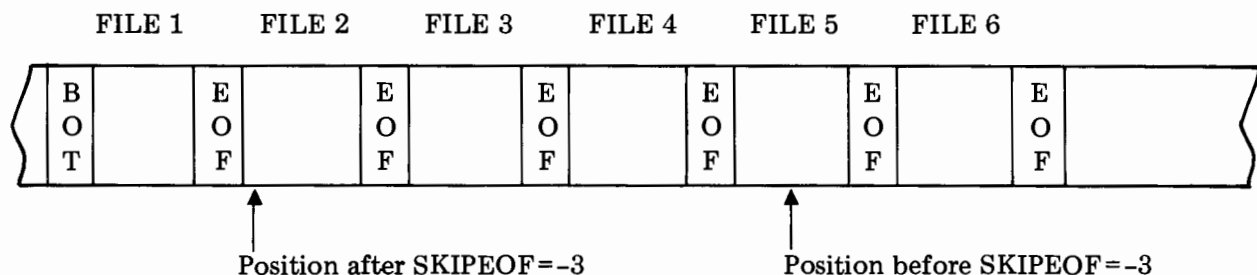
- a) When +0 or -0 is specified, SKIPEOF= positions the tape at the beginning of the current file:



- b) For any non-zero positive integer, a, SKIPEOF=+a simply skips forward over the specified number of EOF marks. For SKIPEOF=+3:



- c) For any non-zero positive integer, b, SKIPEOF=-b first skips b+1 EOF marks in the reverse direction, and then skips forward one EOF mark. This is done to position the tape at the beginning of a file, rather than at the end of the preceeding file. For SKIPEOF=-3:

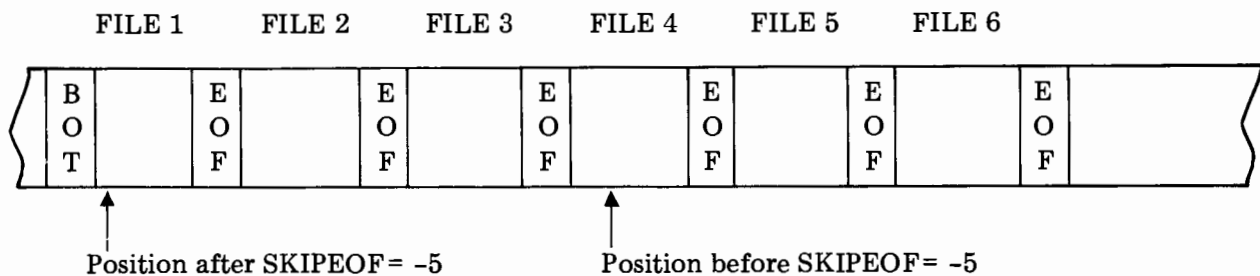


SKIP END-OF-FILE

SKIPEOF =

d) Note that whenever an integer is specified that would skip beyond the beginning-of-tape mark (BOT),

- 1) In a job, an error message is issued and the FCOPY program is aborted.
- 2) In a session, SKIPEOF= positions the tape at the beginning of the first file on the tape after issuing a warning message. For SKIPEOF= -5:



2-25. EXAMPLES

> FROM=*TAPES, TO=*TAPE2, SKIPEOF=+3, 7

In the above example, FCOPY positions the magnetic tape *TAPES three files ahead of its current position, and copies the information on that file onto the seventh file of *TAPE2. Note that if, for example, skipping on *TAPES starts from the second file mark, FCOPY begins to copy data from the sixth file, whereas no matter where skipping begins on *TAPE2, FCOPY copies data onto its seventh file.

> FROM=*TAPEA, TO=*TAPEB; SKIPEOF=5, -4

In the above example, FCOPY skips to the fifth file of *TAPEA, and positions *TAPEB four files back from its current position. FCOPY then copies file 5 of *TAPEA to the selected file on *TAPEB.

> FROM=*THISTAPE; TO=*THATTAPE, SKIPEOF=4,5

In the above example, FCOPY skips to the fourth file of *THISTAPE, and to the fifth file of *THATTAPE before copying the data from *THISTAPE to *THATTAPE.

> FROM=*MYTAPE; TO=*YOURTAPE; SKIPEOF= ,9

In the above example, FCOPY does not move *MYTAPE from its current position, while *YOURTAPE is positioned at the ninth file. FCOPY then copies the file at the current position of *MYTAPE into the ninth file of *YOURTAPE.

> FROM=*TAPEONE; TO=*TAPETWO; SKIPEOF=-2

In the above example, FCOPY skips two positions back from the current file of *TAPEONE, but does not change the position of *TAPETWO. FCOPY then copies a file from *TAPEONE to *TAPETWO.

SPECIFY KEY

KEY =

2-26. SPECIFY KEY

The specify key function is used only when the "from" file is a Keyed Sequential Access Method (KSAM) file. There are no restrictions on the "to" file. This function specifies a key whose value determines the sequence in which the "from" file is copied.

NOTE: Your system must contain the KSAM utility subsystem in order for this keyword to function.

The specify key function has the following format:

KEY=*character location*

where *character location* is a positive integer that identifies the key by its starting character location in the data file. The key indicated may be the primary key or an alternate key.

The use of the specify key function is optional. When KEY= is not specified in an FCOPY command, the KSAM file is copied in primary key sequence to another KSAM file.

If zero is specified as the character location (KEY=0), the file is copied in chronological sequence (the sequence in which the records were originally written in the file) rather than in a key sequence.

For a complete description of KSAM files and how they operate with FCOPY, refer to the *KSAM/3000 Reference Manual*.

2-27. EXAMPLES

Defined Key:

```
>FROM=KSAM1,TO=ALPHA,KEY=21
```

In the above example, assuming that a key has been designated at character 21, the file is copied in key sequence (alphabetical or numerical sequence according to the characters starting in character location 21).

Implied Primary Key:

```
>FROM=KSAM1,TO=KSAM2
```

In the above example, the file is copied in primary key sequence (alphabetical or numerical sequence of the characters in the primary key). Note that omitting the KEY= parameter causes a default to the primary key sequence being used.

Chronological Order:

```
>FROM=KSAM1,TO=DISC1,KEY=0
```

In the above example, the file is copied in chronological sequence (the sequence in which the records were originally placed in the file) rather than in a key sequence.

2-28. SUBSET

The subset function makes it possible for you to perform copy operations using only a specified portion (subset) of a file. You may designate the file subset as being all records that contain a certain data item or as being a range of contiguous records.

If the “from” file is on a disc, and the file does not fill the space designated for it, then when the SUBSET keyword is used, the “to” file is only as large as the actual contents of the “from” file. Thus, unused space is not copied. In addition, when the “from” file is a magnetic tape unit you may use the subset function (the keyword SUBSET by itself) to limit the FCOPY operation to a single file on that tape.

In addition, when the “from” file is a magnetic tape unit you may use the subset function (the keyword SUBSET by itself) to limit the FCOPY operation to a single file on that tape.

FCOPY selects magnetic tape subsets by logical record number. If physical block subsets are needed, they may be gotten by temporarily defining a logical record length equal to the physical record length for the duration of the copying operation.

The subset function has the following four formats:

```
SUBSET="characterstring"[:,column] [,EXCLUDE]]  
SUBSET=# patternlist[:,column] [,EXCLUDE]]  
SUBSET=(range[:range; . . . ])  
SUBSET
```

where: *characterstring* is a string of ASCII characters.

patternlist is a string of decimal and/or octal integers, each representing the bit pattern of one byte.

column is an integer specifying at which character position within each record the particular *characterstring* or *patternlist* is expected to begin. If omitted, the default value 1 (first character position) is assumed.

EXCLUDE specifies that the designated subset is to be excluded from the FCOPY operation (all of the file *except the subset* is to be used in the operation).

range is either of the form

[*starting-record-number*] [,*number-of-records*]

or

[*starting-record-number*] [:*last-record-number*]

and

starting-record-number is an integer record number specifying the start of the subset. If omitted, the default value 0 (first record) is assumed.

number-of-records is an integer specifying the size of the subset in records. If omitted, it is assumed that the subset consists of all records through the end of the file.

last-record-number is an integer record number specifying the last record in the subset. If omitted, the last record in the file is assumed.

SUBSET

SUBSET =

2-29. COMMENTS

characterstring must contain no more than 35 characters within the quotes. Also, a *characterstring* may not be continued from one line or record to the next. If a quote mark is needed with *characterstring*, use two successive quote marks; each such pair is counted as one character.

patternlist must contain no more than 35 integers. Note that although you may continue a *patternlist* from one line or record to the next, the continuation must not be specified between elements of an individual pattern. It must be specified between the separating comma and the beginning of the next pattern. Each decimal integer must be within the range 0 through 255₁₀ and each octal integer must be within the range 0 through 377₈.

If the *characterstring* or *patternlist* specified does not exist entirely within a particular record in the "from" file, the record is considered *not* to be part of the defined subset.

If the "from" file is on magnetic tape, the keyword SUBSET by itself specifies that only one file on the tape is to be copied. If the "from" file is on magnetic tape and the keyword SUBSET is omitted, the entire reel is copied. Note that using the keyword SUBSET by itself is equivalent to using FILES=1; omitting SUBSET is equivalent to using FILES=ALL.

A subset may not extend over an end-of-file (EOF) mark or a tape mark boundary on magnetic tape.

The user may specify up to 255 numerical subsets. The subsets are copied sequentially in the order listed, and must be in numerical order; they must not overlap.

If the DEBLOCK= function is used in the same command line as the subset function, subsets are taken from the deblocked records.

2-30. EXAMPLES

```
> FROM=MASTER; TO=MEN; SUBSET="MALE",17
```

In the above example, all records in the disc file MASTER which contain the character string MALE starting in character position 17 are copied into the disc file MEN.

```
> FROM=MASTER; TO=WOMEN; SUBSET="MALE",17,EXCLUDE
```

In the above example, all records in the disc file MASTER which *do not* contain the character string MALE starting in character position 17 are copied into the disc file WOMEN.

```
> FROM=DATA; TO=SUBFILE3, SUBSET=%144,%141,%164,%141#,77
```

This variation of the SUBSET function is useful for specifying lowercase alphabetic characters from a terminal which has only uppercase characters. For example, the above command causes all records in the disc file DATA which contain the lowercase alphabetic characters "data" (octal codes 144, 141, 164, and 141) in character positions 77 through 80 to be copied into the disc file SUBFILE3.

SUBSET

SUBSET =

```
> FROM=DATA; TO=SUB1; SUBSET=#13,10#.71
```

This variation of the SUBSET function is also useful for specifying non-printing characters. For example, the above command causes all records in the disc file DISC which contain the non-printing control characters CR and LF (decimal codes 13 and 10) in character positions 71 and 72 to be copied into the disc file SUB1.

```
> FROM=MASTER, TO=SUB1; SUBSET=,500
```

In the above example, 500 records starting with record 0 (the first record) are copied from the disc file MASTER to the disc file SUB1.

```
> FROM=MASTER; TO=SUB2; SUBSET=500:1499
```

In the above example, records 500 through 1499 are copied from the disc file MASTER into the disc file SUB2.

```
> FROM=MASTER, TO=SUB3; SUBSET=1500
```

In the above example, all records from record 1500 through the end of the file are copied from the disc file MASTER into the disc file SUB3.

```
> FROM=*TAPE; TO=DISCFILE; SUBSET
```

In the above example, one file (the first one on the tape) is copied from the magnetic tape *TAPE into the disc file DISCFILE.

```
> FROM=*TAPE; TO=DISC7; SKIPEOF=2; SUBSET
```

In the above example, one file (the second one on the tape) is copied from the magnetic tape *TAPE into the disc file DISC7.

```
> FROM=*TAPEA, TO=FILEONE; SUBSET=(0,12,30:75)
```

In the above example, records 0 through 11 and 30 through 75 are copied from TAPEA to FILEONE. Note that the subsets are sequentially ordered and do not overlap. As a counter example, consider the following example:

```
> FROM=DISCDATA; TO=FILEONE; SUBSET=(12,9;8:15)
```

The subsets are incorrectly specified, since the first one is not numerically ordered, and the second overlaps the first.

UPSHIFT

UPSHIFT

2-31. UPSHIFT

The upshift function converts lower-case roman alphabetic characters to upper case as part of a copying operation. Most devices which can only print uppercase roman alphabetic characters automatically take care of upshifting lowercase alphabetic characters. If you are copying to such a device, the FCOPY upshift function need not be used. If you are copying to a device which prints only uppercase alphabetic characters and does *not* automatically upshift, then the FCOPY upshift function should be used.

The upshift function has the following format:

UPSHIFT

In the following example, any lower-case roman alphabetic characters are converted to upper case as the disc file LOWER is copied to the disc file UPPER. The content of LOWER remains unchanged.

```
>FROM=LOWER; TO=UPPER; UPSHIFT
```

VERIFY

VERIFY =

2-32. VERIFY

The verify function compares the content of the "to" file against the content of the "from" file, record-by-record, *immediately after a copying operation*. The two files must reside on either disc or magnetic tape and must have identical record characteristics (record size, record format (fixed/variable/undefined), blocking factor, record type (ASCII/binary)).

The FCOPY compare and verify functions are essentially the same except that the verify function is always performed in conjunction with a file copying operation whereas the compare function is not.

The verify function has the following format:

VERIFY=number-of-errors

where *number-of-errors* is an integer specifying the maximum number of comparison errors to be tolerated. If this number is reached, the operation terminates. If *number-of-errors* is omitted, the default value is 1.

Each time FCOPY detects an error during the verify phase, it prints a message on the standard listing device (your terminal in a session or a line printer in a job) specifying the number of the "from" file record in which the error occurred.

2-33. EXAMPLES

```
>FROM=OLDDISC; TO=COPY; VERIFY
```

In the above example, FCOPY copies the disc file OLDDISC to the disc file COPY and then compares the two files record-by-record. If it finds an error, FCOPY prints a message on the standard listing device and terminates the operation.

```
>FROM=*TAPE2; TO=*TAPE6; VERIFY=10
```

In the above example, FCOPY copies the magnetic tape *TAPE2 to the magnetic tape *TAPE6 and then rewinds the two tapes and compares them record-by-record. Each time it finds an error, FCOPY prints a message on the standard listing device. The tenth error terminates the operation.

This section presents nine annotated examples of the use of FCOPY/3000. In the session examples, all characters entered by the user are underlined; the remaining characters are issued by FCOPY. In the job examples, all commands submitted by the user as part of his job deck are underlined.

Example 1. During a session, a card deck (containing 753 cards) is to be listed on a line printer. In this example, CARD and LP are the device class names for a card reader and line printer, respectively.

```

.
.
.

:FILE CARDS;DEV=CARD
:FILE LIST;DEV=LP
:RUN FCOPY.PUB.SYS
HP32212A,3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977

>FROM=*CARDS;TO=*LIST
*200*L
WARNING: FROMFILE RECSIZE IS 80 BYTES, TOFILE RECSIZE IS 132 BYTES.
CONTINUE OPERATION (Y OR N)? Y
EOF FOUND IN FROMFILE AFTER RECORD 752

753 RECORDS PROCESSED *** 0 ERRORS

>EXIT

.
.
.
```

Example 2. During a job, a card deck (containing 947 cards) is to be copied to a magnetic tape, 10 records (cards) per block. In this example, TAPE is the device class name for a magnetic tape unit.

```

.
.

:FILE T;DEV=TAPE;REC=-80,10,F,ASCII
:RUN FCOPY.PUB.SYS
FROM=;TO=*T

(the card deck)

:EOD

.
.
```

The following messages appear in the job output printed on the standard job listing device \$STDLIST:

EOF FOUND IN FROMFILE AFTER RECORD 946

947 RECORDS PROCESSED *** 0 ERRORS

Example 3. During a session, one record (the 12th) in a disc file called CUSTDATA is to be displayed at the user's terminal in both octal and character formats.

```

      .
      .
      .

:RUN FCOPY.PUB.SYS
HP32212A,3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977

>FROM=CUSTDATA;TO=;SUBSET=11,1;OCTAL;CHAR
CUSTDATA.PUB.ACCOUNT RECORD 11 (%13)

000000: 040512 040530 020115 043107 AJAX MFG
000004: 027040 041517 027040 031054 . CO. 2,
000010: 030460 030456 033463 020040 101.73
000014: 030460 026063 033471 027061 10,379.1
000020: 030040 020070 033463 027061 0 873.1
000024: 034040 020061 026060 030471 8 1,019
000030: 027067 032440 020062 032465 .75 255
000034: 027062 034440 020040 020040 .29
000040: 020040 020040 020040 020040
000044: 020040 020040 020040 020040
1 RECORD PROCESSED *** 0 ERRORS

>EXIT
END OF PROGRAM

      .
      .
      .

```

NOTE: FCOPY always considers the first record to be record number 0.

Example 4. During a job, an ASCII character disc file called SPLTEXT containing both uppercase and lowercase alphabetic characters is to be listed entirely in uppercase on a line printer which is capable of printing both uppercase and lowercase alphabetic characters. The FCOPY upshift function converts any lowercase alphabetic characters to uppercase as the data is copied from the disc to the line printer (the data in the disc file remains unchanged). In this example, the disc file contains 424 records and LP2 is the device class name of the line printer.

```

      .
      .
      .

:FILE LIST;DEV=LP2
:RUN FCOPY.PUB.SYS
FROM=SPLTEXT;TO=*LIST;UPSHIFT
EXIT
      .
      .
      .

```

The following messages appear in the job output printed on the standard job listing device \$STDLIST:

```

      EOF FOUND IN FROMFILE AFTER RECORD 423

      424 RECORDS PROCESSED *** 0 ERRORS

```

Example 5. During a session, two subsets are to be extracted from a disc file called EMPRECS. One subset, consisting of those records which contain the keyword MALE starting in the fortieth byte position, is to be copied to a disc file called MEN. The other subset, consisting of all other records, is to be copied to a disc file called WOMEN. EMPRECS contains a total of 5986 records, 2859 of which contain the keyword MALE starting in position 40 and 3127 of which contain the keyword FEMALE starting in position 40.

```

      .
      .
      .

:RUN FCOPY.PUB.SYS
HP32212A,3.00 FILE COPIER (C) HEWLETT-PACKARD CO, 1977

>FROM=EMPRECS;TO=MEN;SUBSET="MALE",40
EOF FOUND IN FROMFILE AFTER RECORD 5985

2859 RECORDS PROCESSED *** 0 ERRORS

>FROM=EMPRECS;TO=WOMEN;SUBSET="MALE",40,EXCLUDE
EOF FOUND IN FROMFILE AFTER RECORD 5985

3127 RECORDS PROCESSED *** 0 ERRORS

>EXIT
      .
      .
      .

```

On System #1:

>EXIT

The diagram illustrates a tape structure with four segments: FUTIL, DUMPER, SPOOLER, and an unlabeled segment. Arrows point to the boundaries between segments, labeled 'LOAD POINT' and 'EOF'.

On System #2:

```
.  
.  
.  
  
:BUILD FUTIL;DISC=298,1,1;CODE=PROG  
:BUILD DUMPER;DISC=406,1,1;CODE=PROG  
:BUILD SPOOLER;DISC=387,1,1;CODE=PROG  
:FILE T;DEV=TAPE  
:RUN FCOPY.PUB.SYS  
HP32212A,3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977
```

```
>FROM=*T;TO=FUTIL;SUBSET  
EOF FOUND IN FROMFILE AFTER RECORD 297
```

```
298 RECORDS PROCESSED *** 0 ERRORS
```

```
>FROM=*,TO=DUMPER;SKIPEOF=+1;SUBSET  
EOF FOUND IN FROMFILE AFTER RECORD 405
```

```
406 RECORDS PROCESSED *** 0 ERRORS
```

```
>FROM=*,TO=SPOOLER;SKIPEOF=+1;SUBSET  
EOF FOUND IN FROMFILE AFTER RECORD 386
```

```
387 RECORDS PROCESSED *** 0 ERRORS
```

```
>EXIT
```

NOTE: SKIPEOF=+1
because FCOPY left the
tape positioned just be-
fore the EOF found by
the previous command.

```
.  
.  
.
```

Example 7. During a job, three ASCII character data files on disc called DATA1, DATA2, and DATA3 (containing 611, 712, and 473 records, respectively) are to be concatenated onto one file on magnetic tape. The record size is 80 bytes and the magnetic tape is to have a blocking factor of 20 records per block. In this example, TAPE is the device class name of a magnetic tape unit.

```
.  
.  
.  
  
:FILE T;DEV=TAPE;REC=-80,20,F,ASCII  
:RUN FCOPY.PUB.SYS  
FROM=DATA1;TO=*T  
FROM=DATA2;TO=*  
FROM=DATA3;TO=*  
EXIT
```

```
.  
.
```

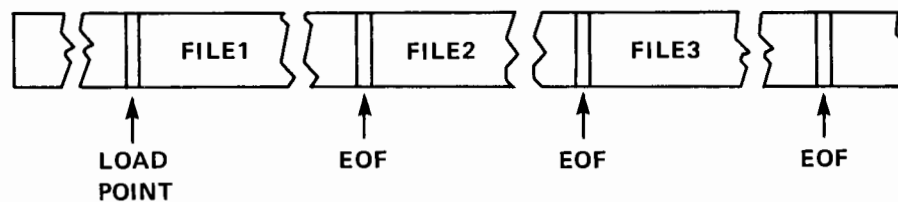
The following messages appear in the job output printed on the standard job listing device \$STDLIST:

```

EOF FOUND IN FROMFILE AFTER RECORD 610
611 RECORDS PROCESSED *** 0 ERRORS
EOF FOUND IN FROMFILE AFTER RECORD 711
712 RECORDS PROCESSED *** 0 ERRORS
EOF FOUND IN FROMFILE AFTER RECORD 472
473 RECORDS PROCESSED *** 0 ERRORS

```

Example 8. Three ASCII character files on magnetic tape look like this:



During a session, two subsets extracted from FILE1 and all of FILE3 are to be copied into an ASCII character disc file called AMALGUM. In this example, TAPE is the device class name of a magnetic tape unit.

```

:FILE AMALGUM;ACC=APPEND
:FILE T;DEV=TAPE;REC=-80,1,F,ASCII
:RUN FCOPY.PUB.SYS
HP32212A,3.00 FILE COPIER (C) HEWLETT-PACKARD CO. 1977

```

```

>FROM=*T;TO=*AMALGUM;SUBSET=4:12
9 RECORDS PROCESSED *** 0 ERRORS
>FROM=*;TO=*;SUBSET=19,16
16 RECORDS PROCESSED *** 0 ERRORS
>FROM=*;TO=*;SKIPEOF=3;SUBSET
EOF FOUND IN FROMFILE AFTER RECORD 302
303 RECORDS PROCESSED *** 0 ERRORS
>EXIT

```

•
•

Example 9. A magnetic tape reel contains only one file (containing 7439 records) which has a record length of 58 bytes and a blocking factor of one record per block. During a job, the file is to be copied to another reel to change only the blocking factor to 20 records per block. In this example TAPE is the device class name for a pair of magnetic tape units.

```
.  
.   
.   
  
:FILE T1;DEV=TAPE;REC=-58,1,F,ASCII  
:FILE T2;DEV=TAPE;REC=-58,20,F,ASCII  
:RUN FCOPY.PUB.SYS  
FROM=*T1;TO=*T2;SUBSET  
EXIT
```



The following messages appear on the job output printed on the standard job listing device \$STDLIST:

```
EOF FOUND IN FROMFILE AFTER RECORD 7438  
  
7439 RECORDS PROCESSED *** 0 ERRORS
```


FCOPY WARNING AND ERROR MESSAGES

SECTION

IV

FCOPY can detect two types of faults in an operation:

1. Errors such as incorrect syntax or parity errors while reading or writing.
2. Potential errors involving files, such as a conflict between the characteristics of the “from” file and those of the “to” file.

Depending upon whether the user is in an interactive session or submits a job, FCOPY reports and acts upon an error or warning message differently.

IN A BATCH JOB:

The complete error or warning message is always displayed on the standard listing device.

FCOPY handles potential problems with files by assuming that the command has been submitted as intended, and attempting to perform the requested operation.

IN AN INTERACTIVE SESSION:

If FCOPY detects an incorrect syntax or a parity error it displays the FCOPY reference number of the error. For example:

54

To request the complete message, type any printing character other than a colon and then press RETURN. FCOPY responds with the error or warning message:

*54*d
SYNTAX ERROR: UNKNOWN OPTION

To skip the message, simply press RETURN.

When FCOPY detects a potential file problem, the display is the same as for the above syntax error except that FCOPY then asks the user if a file information display is desired.

For example, if a user attempts to copy a file named THISFILE to a file named THATFILE, and the “from” file does not exist, FCOPY displays:

*105*a
CAN'T OPEN FROMFILE
DISPLAY FILE INFORMATION (Y OR N)?

A Y response results in an MPE file information display which, on a Series II system, is preceded by the file system error message. For example:

```
*105*
CAN'T OPEN FROMFILE
DISPLAY FILE INFORMATION (Y OR N)?y

NONEXISTENT PERMANENT FILE

+-F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y+
!  FILE NUMBER 0          IS UNDEFINED              !
!  ERROR NUMBER:  52      RESIDUE:  0                !
!  BLOCK NUMBER:  0          NUMREC;  0              !
+-----+
```

The error number within the file information display corresponds to the “NONEXISTENT PERMANENT FILE” message shown immediately above the display.

On a Series I system, the user does not receive the error message and must use the error number appearing in the file information display to look up the message. Messages corresponding to error numbers in the file information display are listed in table 10-4, the File System Error Table, which is in the MPE Intrinsic Reference Manual. A complete explanation of the file information display can also be found in the same manual.

4-1. FCOPY ERROR AND WARNING MESSAGES

Table 4-1 lists the FCOPY warning and error messages in numerical order along with a brief description of the meaning of each message and the recommended corrective action.

Table 4-1. FCOPY Warning and Error Messages

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
None	<CONTROL Y>	Acknowledges receipt of a CONTROL-Y entered during a session.	None.
None	READ ERROR FROM COMMAND INPUTFILE	An error occurred while reading an FCOPY command from \$STDIN.	<i>In a job:</i> Re-submit the job. <i>In a session:</i> Re-enter the command.
None	WRITE ERROR TO COMMAND LISTFILE	An error occurred while writing an FCOPY message to \$STDLIST.	More than likely nothing serious has occurred and all FCOPY operations have been performed successfully. If you want to be sure, however, do the following: <i>In a job:</i> Re-submit the job. <i>In a session:</i> Re-enter the most recent FCOPY command.
3	SYNTAX ERROR: IN SUBSET OPTION	The subset function was not specified properly.	<i>In a job:</i> Correct the command and re-submit the job. <i>In a session:</i> Re-enter the command using the correct format.
4	SYNTAX ERROR: IN TITLE OPTION	The title option of the display function was not specified properly.	
5	SYNTAX ERROR: IN IGNERR OPTION	The ignore errors function was not specified properly.	
6	SYNTAX ERROR: IN VERIFY OPTION	The verify function was not specified properly.	
7	SYNTAX ERROR: IN SKIPEOF OPTION	The skip end-of-file function was not specified properly.	
8	SYNTAX ERROR: IN COMPARE OPTION	The compare function was not specified properly.	
9	SYNTAX ERROR: IN NEW OPTION	The new file function was not specified properly.	
10	SYNTAX ERROR: IN HEX OPTION	The display hexadecimal function was not specified properly.	
11	SYNTAX ERROR: IN EBCDICOUT OPTION	The EBCDICOUT character translate function was not specified properly.	
12	SYNTAX ERROR: IN CHAR OPTION	The display character function was not specified properly.	

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
13	SYNTAX ERROR: IN OCTAL OPTION	The display octal function was not specified properly.	<i>In a job:</i> Correct the command and re-submit the job. <i>In a session:</i> Re-enter the command using the correct format.
14	SYNTAX ERROR: IN UP-SHIFT OPTION	The upshift function was not specified properly.	
15	SYNTAX ERROR: IN BCDICIN OPTION	The BCDICIN character translate was not specified properly.	
16	SYNTAX ERROR: IN NORECNUM OPTION	The NORECNUM option of the display function was not specified properly.	
17	SYNTAX ERROR: IN EBCDICIN OPTION	The EBCDICIN character translate function was not specified properly.	
18	SYNTAX ERROR: IN BCDICOUT OPTION	The BCDICOUT character translate function was not specified properly.	None. FCOPY terminates.
19	SYNTAX ERROR: INVALID FORM OF EXIT COMMAND	The EXIT command was not specified properly.	
51	SYNTAX ERROR: IN QUOTED STRING	The <i>characterstring</i> specified for the subset function is not valid.	<i>In a job:</i> Correct the command and re-submit the job. <i>In a session:</i> Re-enter the command using the correct format.
52	SYNTAX ERROR: IN BIT PATTERN	The <i>patternlist</i> specified for the subset function is not valid.	
53	SYNTAX ERROR: INVALID INTEGER	An integer specified is outside the range allowed for the particular FCOPY function.	
54	SYNTAX ERROR: UNKNOWN OPTION NAME	One of the specified functions was unrecognizable.	
55	SYNTAX ERROR: IN FROM-FILE SPECIFIER	The "from" file was not specified properly.	
56	SYNTAX ERROR: IN TOFILE SPECIFIER	The "to" file was not specified properly.	
57	SYNTAX ERROR: ILLEGAL COMBINATION OF OPTIONS	Two or more <i>functionlist</i> entries conflict with one another.	
58	SYNTAX ERROR: FROM-FILE AND TOFILE NOT BOTH SPECIFIED	FROM= and TO= were not both specified in the FCOPY command.	
59	SYNTAX ERROR: ILLEGAL USE OF NEW	The context used to specify a new "to" file is not valid.	

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
60	SYNTAX ERROR: ILLEGAL USE OF *	The context used to specify * as a "from" file or "to" file is not valid.	<i>In a job:</i> Correct the command and re-submit the job. <i>In a session:</i> Re-enter the command using the correct format.
62	SYNTAX ERROR: FILE NAME TOO LONG	The "from" or "to" file name specified is longer than the 35 characters allowed in a fully-qualified file name with lockword.	
102	CAN'T CLOSE FROMFILE	MPE can't close the "from" file. This message is followed by an MPE file information display containing (among other things) an error number.	Look up the error number in the File System Error Table in the <i>MPE Intrinsic Reference Manual</i> (HP part number 30000-90010) and act accordingly.
103	CAN'T CLOSE TOFILE	MPE can't close the "to" file. This message is followed by an MPE file information display containing (among other things) an error number.	
104	CAN'T SAVE NEW TOFILE	MPE can't close the "to" file as a permanent file. Either you do not have SF capability or there is not enough group, account, or system file space.	If you don't have SF capability, you can't perform the operation. If there is not enough file space, purge some unneeded files to free some file space.
105	CAN'T OPEN FROM FILE	MPE can't open the "from" file. This message is followed by an MPE file information display containing (among other things) an error number.	Look up the error number in the File System Error Table in the <i>MPE Intrinsic Reference Manual</i> (HP part number 30000-90010) and act accordingly.
106	CAN'T OPEN TOFILE	MPE can't open the "to" file. This message is followed by an MPE file information display containing (among other things) an error number.	
107	VERIFY OR COMPARE OPTION: CAN'T GET READ ACCESS TO TOFILE	MPE can't get read access to the "to" file for a verify or compare operation. The read access specified in the file label has been overridden by an MPE :FILE command containing ACC=APPEND, ACC=OUT, or ACC=OUTKEEP.	Reset the particular :FILE command (using the MPE :RESET command) and re-try the operation.

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
108	ERROR IN CALLING FGETINFO FOR FROMFILE	An error prevented MPE from obtaining information from the "from" file's label. This message is followed by an MPE file information display containing (among other things) an error number.	Look up the error number in the File System Error Table in the <i>MPE Intrinsic Reference Manual</i> (HP part number 30000-90010) and act accordingly.
109	ERROR IN CALLING FGETINFOR FOR TOFILE	An error prevented MPE from obtaining information from the "to" file's label. This message is followed by an MPE file information display containing (among other things) an error number.	
110	IGNERR OPTION: FROMFILE NOT TAPE	The "from" file's device is not a magnetic tape unit.	The ignore errors function cannot be used in this case.
111	CAN'T GET READ ACCESS TO FROMFILE	MPE can't get read access to the "from" file. The read access specified in the file label has been overridden by an MPE :FILE command containing ACC=APPEND, ACC=OUT, or ACC=OUTKEEP.	Reset the particular :FILE command (using the MPE :RESET command) and re-try the operation.
112	CAN'T GET WRITE ACCESS TO TOFILE	MPE can't get write access to the "to" file. The write access specified in the file label has been overridden by an MPE :FILE command containing ACC=IN.	
113	SKIPEOF OPTION: FROMFILE NOT TAPE	The skip end-of-file function was specified for the "from" file and the "from" file device is not a magnetic tape unit.	If the intended "from" or "to" file is on magnetic tape, check the associated MPE :FILE command and the back reference to it.
114	SKIPEOF OPTION: TOFILE NOT TAPE	The skip end-of-file function was specified for the "to" file and the "to" file device is not a magnetic tape unit.	
115	SUBSET OPTION: STRING FALLS OUTSIDE OF FROMFILE RECSIZE	The <i>characterstring</i> or <i>patternlist</i> specified is greater than the record size of the "from" file. No such subset can exist in the specified "from" file.	Change the specified subset definition to a valid one and try the operation again.

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
116	CAN'T GET LARGE ENOUGH BUFFER	There is not enough data space for the buffers needed by the requested operation. FCOPY uses the DL-DB area for variable sized buffers.	Ask the system manager what size data area was specified when FCOPY was prepared and rerun FCOPY specifying a larger MAXDATA= parameter. Also make sure that the system configuration will accommodate your record size in the maximum allowed data segment size.
117	SKIPEOF OPTION: ERROR WHILE SKIPPING IN FROMFILE	An error occurred while skipping end-of-file marks in the "from" file.	Retry the operation.
118	SKIPEOF OPTION: ERROR WHILE SKIPPING IN TOFILE	An error occurred while skipping end-of-file marks in the "to" file.	
119	SUBSET OPTION: ERROR WHILE SPACING IN FROMFILE	An error occurred while spacing through the "from" file.	
120	SUBSET OPTION: SUBSET STARTS OVER EOF BOUNDARY	The subset specified extends over an end-of-file mark or a tape mark boundary.	Change the specified subset definition to a valid one and try the operation again.
123	SUBSET OPTION: THIS INPUT DEVICE DOES NOT BACKSPACE	The specified subset requires backspacing in the "from" file but the device for that file is not a disc or magnetic tape.	Check the MPE :FILE command associated with the "from" file and the back reference to it.
124	READ ERROR IN FROM-FILE AT RECORD <i>recnum</i>	An error occurred while spacing through the "from" file in search of the start of a subset.	Retry the operation.
125	SUBSET OPTION: NUMERIC SUBSET IS EMPTY	A subset specified by <i>starting-record-number</i> , <i>number-of-records</i> , and/or <i>last-record-number</i> does not exist or contains no data.	Change the specified subset definition to a valid one and try the operation again.
126	VERIFY OPTION: ERROR WHILE REWINDING FROMFILE	An error occurred while spacing backward to the beginning of the "from" file at the start of a verify operation.	Retry the operation.

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
127	VERIFY OPTION: ERROR WHILE REWINDING TOFILE	An error occurred while spacing backward to the beginning of the "to" file at the start of a verify operation.	Retry the operation.
128	EOF FOUND WHILE SPACING IN FROMFILE	An end-of-file mark was encountered while spacing through the "from" file in search of the start of a subset. This most often occurs when the "from" file is a blocked magnetic tape. For a blocked magnetic tape, the record numbers supplied in the SUBSET= parameter are used as block numbers.	Retry the operation specifying block numbers instead of record numbers. OR Reblock the tape so each block contains one record and then retry the operation.
129	EOF FOUND WHILE SPACING IN TOFILE	An end-of-file mark was encountered while spacing through the "to" file in search of the start of a subset during a compare or verify operation.	<i>Compare operation:</i> The "from" and "to" files are not identical. Display the "to" file to determine what it actually contains. <i>Verify operation:</i> The copy operation was not performed correctly. Retry the operation.
131	ERROR WHILE WRITING EOF TO TOFILE	An error occurred while writing an end-of-file mark in the "to" file.	Retry the operation.
132	VERIFY OPTION: ERROR WHILE SPACING IN THE FROMFILE	An error occurred while spacing through the "from" file during a verify operation.	
133	VERIFY OPTION: ERROR WHILE SPACING IN THE TOFILE	An error occurred while spacing through the "to" file during a verify operation.	
134	WARNING: FOUND EOF IN TOFILE	FCOPY has performed the specified operation but has filled the "to" file before completing the operation.	The "to" file was not large enough. Use the MPE :LISTF filename,2 command to determine the "to" file's size and then increase its size (using the MPE :PURGE and :BUILD commands) and retry the operation.
135	WRITE ERROR TO TOFILE	An error occurred while writing to the "to" file	Retry the operation.

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
136	READ ERROR FROM TOFILE	An error occurred while reading from the "to" file during a compare or verify operation.	<i>Compare operation:</i> Retry the operation. If the error persists, you must try to recreate the "to" file. <i>Verify operation:</i> Retry the operation.
137	WARNING: AN UNLABELLED TAPE OPERATION ENDS ON AN ERROR	An operation involving a magnetic tape "from" file was terminated by reading beyond the end of valid data rather than by sensing an end-of-file mark.	This is not an error. You can avoid this message by reading the "from" tape one file at a time and using the keyword SUBSET.
138	TITLE OPTION: TITLE TOO LONG	The title specified for the list function is longer than the 70 characters allowed or it extended over more than one line (record).	<i>In a job:</i> Correct the command and re-submit the job. <i>In a session:</i> Re-enter the command using the correct format.
139	DUMP OPTION: TOFILE RECSIZE NOT WITHIN LEGAL LIMIT	A file display was directed to an intermediate storage device with an incorrect record size. That record size must be ≥ 60 bytes (30 words).	Change the record size of the intermediate storage file (using the MPE :PURGE and :BUILD commands) so that it is within the allowed range and then retry the operation.
140	COMPARE OR VERIFY OPTION: OPERATION FAILS; DIFFERENT FIXED RECSIZES	The compare or verify operation was not attempted because the record sizes of the "to" and "from" files are not identical.	<i>Compare operation:</i> None. The compare operation revealed that the fixed record sizes of the two files are not identical. <i>Verify operation:</i> Change the record size of the "to" file (using the MPE :PURGE and :BUILD commands) so that it is the same as that of the "from" file and then retry the operation.
141	COMPARE BEGINS	The comparison phase of a verify operation has begun.	None.
143	WARNING: FROMFILE IS EMPTY	The "from" file contained no data. Nothing was copied or compared.	None. You may have accidentally specified the wrong file as the "from" file.

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
144	NEW OPTION: FILE ALREADY EXISTS	The "to" file named for the new file function already exists in the specified (or implied group and account.	Change the name of the "to" file and try the operation again.
145	BACKSPACE ERROR IN FROMFILE	An error occurred while spacing backward to the beginning of the "from" file or a subset within it.	Retry the operation.
146	CAN'T OPEN TEMPORARY LABEL FILE	Possibly a lack of space in your account.	Allocate more space in the account.
147	WRITE ERROR IN TEMPORARY LABEL FILE	An error occurred while writing to the temporary label file.	Retry the operation.
148	READ ERROR IN TEMPORARY LABEL FILE	An error occurred while reading from the "from" file.	Retry the operation.
149	OVERFLOW IN TEMPORARY LABEL FILE	There are more labels in the "from" file than can be copied.	Use the NOUSERLABELS function to copy files without labels.
150	READ ERROR IN FROM FILE LABELS	An error occurred while reading labels in the "from" file.	Retry the operation.
151	WRITE ERROR IN TO FILE LABELS	An error occurred while writing labels in the "to" file.	Retry the operation.
152	OVERFLOW IN TO FILE LABELS	There are too many labels in the "from" file to be copied to the "to" file.	If the "to" file is an old file, retry the operation. If it is a new file, the labels in the overflow are lost.
153	ATTEMPT TO BACKSPACE OVER FROMFILE BOT	The negative number specified in a SKIPEOF= function for the "from" file is too large.	Change the number to a smaller one to avoid skipping too many files back.
154	ATTEMPT TO BACKSPACE OVER TOFILE BOT	The negative number specified in a SKIPEOF= function for the "to" file is too large.	Change the number to a smaller one to avoid skipping too many files back.
200	WARNING: FROMFILE RECSIZE IS <i>number type</i> , TOFILE RECSIZE IS <i>number type</i>	The record sizes of the "from" and "to" files are not identical.	<i>In a job:</i> FCOPY performs the specified operation despite the conflict. <i>In a session:</i> You are given the choice whether or not to continue the operation. Note that if the "from" record size is larger than the "to" record size, the "from" records would be truncated. If

Table 4-1. FCOPY Warning and Error Messages (Continued)

ERROR NO.	MESSAGE	CAUSE	CORRECTIVE ACTION
200 (Continued)			the "to" record size is larger than the "from" record size, the content of the excess byte positions in the "to" records is unpredictable.
201	WARNING: FROMFILE IS ASCII, TOFILE IS BINARY or WARNING: FROMFILE IS BINARY, TOFILE IS ASCII	The data formats of the "from" and "to" files are not identical.	<i>In a job:</i> FCOPY performs the operation despite the conflict. <i>In a session:</i> You are given the choice whether or not to continue the operation.
301	READ ERROR IN FROMFILE AT RECORD <i>recnum</i>	An error occurred while reading from the "from" file at the record number displayed (<i>recnum</i>).	Retry the operation. If the error persists, use the subset function to copy all of the file except the erroneous record.
302	VERIFY OPTION: RAN OUT OF VERIFY ERRORS AT FROMFILE RECORD <i>recnum</i>	The verify function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (<i>recnum</i>).	Retry the operation specifying a larger <i>number-of-errors</i> parameter.
304	COMPARE OPTION: RAN OUT OF COMPARE ERRORS AT FROMFILE RECORD <i>recnum</i>	The compare function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (<i>recnum</i>).	Retry the operation specifying a larger <i>number-of-errors</i> parameter.



USING FCOPY WITH TERMINAL PERIPHERALS

APPENDIX

A

Several types of Hewlett-Packard terminals have peripheral devices of their own. Besides keyboards and CRT displays, such terminals have built-in cartridge tape units, and optionally may have line printers dedicated to them. The terminals that currently have these capabilities are:

HP2641A
HP2644A
HP 2644B
HP 2645A
HP 2645K

When logged on to one of these terminals, FCOPY can be used to copy files between the terminal peripherals and the other system devices. This is done by using file designators that identify each terminal peripheral in the "from" or "to" parameters of the FCOPY command. These file names are listed below.

\$CTUL — Left cartridge tape unit
\$CTUR — Right cartridge tape unit
\$HARD — Terminal Line Printer

All FCOPY operations can be done with terminal peripheral devices. However, the notation "*" cannot be used to specify a prior use of \$CTUL, \$CTUR, or \$HARD.

Although FCOPY permits both the "from" and "to" files to be terminal peripherals, this sort of application can be done more rapidly using the local capabilities of the terminal.

NOTE

FCOPY does not check whether or not the user is actually using one of the peripheral bearing terminals listed above. Consequently, the effects of using FCOPY with another type of terminal are unpredictable.

When using FCOPY to or from a terminal peripheral, FCOPY either makes the following settings or prompts you to make them.

- For HP 2644A terminals, set `termtype` equal to 10
For HP 2644B terminals, set `termtype` equal to 10
For HP 2645A terminals, set `termtype` equal to 10
For HP 2645K terminals, set `termtype` equal to 12
(If `termtype` is not 10 or 12, FCOPY prompts you to log on again with the `;TERM=` parameter correctly set.)
- Set duplex toggle switch to HALF duplex.
(FCOPY prompts you to make this setting and then press RETURN)
- Set message reception off (`SETMSG OFF`).
(FCOPY makes this setting automatically if you have not done so.)

- Set parity toggle switch to NONE.
(FCOPY prompts you to make this setting and then press RETURN)
- Set echo off (ESC ;).
(FCOPY makes this setting automatically if you have not done so.)
- Set automatic line feed off.

When issuing an FCOPY command with \$CTUL, or \$CTUR, or \$HARD in the "from" or "to" parameters and with the termtype incorrectly set, the following dialogue would be typical. The user entered the first line and received the remaining messages from FCOPY by pressing RETURN.

```
>FROM=$CTUR; TO=FILES;NEW
SET AND LEAVE DUPLEX TOGGLE TO HALF - THEN RETURN:

SETMSG OFF ISSUED.
SET AND LEAVE PARITY TOGGLE TO NONE, THEN RETURN.

UNABLE TO DO HP2644A,B,45A/45K I/O WHEN :HELLO ...;TERM<>10/12
*105*
CAN'T OPEN FROMFILE
```

The user should now issue a :HELLO command with the termtype parameter (;TERM=) set to 10 or 12.

A-1. EXAMPLES

```
>FROM=$CTUR; TO=DISC1;SUBV1
```

In the above example, the first file on the right cartridge tape unit is copied to the disc file named DISC1.

```
>FROM=DISC4; TO=MAGTP;OCTAL;CHAR
```

In the above example, the disc file named DISC4 is displayed on the HP 2644A terminal line printer in both octal and character formats.

```
>FROM=$CTUR; TO=DISC3;SUBV1
```

In the above example, the third file from the magnetic tape unit defined by the formal file designator MAGTP is copied to the left cartridge tape unit.

DEFAULT FILE DEFINITION VALUES

APPENDIX

B

This appendix summarizes the default file definition values that are used when FCOPY opens a specified "from" or "to" file. The shaded items are the ones which are most likely to cause problems when copying from one type of device to another. When opening an existing disc file, the information in the file's label always prevails (neither you nor FCOPY can override it). In all other cases, including the new file function, you can override the specified values by supplying the desired values in a :FILE command prior to issuing the particular FCOPY command.

FROM=card reader

Formal File Designator: the “from” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: undefined

Access Type: read only

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? no (shared access)

Inhibit Buffering? no

Record Size: 80 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

FROM=paper tape reader.

Formal File Designator: the “from” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: undefined

Access Type: read only

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? no (shared access)

Inhibit Buffering? no

Record Size: 80 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

FROM=magnetic tape

Formal File Designator: the “from” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): binary

Record Format: undefined

Access Type: read/write

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: 256 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

FROM=named disc file

Formal File Designator: the “from” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: old permanent or temporary

Record Type (ASCII/binary): (as specified when the file was created)

Record Format: (as specified when the file was created)

Access Type: read/write

Carriage Control? (as specified when the file was created)

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: (as specified when the file was created)

Device: (as specified when the file was created)

Userlabels: (as specified when the file was created)

Blocking Factor: (as specified when the file was created)

Number of Buffers: 2

File Size: (as specified when the file was created)

Initial Allocation: (as specified when the file was created)

File Code: (as specified when the file was created)

Allow :FILE Command? yes

FROM=\$CTUL or \$CTUR

Formal File Designator: (empty)

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII (JIS for the HP 2645K terminal)

Record Format: fixed

Access Type: input/output

Carriage Control? yes

Multirecord Access? yes

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? yes

Record Size: 256 bytes

Device: logical device number of log-on terminal (left-justified ASCII number)

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: 1023 records

Initial Allocation: 1

File Code: 0

Allow :FILE Command? yes

TO=card punch

Formal File Designator: the “to” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: undefined

Access Type: write only

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: 80 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes



TO=paper tape punch

Formal File Designator: the “to” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: undefined

Access Type: write only

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: 80 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

TO=magnetic tape

Formal File Designator: the “to” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): binary

Record Format: undefined

Access Type: read/write

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: 256 bytes (same as physical record size defined during system configuration)

Device: a byte array containing logical device number or device class name from :FILE command

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

TO=named disc file

Formal File Designator: the “to” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: (see note 1)

Record Type (ASCII/binary): (as specified when the file was created)

Record Format: (as specified when the file was created)

Access Type: (see note 2)

Carriage Control? (as specified when the file was created)

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? (see note 3)

Inhibit Buffering? no

Record Size: (as specified when the file was created)

Device: (as specified when the file was created)

Userlabels: (as specified when the file was created)

Blocking Factor: (as specified when the file was created)

Number of Buffers: 2

File Size: (as specified when the file was created)

Initial Allocation: (as specified when the file was created)

File Code: (as specified when the file was created)

Allow :FILE Command? yes

Note 1. *For CHAR, OCTAL, and HEX functions:* old permanent or temporary
For all other functions: old permanent

Note 2. *For COMPARE function:* read only
For VERIFY function: input/output
For all other functions: write only

Note 3. *For COMPARE function:* no (shared access)
For all other functions: yes

TO=named disc file;NEW

Formal File Designator: the "to" file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): same as "from" file

Record Format: same as "from" file

Access Type: read/write

Carriage Control? same as "from" file

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: same as "from" file

Device: a byte array containing DISC

Userlabels: same as "from" file, or 0 if NOUSERLABELS is specified.

Blocking Factor: same as "from" file

Number of Buffers: 2

File Size: (see note 1)

Initial Allocation: 1

File Code: same as "from" file

Allow :FILE Command? yes

Note 1. If "from" file is *not* a disc file and SUBSET is *not* used, the default file size is 1023 records.

If "from" file is a disc file and SUBSET is *not* used, the "to" file contains the same number of records, extents, and allocated extents as the "from" file.

TO=line printer

Formal File Designator: the “to” file name in the FCOPY command

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: undefined

Access Type: write only

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

Record Size: 132 bytes (same as physical record size defined during system configuration)

Device: A byte array containing logical device number or device class name from :FILE command

Userlabels: (not applicable)

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

TO=\$CTUL or \$CTUR

Formal File Designator: (empty)

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII (JIS for the HP 2645K terminal)

Record Format: fixed

Access Type: input/output

Carriage Control? yes

Multirecord Access? yes

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? yes

Record Size: 266 bytes (256 bytes of data plus a 10-byte escape sequence)

Device: logical device number of log-on terminal (left-justified ASCII number)

Userlabels: 0

Blocking Factor: default (divide 266 into configured physical record size)

Number of Buffers: 2

File Size: 1023 records

Initial Allocation: 1

File Code: 0

Allow :FILE Command? no

TO=\$HARD

Formal File Designator: (empty)

Default File Designator: actual file designator is same as formal file designator

Domain: new

Record Type (ASCII/binary): ASCII

Record Format: fixed

Access Type: input/output

Carriage Control? yes

Multirecord Access? yes

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? yes

Record Size: 86 bytes (80 bytes of data plus a 6-byte escape sequence)

Device: logical device number of log-on terminal (left-justified ASCII number)

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? no

CONVERSION TABLES

APPENDIX

C

This appendix contains three conversion code tables. The first is a conversion table for ASCII to EBCDIC or EBCDIC to ASCII. The second is for JIS to EBCDIC or EBCDIC to JIS, and the third table is for ASCII to BCDIC or BCDIC to ASCII.



HOW TO USE THIS TABLE

- The table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalent.
- Each row of the table gives the ASCII and EBCDIC meaning of the character code, the ASCII↔EBCDIC conversion code, and the Hollerith representation (punched card code) for the ASCII character.

The following examples describe several ways of using the table:

Example 1: Suppose you want to determine the ASCII code for the \$ character. Scan down the ASCII graphic column until you locate \$, then look left on that row to find the character code – 36 (dec), 044 (oct), and 24 (hex). This is the code used by an ASCII device (terminal, printer, computer, etc.) to represent the \$ character. Its Hollerith punched card code is 11-3-8.

Example 2: The character code 5B (hex) is the EBCDIC code for what character? Also, when 5B is converted to ASCII (for example, by FCOPY with the EBCDICIN option), what is the octal character code? First, locate 5B in the hex character code column and move right on that row to the EBCDIC graphic which is \$. The next column to the right gives the conversion to ASCII, 044. As a check, find 044 (oct) in the character code column, look right to the ASCII graphic column and note that \$ converted to EBCDIC is 133 (oct) which equals 5B (hex).

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
0	000	00	NUL	000	12-0-1-8-9	NUL	000
1	001	01	SOH	001	12-1-9	SOH	001
2	002	02	STX	002	12-2-9	STX	002
3	003	03	ETX	003	12-3-9	ETX	003
4	004	04	EOT	067	7-9	PF	234
5	005	05	ENQ	055	0-5-8-9	HT	011
6	006	06	ACK	056	0-6-8-9	LC	206
7	007	07	BEL	057	0-7-8-9	DEL	177
8	010	08	BS	026	11-6-9		227
9	011	09	HT	005	12-5-9		215
10	012	0A	LF	045	0-5-9	SMM	216
11	013	0B	VT	013	12-3-8-9	VT	013
12	014	0C	FF	014	12-4-8-9	FF	014
13	015	0D	CR	015	12-5-8-9	CR	015
14	016	0E	SO	016	12-6-8-9	SO	016
15	017	0F	SI	017	12-7-8-9	SI	017
16	020	10	DLE	020	12-11-1-8-9	DLE	020
17	021	11	DC1	021	11-1-9	DC1	021
18	022	12	DC2	022	11-2-9	DC2	022
19	023	13	DC3	023	11-3-9	TM	023
20	024	14	DC4	074	4-8-9	RES	235
21	025	15	NAK	075	5-8-9	NL	205
22	026	16	SYN	062	2-9	BS	010
23	027	17	ETB	046	0-6-9	IL	207
24	030	18	CAN	030	11-8-9	CAN	030
25	031	19	EM	031	11-1-6-9	EM	031
26	032	1A	SUB	077	7-8-9	CC	222
27	033	1B	ESC	047	0-7-9	CU1	217
28	034	1C	FS	034	11-4-8-9	IFS	034
29	035	1D	GS	035	11-5-8-9	IGS	035
30	036	1E	RS	036	11-6-8-9	IRS	036
31	037	1F	US	037	11-7-8-9	IUS	037
32	040	20	SP	100	Blank	DS	200
33	041	21	!	117	12-7-8	SOS	201
34	042	22	"	177	7-8	FS	202
35	043	23	#	173	3-8		203
36	044	24	\$	133	11-3-8	BYP	204
37	045	25	%	154	0-4-8	LF	012
38	046	26	&	120	12	ETB	027
39	047	27	'	175	5-8	ESC	033
40	050	28	(	115	12-5-8		210
41	051	29	)	135	11-5-8		211
42	052	2A	*	134	11-4-8	SM	212
43	053	2B	+	116	12-6-8	CU2	213
44	054	2C	,	153	0-3-8		214
45	055	2D	-	140	11	ENQ	005
46	056	2E	.	113	12-3-8	ACK	006
47	057	2F	/	141	0-1	BEL	007

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
48	060	30	0	360	0		220
49	061	31	1	361	1		221
50	062	32	2	362	2	SYN	026
51	063	33	3	363	3		223
52	064	34	4	364	4	PN	224
53	065	35	5	365	5	RS	225
54	066	36	6	366	6	UC	226
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A	:	172	2-8		232
59	073	3B	;	136	11-6-8	CU3	233
60	074	3C	<	114	12-4-8	DC4	024
61	075	3D	=	176	6-8	NAK	025
62	076	3E	>	156	0-6-8		236
63	077	3F	?	157	0-7-8	SUB	032
64	100	40	@	174	4-8	SP	040
65	101	41	A	301	12-1		240
66	102	42	B	302	12-2		241
67	103	43	C	303	12-3		242
68	104	44	D	304	12-4		243
69	105	45	E	305	12-5		244
70	106	46	F	306	12-6		245
71	107	47	G	307	12-7		246
72	110	48	H	310	12-8		247
73	111	49	I	311	12-9		250
74	112	4A	J	321	11-1		133
75	113	4B	K	322	11-2		056
76	114	4C	L	323	11-3	<	074
77	115	4D	M	324	11-4	(	050
78	116	4E	N	325	11-5	+	053
79	117	4F	O	326	11-6		041
80	120	50	P	327	11-7	&	046
81	121	51	Q	330	11-8		251
82	122	52	R	331	11-9		252
83	123	53	S	342	0-2		253
84	124	54	T	343	0-3		254
85	125	55	U	344	0-4		255
86	126	56	V	345	0-5		256
87	127	57	W	346	0-6		257
88	130	58	X	347	0-7		260
89	131	59	Y	350	0-8		261
90	132	5A	Z	351	0-9	!	135
91	133	5B	[	112	12-2-8	\$	044
92	134	5C	\	340	0-2-8	*	052
93	135	5D	]	132	11-2-8	)	051
94	136	5E	^	137	11-7-8	;	073
95	137	5F	_	155	0-5-8	—	136

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
96	140	60	'	171	1-8	-	055
97	141	61	a	201	12-0-1	/	057
98	142	62	b	202	12-0-2		262
99	143	63	c	203	12-0-3		263
100	144	64	d	204	12-0-4		264
101	145	65	e	205	12-0-5		265
102	146	66	f	206	12-0-6		266
103	147	67	g	207	12-0-7		267
104	150	68	h	210	12-0-8		270
105	151	69	i	211	12-0-9		271
106	152	6A	j	221	12-11-1	--	174
107	153	6B	k	222	12-11-2	,	054
108	154	6C	l	223	12-11-3	%	045
109	155	6D	m	224	12-11-4		137
110	156	6E	n	225	12-11-5	>	076
111	157	6F	o	226	12-11-6	~	077
112	160	70	p	227	12-11-7		272
113	161	71	q	230	12-11-8		273
114	162	72	r	231	12-11-9		274
115	163	73	s	242	11-0-2		275
116	164	74	t	243	11-0-3		276
117	165	75	u	244	11-0-4		277
118	166	76	v	245	11-0-5		300
119	167	77	w	246	11-0-6		301
120	170	78	x	247	11-0-7		302
121	171	79	y	250	11-0-8		140
122	172	7A	z	251	11-0-9		072
123	173	7B	{	300	12-0	#	043
124	174	7C	}	152	12-11	@	100
125	175	7D	~	320	11-0		047
126	176	7E		241	11-0-1	=	075
127	177	7F	DEL	007	12-7-9		042
128	200	80		040	11-0-1-8-9		303
129	201	81		041	0-1-9	a	141
130	202	82		042	0-2-9	b	142
131	203	83		043	0-3-9	c	143
132	204	84		044	0-4-9	d	144
133	205	85		025	11-5-9	e	145
134	206	86		006	12-6-9	f	146
135	207	87		027	11-7-9	g	147
136	210	88		050	0-8-9	h	150
137	211	89		051	0-1-8-9	i	151
138	212	8A		052	0-2-8-9		304
139	213	8B		053	0-3-8-9		305
140	214	8C		054	0-4-8-9		306
141	215	8D		011	12-1-8-9		307
142	216	8E		012	12-2-8-9		310
143	217	8F		033	11-3-8-9		311
144	220	90		060	12-11-0-1-8-9		312
145	221	91		061	1-9	j	152
146	222	92		032	11-2-8-9	k	153
147	223	93		063	3-9	l	154
148	224	94		064	4-9	m	155
149	225	95		065	5-9	n	156
150	226	96		066	6-9	o	157
151	227	97		010	12-8-9	p	160
152	230	98		070	8-9	q	161
153	231	99		071	1-8-9	r	162
154	232	9A		072	2-8-9		313
155	233	9B		073	3-8-9		314
156	234	9C		004	12-4-9		315
157	235	9D		024	11-4-9		316
158	236	9E		076	6-8-9		317
159	237	9F		341	11-0-1-9		320
160	240	A0		101	12-0-1-9	~	321
161	241	A1		102	12-0-2-9	s	176
162	242	A2		103	12-0-3-9	t	163
163	243	A3		104	12-0-4-9		164
164	244	A4		105	12-0-5-9	u	165
165	245	A5		106	12-0-6-9	v	166
166	246	A6		107	12-0-7-9	w	167
167	247	A7		110	12-0-8-9	x	170
168	250	A8		111	12-1-8	y	171
169	251	A9		121	12-11-1-9	z	172
170	252	AA		122	12-11-2-9		322
171	253	AB		123	12-11-3-9		323
172	254	AC		124	12-11-4-9		324
173	255	AD		125	12-11-5-9		325
174	256	AE		126	12-11-6-9		326
175	257	AF		127	12-11-7-9		327

CHAR CODE			ASCII			EBCDIC	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIC (Oct)	Hollerith	Cntl/ Gph	to ASCII (Oct)
176	260	B0		130	12-11-8-9		330
177	261	B1		131	11-1-8		331
178	262	B2		142	11-0-2-9		332
179	263	B3		143	11-0-3-9		333
180	264	B4		144	11-0-4-9		334
181	265	B5		145	11-0-5-9		335
182	266	B6		146	11-0-6-9		336
183	267	B7		147	11-0-7-9		337
184	270	B8		150	11-0-8-9		340
185	271	B9		151	0-1-8		341
186	272	BA		160	12-11-0		342
187	273	BB		161	12-11-0-1-9		343
188	274	BC		162	12-11-0-2-9		344
189	275	BD		163	12-11-0-3-9		345
190	276	BE		164	12-11-0-4-9		346
191	277	BF		165	12-11-0-5-9		347
192	300	C0		166	12-11-0-6-9	{	173
193	301	C1		167	12-11-0-7-9	A	101
194	302	C2		170	12-11-0-8-9	B	102
195	303	C3		200	12-0-1-8	C	103
196	304	C4		212	12-0-2-8	D	104
197	305	C5		213	12-0-3-8	E	105
198	306	C6		214	12-0-4-8	F	106
199	307	C7		215	12-0-5-8	G	107
200	310	C8		216	12-0-6-8	H	110
201	311	C9		217	12-0-7-8	I	111
202	312	CA		220	12-11-1-8		350
203	313	CB		232	12-11-2-8		351
204	314	CC		233	12-11-3-8	J	352
205	315	CD		234	12-11-4-8		353
206	316	CE		235	12-11-5-8	K	354
207	317	CF		236	12-11-6-8		355
208	320	D0		237	12-11-7-8	}	175
209	321	D1		240	11-0-1-8	J	112
210	322	D2		252	11-0-2-8	K	113
211	323	D3		253	11-0-3-8	L	114
212	324	D4		254	11-0-4-8	M	115
213	325	D5		255	11-0-5-8	N	116
214	326	D6		256	11-0-6-8	O	117
215	327	D7		257	11-0-7-8	P	120
216	330	D8		260	12-11-0-1-8	Q	121
217	331	D9		261	12-11-0-1	R	122
218	332	DA		262	12-11-0-2		356
219	333	DB		263	12-11-0-3		357
220	334	DC		264	12-11-0-4		360
221	335	DD		265	12-11-0-5		361
222	336	DE		266	12-11-0-6		362
223	337	DF		267	12-11-0-7		363
224	340	E0		270	12-11-0-8	\	134
225	341	E1		271	12-11-0-9		237
226	342	E2		272	12-11-0-2-8	S	123
227	343	E3		273	12-11-0-3-8	T	124
228	344	E4		274	12-11-0-4-8	U	125
229	345	E5		275	12-11-0-5-8	V	126
230	346	E6		276	12-11-0-6-8	W	127
231	347	E7		277	12-11-0-7-8	X	130
232	350	E8		312	12-0-2-8-9	Y	131
233	351	E9		313	12-0-3-8-9	Z	132
234	352	EA		314	12-0-4-8-9		364
235	353	EB		315	12-0-5-8-9		365
236	354	EC		316	12-0-6-8-9		366
237	355	ED		317	12-0-7-8-9		367
238	356	EE		332	12-11-2-8-9		370
239	357	EF		333	12-11-3-8-9		371
240	360	F0		334	12-11-4-8-9	0	060
241	361	F1		335	12-11-5-8-9	1	061
242	362	F2		336	12-11-6-8-9	2	062
243	363	F3		337	12-11-7-8-9	3	063
244	364	F4		352	11-0-2-8-9	4	064
245	365	F5		353	11-0-3-8-9	5	065
246	366	F6		354	11-0-4-8-9	6	066
247	367	F7		355	11-0-5-8-9	7	067
248	370	F8		356	11-0-6-8-9	8	070
249	371	F9		357	11-0-7-8-9	9	071
250	372	FA		372	12-11-0-2-8-9		372
251	373	FB		373	12-11-0-3-8-9		373
252	374	FC		374	12-11-0-4-8-9		374
253	375	FD		375	12-11-0-5-8-9		375
254	376	FE		376	12-11-0-6-8-9		376
255	377	FF		377	12-11-0-7-8-9	EO	377

HOW TO USE THIS TABLE

- The table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalent.
- Each row of the table gives the JIS and EBCDIK meaning of the character code, the JIS EBCDIK conversion code, and the Hollerith representation (punched card code) for the ASCII character.

The following examples describe several ways of using the table:

Example 1: Suppose you want to determine the JIS code for the S character. Scan down the JIS graphic column until you locate \$, then look left on that row to find the character code — 36 (dec), 044 (oct), and 24 (hex). This is the code used by an JIS device (terminal, printer, computer, etc.) to represent the S character. Its Hollerith punched card code is 11-3-8.

Example 2: The character code 5B (hex) is the EBCDIK code for what character? Also, when 5B is converted to JIS (for example, by FCOPY with the EBCDIKIN option), what is the octal character code? First, locate 5B in the hex character code column and move right on that row to the EBCDIK graphic which is \$. The next column to the right gives the conversion to JIS, 044. As a check, find 044 (oct) in the character code column, look right to the JIS graphic column and note that \$ converted to EBCDIK is 133 (oct) which equals 5B (hex).

CHAR CODE			JIS			EBCDIK	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIK (Oct)	Hollerith	Cntl/ Gph	to JIS (Oct)
0	000	00	NUL	000	12-0-1-8-9	NUL	000
1	001	01	SOH	001	12-1-9	SOH	001
2	002	02	STX	002	12-2-9	STX	002
3	003	03	ETX	003	12-3-9	ETX	003
4	004	04	EOT	067	7-9	PF	234
5	005	05	ENQ	055	0-5-8-9	HT	011
6	006	06	ACK	056	0-6-8-9	LC	206
7	007	07	BEL	057	0-7-8-9	DEL	177
8	010	08	BS	026	11-6-9		227
9	011	09	HT	005	12-5-9		215
10	012	0A	LF	045	0-5-9	SMM	216
11	013	0B	VT	013	12-3-8-9	VT	013
12	014	0C	FF	014	12-4-8-9	FF	014
13	015	0D	CR	015	12-5-8-9	CR	015
14	016	0E	SO	016	12-6-8-9	SO	016
15	017	0F	SI	017	12-7-8-9	SI	017
16	020	10	DLE	020	12-11-1-8-9	DLE	020
17	021	11	DC1	021	11-1-9	DC1	021
18	022	12	DC2	022	11-2-9	DC2	022
19	023	13	DC3	023	11-3-9	TM	023
20	024	14	DC4	074	4-8-9	RES	235
21	025	15	NAK	075	5-8-9	NL	205
22	026	16	SYN	062	2-9	BS	010
23	027	17	ETB	046	0-6-9	IL	207
24	030	18	CAN	030	11-8-9	CAN	030
25	031	19	EM	031	11-1-8-9	EM	031
26	032	1A	SUB	077	7-8-9	CC	222
27	033	1B	ESC	047	0-7-9	CU1	217
28	034	1C	FS	034	11-4-8-9	IFS	034
29	035	1D	GS	035	11-5-8-9	IGS	035
30	036	1E	RS	036	11-6-8-9	IRS	036
31	037	1F	US	037	11-7-8-9	IUS	037
32	040	20	SP	100	Blank	DS	200
33	041	21	!	117	12-7-8	SOS	201
34	042	22	"	177	7-8	FS	202
35	043	23	#	173	3-8		203
36	044	24	\$	133	11-3-8	8YP	204
37	045	25	%	154	0-4-8	LF	012
38	046	26	&	120	12	ETB	027
39	047	27	'	175	5-8	ESC	033
40	050	28	(	115	12-5-8		210
41	051	29	)	135	11-5-8		211
42	052	2A	*	134	11-4-8	SM	212
43	053	2B	+	116	12-6-8	CU2	213
44	054	2C	,	153	0-3-8		214
45	055	2D	-	140	11	ENQ	005
46	056	2E	.	113	12-3-8	ACK	006
47	057	2F	/	141	0-1	BEL	007

CHAR CODE			JIS			EBCDIK	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIK (Oct)	Hollerith	Cntl/ Gph	to JIS (Oct)
48	060	30	0	360	0		220
49	061	31	1	361	1		221
50	062	32	2	362	2	SYN	026
51	063	33	3	363	3		223
52	064	34	4	364	4	PN	224
53	065	35	5	365	5	RS	225
54	066	36	6	366	6	UC	226
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A	:	172	2-8		232
59	073	3B	;	136	11-6-8	CU3	233
60	074	3C	<	114	12-4-8	DC4	024
61	075	3D	=	176	6-8	NAK	025
62	076	3E	>	156	0-6-8		236
63	077	3F	?	157	0-7-8	SUB	032
64	100	40	@	174	4-8	SP	040
65	101	41	A	301	12-1		240
66	102	42	B	302	12-2	*	241
67	103	43	C	303	12-3	'	242
68	104	44	D	304	12-4	]	243
69	105	45	E	305	12-5	^	244
70	106	46	F	306	12-6	~	245
71	107	47	G	307	12-7	?	246
72	110	48	H	310	12-8	?	247
73	111	49	I	311	12-9	!	250
74	112	4A	J	321	11-1	@	133
75	113	4B	K	322	11-2	.	056
76	114	4C	L	323	11-3	<	074
77	115	4D	M	324	11-4	(	050
78	116	4E	N	325	11-5	+	053
79	117	4F	O	326	11-6		041
80	120	50	P	327	11-7	&	046
81	121	51	Q	330	11-8	?	251
82	122	52	R	331	11-9	!	252
83	123	53	S	342	0-2	'	253
84	124	54	T	343	0-3	+	254
85	125	55	U	344	0-4	^	255
86	126	56	V	345	0-5	~	256
87	127	57	W	346	0-6	?	257
88	130	58	X	347	0-7	-	260
89	131	59	Y	350	0-8	?	261
90	132	5A	Z	351	0-9	!	135
91	133	5B	[	112	12-2-8	\$	044
92	134	5C	¥	340	0-2-8	*	052
93	135	5D	¦	132	11-2-8	)	051
94	136	5E	^	137	11-7-8	;	073
95	137	5F	_	155	0-5-8	~	136

CHAR CODE			JIS			EBCDIK	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIK (Oct)	Hollerith	Cntl/ Gph	to JIS (Oct)
96	140	60		171	1-8	-	055
97	141	61	a	201	12-0-1	/	057
98	142	62	b	202	12-0-2	イ	262
99	143	63	c	203	12-0-3	ウ	263
100	144	64	d	204	12-0-4	エ	264
101	145	65	e	205	12-0-5	オ	265
102	146	66	f	206	12-0-6	カ	266
103	147	67	g	207	12-0-7	キ	267
104	150	68	h	210	12-0-8	ク	270
105	151	69	i	211	12-0-9	ケ	271
106	152	6A	j	221	12-11-1	コ	174
107	153	6B	k	222	12-11-2	サ	054
108	154	6C	l	223	12-11-3	%	045
109	155	6D	m	224	12-11-4	>	137
110	156	6E	n	225	12-11-5	?	076
111	157	6F	o	226	12-11-6	?	077
112	160	70	p	227	12-11-7	コ	272
113	161	71	q	230	12-11-8	シ	273
114	162	72	r	231	12-11-9	ス	274
115	163	73	s	242	11-0-2		275
116	164	74	t	243	11-0-3	セ	276
117	165	75	u	244	11-0-4	ソ	277
118	166	76	v	245	11-0-5	タ	300
119	167	77	w	246	11-0-6	チ	301
120	170	78	x	247	11-0-7	ツ	302
121	171	79	y	250	11-0-8	、	140
122	172	7A	z	251	11-0-9	:	072
123	173	7B	{	300	12-0	#	043
124	174	7C	{	152	12-11	@	100
125	175	7D	~	320	11-0	、	047
126	176	7E	~	241	11-0-1	=	075
127	177	7F	DEL	007	12-7-9	"	042
128	200	80		040	11-0-1-8-9	テ	303
129	201	81		041	0-1-9	a	141
130	202	82		042	0-2-9	b	142
131	203	83		043	0-3-9	c	143
132	204	84		044	0-4-9	d	144
133	205	85		025	11-5-9	e	145
134	206	86		006	12-6-9	f	146
135	207	87		027	11-7-9	g	147
136	210	88		050	0-8-9	h	150
137	211	89		051	0-1-8-9	i	151
138	212	8A		052	0-2-8-9	ト	304
139	213	8B		053	0-3-8-9	ナ	305
140	214	8C		054	0-4-8-9	ニ	306
141	215	8D		011	12-1-8-9	ヌ	307
142	216	8E		012	12-2-8-9	ネ	310
143	217	8F		033	11-3-8-9	ノ	311
144	220	90		060	12-11-0-1-8-9	ハ	312
145	221	91		061	1-9	j	152
146	222	92		032	11-2-8-9	k	153
147	223	93		063	3-9	l	154
148	224	94		064	4-9	m	155
149	225	95		065	5-9	n	156
150	226	96		066	6-9	o	157
151	227	97		010	12-8-9	p	160
152	230	98		070	8-9	q	161
153	231	99		071	1-8-9	r	162
154	232	9A		072	2-8-9	ヒ	313
155	233	9B		073	3-8-9	フ	314
156	234	9C		004	12-4-9	ハ	315
157	235	9D		024	11-4-9	ニ	316
158	236	9E		076	6-8-9	マ	317
159	237	9F		341	11-0-1-9	ミ	320
160	240	A0		101	12-0-1-9	ム	321
161	241	A1	.	102	12-0-2-9	~	176
162	242	A2	、	103	12-0-3-9	s	163
163	243	A3	、	104	12-0-4-9	t	164
164	244	A4	、	105	12-0-5-9	u	165
165	245	A5	、	106	12-0-6-9	v	166
166	246	A6	、	107	12-0-7-9	w	167
167	247	A7	、	110	12-0-8-9	x	170
168	250	A8	、	111	12-1-8	y	171
169	251	A9	、	121	12-11-1-9	z	172
170	252	AA	、	122	12-11-2-9	メ	322
171	253	AB	、	123	12-11-3-9	モ	323
172	254	AC	、	124	12-11-4-9	ヤ	324
173	255	AD	、	125	12-11-5-9	ユ	325
174	256	AE	、	126	12-11-6-9	ヨ	326
175	257	AF	、	127	12-11-7-9	ラ	327

CHAR CODE			JIS			EBCDIK	
Dec	Oct	Hex	Cntl/ Gph	to EBCDIK (Oct)	Hollerith	Cntl/ Gph	to JIS (Oct)
176	260	80	-	130	12-11-8-9	リ	330
177	261	81	ア	131	11-1-8	ル	331
178	262	82	イ	142	11-0-2-9	レ	332
179	263	83	ウ	143	11-0-3-9	ロ	333
180	264	84	エ	144	11-0-4-9	ワ	334
181	265	85	オ	145	11-0-5-9	ン	335
182	266	86	カ	146	11-0-6-9	フ	336
183	267	87	キ	147	11-0-7-9	ホ	337
184	270	88	ク	150	11-0-8-9		340
185	271	89	ケ	151	0-1-8		341
186	272	8A	コ	160	12-11-0		342
187	273	8B	サ	161	12-11-0-1-9		343
188	274	8C	シ	162	12-11-0-2-9		344
189	275	8D	ス	163	12-11-0-3-9		345
190	276	8E	セ	164	12-11-0-4-9		346
191	277	8F	ソ	165	12-11-0-5-9		347
192	300	C0	タ	166	12-11-0-6-9	{	173
193	301	C1	チ	167	12-11-0-7-9	A	101
194	302	C2	ツ	170	12-11-0-8-9	B	102
195	303	C3	テ	200	12-0-1-8	C	103
196	304	C4	ト	212	12-0-2-8	D	104
197	305	C5	ナ	213	12-0-3-8	E	105
198	306	C6	ニ	214	12-0-4-8	F	106
199	307	C7	ヌ	215	12-0-5-8	G	107
200	310	C8	ネ	216	12-0-6-8	H	110
201	311	C9	ノ	217	12-0-7-8	I	111
202	312	CA	ハ	220	12-11-1-8		350
203	313	CB	ヒ	232	12-11-2-8		351
204	314	CC	フ	233	12-11-3-8	』	352
205	315	CD	ヘ	234	12-11-4-8		353
206	316	CE	ホ	235	12-11-5-8	『	354
207	317	CF	マ	236	12-11-6-8		355
208	320	D0	ミ	237	12-11-7-8	}	175
209	321	D1	ム	240	12-11-0-1-8	J	112
210	322	D2	メ	252	11-0-2-8	K	113
211	323	D3	モ	253	11-0-3-8	L	114
212	324	D4	ヤ	254	11-0-4-8	M	115
213	325	D5	ユ	255	11-0-5-8	N	116
214	326	D6	ヨ	256	11-0-6-8	O	117
215	327	D7	ラ	257	11-0-7-8	P	120
216	330	D8	リ	260	12-11-0-1-8	Q	121
217	331	D9	ル	261	12-11-0-1	R	122
218	332	DA	レ	262	12-11-0-2		356
219	333	DB	ロ	263	12-11-0-3		357
220	334	DC	ワ	264	12-11-0-4		360
221	335	DD	ン	265	12-11-0-5		361
222	336	DE	フ	266	12-11-0-6		362
223	337	DF	ホ	267	12-11-0-7		363
224	340	E0		270	12-11-0-8	〒	134
225	341	E1		271	12-11-0-9		237
226	342	E2		272	12-11-0-2-8	S	123
227	343	E3		273	12-11-0-3-8	T	124
228	344	E4		274	12-11-0-4-8	U	125
229	345	E5		275	12-11-0-5-8	V	126
230	346	E6		276	12-11-0-6-8	W	127
231	347	E7		277	12-11-0-7-8	X	130
232	350	E8		312	12-0-2-8-9	Y	131
233	351	E9		313	12-0-3-8-9	Z	132
234	352	EA		314	12-0-4-8-9		364
235	353	EB		315	12-0-5-8-9		365
236	354	EC		316	12-0-6-8-9	ハ	366
237	355	ED		317	12-0-7-8-9		367
238	356	EE		332	12-11-2-8-9		370
239	357	EF		333	12-11-3-8-9		371
240	360	F0		334	12-11-4-8-9	0	060
241	361	F1		335	12-11-5-8-9	1	061
242	362	F2		336	12-11-6-8-9	2	062
243	363	F3		337	12-11-7-8-9	3	063
244	364	F4		352	11-0-2-8-9	4	064
245	365	F5		353	11-0-3-8-9	5	065
246	366	F6		354	11-0-4-8-9	6	066
247	367	F7		355	11-0-5-8-9	7	067
248	370	F8		356	11-0-6-8-9	8	070
249	371	F9		357	11-0-7-8-9	9	071
250	372	FA		372	12-11-0-2-8-9		372
251	373	FB		373	12-11-0-3-8-9		373
252	374	FC		374	12-11-0-4-8-9		374
253	375	FD		375	12-11-0-5-8-9		375
254	376	FE		376	12-11-0-6-8-9		376
255	377	FF		377	12-11-0-7-8-9	EO	377

ASCII-BCDIC CONVERSION TABLE

This table cross references all of the BCDIC character set with the ASCII character set. Only those ASCII characters which have a corresponding BCDIC code are listed.

Each row gives the decimal, octal, and hexadecimal character code, the corresponding graphic representation, and the Hollerith punch code for the ASCII character.

Note that in the Hollerith column, there are dual entries for some of the punches. In such cases, the entry on the right side is for ASCII punches; the entry on the left is for BCDIC punches.

In five cases, BCDIC character codes can be interpreted in either of two ways. In these cases both the BCDIC graphics are shown in the same row, and are referenced by ASCII decimals 39, 40, 41, 43 and 61.

ASCII					BCDIC				
Char Code			Graphic	Hollerith	Graphic	Char Code			
Dec	Oct	Hex				Dec	Oct	Hex	
32	040	20	(space)	blank	(space)	64	100	40	
34	042	22	"	7-8	✓	127	177	7F	
36	044	24	\$	11-3-8	\$	91	133	5B	
39	047	27	'	5-8/4-8	@	124	174	7C	
39	047	27	'	5-8	:	125	175	7D	
40	050	28	(	12-5-8	[	77	115	4D	
40	050	28	(	12-5-8/0-4-8	% (	108	154	6C	
41	051	29	)	11-5-8/12-4-1	□)	76	114	4C	
41	051	29	)	11-5-8	]	93	135	5D	
42	052	2A	*	11-4-8	*	92	134	5C	
43	053	2B	+	12-6-8	<	78	116	4E	
43	053	2B	+	12-6-8/12	& +	80	120	50	
44	054	2C	' (comma)	0-3-8	' (comma)	107	153	6B	
45	055	2D	-	11	-	96	140	60	
46	056	2E	.	12-3-8	.	75	113	4B	
47	057	2F	/	0-1	/	97	141	61	
48	060	30	0	0	0	240	360	F0	
49	061	31	1	1	1	241	361	F1	
50	062	32	2	2	2	242	362	F2	
51	063	33	3	3	3	243	363	F3	
52	064	34	4	4	4	244	364	F4	
53	065	35	5	5	5	245	365	F5	
54	066	36	6	6	6	246	366	F6	
55	067	37	7	7	7	247	367	F7	
56	070	38	8	8	8	248	370	F8	
57	070	39	9	9	9	249	371	F9	
58	071	3A	:	2-8	⌚	122	172	7A	
59	072	3B	;	11-6-8	;	94	136	5E	
61	075	3D	=	6-8/3-8	# =	123	173	7B	
61	075	3D	=	6-8	>	126	176	7E	
62	076	3E	>	0-6-8	\	110	156	6E	
63	077	3F	?	0-7-8	⦿	111	157	6F	
65	101	41	A	12-1	A	193	301	C1	
66	102	42	B	12-2	B	194	302	C2	
67	103	43	C	12-3	C	195	303	C3	
68	104	44	D	12-4	D	196	304	C4	
69	105	45	E	12-5	E	197	305	C5	
70	106	46	F	12-6	F	198	306	C6	
71	107	47	G	12-7	G	199	307	C7	
72	110	48	H	12-8	H	200	310	C8	
73	111	49	I	12-9	I	201	311	C9	
74	112	4A	J	11-1	J	209	321	D1	

ASCII-BCDIC CONVERSION TABLE (Continued)

ASCII						BCDIC					
Char Code			Graphic	Hollerith	Graphic	Char Code			Graphic		
Dec	Oct	Hex				Dec	Oct	Hex			
75	113	4B	K	11-2	K	210	322	D2			
76	114	4C	L	11-3	L	211	323	D3			
77	115	4D	M	11-4	M	212	324	D4			
78	116	4E	N	11-5	N	213	325	D5			
79	117	4F	O	11-6	O	214	326	D6			
80	120	50	P	11-7	P	215	327	D7			
81	121	51	Q	11-8	Q	216	330	D8			
82	122	52	R	11-9	R	217	331	D9			
83	123	53	S	0-2	S	226	342	E2			
84	124	54	T	0-3	T	227	343	E3			
85	125	55	U	0-4	U	228	344	E4			
86	126	56	V	0-5	V	229	345	E5			
87	127	57	W	0-6	W	230	346	E6			
88	130	58	X	0-7	X	231	347	E7			
89	131	59	Y	0-8	Y	232	350	E8			
90	132	5A	Z	0-9	Z	233	351	E9			
92	134	5C	\	0-2-8	‡	224	340	E0			
95	137	5F	~	0-5-8	√	109	155	6D			
123	173	7B	{	12-0	?	192	300	C0			
125	175	7D	}	11-0	!	208	320	D0			
136	210	88	^	0-8-9	>	95	137	5F			
250	372	FA		12-11-0-2-8-9	⦿	79	117	4F			

Attributes, File, 2-1, B-1

BCDICIN, 2-10

BCDICOUT, 2-10

Carriage Control Function, 2-6 through 2-9

CCTL, 2-6

CHAR, 2-17 through 2-22

Characters Function, Display, 1-1, 1-2, 2-1, 2-5, 2-17

Character Translate Function, 1-1, 1-2, 2-1, 2-10, 2-12

CLEAR, 2-17 through 2-22

Code Conversions, C-1 through C-7

Code Translate Function, 1-1, 1-2, 2-1, 2-5, 2-11

COMPARE, 2-13

Compare Function, 1-2, 2-1, 2-5, 2-13

Control-Y, 2-2, 2-28

Copy Function, 1-2, 2-1, 2-4, 2-5

\$CTUL, 2-5, A-1, B-6, B-13

Deblock Records Function, 2-14 through 2-16

DEBLOCK, 1-2, 2-14 through 2-16

Default File Definition Values, B-1

Display Functions, 1-1, 1-2, 2-1, 2-5, 2-17 through 2-22

EBCDICIN, 2-10

EBCDICOUT, 2-10

EBCDIKIN, 2-10

EBCDIKOUT, 2-10

Error Messages, FCOPY, 4-1

FCOPY Command Format, 1-2

File Attribute Requirements, 2-1

:FILE Commands, 1-2, 2-4, 2-25

File Definition Values, Default, B-1

File Information Display, MPE, 4-2

FILES, 1-2, 2-24 through 2-27

FROM, 2-4

fromfile, 2-4, B-1

\$HARD, 2-5, A-1, B-14

HEX, 2-17 through 2-20

IGNERR, 2-23

Ignore Errors Function, 1-2, 2-1, 2-5, 2-23

Interrupting FCOPY, 2-2

KANA, 2-17 through 2-20

Katakana Character Display, 2-17

KEY, 1-1, 1-2, 2-34

KSAM Files, 1-2, 2-5, 2-28, 2-29, 2-34

Messages, FCOPY, 4-1

Multiple Tape File Copying, 2-24 through 2-27

NEW, 2-28

New File Function, 1-2, 2-1, 2-5, 2-28

No Carriage Control Function, 2-6 through 2-9

NOCCTL, 1-2, 2-6 through 2-9

NOKSAM, 1-1, 1-2, 2-29

NORECNUM, 2-17 through 2-22

No User Labels Function, 1-2, 2-30

NOUSERLABELS, 1-2, 2-30

Numerical Code Function, Display, 2-20 through 2-22

OCTAL, 2-17, 2-18, 2-20 through 2-22

Octal Function, Display, 1-1, 1-2, 2-1, 2-5, 2-17, 2-18, 2-20 to 2-22

Peripherals, Terminal, A-1 A-1

Range, 2-35

Running FCOPY, 1-1

SKIPEOF, 2-31 through 2-33

Skip End-Of-File Function, 1-1, 1-2, 2-1, 2-5, 2-31 through 2-33

Specify Key, 2-34

SUBSET, 2-35

Subset Function, 1-1, 1-2, 2-1, 2-5, 2-35 through 2-37

Syntax, 2-3

Terminal Peripherals, A-1

TITLE, 2-17 through 2-20

TO, 2-4

Tofile, 2-4, B-1

UPSHIFT, 2-38

Upshift Function, 1-2, 2-1, 2-38

VERIFY, 2-39

Verify Function, 1-2, 2-1, 2-5, 2-39

Warning Messages, FCOPY, 4-1

