FCOPY Reference Manual

HP 3000 MPE/iX Computer Systems

Edition 3

Manufacturing Part Number: 32212-90008
E1290

U.S.A. December 1990

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 or fitness for a particular purpose. Hewlett-Packard shall not be liable for errors contained herein or for direct, indirect, special, incidental or consequential damages in connection with the furnishing 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.

Restricted Rights Legend

Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013. Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c) (1,2).

Acknowledgments

UNIX is a registered trademark of The Open Group.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Contents

Contents

Contents

Preface

This manual is designed to serve a wide variety of users, from the occasional MPE V/E or MPE XL user to the experienced system operator. Because FCOPY runs in almost exactly the same way on MPE V/E and MPE XL, this manual is used for both systems.

MPE V/E and XL systems both support a KSAM *tofile* that consists of a pair of key and data files. It is referred to on MPE V/E as KSAM V/E or KSAM/3000, and on MPE XL as CM KSAM. MPE XL also supports a single file format, called KSAM XL. This single file KSAM structure is available on MPE XL only. FCOPY access to both types of KSAM files is described in this manual.

Manual Organization

The information in the manual is presented as follows:

Chapter 1	**Introduction** describes how to use the manual.
Chapter 2	**Using FCOPY** gives an overview of the FCOPY command and lists the FCOPY functions.
Chapter 3	**FCOPY Applications and Examples** provides instructions for using FCOPY and gives examples of typical transactions.
Chapter 4	**FCOPY Syntax** summarizes the syntax of the FCOPY command and the FCOPY functions.
Chapter 5	**FCOPY Functions** describes each of the FCOPY functions in detail.
Chapter 6	**Using FCOPY with Terminal Peripherals** discusses terminal peripheral file designators, terminal settings, and copying files between peripherals.
Appendix A	**FCOPY Messages** contains all the FCOPY error, warning, and status messages.
Appendix B	**Default File Definition Values** summarizes the values that FCOPY uses to open *fromfiles* or *tofiles*.
Appendix C	**Code Conversion Tables** contains three tables showing ASCII/EBCDIC, JIS/BCDIK, and ASCII/BCDIC code conversions.

How to Use This Manual

The manual is organized into three major parts.

- Chapters 1 through 3 explain how to use FCOPY. Chapter 3 contains several examples, so you might look at chapter 3 when you have a specific task to perform, and follow an example before carrying out your task. Read these chapters first if you are a new FCOPY user.
- Chapters 4 through 6 contain the FCOPY syntax and descriptions of all the FCOPY functions. Use these chapters as a reference guide if you want to look up the syntax for an FCOPY function.
- The appendices contain FCOPY message descriptions, default file definitions and code conversion tables. Use these chapters for general reference and for information when troubleshooting or converting codes.

If you have any general comments about this edition, find any technical errors, or encounter any situations that you would like to see as examples in future editions, please fill in and mail the Reader Comment Card at the front of the manual.

Additional Information

You may also find the following manuals useful:

- *MPE V General User's Reference Manual* for MPE V/E.
- *MPE V/E Commands Reference Manual* or *MPE XL Commands Reference Manual*.
- *MPE Intrinsics Reference Manual* for MPE V/E or *MPE XL Intrinsics Reference Manual* for MPE XL.
- *MPE File System Reference Manual* for MPE V/E or *Accessing Files Programmer's Guide*) for MPE XL.
- *KSAM/3000 Reference Manual* for KSAM/3000 files on MPE V/E and for CM KSAM files on MPE XL.
- *Using KSAM XL* for KSAM XL files on MPE XL.
- *Native Language Support Reference Manual* for MPE V/E or *Native Language Programmer's Guide* for MPE XL.
- *MPE XL Error Message Manual Volumes 1 and 2* for MPE XL.

1 Introduction

This manual describes the HP 3000 file copy utility, FCOPY/3000. FCOPY is a utility program within the HP 3000 multiprogramming executive operating systems (MPE V/E and MPE XL) that lets you copy data from one file to another. This manual covers the use of FCOPY on MPE V/E and on MPE XL, since FCOPY works the same way on both operating systems. Because MPE V/E and MPE XL treat devices as files, you can use FCOPY to copy data from any input device to any output device. FCOPY is useful for making multiple copies of files, for making account-independent magnetic tape copies of disk files to move from one HP 3000 system to another, and for transferring programs or data from one medium to another; for example, from magnetic tape to disk.

In addition, FCOPY has several powerful features that extend its usefulness beyond these basic functions. For example, FCOPY lets you convert data from one computer code system to another as you copy it. This function makes it easy to prepare files on magnetic tapes for use at various computer installations or to incorporate files on magnetic tape from other computer installations into your HP 3000 system. Another FCOPY function lets you copy portions of a file. You might use this function to divide an existing file into several separate files.

Before You Use FCOPY

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

- The MPE account structure.
- File security.
- How to specify and modify file characteristics.
- File and device relationships.
- How to run jobs and sessions.

2 Using FCOPY

You use FCOPY by issuing commands that identify the files you want to copy and the way in which you want to copy them. This chapter describes the components of an FCOPY command, how to identify the files that you reference in a command, how to issue an FCOPY command, and the type of message you might see in response to a command.

Components of an FCOPY Command

You can issue an FCOPY command in several ways, but, whichever method you choose, the command always has the same components. It identifies a *fromfile*, a *tofile*, and one or more functions that you want FCOPY to perform. You enter the FCOPY subsystem from MPE by typing FCOPY at the MPE colon prompt (:). At the FCOPY prompt (>), you set up the command. An example of an FCOPY command is:

```
:FCOPY
>FROM=OLDFILE;TO=NEWFILE;NEW
```

"FROM" and "TO" Files

A *fromfile* is the input file for an FCOPY command. It contains the data you want to copy. A *tofile* is the output file to which you want to copy the data.

You identify a *fromfile* for an FCOPY command with the FROM parameter. It has the following format:

```
>FROM[={fromfile
              * <file>
              *
              <empty>}]
```

The value you assign to FROM can be an input file name (*fromfile*); an asterisk (*) plus a file name, if desired; an asterisk (*); or nothing at all (<*empty*>). An asterisk preceding the *fromfile* backreferences the *fromfile* named in a previously set file equation. An asterisk, alone, specifies continued use of the *fromfile* specified in the previous command. If you leave FROM empty, you can use your terminal (or a spoolfile during a job) as the input file.

In the example below, the FROM parameter describes an input file named INFO1 for an FCOPY command.

```
>FROM=INFO1
```

You specify a *tofile* with the TO parameter. TO has the following format in MPE V/E:

```
;TO[={(datafile,keyfile)
        tofile
        * <file>
        *
       <empty>}]
```

The value that you assign to TO can be the names of data and key files for a new KSAM V/E file (*datafile* and *keyfile*); the name of the output file (*tofile*); an asterisk (*) plus a file name, if desired; an asterisk; or nothing at all (<*empty*>). An asterisk preceding a *tofile* name backreferences the *tofile* named in a previously set file equation. An asterisk, alone, specifies continued use of the *tofile* specified in the previous command. For details, see "Defining Files" later in this chapter. If you leave TO empty, you can copy files to your terminal (or to a printer during a job).

TO has the following format in MPE XL:

```
;TO [={{(datafile,keyfile)
        (tofile)}
        tofile
        *<file>
       <empty>}]
```

You use a single file name enclosed in parentheses (*tofile*) to create a KSAM XL file. You still use the names of data files and key files to create a new CM KSAM file (*datafile* and *keyfile*). The other values are the same as those described above. For more information, refer to *Using KSAM XL* (32650-90168).

The example below describes an output file named OUTFILE for an FCOPY command.

```
;TO=OUTFILE
```

For more information on the FILE command, refer to the *MPE V/E Commands Reference Manual* (32033-90006) or the *MPE XL Commands Reference Manual* (32650-90003).

FCOPY Commands

In addition to specifying input and output files in an FCOPY command, you also define the functions that you want FCOPY to perform. Each function has its own syntax, and there are guidelines for its use. The NEW function, for example, lets you create a new disk file. An example of an FCOPY command using the NEW function is shown below:

```
:FCOPY
>FROM=OLDFILE;TO=NEWFILE;NEW
```

Chapter 3 contains more examples that illustrate the use of FCOPY commands, and chapter 5 describes all FCOPY functions in detail. Table 2-1 lists the FCOPY functions and brief descriptions in alphabetical order.

Table 2-1. FCOPY Functions

Function	Description
BCDICIN	Translates from BCDIC to ASCII
BCDICOUT	Translates from ASCII to BCDIC
CCTL	Specifies the first character of each record as a carriage control character in *tofile*
CHAR	Displays the contents of a file as ASCII symbols
CLEAR	Displays the contents of a file as character codes
COMPARE	Compares two files
COPYACD	Copies an ACD associated with a file (available on MPE XL release 3.0 or later and on MPE V/E)
DEBLOCK	Deblocks blocked records
EBCDICIN	Translates from EBCDIC to ASCII
EBCDICOUT	Translates from ASCII to EBCDIC
EBCDIKIN	Translates from EBCDIK to JIS
EBCDIKOUT	Translates from JIS to EBCDIK
EXIT	Leaves the FCOPY subsystem and returns you to MPE
FILES	Copies multiple files from tape
HEX	Displays the contents of a file in hexadecimal form
HEXO	Displays the contents of a file in hexadecimal form and the sequential record number in octal form
IGNERR	Bypasses and reports magnetic tape errors
KANA	Displays the contents of a file as JIS character symbols
KEY	Specifies the key sequence in which to copy a KSAM file
LANG	Represents a native language name or number
NEW	Creates a new permanent disk file
NOCCTL	Specifies that the first character of each record in the *fromfile* is not to be a carriage control character
NOKSAM	Copies a KSAM V/E or CM KSAM data file to a non-KSAM file
NOUSERLABELS	Omits user labels when copying between disk and tape
OCTAL	Displays the contents of a file in octal form
SKIPEOF	Positions a serial storage device at a desired file
SUBSET	Copies a subset of a file

Table 2-1. FCOPY Functions

Function	Description
UPSHIFT	Converts lowercase characters to uppercase
VERIFY	Compares files after copying

Defining Files

If you are copying files from or to devices other than disk, you must define the files and their associated devices with MPE FILE commands before you issue an FCOPY command. For example, to copy a file from magnetic tape to a line printer, you define two device files as follows:

```
FILE TAPEFILE;DEV=TAPE;REC=-80,25,F,ASCII
FILE PRINTER;DEV=LP
```

TAPEFILE and PRINTER are the formal file designators you use in the FCOPY command. TAPE and LP are device class names for a magnetic tape unit and a line printer respectively. Device class names are defined when your system is configured and may vary from one installation to another.

Use the two formal file designators as the *fromfile* and *tofile* in an FCOPY command. Type an asterisk (*) before each file name to tell FCOPY to refer to the previous FILE commands for the file's description. The FROM and TO parameters below reference the two files defined above:

```
>FROM=*TAPEFILE;TO=*PRINTER
```

Appendix B contains a list of default file characteristics; FCOPY assumes files to have these default characteristics unless you define the files with other characteristics. For more information on the FILE command, refer to the *MPE V/E Commands Reference Manual* (32033-90006) or the *MPE XL Commands Reference Manual* (32650-90003).

General Guidelines

An FCOPY command must follow these general guidelines:

Semicolons always separate the different components of a command. You can leave spaces between components. However, there can be no more than 70 characters between two semicolons. The three examples of an FCOPY command shown below are all valid.

```
>FROM=A;         TO=B; NEW
>FROM=C;TO=D;NEW
>FROM=E; TO=F; NEW
```

NOTE For the sake of clarity, the examples in this manual show user input without spaces between parameters, unless the spaces are required as part of the syntax.

To continue an FCOPY command onto more than one line, use an ampersand (&) at the end of each line except the last. For example,

```
>FROM=A;&
TO=B;&
NEW
```

An FCOPY command has no maximum length; however, a single line in an FCOPY command cannot be more than 72 characters long.

To interrupt FCOPY while it is copying a file, press **CTRL** Y. If you use this procedure to cancel a command, you cannot use an asterisk (*) to specify reuse of a previous file in the next command.

Numeric parameters in FCOPY functions can be in either octal or decimal form. Precede octal numbers with a percent sign (%).

The numbering of files and columns of data begins with 1; numbering of records within a file begins with 0.

Issuing an FCOPY Command

You can issue an FCOPY command directly from MPE, or you can enter the FCOPY subsystem and issue one or more commands before returning to MPE.

Using FCOPY from MPE

You can issue a single FCOPY command directly from MPE using either the RUN or FCOPY command. If you use the RUN command, include FROM, TO, and any functions you want in the INFO=" " parameter. For example:

```
RUN FCOPY.PUB.SYS;INFO="FROM=A;TO=B;NEW"
```

When you use the FCOPY command, you list FROM, TO, and any functions you want after the command. For example:

```
FCOPY FROM=A;TO=B;NEW
```

The FCOPY Subsystem

To enter the FCOPY subsystem, type RUN FCOPY.PUB.SYS in response to the MPE prompt. For example,

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

Or you can simply type FCOPY. The FCOPY subsystem then prompts you for commands with a greater-than symbol (>).

Once you are in the FCOPY subsystem, you can issue one or more commands before returning to MPE. Within the FCOPY subsystem, an FCOPY command consists of the FROM and TO parameters and any functions that you want to perform. For example,

```
>FROM=A;TO=*PRINTER
```

Exiting FCOPY

To leave the FCOPY subsystem, type EXIT, or E, in response to the FCOPY prompt. For example,

```
>EXIT
```

Interrupting FCOPY

You can temporarily leave the FCOPY subsystem and return to MPE by pressing the **BREAK** key. When you see the MPE colon prompt (:), you can enter any MPE command that does not create another process. Use the RESUME command to return to the FCOPY subsystem, or use the ABORT command to terminate FCOPY.

MPE Commands

From the FCOPY subsystem, you can issue any MPE command that does not create another process. To do so, type a colon immediately after the FCOPY prompt, and then

type the MPE command. This feature of FCOPY is handy when you wish to use the FILE command to define the files you intend to copy. For example,

```
>:FILE PRINTER;DEV=LP
>FROM=A;TO=*PRINTER
```

Using FCOPY in a Job

You can use the MPE STREAM command to create jobs that contain FCOPY commands. Follow the instructions for streaming jobs in the *MPE V/E Commands Reference Manual* (32033-90006) or the *MPE XL Commands Reference Manual* (32650-90003). Your job input file may contain MPE FCOPY commands, FCOPY subsystem commands, or both. Be sure to provide a substitute character (usually an exclamation point) for the MPE command prompt. Do not enter FCOPY prompts. For example,

```
:STREAM EXAMPLE
!JOB FCOPYJOB,JOE.FCOPY.EXAMPLES,STREAM
!RUN FCOPY.PUB.SYS
FROM=A;TO=B;NEW
EXIT
!FCOPY FROM=B;TO=C;NEW
!EOJ
#JI44
:
```

Customizing Commands

By combining FCOPY functions and using different forms of the functions, you can use FCOPY for many purposes. The examples in chapter 3 illustrate how to use FCOPY commands for several common situations. If you cannot find the task you need to perform in chapter 3, use the steps described below to find the information you need.

In general, to determine which FCOPY command to use for a specific purpose, do the following:

- Use the index to look up the task you want to perform.
- Use chapter 3 to find instructions for the task.
- Check table 2-1 to determine which functions you might need.
- Use chapter 5 to find descriptions of the functions you plan to use.

FCOPY Messages

FCOPY displays three types of messages: status, warning, and error messages. Status messages report on the progress of a copying operation. Warning messages warn you of problems that were not serious enough to cause an operation to terminate. Error messages report errors that caused an FCOPY command to terminate (in a session) or caused the FCOPY subsystem to terminate (in a job stream). For more information on FCOPY messages, refer to appendix A.

3 FCOPY Applications and Examples

This chapter contains instructions for using FCOPY and its functions for several common purposes. It also provides examples of these operations. Although it explains the use of several FCOPY functions, it does not completely describe the characteristics of any function. Refer to chapter 5 for complete descriptions of FCOPY functions.

Copying Files

You can use FCOPY to create new files, append one or more files onto an existing file, or extract portions from files.

New Disk Files

To create a new disk file as you copy information to it, use the NEW function. For example,

```
>FROM=OLDDATA;TO=NEWDATA;NEW
```

In this example, the new file called NEWDATA has the default file characteristics described in appendix B and exists in the current account and group.

Appending Files

Often you need to string several files together. Use FCOPY to append a file to another as follows:

1. Use the MPE FILE command to give the *tofile* append access (ACC=APPEND), and set its limit large enough to hold the contents of all the files you are appending to it.

   For example,

   ```
   >:FILE INFO3;REC=-80,10,F,ASCII;ACC=APPEND;DISC=500,2
   ```

   Refer to the *MPE File System Reference Manual* (30000-90236) for MPE V/E or the *Accessing Files Programmers Guide* (32650-90017) for MPE XL for instructions on calculating the amount of file space you would require.

2. Copy the first file you want to append to the *tofile*. Type an asterisk in front of the *tofile* name to use the file characteristics described in the preceding FILE command instead of the default file characteristics. If the *tofile* is new, use the NEW function. For example,

   ```
   >FROM=INFO1;TO=*INFO3;NEW
   ```

   The system responds with the following message:

   ```
   EOF FOUND IN FROMFILE AFTER RECORD 19

   20 RECORDS PROCESSED * * * 0 ERRORS
   ```

3. Copy any subsequent files in the order you want to append them. Use an asterisk alone to reuse the *tofile* in the previous command. For example,

   ```
   >FROM=INFO2;TO=*

   EOF FOUND IN FROMFILE AFTER RECORD 24

   25 RECORDS PROCESSED * * * 0 ERRORS
   ```

Extracting Parts of Files

Use the SUBSET function to extract portions of a file. You can define a subset as a set of continuous records or as a set of records with a certain pattern of characters or numbers. For example, a file may contain information about the employees in your company. Each record in the file describes one employee, and the employee's last name begins in column 1 of each record. To create a file containing all employees whose last name begins with "S," copy the subset of the employee file that contains the character "S" in column 1. To create a file containing all employees whose last name is "Smith," copy the subset of the employee file that contains the characters "Smith" beginning in column 1.

Defining Subsets

To copy a subset containing a particular character string from a file, use the following form of the SUBSET function:

```
;SUBSET [="characterstring"[,[column]][,EXCLUDE]]
```

Substitute the series of characters that you want to match for *characterstring*, and the beginning column number of the character string for *column*. For example, the command below copies the subset of all records with the characters "COLLEGE" beginning in column 40 from the EMPRECS file to the GRADS file:

```
>FROM=EMPRECS;TO=GRADS;SUBSET="COLLEGE",40

EOF FOUND IN FROMFILE AFTER RECORD 5985

2859 RECORDS PROCESSED * * * 0 ERRORS
```

To copy a subset excluding specific records, use the EXCLUDE parameter. For example, the command below copies all records in EMPRECS, except those with the characters "COLLEGE" beginning in column 40, to the file NONGRADS.

```
>FROM=EMPRECS;TO=NONGRADS;SUBSET="COLLEGE",40,EXCLUDE

EOF FOUND IN FROMFILE AFTER RECORD 5985

3127 RECORDS PROCESSED * * * 0 ERRORS
```

Displaying Disk Files

You display the contents of your files by copying them to your terminal. FCOPY displays files in the following formats:

- Character.
- Octal.
- Hexadecimal.

ASCII Files

If you know that the file is an ASCII file, simply copy it to your terminal by leaving TO empty. For example,

```
>FROM=ASCITEST;TO=
This is an ASCII file.
It contains two lines of text.

EOF FOUND IN FROMFILE AFTER RECORD 1

2 RECORDS PROCESSED * * * 0 ERRORS
```

Hexadecimal and Octal Codes

Use the CHAR and HEX, HEXO, or OCTAL functions to display both ASCII characters and hexadecimal or octal codes. You might display a file in both ASCII and octal, for example, when your terminal upshifts lowercase characters. Combining octal with ASCII would let you compare octal values and determine when characters are actually lowercase. The example below displays both octal codes and ASCII characters. The numbers at the end of each record are text editor line numbers that start at 1 (00001000):

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

TEXT3.PUB.ACCOUNT RECORD 0 (%0)

000000: 052110 044523 020111 051440 046111 047105 020061 27040 THIS IS LINE 1
000010: 020124 044105 020070 026504 044507 044524 020116 52515 THE 8-DIGIT NUM
000020: 041105 051123 020101 052040 052110 042440 051111 43510 BERS AT THE
000030: 052040 040522 042440 052110 042440 020040 020040 20040 RIGHT ARE THE
000040: 020040 020040 020040 020040 030006 030006 030006 30006 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 051140 020040 020040 020040 020040 NUMBERS
000040: 020040 020040 020040 020040 030060 030061 030460 030060 00002000
```

Binary Files

To display the contents of a binary disk file at your terminal, do the following:

1. Leave the TO parameter empty.
2. Use the HEX or OCTAL function to display the hexadecimal or octal representation of the file's contents. Use the CHAR function to display any ASCII characters that may exist.

For example, you might use the commands below to display a file on your terminal.

```
>FROM=FOO;TO=;OCTAL;CHAR

FOO RECORD 0 (%0, #0)

00000: 000000 000000 000000 044105 053514 042524 052055 050101 ......HEWLETT-PA
00010: 041513 040522 042040 031462 031061 031101 027060 031456 CKARD 32212A.03.
00020: 031460 020106 044514 042440 041517 050111 042522 020040 30 FILE COPIER
00030: SAME: TO 000040-1
00040: 020040 020040 020040 020040 020040 020040 024103 024440             (C)
00050: 044105 053514 042524 052055 050101 041513 040522 042040 HEWLETT-PACKARD
```

The NORECNUM function can be used to suppress the display record numbers.

File Subsets

To display a subset of a file on your terminal, use the SUBSET function. The example below copies a subset of five records (records 30 through 34) from the file CUSTDATA to a terminal. CUSTDATA is an ASCII file containing the names and addresses of customers.

```
>FROM=CUSTDATA;TO=;SUBSET=29:33

XYZ MANUFACTURING CO.                   131 PINE STREET SF
WIDGET, INC.                            43721 OAK STREET DET
ROBERT ALCOTT & ASSOCIATES              752 SURF AVE. CHI
WILLIAM LEWIS & SON                     3217 W. MADISON AVENUE NY
SUPER GIZMO OF NEVADA, INC.             431 S. JACKPOT LANE RENO
```

Printing Files

You can print a file by copying it to a printer. For example, suppose you ran a program or utility that normally displays messages on your terminal, but you redirected the program's output to a disk file to save it for later study. To print such a file, use FCOPY to copy it to a printer.

Printing an ASCII File:

To print an ASCII file, follow these steps:

1. Use the MPE FILE command to define a file name for your printer. For example,

   ```
   >:FILE LIST;DEV=LP2
   ```

2. Use the printer's file name, LIST, as the *tofile* name. Precede the file name with an asterisk to backreference the FILE command. For example,

   ```
   >FROM=TEXT1;TO=*LIST
   ```

Printing a Binary File:

To print a binary file, follow these steps:

1. Use the MPE FILE command to define a file name for your printer. For example,

   ```
   >:FILE LIST;DEV=LP2
   ```

2. Use the HEX or OCTAL function to display the hexadecimal or octal representation of the binary content.
3. Use the CHAR function to print any ASCII characters that may exist.
4. Use the NORECNUM function if you wish to suppress line numbers from printing. For example,

   ```
   >FROM=FILE1;TO=*LIST;OCTAL;CHAR;NORECNUM
   ```

Using Your Terminal as a Fromfile

You can enter a small amount of information into a file directly from your terminal keyboard. (You may enter a large amount if you want to, but that is easier with an editor).

To copy information from your terminal:

1. Leave the FROM parameter empty. If the *tofile* is a new disk file, use the NEW function. For example,

   ```
   >FROM=;TO=NEWFILE;NEW
   ```

2. FCOPY waits for you to enter records at your terminal. Use a **Return** to mark the end of each record. For example,

   ```
   ABC MOVING COMPANY     Return          **Record 1 **
   123 MAIN STREET SF     Return          **Record 2 **
   POP MOVING COMPANY     Return          **Record 3 **
   789 PINE ST SJ         Return          **Record 4 **
   ```

3. Type **CTRL** Y or :EOD to mark the end of the *fromfile*. (If you use **CTRL** Y, FCOPY can accept additional commands. If you use :EOD, FCOPY terminates.)

Copying Files of Varying Sizes

You can determine the block size of a file by multiplying its record size by its block factor.

To copy a file when you do not know the block size or when the block size varies (as it could on a magnetic tape), use records of undefined length for both files.

In the file equations, specify a record size at least as large as the largest record you expect. For example,

```
>:FILE OLDTAPE;REC=-16384,,U,ASCII;DEV=TAPE

>:FILE NEWTAPE;REC=-16384,,U,ASCII;DEV=TAPE

>FROM=*OLDTAPE;TO=*NEWTAPE
```

This copies all records from the first file of one tape to another tape, preserving the record structure. You can use the FILES parameter to copy more than one file. If you are copying an MPE V/E store tape, specify the number of files stored plus six. These extra files contain tape header information for the STORE program.

Copying Tapes

The general methods for copying tapes described in this chapter apply to unlabeled magnetic tapes, serial disks, and cartridge tapes. If you need to copy files from or to labeled tapes, refer to the instructions for using labeled tapes later in this chapter.

NOTE Labeled tapes must be used if the file being written will extend beyond one reel.

Specifying Tape Files

Use the SKIPEOF function to position an unlabeled tape, cartridge tape, or serial disk at the beginning of the *fromfile* or *tofile* you want to copy. The SKIPEOF function "skips" over a specified number of end-of-file (EOF) marks to position the tape at the beginning of a particular file.

You can specify the number of EOF marks to be skipped as either absolute file numbers (for example, the fifth file on the tape) or relative file numbers (for example, the fifth file from the tape's current position). A plus (+) or minus (–) sign distinguishes a relative file number from an absolute file number. A plus sign tells FCOPY to skip forward, and a minus sign tells FCOPY to skip backward. A comma before the plus or minus sign (for example, SKIPEOF=,+1) indicates that the file is a *tofile*; no comma (for example, SKIPEOF=+1) indicates a *fromfile*.

The example below tells FCOPY to copy the file three files before TAPEA's current position and place it in the fifth file on TAPEB.

```
>FROM=*TAPEA;TO*TAPEB;SKIPEOF=-3,5
```

NOTE When FCOPY copies files from or to magnetic tapes, it leaves the tape positioned before the EOF mark for the file it just copied. Thus, if you want to copy a series of files from or to the same magnetic tape, be sure to include SKIPEOF=+1 in the commands for copying each file after the first. SKIPEOF=+1 instructs FCOPY to position the tape at the beginning of the next file.

Ignoring Tape Errors

Use the IGNERR function when you want certain file system errors to be ignored during the process of copying files from magnetic tape. This allows the recovery of as much data as possible from faulty media. The following errors can be ignored:

```
21    Data Parity
26    Transmission
27    I/O Timeout
38    Type Parity
```

Using IGNERR, you can instruct FCOPY to terminate the copy operation after it encounters a given number of errors. The example below instructs FCOPY to ignore errors, but to

terminate if it encounters more than five errors.

```
>FROM=*TAPE;TO=DISC3;IGNERR=5
```

Comparing Files

To compare the contents of two files without changing either file, use the COMPARE function. You can compare the contents of a disk file and a tape file or the contents of two files stored on tape. As with the IGNERR function, COMPARE lets you specify a number of errors after which it terminates. The default is 1. The example below compares the disk file TEXTA with the first file on TAPE1:

```
>:FILE TAPE1;DEV=TAPE;REC=-80,16,F,ASCII
>FROM=TEXTA;TO=*TAPE1;COMPARE

COMPARE ERROR FOUND AT RECORD 52, BYTE 19

*304* COMPARE OPTION: RAN OUT OF COMPARE ERRORS AT FROMFILE RECORD
53

53 RECORDS PROCESSED * * * 0 ERRORS
```

Verifying Copied Files

Use the VERIFY function to compare the contents of two files immediately after copying them. As with IGNERR and COMPARE, VERIFY lets you control the number of errors that FCOPY reports. If you do not specify a maximum number of errors, FCOPY terminates after it encounters the first error. The example below copies the third file on TAPE2 to the next file on TAPE6 and then compares the two files.

```
>:FILE TAPE2;DEV=TAPE
>:FILE TAPE6;DEV=TAPE
>FROM=*TAPE2;TO=*TAPE6;SKIPEOF=3,+1;VERIFY

EOF FOUND IN FROMFILE AFTER 19 RECORDS

20 RECORDS PROCESSED * * * 0 ERRORS
```

Changing the Tape Blocking Factor

Use the DEBLOCK function to change the record blocking factor as you copy a file. DEBLOCK is most useful when copying tapes from other computer systems to the HP 3000. The HP 3000 blocks records to tape with an even number of bytes per record. Other systems may block records to tape with an odd number of bytes per record. When you copy a foreign file to the HP 3000, you must provide a tape format compatible with MPE. For more information on copying foreign tapes, refer to "Copying Foreign Tapes" later in this chapter.

To deblock a file with odd-byte records, perform the following steps:

1. Use the MPE FILE command to treat each block of the *fromfile* as a single record and set the blocking factor to 1. The record length should be the old record length times the

old blocking factor. For example, a block of ten 79-byte records would be represented as a 790-byte record with a blocking factor of 1:

```
>:FILE TAPEFILE;REC=-790,1,U,ASCII
```

2. In the FCOPY command, set DEBLOCK equal to the original record length. For example,

```
>FROM=*TAPEFILE;TO=DISCFILE;DEBLOCK=-79

EOF FOUND IN FROMFILE AFTER RECORD 789

790 RECORDS PROCESSED * * * 0 ERRORS
```

Combining Disk Files on Tape

To combine several disk files into a single tape file, follow the steps described below:

1. Use the MPE FILE command to describe the tape file to be written. For example,

```
>:FILE T;DEV=TAPE;REC=-80,20,F,ASCII
```

2. Copy the first disk file to tape using an asterisk before the *tofile* name to backreference the FILE command. For example,

```
>FROM=DATA1;TO=*T

EOF FOUND IN FROMFILE AFTER RECORD 610

611 RECORDS PROCESSED * * * 0 ERRORS
```

3. Copy the subsequent disk files to tape without repositioning the tape drive. Use an asterisk as the *tofile* name. For example,

```
>FROM=DATA2;TO=*

EOF FOUND IN FROMFILE AFTER RECORD 711

712 RECORDS PROCESSED * * * 0 ERRORS

>FROM=DATA3;TO=*

EOF FOUND IN FROMFILE AFTER RECORD 472

473 RECORDS PROCESSED * * * 0 ERRORS
```

Combining Tape Files on Disk

To combine several tape files, or subsets of tape files, in a single disk file, perform the following steps:

1. Use the MPE FILE command to give the disk file append access (ACC=APPEND). The disk file must be large enough to hold the contents of all of the tape files, and you must specify its record structure. For example,

```
>:FILE AMALGAM;ACC=APPEND;REC=-80,16,F,ASCII;DISC=5000
```

2. Use a second FILE command to describe the tape file. For example,

```
>:FILE T;DEV=TAPE;REC=-80,1,F,ASCII
```

3. Copy the first tape file to the disk file. Precede the file names with asterisks to reference the two file commands. If you intend to copy only a subset of the tape file, use the SUBSET function. For example, the command below copies all records with the character "C" in column 1.

```
>FROM=*T;TO=*AMALGAM;SUBSET="C",1

9 RECORDS PROCESSED * * * 0 ERRORS
```

4. In subsequent FCOPY commands, use an asterisk to reference the same *tofile*. Use SKIPEOF to specify the position of the *fromfile* on tape. For example,

```
>FROM=*T;TO=*;SKIPEOF=1

16 RECORDS PROCESSED * * * 0 ERRORS

>FROM=*;TO=*;SKIPEOF=3

EOF FOUND IN FROMFILE AFTER RECORD 302

303 RECORDS PROCESSED * * * 0 ERRORS
```

Copying Labeled Tapes

Instructions for copying to and from labeled tapes are given below, as well as an example of how to deblock a labeled tape.

Copying Files to Labeled Tapes

The default parameter NEXT specifies that a file is to be written to the current position on the tape. If this is a newly mounted tape, the position will be the beginning of the tape.

To copy a single file starting at the beginning of the tape:

1. Use the NEXT parameter in the FILE command describing the labeled tape. For example,

```
>:FILE LTAPE;DEV=TAPE;LABEL=TLABEL,,,NEXT;REC=-80,10,F,ASCII
```

2. Copy the first disk file to the tape, using an asterisk before the tape file name to backreference the file command. For example,

```
>FROM=FILEA;TO=*LTAPE

EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

To copy a second file to the same labeled tape, specify the desired position of the file.

1. Enter a number, in this case 2, in place of the NEXT parameter in the FILE command.

```
>:FILE LTAPE;DEV=TAPE;LABEL=TLABEL,,,2;REC=-80,10,F,ASCII
```

2. Copy the file to the tape, using an asterisk before the tape file name to backreference the file command. For example,

```
>FROM=SRCFILE;TO=*LTAPE

EOF FOUND IN FROMFILE AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

To combine data from two disk files into a single file on tape, use an asterisk alone as the *tofile* in the command for copying the second disk file. For example,

```
>FROM=FILEC;TO=*LTAPE

EOF FOUND IN FROMFILE AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS

>FROM=FILED;TO=*
```

```
EOF FOUND IN FROMFILE AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

The parameter ADDF specifies that a file is to be written following all existing files on the tape. To add a new file to a tape without overwriting existing files, perform the following steps:

1. Use the ADDF parameter in the MPE FILE command. ADDF positions the tape drive after the last file on the tape. For example,

   ```
   >:FILE LT2=FILE4;DEV=TAPE;LABEL=TLABEL,ANS,,ADDF
   ```

2. Copy the file to the tape, preceding the *tofile* name with an asterisk to backreference the FILE command. For example,

   ```
   >FROM=FILEF;TO=*LT2

   EOF FOUND IN FROMFILE AFTER RECORD 9

   10 RECORDS PROCESSED * * * 0 ERRORS
   ```

Copying Files from Labeled Tapes

You may copy a particular file from a labeled tape by using its number or its file name. When reading a labeled tape, it is usually not necessary to specify the record structure of the tape file in a file equation because the information is read from the labels.

Copying Files by Number

You can copy a file from labeled tape using its number, as follows:

1. Specify its file number in a FILE command. For example, the command below specifies the third tape file:

   ```
   >:FILE LT;DEV=TAPE;LABEL=TLABEL,ANS,,3
   ```

   Copy the file from the tape, preceding the *fromfile* name with an asterisk to backreference the FILE command. For example,

   ```
   >FROM=*LT;TO=NEWDISK;NEW

   EOF FOUND IN FROMFILE AFTER RECORD 19

   20 RECORDS PROCESSED * * * 0 ERRORS
   ```

Copying Named Files

You can also copy a named file from a labeled tape by using its name, as follows:

1. Specify the file name in the MPE FILE command describing the tape file. For example, the command below specifies a file named FILE4:

   ```
   >:FILE LTNAME=FILE4;DEV=TAPE;LABEL=TLABEL,ANS,,0
   ```

2. In the FCOPY command, backreference the FILE command by preceding the *fromfile* name with an asterisk. For example,

```
>FROM=*LTNAME;TO=FILED;NEW

EOF FOUND IN FROMFILE AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

Concatenating Files from a Labeled Tape

You can concatenate files from a labeled tape into a single file, as follows:

1. Use the NEXT parameter in the MPE FILE command describing the tape. For example,

```
>:FILE LTAPE;DEV=TAPE;LABEL=TLABEL,ANS,,NEXT;REC=-80,10,F,ASCII
```

2. In your FCOPY command, backreference the FILE command by preceding the *fromfile* name with an asterisk. To copy the labeled tape files to a single file, use the FCOPY FILES function.

```
>FROM=*LTAPE;TO=CONCAT;FILES=4

EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS

EOF FOUND IN FROMFILE AFTER RECORD 29

30 RECORDS PROCESSED * * * 0 ERRORS

EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS

EOF FOUND IN FROMFILE AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

Deblocking Labeled Tapes

You can read and deblock labeled tapes with blocked odd-byte records similarly to unlabeled tapes.

1. It is necessary to specify the tape characteristics as undefined records so that the byte count in the header labels will be ignored. For example, if the tape has 133-byte records, blocked in twenty-record blocks, you can deblock the records as follows:

```
>FILE LTAPE;DEV=TAPE;LABEL=TLABEL;REC=-2660,1,U,ASCII
```

2. In the FCOPY command, set DEBLOCK equal to the original record length.

```
>FROM=*LTAPE;TO=EVENBYTE;DEBLOCK=-133;NEW
```

Note that this method is not necessary with labeled tapes created on MPE. MPE always specifies an even byte count in the header label if there is more than one record per tape block.

Copying Files Between HP 3000 Systems

You can use magnetic tapes to copy disk files from one HP 3000 system to another. If all files on a single tape have the same record size and type, copy the files from the first system to tape. Then copy the tape files onto the second system, as follows:

1. Issue an MPE FILE command describing the tape to which you want to copy the disk files. You must specify the record size and structure. For example,

```
>:FILE T;DEV=TAPE;REC=-80,16,F,ASCII
```

2. Copy each of the disk files to the tape. Use an asterisk to backreference the FILE command describing the tape. Use SKIPEOF=,+1 to position the tape at the beginning of the next file. For example,

```
>FROM=FUTIL;TO=*T

EOF FOUND IN FROMFILE AFTER RECORD 297

298 RECORDS PROCESSED * * * 0 ERRORS

>FROM=DUMPER;TO=*;SKIPEOF=,+1

EOF FOUND IN FROMFILE AFTER RECORD 405

406 RECORDS PROCESSED * * * 0 ERRORS

>FROM=SPOOLER;TO=*;SKIPEOF=,+1

EOF FOUND IN FROMFILE AFTER RECORD 386

387 RECORDS PROCESSED * * * 0 ERRORS
```

3. Use the FILE command to describe the tape drive on the second system. The record size and structure must be specified and must match how the tape was written.

```
>:FILE T;DEV=TAPE;REC=-80,16,F,ASCII
```

4. Copy the files from tape to disk. Use an asterisk to backreference the FILE command describing the tape drive. Use SKIPEOF=,+1 to position the tape at the beginning of the next file. For example,

```
>FROM=*T TO=FUTIL;NEW

EOF FOUND IN FROMFILE AFTER RECORD 297

298 RECORDS PROCESSED * * * 0 ERRORS

>FROM=*;TO=DUMPER;SKIPEOF=,+1

EOF FOUND IN FROMFILE AFTER RECORD 405

406 RECORDS PROCESSED * * * 0 ERRORS

>FROM=*;TO=SPOOLER;SKIPEOF=,+1

EOF FOUND IN FROMFILE AFTER RECORD 386

387 RECORDS PROCESSED * * * 0 ERRORS
```

NOTE If the files are of varying structure, or have user labels or varying file codes, it is better to use the MPE STORE and RESTORE commands to move files between systems.

Copying Foreign Tapes

Tapes written on foreign machines often have to be read on HP 3000 systems. If you don't know the exact data format, here are some ideas that might help you determine what it is.

1. Mount the tape on your HP 3000 (without a write ring). The automatic volume recognition message tells you if it is labeled, and if so, what the label is. If the tape is ANSI-labeled, the character data is probably in ASCII, but if the tape is IBM-labeled, the character data is almost certainly in EBCDIC.
2. Use FCOPY to dump one or two blocks of data from the tape. If the data is likely to be EBCDIC characters, then use EBCDICIN. For example, if the tape was reported as "Vol 000001,IBM", you could do this:

```
FILE L;DEV=TAPE;LABEL=000001,IBM;REC=-16384,1,U,ASCII

FILE LP;DEV=LP

FCOPY FROM=*L;TO=*LP;OCTAL;CHAR;EBCDICIN;SUBSET=,1
```

   This reads one block from the tape, converts it to ASCII, formats it as both octal and characters, and prints it, telling you how big it is. The tape records may be blocked, and examination of the data will probably disclose a pattern of repetitions from which the blocking factor can be deduced. This is necessary more often with unlabeled tapes or IBM-format labeled tapes, which may not have the HDR2 records that give the record and block size.
3. Once you know the format of the data, use FCOPY to copy the data to where you want it--for example, to disk. Suppose that the above tape is found to have blocks of 2640 bytes, consisting of 20 records of 132 bytes. You could do the following:

```
FILE LL;DEV=TAPE;LABEL=000001,IBM;REC=-132,20,F,ASCII

FILE IBMDATA;REC=-132,F,ASCII;DISC=12000,32

FCOPY FROM=*LL;TO=*IBMDATA;NEW;EBCDICIN
```

   You need an estimate of the amount of data on the tape. If you underestimate, the disk file fills up and FCOPY terminates, leaving some data unread. If you overestimate, you waste some disk space, but you can reduce the amount wasted if you specify a large number of extents (32 in the example).

NOTE If you are translating data from EBCDIC to ASCII, use the EXCLUDE option to ensure that "Packed Fields (comp-3)" are not translated. For more information, refer to the SUBSET function.

Remote Copies

The following example copies a local file to a line printer connected to a remote system. First, the line must be made available.

1. After you have logged on, at the MPE colon prompt (:) type `DSLINE` *`nodename`* to open the line.
2. Log on to the remote system by typing `REMOTE HELLO` *`user.account`* at the prompt, as follows:

```
DSLINE nodename
REMOTE HELLO user.account
```

3. Then type the following commands:

```
FILE OUTPUT;DEV=nodename # LP
FCOPY FROM=diskfile;TO=*OUTPUT
```

Copying KSAM Files

If your system supports the KSAM/3000 subsystem, you can use FCOPY to copy KSAM files. With some exceptions, you can copy KSAM files in the same way that you copy regular HP 3000 files. The functions SKIPEOF, IGNERR, and SUBSET do not apply to KSAM files. The KEY option applies to all KSAM files. The NOKSAM option applies only to KSAM V/E and CM KSAM files. Refer to the *KSAM/3000 Reference Manual* (30000-90079) and *Using KSAM XL* (32650-90168) for more information on using KSAM files.

Copying to Existing KSAM Files

To copy a KSAM V/E file or a CM KSAM file to another existing KSAM V/E or CM KSAM file, use the FROM and TO parameters, with the value of the TO parameter as *tofile*. For example,

```
>FROM=KSAM1;TO=KSAMA
```

FCOPY copies the file in primary key sequence.

To copy a KSAM XL file to another existing KSAM XL file on an MPE XL system, use the FROM and TO parameters, with the value of the TO parameter as *tofile*, a single file name. For example,

```
>FROM=KSXL5;TO=KSXL6
```

NOTE You cannot use the NOKSAM option when copying KSAM XL files.

Copying to New KSAM Files

To copy an existing KSAM V/E or CM KSAM file to a new KSAM V/E or CM KSAM file, include the names of the new KSAM V/E or CM KSAM data file and key file in the TO parameter, as shown in the example below:

```
>FROM=OLDKSAM;TO=(NEWKSAM,NEWKSAMKEY)
```

To copy an existing KSAM XL file to a new KSAM XL file on an MPE XL system, include the name of the new KSAM XL file in the TO parameter, as shown in the example below:

```
>FROM=OLDKSXL;TO=(NEWKSXL)
```

Copying from KSAM XL Files to CM KSAM Files

You can copy a KSAM XL file to a CM KSAM file on an MPE XL system using the FROM and TO parameters as described under "Copying to Existing KSAM Files" above, with these stipulations:

- You may not use the NOKSAM option.
- The *tofile* must be specified as a (*datafile,keyfile*) pair.
- The *fromfile* must be an existing KSAM XL file.

For example,

```
>FROM=OLDKSXL;TO=(NEWKSAM,NEWKSAMKEY)
```

FCOPY estimates the size of the *keyfile* if you use the NEW option.

Copying from CM KSAM Files to KSAM XL Files

You can copy a CM KSAM file to a KSAM XL file on an MPE XL system using the FROM and TO parameters as described under "Copying to Existing KSAM Files" above, with these stipulations:

- The *tofile* must be specified as an existing KSAM XL file, unless you are creating a new file. If the *tofile* is a new file, you must specify it as (*targetfile*). You can backreference the *targetfile* to a KSAM XL file using a file equation.
- The *fromfile* must be an existing CM KSAM file.

For example,

```
> FROM=FILECM;TO=(FILENM)
```

If you open the CM KSAM file with the NOKSAM option, the deleted data records of the CM KSAM file as well as its active data records are included as active data records in the KSAM XL *tofile* (*targetfile*).

If you open the CM KSAM file with the KEY=0 option (see "Copying Key Sequences" below), the KSAM XL *tofile* contains only the active data records of the CM KSAM file.

Copying Key Sequences

To copy records from a KSAM file in a particular key sequence, use the KEY function. This option is valid for all types of KSAM files. The KEY function lets you specify the starting character position for the key you want to use to copy the file. You can copy the file in primary key sequence, any alternate key sequence, or physical record sequence. For example, the FCOPY command below instructs FCOPY to copy the KSAM V/E file KSAM1 to the KSAM V/E file KSAM2 in the sequence determined by the key beginning in position 21 of the data records of KSAM1:

```
>FROM=KSAM1;TO=KSAM2;KEY=21
```

To copy a KSAM file in physical record order, specify zero as the character location. For example, the command below copies the file KSAM3 to the file KSAM4 in the same physical record order:

```
>FROM=KSAM3;TO=KSAM4;KEY=0
```

To copy a file in primary key sequence (the default), omit the KEY function from your FCOPY command. The command below copies KSAM5 to KSAM6 in primary key order:

```
>FROM=KSAM5;TO=KSAM6
```

Copying to a Non-KSAM File

To copy data in a particular key sequence from a KSAM file to a non-KSAM file, use the KEY function. For example, the command below copies the data file of the KSAM V/E file KSAM7 to the disk file DATA in the order determined by the key beginning in location 20 of the KSAM7 data file:

```
>FROM=KSAM7;TO=DATA;KEY=20
```

To copy the data file of a KSAM V/E or CM KSAM file to a non-KSAM file, use the NOKSAM and NOUSERLABELS functions. NOKSAM instructs FCOPY to copy the CM KSAM data file to a non-KSAM file. (This option is not valid for KSAM XL files.) NOUSERLABELS instructs FCOPY not to copy the CM KSAM file's user labels. For example, the command below copies the data file of the CM KSAM file KSAM8 to the disk file NKSAM1:

```
>FROM=KSAM8;TO=NKSAM1;NOKSAM;NOUSERLABELS
```

Copying Inactive Records

The file system marks inactive records in relative I/O (RIO) and in message files. Normally when you copy such a file, FCOPY does not copy inactive records.

To copy exactly an RIO or message file, including inactive records, follow these steps:

1. Specify NOBUF and COPY in FILE commands describing both the *fromfile* and *tofile*. For example,

```
>:FILE OLD;NOBUF;COPY
>:FILE NEW;NOBUF;COPY
```

2. Backreference the FILE commands with an asterisk before the *fromfile* and *tofile* names. For example,

```
>FROM=*OLD;TO=*NEW
```

Copying Files in Native Languages

You can use the LANG parameter with four FCOPY functions to copy files in a particular native language. The four functions are CHAR, EBCDICIN, EBCDICOUT, and UPSHIFT. You can use LANG to describe a language name or number in the form:

LANG=*language*

where *language* represents a language name or number. The language you specify must be installed on your system.

If you omit the LANG parameter, FCOPY uses the current language. If there is no current language, FCOPY uses the default, NATIVE-3000. Refer to the *Native Language Support Reference Manual* (32414-90001) for MPE V/E or (32650-90022) for MPE XL for more information on using native languages.

Use with CHAR

When you specify a language with the CHAR function, FCOPY uses the character definition table associated with the language to display these characters. It replaces undefined graphic characters and control codes with periods. For example, the command below instructs FCOPY to display the contents of a file in German characters:

```
>FROM=DEUTSCH;TO=;LANG=GERMAN;CHAR;NORECNUM
```

Use with EBCDICIN/OUT

EBCDICIN converts a file from EBCDIC to ASCII and EBCDICOUT converts a file from ASCII to EBCDIC. When you use LANG with either of these functions, FCOPY uses the translation table associated with the given language to convert the file. For example, using an EBCDIC-to-ASCII conversion table, FCOPY converts data from German EBCDIC to ROMAN8:

```
>FROM=MYGEBCFL;TO=MYROM8FL;LANG=GERMAN;EBCDICIN

EOF FOUND IN FROMFILE AFTER RECORD 29

30 RECORDS PROCESSED * * * 0 ERRORS
```

Use with UPSHIFT

UPSHIFT converts lowercase characters to uppercase as you copy them to another file. When you specify a language together with UPSHIFT, FCOPY converts characters according to the rules of the given language. For example, the command below copies the file DEUTSCH to a printer, converting lowercase German characters to uppercase German characters:

```
>FROM=DEUTSCH;TO=*LP;LANG=GERMAN;UPSHIFT
```

4 FCOPY Syntax

This chapter summarizes the syntax of the FCOPY command and the FCOPY functions. The prompt in the FCOPY subsystem is >.

FCOPY Command Syntax

This is the syntax for the FCOPY command.

```
FROM  [ = {fromfile
          *
          * <file>
          <empty>} ];TO [ = { {(datafile,keyfile)
                               (tofile)       }
                             tofile
                             *
                             * <file>
                            <empty>           } ] [; functionlist ]
```

where:

fromfile Is the name of the input file.

tofile Is the name of the output file.

(*tofile*) Is the name of a KSAM XL file.

(*datafile,keyfile*) Are the names of the data and key files of a new KSAM V/E or CM KSAM file.

* Indicates the *fromfile* or *tofile* designated in the FCOPY command issued immediately before the current command. The file continues at the record after the one at which the prior operation stopped.

*<*file*>* Indicates the *fromfile* or *tofile* specified in a previous file equation.

<*empty*> Represents no entry. By leaving this blank, you specify the standard job or session device: $STDIN for the *fromfile* and $STDLIST for the *tofile*.

functionlist Represents one or more FCOPY functions. The functions may appear in any order; however, there are some restrictions for combining functions. The syntax of each individual function is given in chapter 5.

Syntax of FCOPY Functions

This section shows the syntax for each FCOPY function. For convenience, functions that share the same syntax (for example, HEX, HEXO, and OCTAL) are grouped together. For more information on a particular function, refer to chapter 5.

```
[;NOUSERLABELS][;CCTL
                ;NOCCTL][;NEW]
[;{CLEAR
   KANA}[;{HEX
           OCTAL
           HEXO}][;NORECNUM][;TITLE=title] ]

[;CHAR [;{HEX
          OCTAL
          HEXO}][;NORECNUM][;TITLE=title][;LANG=language]]

[;{HEX
   OCTAL
   HEXO}[;{CHAR
           CLEAR
           KANA}][;NORECNUM][;TITLE=title] ]

[;DEBLOCK=logical-record-length]

[;{EBCDICIN
   EBCDICOUT} [={field
               (field[;field[;..] ]) }[,EXCLUDE][;LANG=language] ] ]

[;{BCDICIN
   BCDICOUT
   EBCDIKIN
   EBCDIKOUT} [={field
                 (field[;field[;...] ])} [,EXCLUDE] ] ]

[;FILES= {number-of-files
          ALL           }]

[;IGNERR [=number-of-errors] ]

[;COMPARE[=number-of-errors] ]

[;SKIPEOF=[{+
           -}from-eofs
           from-file-number][,{ {+
                                -}to-eofs
                                 to-file-number    }] ]

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

[;NOKSAM]

[;KEY [=character-location] ]

[;UPSHIFT [;LANG=language] ]

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

[;COPYACD]

[;NOACD]
```

5 FCOPY Functions

This chapter describes each of the FCOPY functions in detail. It presents the functions in alphabetical order, so you can find information about a function quickly. If you do not know the name of the function you need to use, look up the task you want to perform in the index at the end of the manual.

The description of each function includes its syntax, instructions for using it, with restrictions (if any) and examples for its use. In each case, unless otherwise specified, the *fromfile* must have read access and the *tofile* must have write or append access. If the *tofile* has append access, the new data is added following the old data.

BCDICIN/BCDICOUT

BCDICIN translates code from BCDIC to ASCII. BCDICOUT translates code from ASCII to BCDIC.

Syntax

```
;{BCDICIN
  BCDICOUT} [={field
              (field[;field[;...] ] ) } [,EXCLUDE] ]
```

Where:

field Is a single unsigned integer or a pair of unsigned integers, representing column numbers within the records in the form *a*; *a*, *b*; or *a:b*. (The first column of a record is column #1.)

In the form *a*

a is the starting column

the last column in the record is the ending column

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 identified fields are not to be translated. All other fields are translated.

Operation

In the following example you have a tape file of employee information. The name and address occupy the first 75 character positions and are in BCDIC. The social security number takes up the next five character positions as a packed decimal. You can convert the BCDIC fields to ASCII, and retain the packed decimal file unchanged, by using this command:

```
FCOPY FROM=*TAPEFILE;TO=*EMPDAT;NEW;BCDICIN=1,75
```

The *field* and EXCLUDE parameters let you translate specific data in the file. If you do not specify fields or EXCLUDE, the system converts all data in each record in the file to ASCII. If you specify *fields* without specifying EXCLUDE, the system converts only the data within those fields. If you specify *fields* and EXCLUDE, the system converts all data in each record *except* for data in the specified fields.

Restrictions

You cannot use BCDICOUT with CHAR, CLEAR, KANA, OCTAL, HEXO, or HEX.

Notes

When you use BCDICIN or BCDICOUT with the SUBSET function, the system makes conversions first.

Five BCDIC character codes can be converted to ASCII in either of two ways. FCOPY translates them as shown in table 5-1:

Table 5-1. BCDIC to ASCII Conversion

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

Refer to appendix C for a complete list of conversion codes.

The maximum number of fields you can specify is 255.

The translation function does not apply to user labels. If a file contains user labels, FCOPY does not translate them.

Examples

The example below copies FILE1 to FILE2, converting all characters from BCDIC to ASCII, except for those in columns 1 through 5 and 10 through 30:

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

The example below copies FILE1 to FILE2, converting only characters in columns 1 through 5 and 10 through 30 from ASCII to BCDIC:

```
>FROM=FILE1;TO=FILE2; BCDICOUT=(1,5;10:30)
```

CCTL/NOCCTL

CCTL designates the first character of each record in the *fromfile* as a carriage control character in the *tofile*. NOCCTL specifies that the first character of each record in the *fromfile* is not to be used as a carriage control character in the *tofile*.

Syntax

```
;{CCTL
  NOCCTL}
```

Operation

The effect of CCTL or NOCCTL depends upon whether or not the file equations or disk labels of the files you are copying specify carriage control. If you do not use either function, FCOPY treats carriage control characters as shown in table 5-2.

Table 5-2. Copying Files without Using CCTL or NOCCTL

CCTL specified in the file equation or disk label for		Default Effect
fromfile	*tofile*	
NO	NO	The entire record of the *fromfile* is copied to the *tofile*.
NO	YES	The entire record of the *fromfile* is copied to the *tofile*; a single spacing character is supplied.
YES	NO	The entire record of the *fromfile*, including the carriage control character, is copied into the *tofile* as data.
YES	YES	The entire record of the *fromfile* is copied into the *tofile*. The carriage control character for the *fromfile* becomes the carriage control character for the *tofile*.

Table 5-3 shows the effect of using the CCTL function with files that have CCTL or NOCCTL specified. When an equation or disk file label for the *tofile* specifies NOCCTL (no carriage control) and an FCOPY command specifies CCTL, the NOCCTL specification in the file equation or disk label overrides the FCOPY command.

NOTE Asterisks (***) show where a file equation or disk label overrides the FCOPY command in tables 5-3 and 5-4.

Table 5-3. Copying Files Using CCTL

CCTL specified in the file equation or disk label for		Effect
fromfile	*tofile*	
NO	NO***	The entire record of the *fromfile* is copied to the *tofile*. No carriage control characters are supplied.
NO	YES	The entire record of the *fromfile* is copied to the *tofile*; the first character of the *fromfile* record is interpreted as a carriage control character, not as data.
YES	NO***	The entire record of the *fromfile*, including the carriage control character, is copied as data.
YES	YES	The entire record of the *fromfile* is copied into the *tofile*. The carriage control character for the *fromfile* becomes the carriage control character for the *tofile*.

Table 5-4 shows the effect of using NOCCTL.

Table 5-4. Copying Files Using NOCCTL

CCTL specified in the file equation or disk label for		Effect
fromfile	*tofile*	
NO	NO	The entire record of the *fromfile* is copied to the *tofile*. No carriage control character is supplied.
NO	YES***	The entire record of the *fromfile* is copied to the *tofile*. A single spacing carriage control character is supplied for each record of the *tofile*.
YES	NO	The entire record of the *fromfile*, including the carriage control character, is copied as data.
YES	YES***	The entire record of the *fromfile*, including the carriage control character of the *fromfile*, is copied into the *tofile*. A single spacing carriage control character is supplied for each *tofile* record.

Examples

ALPHA is a disk file, and its label specifies the CCTL option. BETA is a disk file created without CCTL.

The command below copies each record from BETA to ALPHA, interpreting the first data character in each record in BETA as the carriage control character:

```
>FROM=BETA;TO=ALPHA;CCTL
```

In the next example, the command copies each record from ALPHA to GAMMA, a new file, using the carriage control character in each record in ALPHA as the carriage control character in each record in GAMMA. File GAMMA is created with the CCTL attribute, as follows:

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

As we have seen above, ALPHA is a disk file that has CCTL specified in its label, and BETA is a disk file without CCTL. In the following example, the FCOPY command copies each record from ALPHA to BETA, including the carriage control character, as data.

```
>FROM=ALPHA;TO=BETA;NOCCTL
```

CHAR

Displays the contents of a file, record by record, in the form of character symbols in ASCII code. CHAR displays character codes not represented by symbols in ASCII code as decimal points.

Syntax

```
;CHAR [;{HEX
        HEXO
        OCTAL} ] [;NORECNUM] [;TITLE=title] [;LANG=language]
```

Where:

OCTAL	Displays octal codes. Refer to the description of OCTAL in this section.
HEX	Displays hexadecimal codes. Refer to the HEX description in this section.
HEXO	Displays the data in hexadecimal and the record number in octal. Refer to the HEXO description in this section.
NORECNUM	Omits file identification information, record numbers, and word offset numbers from the display.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters, but may not be broken to be continued on the following line.
language	Is the name or number of the language in which to display character symbols. FCOPY uses the character definition table associated with the specified language. The default is NATIVE-3000. For more information, refer to the *Native Language Support Reference Manual* (32414-90001) for MPE V/E or the *Native Language Programmer's Guide* (32650-90022) for MPE XL.

File Attributes

The *fromfile* must have read access, and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that will ultimately print the data, and must be equal to or greater than 60 bytes.

Operation

You can combine the CHAR function with the OCTAL, HEX, or HEXO functions, one at a time, to display numerical codes as well as character symbols. You may want to display numerical codes along with character symbols when, for example, the output device to which you are copying automatically upshifts lowercase characters, or when you encounter a disk or magnetic tape file with unknown contents (refer to "Notes" below).

The *tofile* can be a printer, terminal, or an intermediate disk file. If the *tofile* is a disk file, its records must be at least 60 bytes in length.

If you do not specify NORECNUM and words are repeated in the file, FCOPY suppresses the duplicate lines and displays a message in the form SAME TO XXXX-1, where XXXX is in octal. If the entire record consists of the same character, FCOPY displays at least one line of the repeated character before printing the SAME TO message, as shown in the following example. However, if all the characters are blanks in an ASCII file, or zeros in a binary file, FCOPY displays only the SAME TO message.

Restrictions

You cannot combine CHAR with VERIFY, NEW, EBCDICOUT, EBCDIKOUT, or BCDICOUT.

Notes

The FCOPY display functions that show both numeric codes and character symbols are particularly useful because they let you examine the contents of a file either at a terminal or in a printer listing.

Some devices are capable of displaying only uppercase Roman alphabetic characters. When you copy to such a device, the device controller automatically converts lowercase characters to uppercase. In such instances, you may want to combine character and numerical display functions to display the original character codes in addition to the character symbols.

Example

In the following example, the CHAR and OCTAL functions have been combined to show the correspondence between the two forms.

This combined command displays the contents of a file. CHAR suppresses terminal recognition of such things as control sequences imbedded in the file to which the terminal would otherwise be sensitive. Escape codes and other nongraphic control characters are shown as dots, allowing you to see the exact position within the file of all the displayable characters. Refer to the CLEAR function for relevant information.

For example,

```
FCOPY FROM=DISPL;TO=;OCTAL;CHAR
HP32212A.03.24 FILE COPIER (C) HEWLETT-PACKARD CO. 1984

DISPL RECORD 0 (%0, #0)

00000: 047157 073440 065563 020164 064145 020164 064555 062440 Now is the time
00010: 063157 071040 060554 066040 063557 067544 020155 062556 for all good men
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030: SAME:TO 000044-1

DISPL RECORD 1 (%1, #1)

00000: 052157 020143 067555 062440 072157 020164 064145 020141 to come to the
00010: 064544 020157 063040 072150 062440 070141 071164 074456 aid of the party.
00020: 020040 020040 020040 020040 020040 020040 020040 020040
```

```
00030: SAME: TO 000044-1

DISPL RECORD 2 (%2, #2)


00000: 015446 062104 052150 064563 020154 064556 062440 064563 .&dBThis line is|
00010: 020165 067144 062562 066151 067145 062056 015446 062100 underlined..&d@
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030: SAME: TO 000044-1
EOF FOUND IN FROMFILE AFTER RECORD 2

3 RECORDS PROCESSED *** 0 ERRORS

END OF SUBSYSTEM
```

CLEAR

Displays the contents of a file, record by record, in the form of character symbols for all codes in the file. The graphic representation of each code depends upon how the displaying device interprets the code: whether as a command, a displayable character, or a nondisplayable character. If an output device interprets a code as a command, it displays nothing unless you have entered a "display functions" mode at a terminal that has such a mode.

Syntax

```
;CLEAR [;{HEX
         HEXO
         OCTAL}] [;NORECNUM] [;TITLE=title]
```

Where:

OCTAL	Displays octal codes. Refer to the OCTAL description in this section.
HEX	Displays hexadecimal codes. Refer to the HEX description in this section.
HEXO	Displays the data in hexadecimal and the record number in octal. Refer to the HEXO description in this section.
NORECNUM	Omits file identification information, record numbers, and word offset numbers from the display.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attributes

The *fromfile* must have read access, and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST, but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that will ultimately print the data, and must be equal to or greater than 60 bytes.

Operation

Use CLEAR to display files that contain codes other than ASCII. Unlike CHAR, CLEAR does not replace character codes with decimal points.

You can combine the CLEAR function with the OCTAL, HEX, or HEXO functions, one at a time, to display numerical codes as well as character symbols. You may want to display numerical codes along with character symbols when, for example, the output device to which you are copying automatically upshifts lowercase characters (refer to "Notes" below).

The *tofile* can be a printer, terminal, or an intermediate disk file. If the *tofile* is a disk file, it must be at least 60 bytes in length.

If you do not specify NORECNUM and there are words repeated in the file, FCOPY suppresses the duplicate lines and displays a message in the form SAME TO XXXX-1, where XXXX is in octal. This is shown in the example below. If the entire record contains the same character, FCOPY displays at least one line of the repeated character before printing the SAME TO message. However, if all the characters are blanks in an ASCII file, or zeros in a binary file, FCOPY displays only the SAME TO message.

Restrictions

You cannot combine CLEAR with VERIFY, NEW, EBCDICOUT, EBCDIKOUT, or BCDICOUT.

Notes

Some devices are capable of displaying only uppercase Roman alphabetic characters. When you copy to such a device, the device controller automatically converts lowercase characters to uppercase. In such instances, you may want to combine character and numerical display functions to display the original character codes in addition to the character symbols.

Example

In the following example, the CLEAR and OCTAL functions have been combined to show the correspondence between the two forms.

This combined function displays the contents of a file where the effects of terminal control sequences need to be seen. It can be used to display data on a terminal so that video enhancements, cursor positioning, and other control functions are made visible. See also the CHAR function.

For example,

```
FCOPY FROM=DISPL;TO=;OCTAL;CLEAR

HP32212A.03.24 FILE COPIER (C) HEWLETT-PACKARD CO. 1984
DISPL RECORD 0 (%0, #0)

00000: 047157 073440 065563 020164 064145 020164 064555 062440 Now is the time
00010: 063157 071040 060554 066040 063557 067544 020155 062556 for all good men
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030:  SAME: TO 000044-1

DISPL RECORD 1 (%1, #1)

00000: 052157 020143 067555 062440 072157 020164 064145 020141 To come to the
00010: 064544 020157 063040 072150 062440 070141 071164 074456 aid of the party.
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030:  SAME: TO 000044-1

DISPL RECORD 2 (%2,  #2)

00000: 015446 062104 052150 064563 020154 064556 062440 064563 This line is
```

```
00010: 020165 067144 062562 066151 067145 062056 015446 062100 underlined..
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030:  SAME: TO 000044-1
EOF FOUND IN FROMFILE AFTER RECORD 2

3 RECORDS PROCESSED *** 0 ERRORS

END OF SUBSYSTEM
```

COMPARE

Compares the contents of the *fromfile* with the contents of the *tofile*, record by record, without changing either file.

Syntax

```
;COMPARE [=number-of-errors]
```

Where:

number-of-errors Is an integer designating the number of mismatches after which FCOPY should terminate. The default value is 1.

File Attributes

The *fromfile* and the *tofile* must both have read access. The *tofile* record size must be equal to that of the *fromfile*, or else both files must have variable-length records.

Operation

The two files may reside on any device, but their record characteristics (record size, record format, and record type) must be identical.

Each time FCOPY detects an error during file comparison, it prints a message on the standard listing device (your terminal during a session or a printer during a job). The message shows the number of the record in the *fromfile* in which the error occurred.

Restrictions

The only FCOPY functions that can be used with COMPARE are SUBSET, SKIPEOF, and IGNERR.

Examples

In the example below, FCOPY compares the contents of the disk file FILEA with the contents of the disk file DUP1. If it finds a mismatch, FCOPY prints a message on the standard listing device (your terminal during a session, or a printer during a job) and terminates the operation.

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

In the next example, FCOPY compares the contents of the first file on the magnetic tape TAPE4 with the contents of the disk file DUP2. Each time it finds a mismatch, FCOPY prints a message on the standard listing device. FCOPY terminates the operation after the fifth mismatch.

```
>FROM=TAPE4;TO=DUP2;COMPARE=5
```

COPYACD

Copies the access control definition (ACD) associated with a file when the file is being copied.

NOTE COPYACD applies only to MPE V/E Delta 4 and subsequent releases. COPYACD is the default on MPE XL release 3.0 or later.

Syntax

```
;COPYACD
```

Operation

If the *fromfile* has an access control definition (ACD) and the user is authorized to copy that file, both the *fromfile* and the ACD are copied to the *tofile*. If the *fromfile* does not have an ACD or the user is not authorized to copy the file, the user is given the option to copy the data only.

NOTE On MPE XL release 3.0 or later if COPYACD has not been entered, error messages 915 and 916 will not be displayed. To ensure that ACD status has successfully been duplicated, enter the COPYACD parameter to copy the files.

Restrictions

You cannot use COPYACD with the COMPARE function.

Example

```
>FROM=SOURCEF;TO=TARGETF;COPYACD
```

DEBLOCK

Lets you deblock blocked records. For example, you can use DEBLOCK to make smaller records from larger ones or to copy files from systems with records that are not aligned on two-byte word boundaries.

Syntax

```
;DEBLOCK=logical-record-length
```

Where:

logical-record-length Is a signed integer representing the length of each logical record in the *tofile*. A positive integer represents the length in two-byte words, and a negative integer represents the length in bytes.

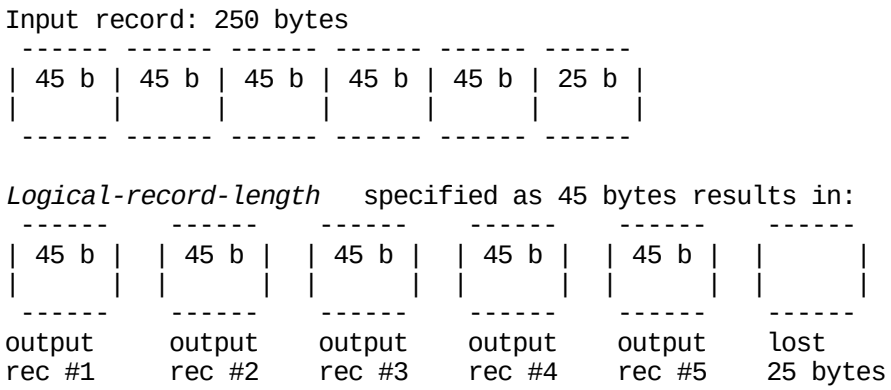
Operation

DEBLOCK is principally used for copying tapes from other systems to an HP 3000 system. For more information on copying foreign tapes, refer to chapter 3.

FCOPY assumes that each group of bytes or words you specify with DEBLOCK is a record. FCOPY copies the record to the *tofile* and then aligns it on word boundaries.

Be careful when specifying the *logical-record-length*. If the input record length is not a multiple of the *logical-record-length* you specify, any fractional remainder of the input record is lost. Figure 5-1 illustrates this principle.

Figure 5-1. Logical Record Length



If you deblock labeled tapes, you must specify "undefined" records in the FILE command describing the tape file. For example:

```
FILE TAPEBYTE;DEV=TAPE;LABEL=TLABEL;REC=-6000,1,U,ASCII
```

Example

The example below copies a file from tape to disk. The original tape file, TAPEBYTE, is a foreign tape that has records grouped in 79-byte blocks, and its blocking factor is 10. The FILE command in the example instructs FCOPY to treat TAPEFILE as if each block of the file were a single record, by defining the record length as the original record length times the blocking factor (790), and defining the new blocking factor as 1. The FCOPY command instructs FCOPY to copy the file, extracting 79-byte records from each block.

```
>:FILE TAPEBYTE;REC=-790,1,U,ASCII
>FROM=*TAPEBYTE;TO=DISC1;DEBLOCK=-79
```

EBCDICIN/EBCDICOUT

EBCDICIN translates code from EBCDIC to the character code specified in the translation table of the language you select. EBCDICOUT translates code from the character code specified in the translation table of the language you select to EBCDIC. When you do not specify a language, EBCDICIN translates code from EBCDIC to ASCII, and EBCDICOUT translates code from ASCII to EBCDIC.

Syntax

```
;{EBCDICIN
  EBCDICOUT}[={field
              (field[;field [;...] ])}[,EXCLUDE][;LANG=language] ]
```

Where:

field Is a single unsigned integer or a pair of unsigned integers representing column numbers within the records in the form *a*; *a*, *b*; or *a:b*. (The first column of a record is column #1.)

In the form *a*

a is the starting column

the last column in the record is the ending column

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, but all other fields will be converted.

language Is the name or number of a supported language. FCOPY uses the translation table associated with the specified language. The default is NATIVE-3000. When using a 16-bit character language, you get an error (#971).

Operation

Use EBCDICIN and EBCDICOUT to translate information from a different computer code system to the HP 3000 code system. For example, you would use EBCDICIN to translate information from IBM codes to HP 3000 codes, and EBCDICOUT to translate information from HP 3000 codes to IBM codes.

You can select any language installed on your system. The default is NATIVE-3000. For more information, refer to the *Native Language Support Reference Manual* (32414-90001) for MPE V/E or the *Native Language Programmer's Guide* (32650-90022) for MPE XL.

The *field* and EXCLUDE parameters let you select the portion of the file that you want to translate. If you do not specify *fields* or EXCLUDE, the system converts all data within each record in the file. If you specify *fields* without specifying EXCLUDE, the system converts only the data within those fields. If you specify *fields* and EXCLUDE, the system converts all data in each record except for data within the specified fields.

Restrictions

You cannot use EBCDICOUT with the CHAR, CLEAR, KANA, OCTAL, HEX, or HEXO functions.

Notes

When you use EBCDICIN or EBCDICOUT with the SUBSET function, the system makes conversions first.

Refer to appendix C for a complete list of conversion codes.

The maximum number of fields you can specify is 255.

FCOPY does not translate user labels when translating a file.

Examples

The example below copies columns 3 through 7 and 14 through 27 of each record in the file *TAPE1 to the file TEST1, converting characters from EBCDIC to ASCII. It uses the default (NATIVE-3000) translation table:

```
>FROM=*TAPE1;TO=TEST1;EBCDICIN=(3:7;14:27)
```

The second example converts MYGEBCFL to MYROM8FL, converting characters from German EBCDIC to Roman8:

```
>FROM=MYGEBCFL;TO=MYROM8FL;EBCDICIN;LANG=GERMAN
```

The example below copies columns 3 through 7 and 14 through 27 of each record in the file TEST1 to the file *TAPE1, converting characters from ASCII to EBCDIC. It uses the default (NATIVE-3000) translation table:

```
>FROM=TEST1;TO=*TAPE1;EBCDICOUT=(3:7;14:27)
```

The next example converts MYROM8FL to MYGEBCFL, converting characters from Roman8 to German EBCDIC:

```
>FROM=MYROM8FL;TO=MYGEBCFL;EBCDICOUT;LANG=GERMAN
```

EBCDIKIN/EBCDIKOUT

EBCDIKIN translates code from EBCDIK (IBM Standard) to JIS (Japanese Industrial Standard). EBCDIKOUT translates code from JIS to EBCDIK.

Syntax

```
;{EBCDIKIN
  EBCDIKOUT} [={field
                (field[;field[;...] ]) }[,EXCLUDE] ]
```

Where:

field Is a single unsigned integer, or a pair of unsigned integers representing column numbers within the records in the form *a*; *a*, *b*; or *a*:*b*. (The first column of a record is column #1.)

In the form *a*

a is the starting column

the last column in the record is the ending column

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 translated.

Operation

These functions support the Katakana language. Use EBCDIKIN and EBCDIKOUT to translate information between EBCDIC and the HP KANA8 character sets. The mapping between these character sets is defined by JIS and by IBM. For more information, refer to the *Native Language Support Reference Manual* (32414-90001) for MPE V/E or the *Native Language Programmer's Guide* (32650-90022) for MPE XL. Use these functions when, for example, you copy files to magnetic tapes in order to transfer them from one system to another.

The *field* and EXCLUDE parameters let you select a portion of the file to be translated. If you do not specify *fields* or EXCLUDE, the system converts all data within each record in the file to EBCDIK. If you specify *fields* without specifying EXCLUDE, the system converts only the data within those fields. If you specify *fields* and EXCLUDE, the system converts all data in each record except for data within the specified fields.

Restrictions

You cannot combine EBCDIKOUT with the CHAR, CLEAR, KANA, OCTAL, HEX, or HEXO functions.

Notes

When you use EBCDIKIN with the SUBSET function, the system makes the conversions first.

Refer to appendix C for a complete list of conversion codes.

The maximum number of fields you can specify is 255.

FCOPY does not translate user labels when translating a file.

Examples

The example below copies all records except columns 3 through 6 from the file CODFILE1 to the file CODFILE2, converting the columns from EBCDIK to JIS code.

```
>FROM=CODFILE1;TO=CODFILE2;EBCDIKIN=3:6,EXCLUDE
```

The following example copies records 1 through 30 of FILEA to FILEB, converting the data from JIS to EBCDIK code.

```
>FROM=FILEA;TO=FILEB;EBCDIKOUT=1:30
```

FILES

Lets you copy multiple files from labeled and unlabeled magnetic tapes, serial disks, and cartridge tapes. FCOPY copies only one file if you do not use the FILES function.

Syntax

```
;FILES={number-of-files
       ALL             }
```

Where:

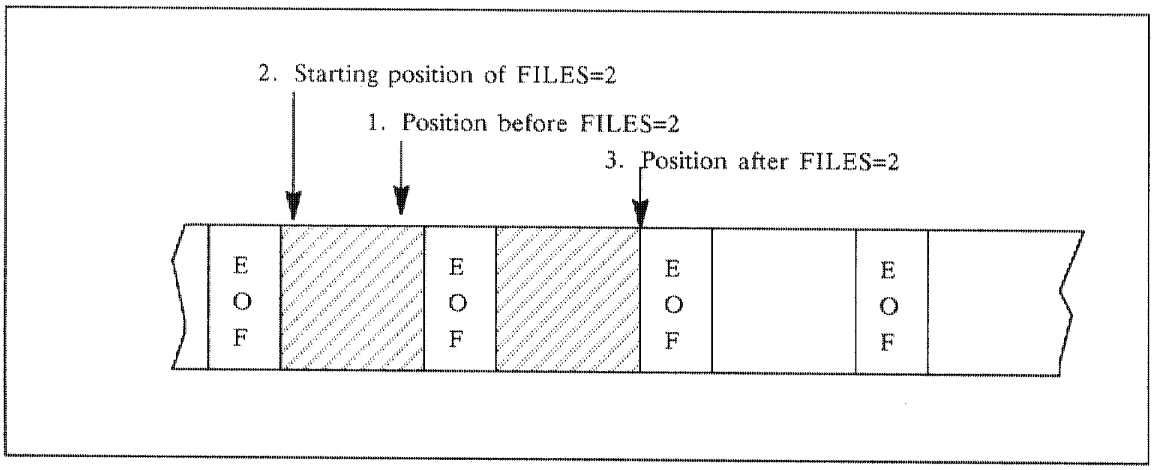
number-of-files Is an integer specifying the number of files to be copied.

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

Operation

Before determining which files to copy, FCOPY positions the tape at the beginning of the current file. Figure 5-2 illustrates an example. Before you issue the FCOPY command with FILES=2, the tape is positioned at the end of file 1. When you issue the FCOPY command, FCOPY positions the tape at the beginning of file 1 and copies two files. In figure 5-2, the shaded files represent the files copied:

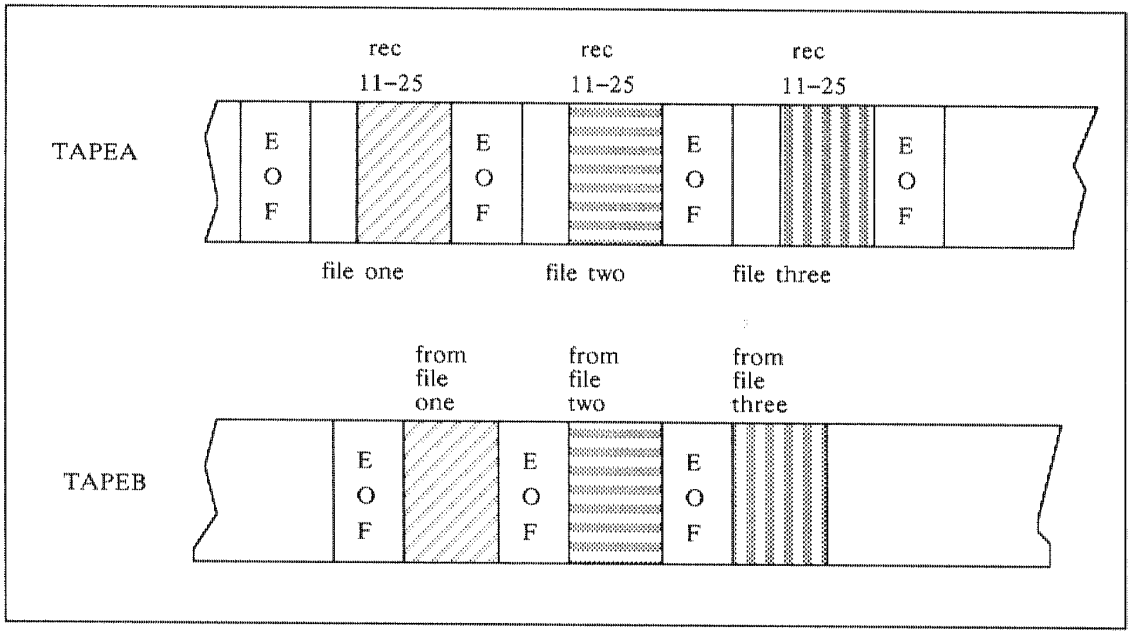
Figure 5-2. Tape Position



If you use the FILES function together with the SUBSET function, FCOPY assumes that the SUBSET function applies to every file to be copied. For example, the command below produces the results illustrated in figure 5-3.

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

Figure 5-3. Copying Subsets of Files to Tape



If you use the FILES function together with the SKIPEOF function, FCOPY positions the tape as specified with SKIPEOF first.

If the *fromfile* is an unlabeled tape, there is no standard that defines the mark used at the end of the last file on the tape. If you copy such a tape with FILES=ALL, the copy proceeds until an error occurs from trying to read unrecorded media.

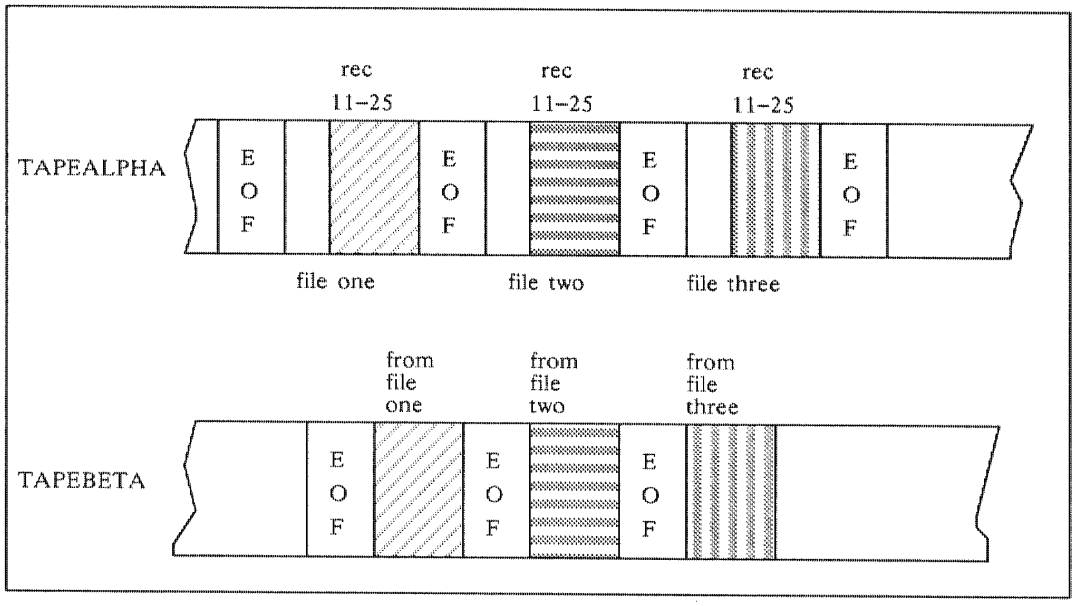
On a labeled tape, two consecutive tapemarks following the trailer labels denote the end of the recorded media. In this case, the copy process ends when the consecutive tapemarks are encountered.

Copying Files on Labeled Tape

Unless you want to copy the same labeled tape file repeatedly, use NEXT as the sequence option for the LABEL parameter in the *fromfile* tape's file equation. For example, the commands below copy subsets of three files from labeled tape TAPEALFA to TAPEBETA, as shown in figure 5-4.

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

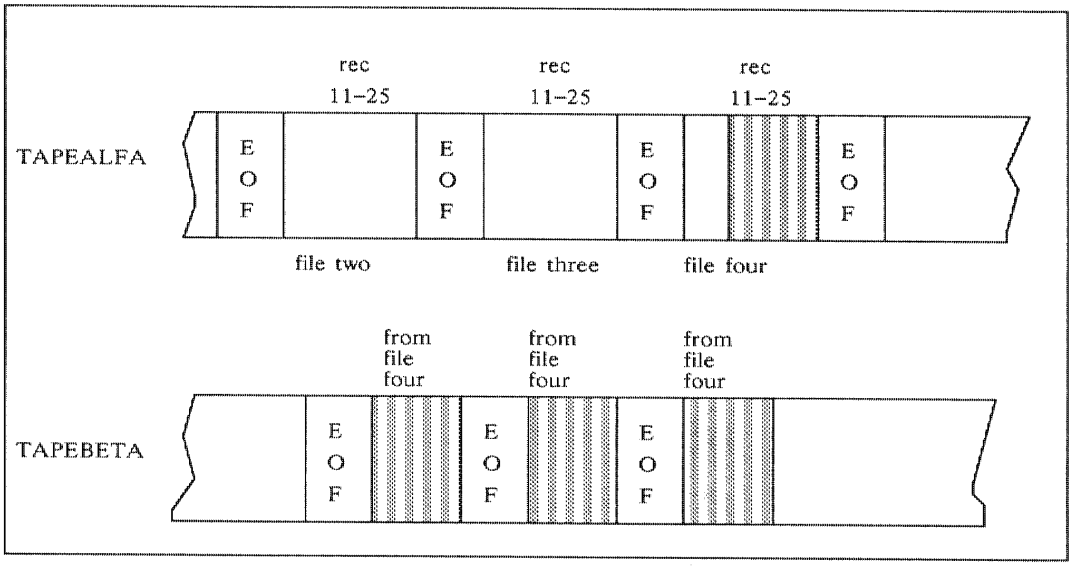
Figure 5-4. Copying Subsets of Files to Labeled Tape



If you specify a particular file as the sequence option for the labeled tape, FCOPY copies data from that file to all the new files. For example, the commands below produce the result shown in figure 5-5:

```
>:FILE TAPEALFA;LABEL=,,,4
>FROM=*TAPEALFA;TO=TAPEBETA;FILES=3;SUBSET=11:25
```

Figure 5-5. Copying from a Single Labeled Tape File



For more information on the sequence options of the FILE command, refer to the FILE command in the *MPE V/E Commands Reference Manual* (32033-90006) or the *MPE XL Commands Reference Manual* (32650-90003). Refer to chapter 3 for examples of using FCOPY with labeled tapes.

Examples

In the example below, FCOPY copies four files, starting with the first file, from magnetic tape *TM1 to magnetic tape *TM7:

```
>FROM=*TM1;TO=*TM7;FILES=4
```

The second example combines the SKIPEOF and FILES functions. FCOPY positions the unlabeled tape file *TM5 at its third file, positions the unlabeled tape file *TM2 four files forward from its current position, and then copies two files (files three and four) from *TM5 to *TM2:

```
>FROM=*TM5;TO=*TM2;SKIPEOF=3,+4;FILES=2
```

In the final example, FCOPY copies all of the files on *ME to the files of *YOU:

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

HEX

Displays the contents of a file, record by record, as character code numbers in hexadecimal form.

Syntax

```
;HEX [;{CHAR
       CLEAR
       KANA} ] [;NORECNUM] [;TITLE=title]
```

Where:

CHAR	Displays character symbols in ASCII code. Refer to the CHAR description in this chapter.
CLEAR	Displays all character symbols. Refer to the CLEAR description in this chapter.
KANA	Displays JIS character symbols. Refer to the KANA description in this chapter.
NORECNUM	Omits file identification information, record numbers, and word offset numbers.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attributes

The *fromfile* must have read access, and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST, but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that will ultimately print the data, and must be equal to or greater than 60 bytes.

Operation

You can combine the HEX function with the CHAR, CLEAR, or KANA functions, one at a time, to display character symbols as well as numerical codes.

The *tofile* can be a printer, terminal, or an intermediate disk file. If the *tofile* is a disk file, it must be at least 60 bytes in length.

Example

In the example below, the file TEXT3.PUB.ACCOUNT is an ASCII file created by using EDIT/3000. The hexadecimal word number at the beginning of each line of data specifies the sequential number within the record of the first word of data in the line.

The example displays the file TEXT3 in both hexadecimal and character format. Since it does not specify a language for the CHAR function, FCOPY uses the default, NATIVE-3000. (Refer to the CHAR description in this chapter.) FCOPY prints the title line described by the *title* parameter.

```
>FROM=TEXT3;TO=*LP;HEX;CHAR; &
TITLE="EXAMPLE OF CHAR/HEX DISPLAY"

EXAMPLE OF CHAR/HEX DISPLAY

TEXT3.PUB.ACCOUNT          RECORD       0 (%0)


0000:  5448  4953  2049  5320  4C49  4E45  2031  2E20  THIS IS LINE 1.
0008:  2054  4845  2038  2D44  4947  4954  204E  554D  THE 8-DIGIT NUM
0010:  4245  5253  2041  5420  5448  4520  5249  4748  BERS AT THE RIGH
0018:  5420  4152  4520  5448  4520  2020  2020  2020  T ARE THE
0020:  2020  2020  2020  2020  3030  3030  3130  3030    00001000
```

HEXO

Displays the contents of a file, record by record, in the form of character code numbers. The data is in hexadecimal form and the record number is in octal form.

Syntax

```
;HEXO [;{CHAR
        CLEAR
        KANA}] [;NORECNUM] [;TITLE=title]
```

Where:

CHAR	Displays character symbols in ASCII code. Refer to the CHAR description in this chapter.
CLEAR	Displays all character symbols. Refer to the CLEAR description in this chapter.
KANA	Displays JIS character symbols. Refer to the KANA description in this chapter.
NORECNUM	Omits file identification information, record numbers, and word offset numbers.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attributes

The *fromfile* must have read access, and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST, but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that will ultimately print the data and must be equal to or greater than 60 bytes.

Operation

You can combine the HEXO function with the CHAR, CLEAR, or KANA functions, one at a time, to display character symbols, as well as numerical codes.

The *tofile* can be a printer, terminal, or an intermediate disk file. If the *tofile* is a disk file, it must be at least 60 bytes in length.

Example

In the example below, file TEXT3.PUB.ACCOUNT is an ASCII file created by using EDIT/3000. The octal word number at the beginning of each line of data specifies the sequential number within the record of the first word of data in the line.

The example displays the file TEXT3 in both hexadecimal and character format. Because it does not specify a language for the CHAR function, FCOPY uses the default, NATIVE-3000 (refer to the CHAR description in this chapter). FCOPY prints the title as it appears in the title parameter.

```
>FROM=TEXT3;TO=*LP;HEXO;CHAR;&
TITLE="EXAMPLE OF CHAR/HEX DISPLAY"

EXAMPLE OF CHAR/HEX DISPLAY

TEXT3.PUB.ACCOUNT    RECORD     0 (%0)

0000:   5448  4953  2049  5320  4C49  4E45  2031  2E20  THIS IS LINE 1.
0010:   2054  4845  2038  2D44  4947  4954  204E  554D  THE 8-DIGIT NUM
0020:   4245  5253  2041  5420  5448  4520  5249  4748  BERS AT THE RIGH
0030:   5420  4152  4520  5448  4520  2020  2020  2020  T ARE THE
0040:   2020  2020  2020  2020  3030  3030  3130  3030  00001000
```

IGNERR

Bypasses errors in a magnetic tape *fromfile* and counts ignored errors up to a specified maximum.

Syntax

```
;IGNERR [=number-of-errors]
```

Where:

number-of-errors Is an integer specifying the maximum number of errors to be ignored. If the number of errors exceeds the maximum, the operation terminates.

File Attributes

The *fromfile* must be a magnetic tape unit with read access, and the *tofile* must have write access.

Operation

You can use IGNERR only with magnetic tape *fromfiles*. IGNERR ignores the following file system errors:

```
21    Data Parity Error
26    Transmission Error
27    I/O Timeout Error
38    Tape Parity Error
```

Errors other than these cause the operation to terminate.

Example

In the following example, the system ignores up to five errors while copying the first file on the magnetic tape unit *TAPE to the disk file FILE3. The operation terminates if FCOPY encounters a sixth error:

```
>FROM=*TAPE;TP=FILE3;SUBSET;IGNERR=5
```

KANA

Displays the contents of a file, record by record, in the form of JIS character symbols. KANA displays symbols not represented by characters in JIS code as decimal points.

Syntax

```
;KANA [;{HEX
        HEXO
        OCTAL}] [;NORECNUM] [;TITLE=title]
```

Where:

OCTAL	Displays octal codes. Refer to the OCTAL description in this chapter.
HEX	Displays hexadecimal codes. Refer to the HEX description in this chapter.
HEXO	Displays the data in hexadecimal and the record number in octal. Refer to the HEXO description in this chapter.
NORECNUM	Omits file identification information, record numbers, and word offset numbers.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters but may not be broken to be continued on the following line.

File Attributes

The *fromfile* must have read access and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST, but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that ultimately prints the data, and must be equal to or greater than 60 bytes.

Operation

You can combine the KANA function with the OCTAL, HEX, or HEXO functions, one at a time, to display numerical codes as well as character symbols. You may want to display numerical codes along with character symbols when, for example, the output device to which you are copying automatically upshifts lowercase characters (refer to "Notes" below).

The *tofile* can be a printer, a terminal, or an intermediate disk file. If the *tofile* is a disk file, it must be at least 60 bytes in length.

If you do not specify NORECNUM and there are words repeated in the file, FCOPY suppresses the duplicate lines and displays a message in the form SAME TO XXXX-1, where XXXX is in octal. If the entire record consists of the same character, FCOPY displays at least one line of the repeated character before printing the SAME TO message. However, if all the characters are blanks in an ASCII file or zeros in a binary file, FCOPY displays only the SAME TO message.

KEY

Lets you choose a key sequence in which to copy KSAM files. The KEY function works with KSAM V/E, CM KSAM, and KSAM XL *fromfiles*.

Syntax

```
;KEY [=character-location]
```

Where:

character-location Is a positive integer that identifies the key by the location of its starting character in the data file. The key may be a primary or an alternate key.

File Attributes

The *fromfile* must be a KSAM file.

Operation

Your system must have the KSAM subsystem in order for the KEY parameter to function. If you do not use the KEY function when copying a KSAM file, FCOPY copies the file in primary key sequence to another KSAM file.

If you specify zero as the character location (KEY=0), FCOPY copies the file in physical record sequence (the sequence in which the records are located in the file) rather than key sequence.

Restrictions

You cannot combine KEY with NOKSAM.

Notes

For a complete description of KSAM file characteristics and their operation with FCOPY, refer to the *KSAM/3000 Reference Manual* (30000-90079) and *Using KSAM XL*.

Examples

In the example below, FCOPY copies the file KSAM in key sequence (alphabetical or numerical) according to the key value starting in character location 21 (assuming that a key has been designated at character 21).

```
>FROM=KSAM;TO=ALPHA;KEY=21
```

The FCOPY command below copies the KSAM1 file in primary key sequence (the default) by omitting the KEY parameter:

```
>FROM=KSAM1;TO=KSAM2
```

The next example copies the KSAM1 file in chronological sequence, that is, the sequence in

which the records were written to the file:

```
>FROM=KSAM1;TO=DUPL1;KEY=0
```

NEW

Creates a new permanent disk file as the *tofile*.

Syntax

```
;NEW
```

Operation

The new file is created in the account and group specified at the start of your job or session (unless otherwise specified). Its name is the name specified with the TO parameter.

File Characteristics

The default characteristics of the new disk file are summarized in appendix B. If you want to predefine the characteristics of a new file, use the MPE FILE command to define them and then backreference that file in the FCOPY command.

Terminating or Interrupting an Operation

If the FCOPY operation containing the NEW function terminates because of an error, FCOPY discards the new file. If you interrupt the operation by pressing **CTRL** Y, FCOPY saves the new file.

Restrictions

You cannot combine NEW with CHAR, CLEAR, KANA, OCTAL, HEX, or HEXO.

Examples

In the following example, FCOPY creates a new disk file named NEWFILE in the user's account and group, and then copies the disk file OLDFILE into NEWFILE.

```
>FROM=OLDFILE;TO=NEWFILE;NEW
```

NOACD

Does not copy the access control definition (ACD) associated with a file when the file is being copied.

NOTE NOACD applies only to MPE XL release 3.0 and subsequent releases.

Syntax

```
;NOACD
```

Operation

If the *fromfile* has an access control definition (ACD) and the user is authorized to copy that file, only the *fromfile* is copied to the *tofile*.

NOTE On MPE XL release 3.0 or later, COPYACD is the default parameter. To keep the ACD from copying, NOACD must be used.

Restrictions

None.

Example

```
>FROM=SOURCEF;TO=TARGETF;NOACD
```

NOKSAM

Copies the data file of a KSAM V/E or CM KSAM file into another, non-KSAM file. This option is not valid for KSAM XL files.

Syntax

```
;NOKSAM
```

File Attributes

The *fromfile* must be a KSAM V/E or CM KSAM file. The *tofile* cannot be a new KSAM file.

Operation

Use the NOUSERLABELS option together with NOKSAM to prevent FCOPY from copying KSAM user labels.

Restrictions

You cannot use NOKSAM with the KEY function.

Notes

NOKSAM is not recognized by systems that do not contain the KSAM subsystem.

For a complete description of KSAM V/E and CM KSAM files and how they operate with FCOPY, refer to the *KSAM/3000 Reference Manual* (30000-90079).

Examples

The command below copies all records in the data file of KSAMFILE, in order, into FILEX:

```
>FROM=KSAMFILE;TO=FILEX;NOKSAM;NOUSERLABELS
```

The command below lists all deleted records, that is, records with the octal number 377 in column 1, in the KSAM file KSAMEX on the standard list device.

```
>FROM=KSAMEX;TO=;NOKSAM;SUBSET=#%377,%377#
```

NOUSERLABELS

Lets you omit user labels when copying from a tape or disk file to another file.

Syntax

```
;NOUSERLABELS
```

Operation

If you do not specify NOUSERLABELS and both the *fromfile* and *tofile* are labeled, FCOPY copies the *fromfile* labels to the *tofile* labels.

Use NOUSERLABELS when copying from tape to disk, or disk to tape, to prevent FCOPY from changing the user label. A disk file's user label is 256 bytes long, while a tape file's user label is 80 bytes long. If you omit NOUSERLABELS when copying from disk to tape, FCOPY copies only the first 80 bytes of the disk file's user label. If you omit NOUSERLABELS when copying from tape to disk, FCOPY pads the additional 176 bytes with zeros. FCOPY issues a warning message whenever you copy user labels from a disk file to a tape file or from tape to disk.

Examples

All the files in the examples below have at least one user label.

The command below copies the first file on TAPEA to a disk file named MYFILE1 without copying the user label of the tape file.

```
>FROM=*TAPEA;TO=MYFILE1;NOUSERLABELS
```

The command below copies the second file on TAPEA, including user labels, to the disk file MYFILE2. The last 176 bytes of MYFILE2's user label are zeros.

```
>FROM=*;TO=MYFILE2;SKIPEOF=2
```

The command below copies the first 80 bytes of the disk file DLBLD's user label to the tape file *TAPE1's user label and then copies the file.

```
>FROM=DLBLD;TO=*TAPE1
```

OCTAL

Displays the contents of a file, record by record, as character code numbers in octal form.

Syntax

```
;OCTAL [;{CHAR
         CLEAR
         KANA}] [;NORECNUM] [;TITLE=title]
```

Where:

CHAR	Displays character symbols in ASCII code. Refer to the CHAR description in this chapter.
CLEAR	Displays all character symbols. Refer to the CLEAR description in this chapter.
KANA	Displays JIS character symbols. Refer to the KANA description in this chapter.
NORECNUM	Omits file identification information, record numbers, and word offset numbers from the display.
title	Is a character string to be used as the title of the display. If the *tofile* is a printer, the title is printed at the top of each page. Otherwise, the title is written only once, at the beginning of the *tofile*. The title may contain up to 62 characters but may not be broken to be continued on the following line.

File Attributes

The *fromfile* must have read access and the *tofile* must have write access. The *tofile* is normally a list device such as $STDLIST, but it can also be another device for intermediate storage. The record size for an intermediate storage device should be the same as that of the list device that ultimately prints the data and must be greater than or equal to 60 bytes.

Operation

You can combine the OCTAL function with the CHAR, CLEAR, or KANA functions, one at a time, to display character symbols as well as numerical codes. The *tofile* can be a printer, terminal, or an intermediate disk file. If the *tofile* is a disk file, it must be at least 60 bytes in length.

Examples

In the examples below, the file TEXT3.PUB.ACCOUNT is an ASCII file created by using EDIT/3000. The octal word number at the beginning of each line of data specifies the sequential number within the record of the first word of data in the line.

The command below produces the following display on an 80-column line printer:

```
>FROM=TEXT3;TO=*LP;OCTAL

TEXT3.PUB.ACCOUNT RECORD 0 (%0)

00000: 052110  044523  020111  051440  046111  047105  020061 027040
00010: 020124  044105  020070  026504  044507  044524  020116 052515
00020: 041105  051123  020101  052040  052110  042440  051111 043510
00030: 052040  040522  042440  052110  042440  020040  020040 020040
00040: 020040  020040  020040  020040  030060  030060  030460 030006

TEXT3.PUB.ACCOUNT RECORD 1 (%1)

00000: 052105  054124  020105  042111  052117  051040  046111 047105
00010: 020116  052515  041105  051123  027040  020116  047524 042440
00020: 052110  040524  020124  044105  020122  042503  047522 042040
00030: 047125  046502  042522  051140  020040  020040  020040 020040
00040: 020040  020040  020040  020040  030060  030060  031060 030060

TEXT3.PUB.ACCOUNT RECORD 2 (%2)

00000: 051525  050120  046111  042504  020102  054440  043103 047520
00010: 054440  051524  040522  052040  040524  020132  042522 047440
00020: 053510  044514  042440  052110  042440  052105  054124 020105
00030: 042111  052117  051040  046111  047105  020040  020040 020040
00040: 020040  020040  020040  020040  030060  030060  031460 030060
```

The following example combines octal and character display.

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

The command produces the following display. The 8-digit numbers at the end of each record are text editor line numbers, starting at 1.

```
EXAMPLE OF CHAR/OCTAL DISPLAY

TEXT3.PUB.ACCOUNT RECORD 0 (%0)

00000: 052110 044523 020111 051440 046111 047105 020061 027040 THIS IS LINE 1
00010: 020124 044105 020070 028504 044507 044524 020116 052515 THE 8-DIGIT NUM
00020: 041105 051123 020101 052040 052110 042440 051111 043510 BERS AT THE RIGHT
00030: 052040 040522 042440 052110 042440 020040 020040 020040 ARE THE
00040: 020040 020040 020040 020040 030060 030060 030460 030060 00001000

TEXT3.PUB.ACCOUNT RECORD 1 (%1)

00000: 052105 054124 020105 042111 052117 051040 046111 047105 TEXT EDITOR LINE
00010: 020116 052515 041105 051123 027040 020116 047524 042440 NUMBERS. NOTE
00020: 052110 040524 020124 044105 020122 042503 047522 042040 THAT THE RECORD
00030: 047125 046502 042522 051140 020040 020040 020040 020040 NUMBERS
00040: 020040 020040 020040 020040 030060 030060 031060 030060 00002000

TEXT3.PUB.ACCOUNT RECORD 2 (%2)

00000: 051525 050120 046111 042504 020102 054440 043103 047520 SUPPLIED BY FCOP
00010: 054440 051524 040522 052040 040524 020132 042522 047440 Y START AT ZERO
00020: 053510 044514 042440 052110 042440 052105 054124 020105 WHILE THE TEXT
00030: 042111 052117 051040 046111 047105 020040 020040 020040 EDITOR LINE
00040: 020040 020040 020040 020040 030060 030060 031460 030060 00003000
```

SKIPEOF

Instructs FCOPY to skip end-of-file (EOF) markers on a serial storage device in order to position the device at the desired file before copying.

Syntax

```
;SKIPEOF=[{+
          -}from-eofs
          from-file-number][,{ {+
                              -}to-eofs
                              to-file-number}]
```

Where:

+/– Specifies forward (+) or reverse (–) movement.

from-eofs Is a positive integer specifying how many files to skip on the "from" device.

from-file-number Is an integer specifying an absolute file number on the "from" device. Absolute file numbers begin with 1.

to-eofs Is a positive integer specifying how many files to skip on the "to" device.

to-file-number Is an integer specifying an absolute file number on the "to" device.

File Attributes

The files that use the SKIPEOF function (either the *fromfile* or the *tofile*, or both) must reside on unlabeled magnetic tapes, serial disks, or cartridge tapes. The *fromfile* must have read access, and the *tofile* must have write access.

Operation

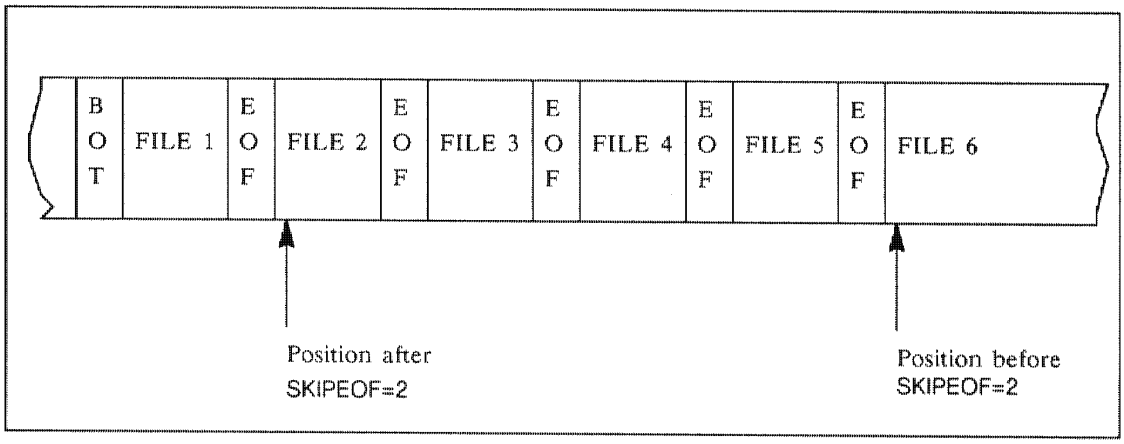
SKIPEOF lets you specify particular files on unlabeled magnetic tapes, cartridge tapes, and serial disks.

Absolute File Numbers

In the syntax description of the SKIPEOF function, *from-file-number* and *to-file-number* represent absolute file numbers. Absolute file numbers refer to specific files on a tape or serial disk. Tape and serial disk files are sequentially numbered, beginning with 1. You can use file numbers to identify the files you want to copy. For example, to copy from the fourth file on a tape, specify 4 as the *from-file-number*, SKIPEOF=4.

When you specify a *from-file-number* or a *to-file-number*, such as SKIPEOF=,2, the SKIPEOF function positions the tape at the beginning of the specified file. The tape's position before you issue the command does not affect the file copied. Figure 5-6 illustrates this process.

Figure 5-6. Skipping Absolute File Numbers

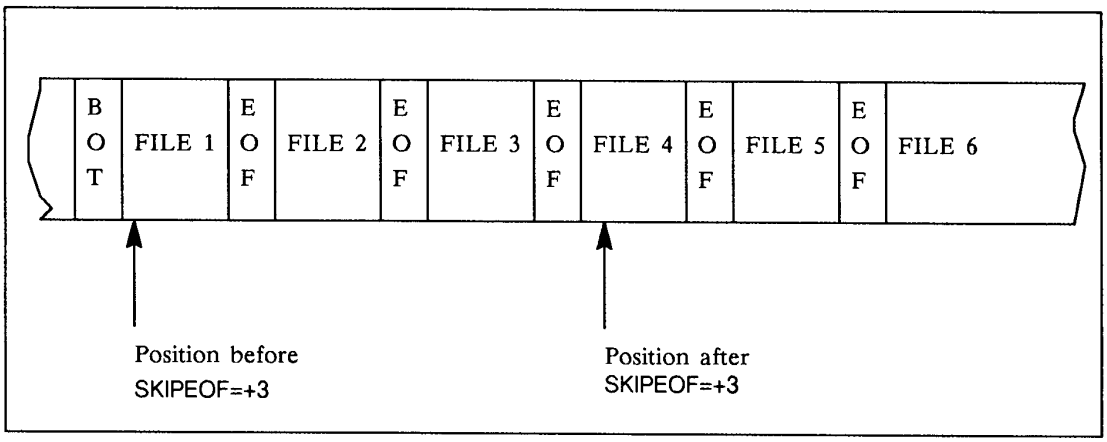


If you specify zero as a *from-file-number* or a *to-file-number*, FCOPY copies from or to the first file on the tape or serial disk.

Relative File Numbers

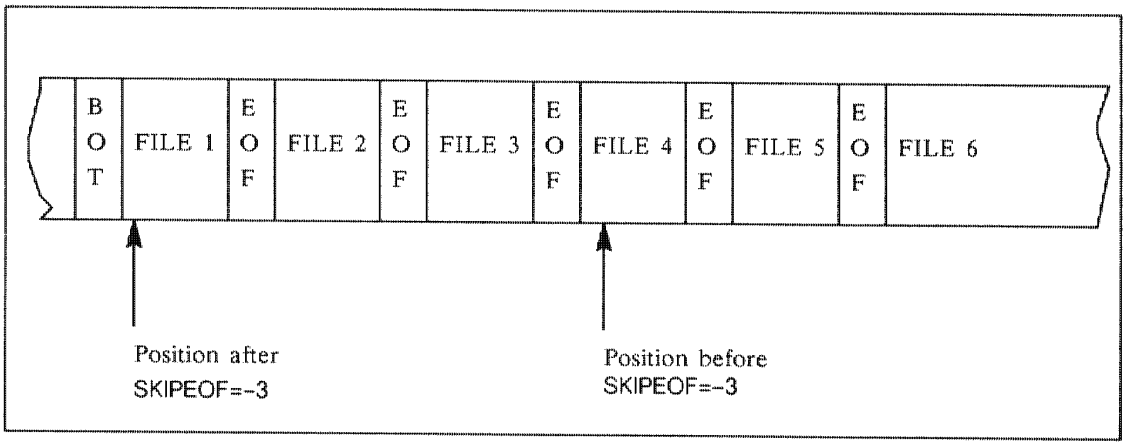
Both *from-eofs* and *to-eofs* represent the number of files to skip relative to the tape's current position. If you use a plus sign (+) to specify forward movement, the SKIPEOF function skips forward over the specified number of EOF marks. Figure 5-7 shows the effect of specifying SKIPEOF=+3.

Figure 5-7. SKIPEOF=+3



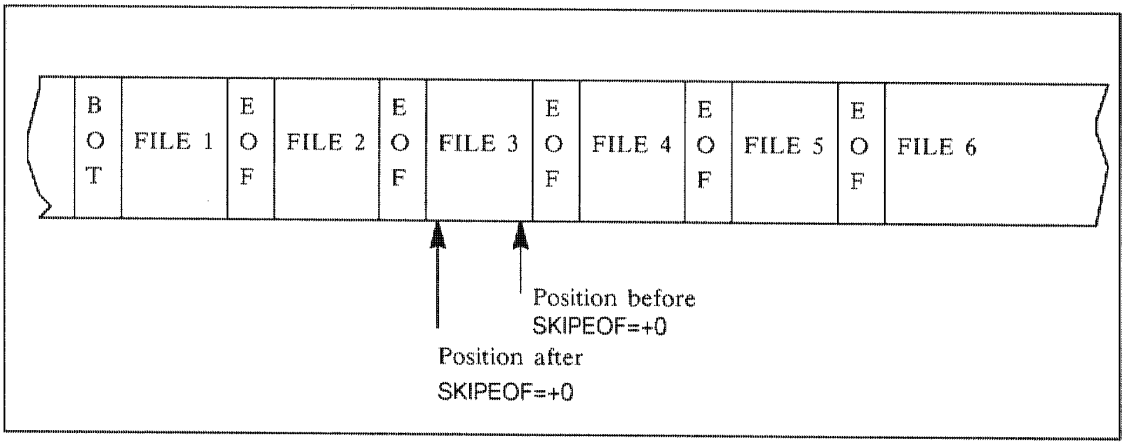
If you use a minus sign (–) to specify backward movement, the SKIPEOF function skips backward over the specified number of EOF marks plus 1, and then skips forward one EOF mark. This action positions the tape at the beginning of a file, rather than at the end of the preceding file. Figure 5-8 shows the effect of specifying SKIPEOF=–3.

Figure 5-8. SKIPEOF=–3



If you specify +0 or –0 for either of these parameters, SKIPEOF positions the tape at the beginning of the current file, as illustrated in figure 5-9.

Figure 5-9. SKIPEOF=+0



Examples

The command below instructs FCOPY to skip to the fourth file of *THISTAPE and to the fifth file of *THATTAPE before copying data from *THISTAPE to *THATTAPE:

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

In the example below, FCOPY copies information from the file three files ahead of *TAPES's current position to the seventh file of *TAPE2. If, for example, the current position of *TAPES is at the second file mark, FCOPY begins to copy from the sixth file. No matter

where skipping begins on *TAPE2, FCOPY copies data to its seventh file:

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

The FCOPY command below skips to the fifth file of *TAPEA and positions *TAPEB four files back from its current position. FCOPY copies file #5 of *TAPEA to the selected file on *TAPEB:

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

If you are copying several files from or to the same device, you can use an asterisk (*) to refer to the "from" or "to" device in the previous command. For example, the command below copies file #2 of *TAPES, the "from" device in the previous example, to file #8 of *TAPE2, the "to" device in the previous example:

```
>FROM=*;TO=*;SKIPEOF=2,8
```

In the example below, FCOPY does not move *MYTAPE from its current position, but positions $CTUL, a terminal cartridge tape unit (refer to chapter 6), at the ninth file. FCOPY then copies the file at the current position of *MYTAPE to the ninth file of the cartridge tape:

```
>FROM=*MYTAPE;TO=$CTUL;SKIPEOF=,9
```

SUBSET

Lets you copy only a specific portion, or subset, of a file. You can define the subset in one of two ways, either as all records with a certain character string or numeric pattern beginning in a specific column, or as a set or sets of continuous records.

Syntax

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

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, FCOPY assumes the default value, which is 1 (first character position).

EXCLUDE Specifies that the designated subset is to be excluded from the FCOPY operation, that is, all of the file *except* for the subset is to be copied.

range Can be in one of two forms: either [*starting-record-number*] [,*number-of-records*] or [*starting-record-number*] [:*last-record-number*]

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 number of the records in the subset. If omitted, it is assumed that the subset consists of all records through the end of the file.

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

Operation

You can define the subset in one of two ways, either as all records that have a certain character string or numeric pattern beginning in a specific column, or as a continuous set of records.

Defining Subsets from Character Strings and Pattern Lists

A character string describes a set of characters. Quotation marks delimit a character string. A character string can contain no more than 35 characters within the quotes, and it may not be continued from one line or record to the next. If you need to match quotation marks within the string, use two successive quotation marks, as each pair counts as one character.

A pattern list is a set of octal or decimal numbers, each representing one byte. A pattern list is delimited by number signs (#), and can contain no more than 35 integers. You can continue a pattern list from one line or record to the next, but you cannot specify the continuation between elements of an individual pattern. Each decimal integer must be within the range 0 through 255, and each octal integer must be within the range 0 through 377.

If the specified character string or pattern list does not exist entirely within a particular record in the *fromfile*, the record is not considered to be part of the defined subset.

Defining Subsets from Record Ranges

You can use record ranges to specify up to 255 subsets. FCOPY sequentially copies the subsets in the order you list them. They must be in numerical order, and they may not overlap.

Using the SUBSET Function Alone

You can use SUBSET alone to copy a disk file to a duplicate that uses less disk space. If you use SUBSET to copy data from a disk file that does not fill the space allocated to it to a new disk file, the resulting *tofile* is only as large as the actual contents of the *fromfile*. FCOPY does not copy the unused space. For example, the command below only copies 20 records from TEST1:

```
>FROM=TEST1;TO=TEST2;NEW;SUBSET
EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

If you exit FCOPY and return to MPE, you can observe the effect of the FCOPY command by using LISTF to list the characteristics of the two files and compare them. For example:

```
LISTF TEST@,2

ACCOUNT= SUBSYS GROUP= EDITOR

FILENAME CODE--------------LOGICAL RECORD----------SPACE----
           SIZE         TYPE   EOF   LIMIT   R/B   SECTORS #X  MX

TEST1         80B          FA     20    1023    1     128     1   8
TEST2         80B          FA     20    20      1     21      7   7
```

Copying User Labels Only

Use SUBSET=0,0 to copy just the user labels from a disk file. For example, the following command copies the user labels from the disk file LABELD to the labeled magnetic tape *TAPE.

```
>FROM=LABELD;TO=*TAPE;SUBSET=0,0
```

Magnetic Tape Subsets

FCOPY selects magnetic tape subsets by logical record number. If you need to use physical block subsets, you can temporarily define a logical record length equal to the physical record length. A subset may not extend over an EOF mark or a tape mark boundary on magnetic tape.

Using SUBSET with DEBLOCK

If you use SUBSET with DEBLOCK, FCOPY copies subsets from the deblocked records.

Examples

The example below copies all records in the disk file AUTOS that contain the character string "BLUE" starting in character position 17, to the disk file COMPANY:

```
>FROM=AUTOS;TO=COMPANY;SUBSET="BLUE",17
```

The following example copies all records in the file AUTOS that do not contain the character string "BLUE" in position 17, to the disk file RENTAL:

```
>FROM=AUTOS;TO=RENTAL;SUBSET="BLUE",17,EXCLUDE
```

You can also use the SUBSET function to specify lowercase alphabetic characters from a terminal that has only uppercase characters. For example, the command below copies all records in the disk file DATA that contain the lowercase alphabetic characters "data" (octal codes 144, 141, 164, and 141) in character positions 77 through 80 to the disk file SUBFILE3:

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

You might use decimal codes to specify subsets containing nonprinting characters. For example, the command below copies all records in the disk file DATA1 that contain the nonprinting control characters CR and LF (decimal codes 13 and 10) in character positions 71 and 72 to the disk file SUB1:

```
>FROM=DATA1;TO=SUB1;SUBSET=#13,10#,71
```

The example below copies 500 records, beginning with record 0 (the first), from the disk file MAIN to the disk file SUB1:

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

The next example copies records 500 through 1499 of the disk file MAIN into the disk file SUB2:

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

The following example copies all records from 1500 through the end of the file from the disk file MAIN to the disk file SUB3:

```
>FROM=MAIN;TO=SUB3;SUBSET=1500
```

You can copy several ranges of records as long as the ranges are sequentially ordered and do not overlap. For example, the command below copies records 0 through 11 and 30 through 74 from TAPEA to FILEONE:

```
>FROM=*TAPEA;TO=FILEONE; SUBSET=(0,12;30,45)
```

UPSHIFT

Converts lowercase Roman alphabetic characters to uppercase as part of the copying operation.

Syntax

```
;UPSHIFT [;LANG=language]
```

Where:

language Is the name or number of a supported language. FCOPY uses the translation associated with that language. The default is NATIVE-3000. For more information, refer to the *Native Language Support Reference Manual* (32414-90001) for MPE V/E or the *Native Language Programmer's Guide* (32650-90022) for MPE XL.

Operation

You use the *language* parameter to describe the language conventions you want to use when you convert characters. You can use any language installed on your system. If you do not specify a language, FCOPY uses the default, which is NATIVE-3000.

Most devices that can print only uppercase characters automatically upshift lowercase characters when you copy a file to them. If you are copying to such a device, you need not use the UPSHIFT function. If you are copying to a device that prints uppercase characters only but does not automatically upshift, then use UPSHIFT.

Examples

In the following example, lowercase Roman alphabetic characters in the disk file LOWER are converted to uppercase as they are copied to the disk file UPPER. The contents of LOWER, a NATIVE-3000 file, remain unchanged:

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

In the next example, FCOPY copies the contents of the LOW2 disk file to a line printer, converting all characters to uppercase according to German linguistic rules:

```
>FROM=LOW2;TO=LP;UPSHIFT;LANG=GERMAN
```

VERIFY

Compares the contents of the *tofile* with the contents of the *fromfile*, record by record, immediately after a copy operation. The two files must reside on either disk or magnetic tape and must have identical record characteristics, that is, record size, format (fixed, variable, or undefined), and type (ASCII or binary).

The VERIFY function is essentially the same as the COMPARE function, except that VERIFY is always performed in conjunction with the copying of a file, whereas COMPARE is not.

Syntax

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

Where:

number-of-errors Is an integer specifying the number of errors after which the operation terminates. If this parameter is omitted, the default is 1.

File Attributes

Both the *fromfile* and *tofile* must be a disk file or a magnetic tape file. The *fromfile* must have read access, and the *tofile* must have read/write access.

Operation

Each time FCOPY detects an error while verifying files, it prints a message on the standard listing device (your terminal in a session or a printer in a job). The message specifies the number of the *fromfile* record in which the error occurred.

Restrictions

You cannot use VERIFY with the CHAR, CLEAR, KANA, OCTAL, HEX, or HEXO functions.

Examples

In the example below, FCOPY copies the disk file OLDFILE to the disk file DCOPY and then compares the two files, record by record. If FCOPY finds an error, it prints a message on the standard listing device and terminates the operation.

```
>FROM=OLDFILE;TO=DCOPY;VERIFY
```

In the example below, FCOPY copies a file from 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.

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

6 Using FCOPY with Terminal Peripherals

Some HP 3000 terminals have their own peripheral devices. Terminal peripheral devices include cartridge tape units, mini-tape units, floppy disk drives, and printers. If you are using a terminal with such peripherals, you can use FCOPY to copy files between system devices and the terminal peripherals.

Using File Designators

You specify a peripheral device in an FCOPY command by using its file designator as the *fromfile* or *tofile*. Table 6-1 lists terminal peripherals, their corresponding terminals, and their file designators:

Table 6-1. Peripheral Devices and Designators

Peripheral Type	HP 3000 Terminals	File Designator
Left cartridge tape unit	HP 2640, HP 2641, HP 2645, HP 2647A, HP 2648, HP 2649	$CTUL
Right cartridge tape unit	HP 2640, HP 2641, HP 2645, HP 2647A, HP 2648, HP 2649	$CTUR
Line printer	HP 2621, HP 2622, HP 2626	$HARD

For example, you can use the following command to copy a disk file named SAMPLE to the left cartridge tape unit of an HP 2645 terminal:

```
>FROM=SAMPLE;TO=$CTUL
```

Terminal Settings

In order to copy to or from terminal peripherals, your terminal type must be set to 10 or 12 (depending on your terminal), message reception must be turned off, and the parity toggle switch must be set to NONE. FCOPY checks these settings before it begins to copy the files you specified in a command that references a terminal peripheral device. If your terminal is not set correctly, FCOPY prompts you to log on again and set the terminal type using the TERM parameter. For more information on terminal types, refer to the *System Operation and Resources Management Reference Manual* (32033-90005) for MPE V/E or *Controlling System Activity Reference Manual* (32650-90155) for MPE XL. If message reception is not turned off, FCOPY turns it off for you. Before copying the files, FCOPY prompts you to set the parity toggle switch.

Copying Between Terminal Peripheral Devices

Although you can use FCOPY to copy from one peripheral device to another, it is usually more efficient to use the local capabilities of the terminal. Refer to documentation describing your terminal for instructions for copying between local devices.

Examples

The following examples show how to use FCOPY to copy to and from terminal peripheral devices.

The command below copies the first file on the right cartridge tape unit of an HP 2645 terminal to the disk file named DFILE1:

```
>FROM=$CTUR;TO=DFILE1
```

The command in the next example displays the disk file named DFILE4 on the HP 2644A terminal line printer in both octal and character formats:

```
>FROM=DFILE4;TO=$HARD;OCTAL;CHAR
```

The command below copies the third file on the magnetic tape unit MAGTP to the left cartridge tape unit of an HP 2645 terminal:

```
>FROM=*MAGTP;TO=$CTUL;SKIPEOF=3;SUBSET
```

A FCOPY Messages

FCOPY displays three types of messages: error messages, warning messages, and status messages.

- Error messages terminate an FCOPY command during a session or the FCOPY subsystem during a job.
- Warning messages do not terminate FCOPY but, during a job, they set the job control word (JCW) to WARN.
- Status messages supply you with information while FCOPY is copying a file.

Display During Sessions

During a session, FCOPY displays messages and their reference numbers on your terminal. For example,

```
*54* SYNTAX ERROR: UNKNOWN OPTION
```

If the error involves a file, FCOPY asks you whether you want to display the file information. For example, if you try to copy a nonexistent file named THISFILE to a file named THATFILE, FCOPY displays the following message:

```
*105* CANNOT OPEN FROMFILE
DISPLAY FILE INFORMATION (Y OR N)?
```

Answer "yes" by pressing Y and then **Return**, or answer "no" by pressing N and **Return**. If you answer "yes", FCOPY displays the corresponding file system error message and file information. In the example above, the following display appears:

```
NONEXISTENT PERMANENT FILE (FSERR 52)
+-F-I-L-E---I-N-F-O-R-M-A-T-I-O-N---D-I-S-P-L-A-Y+
!                                                 !
!    ERROR NUMBER:     52 RESIDUE: 0             !
!    BLOCK NUMBER:     0     NUMREC: 0           !
+-------------------------------------------------+
```

Refer to the *MPE V Intrinsics Reference Manual* (32033-90007) or the *MPE XL Intrinsics Reference Manual* (32650-90028) for a complete description of the file information display.

Display During Jobs

During a job, FCOPY displays warning and error messages on the standard listing device. It does not give you the option of displaying file information. When FCOPY encounters a potential problem with a file, it assumes that the command was submitted as you intended and attempts to perform the requested operation.

Types of Message

This appendix lists, in numeric order, all FCOPY messages, their causes, and the actions you should take to correct them.

Three types of messages may be received: error, warning, and status messages.

- Errors encountered during a job stream cause FCOPY to terminate. However, only the command is terminated if the errors occur during a session.
- Warnings do not cause an FCOPY command to terminate. In a job, they set the JCW to WARN.
- Status messages do not indicate errors, but simply provide information about what is happening during the copying operation.

The following table lists the FCOPY error, warning, and status messages in numerical order, along with a brief definition of each message and the recommended corrective action, where applicable.

Error, Warning, and Status Messages

These are the messages you are likely to encounter when you use the FCOPY subsystem.

NOTE In the text,

```
* indicates a WARNING message
          ** indicates a STATUS message
```

Message	< CONTROL Y >
Cause	Acknowledges receipt of a **CTRL** Y entered during a session.
Action	None.

Message	READ ERROR FROM COMMAND INPUTFILE
Cause	An error occurred while reading an FCOPY command from $STDIN.
Action	In a job, resubmit the job. In a session, reenter the command.

Message	WRITE ERROR TO COMMAND LISTFILE
Cause	An error occurred while writing an FCOPY message to $STDLIST.
Action	It is likely that nothing serious occurred and all FCOPY operations were performed successfully. To make sure, however, do the following: In a job, resubmit the job. In a session, reenter the most recent FCOPY command.

Message	SYNTAX ERROR: EOF FOUND IN COMMAND FILE, MISSING EXIT COMMAND
Cause	The command input file did not contain an EXIT command.
Action	Correct the input file.

3	Message	SYNTAX ERROR: IN SUBSET OPTION
	Cause	The SUBSET function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.

4	Message	SYNTAX ERROR: IN TITLE OPTION
	Cause	The "title" option of the DISPLAY function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
5	Message	SYNTAX ERROR: IN IGNERR OPTION
	Cause	The "ignore errors" function is not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
6	Message	SYNTAX ERROR: IN VERIFY OPTION
	Cause	The "verify" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
7	Message	SYNTAX ERROR: IN SKIPEOF OPTION
	Cause	The "skip end-of-file" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
8	Message	SYNTAX ERROR: IN COMPARE OPTION
	Cause	The "compare" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
9	Message	SYNTAX ERROR: IN NEW OPTION
	Cause	The "new file" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
10	Message	SYNTAX ERROR: IN HEX OPTION
	Cause	The hexadecimal display function was not specified.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.

11	Message	SYNTAX ERROR: IN EBCDICOUT OPTION
	Cause	The EBDICOUT character translation function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
12	Message	SYNTAX ERROR: IN CHAR OPTION
	Cause	The "display character" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
13	Message	SYNTAX ERROR: IN OCTAL OPTION
	Cause	The "display octal" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
14	Message	SYNTAX ERROR: IN UPSHIFT OPTION
	Cause	The "upshift" function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
15	Message	SYNTAX ERROR: IN BCDICIN OPTION
	Cause	The BCDICIN character translation function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
16	Message	SYNTAX ERROR: IN NORECNUM OPTION
	Cause	The NORECNUM option of the display function was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
17	Message	SYNTAX ERROR: IN EBCDICIN OPTION
	Cause	The EBCDICIN character translation function was not specified properly.

	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
18	Message	SYNTAX ERROR: IN BCDICOUT OPTION
	Cause	The BCDICOUT character translation function was not specified properly
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
19	Message	SYNTAX ERROR: INVALID FORM OF EXIT COMMAND
	Cause	The EXIT command was not specified properly.
	Action	None. FCOPY terminates.
51	Message	SYNTAX ERROR: IN QUOTED STRING
	Cause	The *characterstring* specified for the SUBSET function is not valid.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
52	Message	SYNTAX ERROR: IN BIT PATTERN
	Cause	The *patternlist* specified for the SUBSET function is not valid.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
53	Message	SYNTAX ERROR: INVALID INTEGER
	Cause	An integer specified in the command is outside the range allowed for the particular COPY function.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
54	Message	SYNTAX ERROR; UNKNOWN OPTION
	Cause	One of the specified functions was unrecognizable, or the command passed with the INFO parameter of the RUN command had more than 70 characters between semicolons.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.

55	Message	SYNTAX ERROR: IN FROMFILE SPECIFIER
	Cause	The *fromfile* was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
56	Message	SYNTAX ERROR: IN TOFILE SPECIFIER
	Cause	The *tofile* was not specified properly.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
57	Message	SYNTAX ERROR: ILLEGAL COMBINATION OF OPTIONS
	Cause	Two or more "functionlist" entries conflict with one another.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
58	Message	SYNTAX ERROR: FROMFILE AND TOFILE NOT BOTH SPECIFIED
	Cause	FROM= and TO= were not both specified in the FCOPY command.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
59	Message	SYNTAX ERROR: ILLEGAL USE OF NEW
	Cause	The context used to specify a new *tofile* is not valid.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
60	Message	SYNTAX ERROR: ILLEGAL USE OF *
	Cause	The context used to specify * as a *fromfile* or *tofile* is not valid.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
62	Message	SYNTAX ERROR: FILE NAME TOO LONG
	Cause	The name specified for the *fromfile* or *tofile* is longer than the 35 characters allowed in a fully qualified file name

		with lockword.
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
102	Message	CANNOT CLOSE FROMFILE
	Cause	MPE cannot close the *fromfile*. This message is followed by an MPE file information display containing (among other things) an error number.
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.
103	Message	CANNOT CLOSE TOFILE
	Cause	MPE cannot close the *tofile*. This message is followed by an MPE file information display containing an error number. Usually, this occurs when the file being created already exists. (It was created after the NEW file checking code determined that the file did not exist.)
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.
104	Message	CANNOT SAVE NEW TOFILE
	Cause	MPE cannot close the *tofile* as a permanent file. Either you do not have SF capability or there is not enough group, account, or system file space.
	Action	If you don't have SF capability, you cannot perform the operation. If there is not enough file space, purge some unneeded files to free some file space.
105	Message	CANNOT OPEN FROMFILE
	Cause	MPE cannot open the *fromfile*. This message is followed by an MPE file information display containing an error number.
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.

106	Message	CANNOT OPEN TOFILE
	Cause	MPE cannot open the *tofile*. This message is followed by an MPE file information display containing an error number.
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.
107	Message	VERIFY OR COMPARE OPTION: CANNOT GET READ ACCESS TO TOFILE
	Cause	MPE cannot get read access to the *tofile* for a verify or a 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.
	Action	Reset the FILE command (using the MPE RESET command) and retry the operation.
108	Message	ERROR IN CALLING FGETINFO FOR FROMFILE
	Cause	An error prevented MPE from obtaining information from the *fromfile's* label. This message is followed by an MPE file information display containing an error number.
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.
109	Message	ERROR IN CALLING FGETINFO FOR TOFILE
	Cause	An error prevented MPE from obtaining information from the *tofile's* label. This message is followed by an MPE file information display containing an error number.
	Action	Look up the error number in the file system error table in the *MPE V Intrinsics Reference Manual* (30000-90010) or the *MPE XL Error Message Manual Volumes 1 and 2* (32650-90066 and 32650-90152) and act accordingly.
110	Message	IGNERR OPTION: FROMFILE NOT TAPE
	Cause	The *fromfile's* device is not a magnetic tape unit.
	Action	The "ignore errors" function cannot be used in this case.
111	Message	CANNOT GET READ ACCESS TO FROMFILE

	Cause	MPE cannot get read access to the *fromfile*. The read access specified by FCOPY has been overridden by an MPE FILE command containing ACC=APPEND, ACC=OUT, or ACC=OUTKEEP.
	Action	Reset the FILE command (using the MPE RESET command) and retry the operation.
112	Message	CANNOT GET WRITE ACCESS TO TOFILE
	Cause	MPE cannot get read access to the *tofile*. The write access specified by FCOPY has been overridden by an MPE FILE command containing ACC=IN.
	Action	Reset the particular FILE command (using the MPE RESET command) and retry the operation.
113	Message	SKIPEOF OPTION: FROMFILE NOT AN UNLABELED TAPE
	Cause	The "skip end-of-file" function was specified for the *fromfile*, and the *fromfile* device is not a magnetic tape unit, serial disk, or cartridge tape unit containing unlabeled media.
	Action	If the intended *fromfile* or *tofile* is on magnetic tape or serial disk, check the associated MPE FILE command and its backreference.
114	Message	SKIPEOF OPTION: TOFILE NOT AN UNLABELED TAPE
	Cause	The "skip end-of-file" function was specified for the *tofile*, and the *tofile* device is not a magnetic tape unit, serial disk, or cartridge tape unit containing unlabeled media.
	Action	If the intended *fromfile* or *tofile* is on magnetic tape or serial disk, check the associated MPE FILE command and its backreference.
115	Message	SUBSET OPTION: STRING FALLS OUTSIDE OF FROMFILE RECSIZE
	Cause	The *characterstring* or *patternlist* specified extends beyond the record size of the *fromfile*. No such subset can exist in the specified *fromfile*.
	Action	Change the specified subset definition to a valid one and try the operation again.

116	Message	CANNOT GET LARGE ENOUGH BUFFER
	Cause	There is not enough data space for the buffers needed by the operation you have requested. FCOPY uses the DL-DB area for variable sized buffers.
	Action	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 can accommodate your record size in the maximum size allowed for data segments.
117	Message	SKIPEOF OPTION: ERROR WHILE SKIPPING IN FROMFILE
	Cause	An error occurred while skipping end-of-file marks in the *fromfile*.
	Action	Check the syntax for the SKIPEOF function and retry the operation.
118	Message	SKIPEOF OPTION: ERROR WHILE SKIPPING IN TOFILE
	Cause	An error occurred while skipping end-of-file marks in the *tofile*.
	Action	Check the syntax for the SKIPEOF function and retry the operation.
119	Message	SUBSET OPTION: ERROR WHILE SPACING IN FROMFILE
	Cause	An error occurred while skipping end-of-file marks in the *fromfile*.
	Action	Check the syntax for the SKIPEOF function and retry the operation.
120	Message	SUBSET OPTION: SUBSET STARTS OVER EOF BOUNDARY
	Cause	The specified subset extends over an end-of-file mark or a tape mark boundary.
	Action	Change the specified subset definition to a valid one and try the operation again.
123	Message	SUBSET OPTION: THIS INPUT DEVICE DOES NOT BACKSPACE

	Cause	The specified subset requires backspacing in the *fromfile*, but the device for that file is not a disk or magnetic tape.
	Action	Check the MPE FILE command associated with the *fromfile* and its backreference.
124	Message	READ ERROR IN FROMFILE AT RECORD RECNUM
	Cause	An error occurred while spacing through the *fromfile* in search of the start of a subset.
	Action	Check the syntax of your command and retry the operation.
125	Message	SUBSET OPTION: NUMERIC SUBSET IS EMPTY
	Cause	A subset specified by *starting-record-number, number-of records* and/or *last-record-number* does not exist, or contains no data.
	Action	Change the specified subset definition to a valid one and try the operation again.
126	Message	VERIFY OPTION: ERROR WHILE REWINDING FROMFILE
	Cause	An error occurred while spacing backward to the beginning of the *fromfile* at the start of a "verify" operation.
	Action	Check the syntax of your command and retry the operation.
127	Message	VERIFY OPTION: ERROR WHILE REWINDING TOFILE
	Cause	An error occurred while spacing backward to the beginning of the *tofile* at the start of a "verify" operation.
	Action	Check the syntax of your command and retry the operation.
128	Message	EOF FOUND WHILE SPACING IN FROMFILE
	Cause	An end-of-file mark was encountered while spacing through the *fromfile* in search of the start of a subset.
	Action	Make sure that the starting record number for the subset does not exceed the number of records in the file.

129	Message	EOF FOUND WHILE SPACING IN TOFILE
	Cause	An end-of-file mark was encountered while spacing through the *tofile* in search of the start of a subset during a "compare" or "verify" operation.
	Action	In a compare operation, the *fromfile* and *tofiles* are not identical. Display the *tofile* to determine what it actually contains. In a verify operation, the copy operation was not performed correctly. Retry the operation.
130	Message	FCHECK ERROR
	Cause	The FCHECK intrinsic failed.
	Action	Notify your system manager.
131	Message	ERROR WHILE WRITING EOF TOFILE
	Cause	An error occurred while writing an end-of-file mark in the *tofile*.
	Action	Check the syntax of your command and retry the operation.
132	Message	VERIFY OPTION: ERROR WHILE SPACING IN THE FROMFILE
	Cause	An error occurred while writing an end-of-file mark in the *fromfile* during a "verify" operation.
	Action	Check the syntax of your command and retry the operation.
133	Message	VERIFY OPTION: ERROR WHILE SPACING IN THE TOFILE
	Cause	An error occurred while writing an end-of-file mark in the *tofile* during a "verify" operation.
	Action	Check the syntax of your command and retry the operation.
134	Message	FOUND EOF IN TOFILE
	Cause	FCOPY has performed the specified operation but has filled the *tofile* before completing the operation.
	Action	The *tofile* was not large enough. Use the MPE LISTF *filename*,2 command to determine the *tofile's* size and then increase its size (using the MPE PURGE and BUILD

		commands) and retry the operation. To copy from one file to a smaller file, use SUBSET=.
135	Message	WRITE ERROR TO TOFILE
	Cause	An error occurred while writing to the *tofile*.
	Action	Check the syntax of your command and retry the operation.
136	Message	READ ERROR FROM TOFILE
	Cause	An error occurred while reading from the *tofile* during a "compare" or "verify" operation.
	Action	In a compare operation, retry the operation. If the error persists, you must try to recreate the *tofile*. In a verify operation, retry the operation.
137 *	Message	WARNING: AN UNLABELED TAPE OPERATION ENDS ON AN ERROR
	Cause	An operation involving a magnetic tape *fromfile* was terminated by reading beyond the end of valid data rather than by sensing an end-of-file mark.
	Action	This is not an error. You can avoid this message by specifying FILES=*n* where *n* is the number of files on the tape.
138	Message	TITLE OPTION: TITLE TOO LONG
	Cause	The title specified for the "list" function is longer than the 70 characters allowed, or it extends over more than one line (record).
	Action	In a job, correct the command and resubmit the job. In a session, reenter the command using the correct format.
139	Message	DUMP OPTION: TOFILE RECSIZE NOT WITHIN LEGAL LIMIT
	Cause	A file display was directed to an intermediate storage device with an incorrect record size. That record size must be greater than 60 bytes (30 words).
	Action	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	Message	COMPARE OR VERIFY OPTION: OPERATION FAILS; DIFFERENT FIXED RECSIZES
	Cause	The "compare" or "verify" operation was not attempted because the record sizes of the *tofile* and *fromfile* are not identical.
	Action	In a compare operation, none. The compare operation revealed that the fixed record sizes of the two files are not identical. In a verify operation, change the record size of the *tofile* (using the MPE PURGE and BUILD commands) so that it is the same as that of the *fromfile* and then retry the operation.
141 **	Message	COMPARE BEGINS
	Cause	The comparison phase of a "verify" operation has begun.
	Action	None.
143 *	Message	WARNING: FROMFILE IS EMPTY
	Cause	The *fromfile* contained no data. Nothing was copied or compared.
	Action	None. You may inadvertently have specified the wrong file as the *fromfile*.
144	Message	NEW OPTION: FILE ALREADY EXISTS
	Cause	The *tofile* named for the "new file" function already exists in the specified (or implied) group and account.
	Action	Change the name of the *tofile* and try the operation again.
145	Message	BACKSPACE ERROR IN FROMFILE
	Cause	An error occurred while spacing backward to the beginning of the *fromfile* or a subset within it.
	Action	Check the syntax of your command and retry the operation.
146	Message	CANNOT OPEN TEMPORARY LABEL FILE
	Cause	This may indicate a lack of space in your account.
	Action	Allocate more space in the account.

147	Message	WRITE ERROR IN TEMPORARY LABEL FILE
	Cause	An error occurred while writing to the temporary label file.
	Action	Check the syntax of your command and retry the operation.
148	Message	READ ERROR IN TEMPORARY LABEL FILE
	Cause	An error occurred while reading from the *fromfile*.
	Action	Check the syntax of your command and retry the operation.
149	Message	OVERFLOW IN TEMPORARY LABEL FILE
	Cause	There are more labels in the *fromfile* than can be copied.
	Action	Use the NOUSERLABELS function to copy files without labels.
150	Message	READ ERROR IN FROMFILE LABELS
	Cause	An error occurred while reading labels in the *fromfile*.
	Action	Check the syntax of your command and retry the operation.
151	Message	WRITE ERROR IN TOFILE LABELS
	Cause	An error occurred while reading labels in the *tofile*.
	Action	Check the syntax of your command and retry the operation.
152	Message	OVERFLOW IN TOFILE LABELS
	Cause	There are too many labels in the *fromfile* to be copied to the *tofile*.
	Action	If the *tofile* is an old file, retry the operation using the NOUSERLABELS option. If it is a new file, use the NEW option.
153	Message	ATTEMPT TO BACKSPACE OVER FROMFILEBOF
	Cause	The negative number specified in a "skip end-of-file" function for the *fromfile* is too large.
	Action	Change the number to a smaller one to avoid skipping too many files back.

154	Message	ATTEMPT TO BACKSPACE OVER TOFILEBOF
	Cause	The negative number specified in a "skip end-of-file" function for the *tofile* is too large.
	Action	Change the number to a smaller one to avoid skipping too many files back.
155	Message	DEBLOCK OPTION: INVALID LOGICAL RECORD LENGTH
	Cause	Specified logical record length is greater than the physical record length of the *fromfile*, or it is equal to zero.
	Action	Correct the DEBLOCK option and retry the operation. Make sure that there is a minus sign in front of the logical record length if it is specified in bytes.
200 *	Message	WARNING: FROMFILE RECSIZE IS *number type*, TOFILE RECSIZE IS *number type*
	Cause	The record sizes of the *fromfile* and *tofile* are not identical.
	Action	In a job, FCOPY performs the specified operation despite the conflict. In a session, you are given the choice of whether or not to continue the operation. If the record size of the *fromfile* is larger than that of the *tofile*, the "from" records would be truncated. If the record size of the *tofile* is larger than that of the *fromfile*, the content of the excess byte positions in the "to" records will be blanks if the *tofile* is ASCII, and zeros otherwise.
201 *	Message	WARNING: FROMFILE IS ASCII, TOFILE IS BINARY or WARNING: FROMFILE IS BINARY, TOFILE IS ASCII
	Cause	The data formats of the *fromfile* and *tofile* are not identical.
	Action	In a job, FCOPY performs the specified operation despite the conflict. In a session, you are given the choice of whether or not to continue the operation.
301	Message	READ ERROR IN FROMFILE AT RECORD *recnum*
	Cause	An error occurred while reading from the *fromfile* at the record number displayed (*recnum*).
	Action	Retry the operation. If the error persists, use the SUBSET

function to copy all of the file except the erroneous record.

302 Message VERIFY OPTION: RAN OUT OF VERIFY ERRORS AT FROMFILE RECORD *recnum*

Cause The "verify" function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (*recnum*).

Action Specify a larger *number-of-errors* parameter and retry the operation.

303 Message RAN OUT OF IGNERRS AT FROMFILESREC

Cause More errors were found in the *fromfile* than are permitted by the IGNERR option.

Action Specify a larger *number-of-errors* parameter and retry the operation.

304 Message COMPARE OPTION: RAN OUT OF COMPARE ERRORS AT FROMFILE RECORD *recnum*

Cause The "compare" function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (*recnum*).

Action Retry the operation specifying a larger *number-of-errors* parameter.

601 Message END OF VOLUME SET

Cause The end of the last labeled tape of the volume set has been reached.

Action None.

800 Message EOF IN FROMFILE BUT NOT IN TOFILE

```
{ AT
  AFTER
  BEFORE } RECORD X
```

Cause The *tofile* is larger than the *fromfile* to which it is being compared.

Action None.

801 * Message EOF IN TOFILE BUT NOT IN FROMFILE
{ AT
AFTER
BEFORE } RECORD X

Cause The *tofile* is smaller than the *fromfile* to which it is being compared.

Action None.

802 * Message READ ERROR IN FROMFILE
{ AT
AFTER
BEFORE } RECORD X

Cause A read error occurred while verifying or comparing.

Action Retry the operation and check the MPE file system error.

803 * Message READ ERROR IN TOFILE
{ AT
AFTER
BEFORE } RECORD X

Cause A read error occurred while verifying or comparing,

Action Retry the operation and check the MPE file system error.

901 Message KSAM FROMFILE BOUNDARY (EOF OR BOF)

Cause The beginning or end of the *fromfile* was reached during the FCOPY operation.

Action Try again.

902 Message KSAM FROMFILE POSITIONING ERROR

Cause FCOPY could not find the desired position in the *fromfile*.

Action Try again or check the length of the *fromfile*.

903 Message ERROR: WRONG CONDITIONS FOR OPENING NEW KSAM FILE

	Cause	The *fromfile* is not a KSAM file.
	Action	Create a KSAM file before running FCOPY and copy to that file.
904	Message	KSAM ERROR 904
	Cause	Internal error.
	Action	Notify your system manager.
905	Message	KSAM ERROR 905
	Cause	Internal error.
	Action	Notify your system manager.
906	Message	KSAM ERROR 906
	Cause	Internal error.
	Action	Notify your system manager.
907	Message	KSAM ERROR 907
	Cause	Internal error.
	Action	Notify your system manager.
908	Message	NOKSAM IS NOT A VALID OPTION FOR KSAM XL FILES
	Cause	You have attempted to open a KSAM XL file with the NOKSAM option.
	Action	Correct the command syntax and try again.
909	Message	VARIABLE RECORD FORMAT IS NOT SUPPORTED FOR KSAM XL FILES
	Cause	You are using a variable record format with a KSAM XL file.
	Action	Rebuild your file with a fixed-record format and try again.
910	Message	ERROR DELETING TO FILE ACD
	Cause	Internal error.
	Action	Notify your system manager.

911	Message	ERROR DELETING TO FILE ACD
	Cause	Internal error.
	Action	Notify your system manager.
912	Message	USER CAN'T ACCESS FROMFILE ACD
	Cause	COPYACD was specified for a file to which you have insufficient access rights.
	Action	Retry without specifying COPYACD.
913	Message	USER CAN'T ACCESS FROMFILE ACD
	Cause	COPYACD was specified for a file to which you have insufficient access rights.
	Action	Retry without specifying COPYACD.
914	Message	ERROR COPYING ACD TO NEW TOFILE
	Cause	Internal error.
	Action	Notify your system manager.
915 *	Message	WARNING: NO SOURCE ACD
	Cause	COPYACD option specified but the *fromfile* doesn't have an ACD. Note that on MPE XL release 3.0 or later, this message is not displayed if the COPYACD option is not used and the file does not contain an ACD.
	Action	Quit or continue to copy only the data from the *fromfile*.
916 *	Message	WARNING: COPYACD NOT RELEVANT
	Cause	COPYACD option should be specified only if the *tofile* is a local disk file. Note that on MPE XL release 3.0 or later, this message is not displayed if the COPYACD option is not used and the file does not contain an ACD.
	Action	Quit or continue to copy only the data from the *fromfile*.
930	Message	WRONG ACCESS TYPE WHILE COPYING LABEL EXTENSION OF SPOOL FILES
	Cause	The *tofile* has no write access, or the *fromfile* has no read access.
	Action	Use a file equation to give your file the relevant access

rights.

931	Message	ERROR WHILE COPYING LABEL EXTENSION OF SPOOL FILES
	Cause	Internal error.
	Action	Notify your system manager.
950	Message	CTRANSLATE ERROR
	Cause	Internal error.
	Action	Notify your system manager.
960	Message	LANGUAGE NOT CONFIGURED
	Cause	The language is not installed.
	Action	Run LANGID.PUB.SYS to install it.
961	Message	NLS NOT CONFIGURED
	Cause	NLS is not installed.
	Action	Run LANGID.PUB.SYS to install it.
962	Message	INTERNAL ERROR (NLINFO)
	Cause	Internal error.
	Action	Notify your system manager.
963	Message	INTERNAL ERROR (NLREPCHAR)
	Cause	Internal error.
	Action	Notify your system manager.
964	Message	INTERNAL ERROR (NLSCANMOVE)
	Cause	Internal error.
	Action	Notify your system manager.
965	Message	INTERNAL ERROR (NLTRANSLATE)
	Cause	Internal error.
	Action	Notify your system manager.

966	Message	WARNING: LANG OPTION NOT RELEVANT
	Cause	The *tofile* doesn't need the language.
	Action	Do not use the LANG option.
970	Message	NO EBCDIC TABLE FOR THIS LANGUAGE
	Cause	Some languages do not support the EBCDIC function.
	Action	Do not use the EBCDIC function.
971	Message	INVALID 16-BIT INPUT DATA FOR UPSHIFT
	Cause	16-bit character truncation.
	Action	Check input data and try again.

The following status messages, without assigned numbers, may also be given. They are self-explanatory.

```
EOF FOUND IN FROMFILE AFTER RECORD X

X RECORDS PROCESSED***Y ERRORS

VERIFY ERROR X FOUND AT RECORD Y

COMPARE ERROR X FOUND AT RECORD Y

IGNORED ERROR X: READ ERROR FROM FROMFILE AT RECORD Y

VERIFY ERROR X FOUND AT RECORD Y, BYTE Z

COMPARE ERROR X FOUND AT RECORD Y, BYTE Z

COMPARE ERROR X FOUND AT RECORD Y: DIFFERENT RECORD
```

B File Definitions: Default Values

Appendix B summarizes the default file definition values that FCOPY uses to open a *tofile* or *fromfile*. It contains two tables: table B-1 shows default values for different types of *fromfiles*, and table B-2 shows default values for different types of *tofiles*. The information stored in the file label of an existing disk file always describes its characteristics.

Items marked with a "plus" sign (+) are those most likely to cause problems when you are copying from one device to another. Refer to the *File System Reference Manual* (30000-90236) or *Accessing Files Progammer's Guide* for information about domains, buffers, and record formats.

You can override a file's default definition before you issue an FCOPY command. Simply redefine its characteristics using the MPE FILE command and then use an asterisk (*) to backreference the FILE command in the FCOPY command.

Table B-1. Default "from" Device Values

"From" Device	$CTUL $CTUR	Magnetic Tape	Named Disk File	$STDIN $STDINX
Formal File Designator	(empty)	*fromfile*	*fromfile*	(empty)
Default File Designator	(empty)	*fromfile*	*fromfile*	(empty)
Domain	new	new+	old perm. or temp.	old+
Record Type	see note 1+	binary+	as specified in the file label	ASCII
Record Format	fixed	undefined+	as specified in the file label	fixed
Access Type	I/O	READ/WRITE	READ/WRITE	READ only
Carriage Control	no	no	as specified in the file label	no
Multirecord Access	yes	no	no	no
Dynamic Locking	no	no	no	no
Exclusive Access	yes	yes	yes	no
Inhibit Buffering	yes	no+	no+	no
Record Size	256 bytes	256 bytes+	as specified in the file label	80 bytes
Device	*ldev* # of terminal (left-justified ASCII number)	byte array containing *ldev* # or device class name from the FILE command	as specified in the file label	session or batch input device
Userlabels	0	0	as specified in the file label	0
Blocking Factor	1	1+	as specified in the file label	1
Buffers	2	2+	2+	2
File Size	N/A	N/A	as specified in the file label	N/A

Table B-1. Default "from" Device Values

"From" Device	$CTUL $CTUR	Magnetic Tape	Named Disk File	$STDIN $STDINX
Initial Allocation	N/A	N/A	as specified in the file label	N/A
File Code	0	N/A	as specified in the file label	
Allow `FILE` Command	no	yes	yes	no

Table B-2. Default "to" Device Values

"To" Device	$STDLIST
Formal File Designator	(empty)
Default File Designator	(empty)
Domain	new
Record Type	see note 1+
Record Format	fixed
Access Type	I/O
Carriage Control	no
Multirecord Access	yes
Dynamic Locking	no
Exclusive Access	yes
Inhibit Buffering	yes
Record Size	256 bytes
Device	*ldev* # of terminal (left-justified ASCII number)
Userlabels	0
Blocking Factor	1
Buffers	2
File Size	N/A
Initial Allocation	1
File Code	0
Allow `FILE` Command	no

Table B-2. Default "to" Device Values

"To" Device	$STDLIST
Formal File Designator	(empty)
Default File Designator	(empty)
Domain	new
Record Type	ASCII
Record Format	undefined
Access Type	WRITE only
Carriage Control	no
Multirecord Access	no
Dynamic Locking	no
Exclusive Access	no
Inhibit Buffering	no
Record Size	80 bytes
Device	session or batch output device
Userlabels	N/A
Blocking Factor	1
Buffers	2
File Size	N/A
Initial Allocation	N/A
File Code	N/A
Allow FILE Command	no

1. $CTUL or $CTUR domain is ASCII (JIS for the HP 2645K terminal) or binary, depending on the *tofile*.
2. A named disk file's access type is as follows:

   For COMPARE: READ only.

   For VERIFY: INPUT/OUTPUT.

   APPEND when message files are used.
3. A named disk file has exclusive access for all functions except COMPARE.

C Codes

This appendix contains three tables showing conversions between the following sets of codes:

- ASCII and EBCDIC
- JIS and EBCDIK
- ASCII and BCDIC

ASCII/EBCDIC Conversion Table

This table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalents. 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.

Examples

The following examples illustrate how to use 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 (decimal), 044 (octal), 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

Suppose you want to find out what EBCDIC character the character code 5B (hex) represents. Also, when 5B is converted to ASCII (for example by FCOPY with the EBCDICIN option), you want to know the octal character code. First, locate 5B in the hex character code column and look right on that row to the EBCDIC graphic, which is $. The next column to the right gives the conversion to ASCII, which is 044. As a check, find 044 (octal) in the character code column, look right to the ASCII graphic column and note that $ converted to EBCDIC is 133 (octal), which equals 5B (hex).

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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	GE	227
9	011	09	HT	005	12 5 9	RLF	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	AES	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
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 space	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	1 2	ETB	027
39	047	27	' apost.	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	, comma	153	0 3 8		214
45	055	2D	- hyphen	140	11	ENQ	005
46	056	2E	. period	113	12 3 8	ACK	006
47	057	2F	/	141	0 1	BEL	007
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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
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		023
63	077	3F	?	157	0 7 8	SUB	032
64	100	40	@	174	48	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	. period	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
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	_underline	155	0 5 8		136
96	140	60	` grave	171	1 8	- hyphen	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	, comma	054
108	154	6C	l	223	12 11 3	%	045
109	155	6D	m	224	12 11 4	_unlin	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
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 O 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	' (apos)	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII			EBCDIC	
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	~	176
162	242	A2		103	12 0 3 9	s	163
163	243	A3		104	12 04 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 79	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII		EBCDIC	
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
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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII		EBCDIC	
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		352
205	315	CD	234	12 11 4 8		353
206	316	CE	235	12 11 5 8		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	D8	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

Table C-1. ASCII and EBCDIC Characters

Character Code			ASCII		EBCDIC	
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 78 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 05 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		377

JIS/EBCDIK Conversion Table

This 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 JIS and EBCDIK character. The Hollerith code given in this table is YHP Hollerith and is not equivalent to JIS Hollerith. To determine the JIS Hollerith code, see the Hollerith column in table C-1.

Examples

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 a JIS device (terminal, printer, computer, etc.) to represent the S character. Its Hollerith punched card code is 11-3-8.

Example 2

Suppose you want to find out what EBCDIK character the character code 5B (hex) represents. You also want to know the octal character code when 5B is converted to JIS (for example, by FCOPY with the EBCDIKIN option). First, locate 5B in the hex character code column and look 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 you see that $ converted to EBCDIK is 133 (oct), which equals 5B (hex).

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
0	000	00	NUL	000	12 0 1 8 9	NUL	000
1	001	01	S0H	001	12 1 9	S0H	001
2	002	02	STX	002	12 2 9	STX	002
3	003	03	ETX	003	12 3 9	ETX	003
4	004	04	E0T	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 69		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	SA	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

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII			EBCDIK	
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 space	100	Blank	DS	200
33	041	21	!	320	12 7 8/11 0	SOS	201
34	0	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	' apost.	174	5 8 / 4 8/0 4 8	ESC	033
40	050	28	(	154	12 5 8		210
41	051	29	)	114	11 5 8/12 4 1		211
42	052	2A	*	134	11 4 8	SM	212
43	053	2B	+	120	12 6 8/12	CU2	213
44	054	2C	, comma	153	0 3 8		214
45	055	2D	- hyphen	140	1 1	ENQ	005
46	056	2E	. period	113	12 3 8	ACK	006
47	057	2F	/	141	01	BEL	007
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

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII			EBCDIK	
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	:	175	2 8/5 8		232
59	073	3B	;	136	11 68	CU3	233
60	074	3C	<	116	12 4 8/12 6 8	DC4	024
61	075	3D	=	173	6 8	NAK	025
62	076	3E	>	176	0 6 8/3 8/6 8		236
63	077	3F	?	300	0 7 8/12 0	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	. period	056
76	114	4C	L	323	11 3	)	051
77	115	4D	M	324	11 4	[	133
78	116	4E	N	325	11 5	<	074
79	117	4F	O	326	11 6	?	077
80	120	50	P	327	11 7	+	053
81	121	51	Q	330	11 8		251
82	122	52	R	331	11 9		252
83	123	53	S	342	0 2		253

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII			EBCDIK	
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	[	115	12 2 8/12 5 8	$	044
92	134	5C	\	156	0 2 8/0 6 8	*	052
93	135	5D	]	135	11 2 8/11 5 8	]	135
94	136	5E	^	137	0 2 9	;	073
95	137	5F	_underline	155	0 5 8	/	057
96	140	60	` grave	171	1 8	-hyphen	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	, comma	054
108	154	6C	l	223	12 11 3	%	050
109	155	6D	m	224	12 11 4	_unlin	137
110	156	6E	n	225	12 11 5	\	134
111	157	6F	o	226	12 11 6	?	077
112	160	70	p	227	12 11 7		272

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII			EBCDIK	
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		040
123	173	7B	{	300	12 0	=	075
124	174	7C	|	152	12 11	' apos	047
125	175	7D	}	320	11 0	^	072
126	176	7E	-	241	11 0 1	>	076
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

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII		EBCDIK	
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	~	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		322

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII		EBCDIK	
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
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	?	077
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 04 8	F	106
199	307	C7	215	12 0 5 8	G	107

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII		EBCDIK	
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		352
205	315	CD	234	12 11 4 8		353
206	316	CE	235	12 11 5 8		354
207	317	CF	236	12 11 6 8		355
208	320	D0	237	12 11 7 8	!	041
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

Table C-2. ASCII and EBCDIK Characters

Character Code			ASCII		EBCDIK	
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		377

ASCII/BCDIC Conversion Table

In the Hollerith column in this table, 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.

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
0	000	00	NUL	000	12 0 1 8 9	NUL	000
1	001	01	S0H	001	12 1 9	S0H	001
2	002	02	STX	002	12 2 9	STX	002
3	003	03	ETX	003	12 3 9	ETX	003
4	004	04	E0T	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 69		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	SA	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

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII			BCDIC	
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 space	100	Blank	DS	200
33	041	21	!	320	12 7 8/11 0	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	' apost.	174	5 8 / 4 8/0 4 8	ESC	033
40	050	28	(	154	12 5 8		210
41	051	29	)	114	11 5 8/12 4 1		211
42	052	2A	*	134	11 4 8	SM	212
43	053	2B	+	120	12 6 8/12	CU2	213
44	054	2C	, comma	153	0 3 8		214
45	055	2D	- hyphen	140	1 1	ENQ	005
46	056	2E	. period	113	12 3 8	ACK	006
47	057	2F	/	141	01	BEL	007
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

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII			BCDIC	
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	:	175	2 8/5 8		232
59	073	3B	;	136	11 68	CU3	233
60	074	3C	<	116	12 4 8/12 6 8	DC4	024
61	075	3D	=	173	6 8	NAK	025
62	076	3E	>	176	0 6 8/3 8/6 8		236
63	077	3F	?	300	0 7 8/12 0	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	. period	056
76	114	4C	L	323	11 3	)	051
77	115	4D	M	324	11 4	[	133
78	116	4E	N	325	11 5	<	074
79	117	4F	O	326	11 6	?	077
80	120	50	P	327	11 7	+	053
81	121	51	Q	330	11 8		251
82	122	52	R	331	11 9		252

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII			BCDIC	
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	[	115	12 2 8/12 5 8	$	044
92	134	5C	\	156	0 2 8/0 6 8	*	052
93	135	5D	]	135	11 2 8/11 5 8	]	135
94	136	5E	^	137	0 2 9	;	073
95	137	5F	_underline	155	0 5 8	/	057
96	140	60	` grave	171	1 8	-hyphen	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	, comma	054
108	154	6C	l	223	12 11 3	%	050
109	155	6D	m	224	12 11 4	_unlin	137
110	156	6E	n	225	12 11 5	\	134
111	157	6F	o	226	12 11 6	?	077

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII			BCDIC	
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		040
123	173	7B	{	300	12 0	=	075
124	174	7C	|	152	12 11	' apos	047
125	175	7D	}	320	11 0	^	072
126	176	7E	-	241	11 0 1	>	076
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

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII		BCDIC	
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	~	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

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII		BCDIC	
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
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	?	077
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 04 8	F	106

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII		BCDIC	
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		352
205	315	CD	234	12 11 4 8		353
206	316	CE	235	12 11 5 8		354
207	317	CF	236	12 11 6 8		355
208	320	D0	237	12 11 7 8	!	041
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

Table C-3. ASCII and BCDIC Characters

Character Code			ASCII		BCDIC	
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		377