

900 Series HP 3000 Computer Systems



FCOPY **Reference Manual**



19483 PRUNERIDGE AVENUE, CUPERTINO, CA 95014-9974

Part No. 03000-90064
U0788

Printed in U.S.A. 11/87

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

Notice

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

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

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

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

© Copyright 1987, 1988 by HEWLETT-PACKARD COMPANY

Printing History

New editions are complete revisions of the manual. Update packages, which are issued between editions, contain additional and replacement pages to be merged into the manual by the customer. The dates on the title page change only when a new edition or a new update is published. No information is incorporated into a reprinting unless it appears as a prior update. The edition does not change when an update is incorporated.

The software code printed alongside the date indicates the version level of the software product at the time the manual edition or update was issued. Many product updates and fixes do not require manual changes, and conversely, manual corrections may be made without accompanying product changes. Therefore, do not expect a one-to-one correspondence between product updates and manual updates.

First Edition	November 1987	A.01.00
Update 1	July 1988	A.10.00

List of Effective Pages

The list of Effective Pages give the date of the current edition, and lists the dates of all changed pages. Unchanged pages are listed as "ORIGINAL". Within the manual, any page changed since the last edition is indicated by printing the date the changes were made on the bottom of the page. Changes are marked with a vertical bar in the margin. If an update is incorporated when an edition is reprinted, these bars and dates remain. No information is incorporated into a reprinting unless it appears as a prior update.

First Edition November 1987
Update 1 July 1988

Page	Effective Date	Page	Effective Date
1 to 4	7/88	3-6A	7/88
2-1	7/88	5-14A	7/88
2-3	7/88	5-44	7/88
3-1	7/88	A-20 to A-24	7/88
3-6	7/88	B-2 to B-4	7/88

Documentation Map

General User's Series

MPE XL
Documentation
Guide & Glossary Of
Terms
(5958-9511)

GENERAL REFERENCE

General User's
Reference Manual
(32650-90002)

MPE XL
Commands
Reference Manual
(32650-90003)

MPE XL
System Utilities
Reference Manual
(32650-90081)

QUICK REFERENCE

MPE XL
Quick Reference Guide
(32650-90032)

Pocket Cards
(32650-90033)

(In Progress)

(In Progress)

TOOLS

EDIT/3000
Reference Manual
(03000-90012)

FCOPY
Reference Manual
(03000-90064)

SORT-MERGE/XL
General User's
Guide
(32650-90082)

There are many other manuals applicable to the 900 Series HP 3000. The *MPE XL Documentation Guide & Glossary Of Terms* (5958-9511) contains a complete list of all MPE XL manuals.

Preface

This manual is written to serve a wide range of users, ranging from the occasional MPE V or MPE XL user to the experienced System Operator. Because there are no differences between the way FCOPY runs on MPE V and MPE XL, this manual can be used for both systems.

The manual has three major parts. Chapters 1 through 3 explain how to use FCOPY. Chapters 4 through 6 provide a reference to FCOPY functions, and the Appendices contain FCOPY message descriptions and additional useful information.

The information in the book is presented as follows:

- Chapter 1.** **Introduction** is the brief look at how to use the manual.
- Chapter 2.** **Using FCOPY** provides overview information on the FCOPY command and lists all FCOPY functions.
- Chapter 3.** **FCOPY Applications and Examples** gives instructions and examples for using FCOPY for several common purposes.
- Chapter 4.** **FCOPY Syntax** summarizes the syntax of an 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** gives a brief look at terminal peripheral file designators, terminal settings, and copying between peripherals.
- Appendix A** **FCOPY Messages** contains all error, warning, and status messages.
- Appendix B** **Default File Definition Values** summarizes the values that FCOPY uses to open “from” or “to” files.
- Appendix C** **Code Conversion Tables** contains three tables showing conversions between the following sets of codes: ASCII and EBCDIC, JIS and EBCDIK, ASCII and BCDIC.

Use Chapters 1 through 3 as an introduction if you have never used FCOPY. Read Chapter 3 if you want to use FCOPY for a special purpose and would like to follow an example. Use Chapters 4 through 6 and the Appendices as an FCOPY reference guide, and refer to them for information as needed.

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

You may also need to consult the following:

- *MPE Guide for the New User* (32033-90009)
- *MPE Commands Reference Manual* (32033-90006) or
 MPE XL Commands Reference Manual (32650-90003)

Preface (continued)

- *MPE File System Reference Manual* (30000-90236) or
Accessing Files Programmer's Guide (32650-90017)
- *KSAM/3000 Reference Manual* (30000-90079)
- *Native Language Support Reference Manual* (32414-90001) or
Introduction to MPE XL for MPE V Programmers (30367-90005)

Table of Contents

Chapter 1

Introduction

How to Use this Manual	1-1
Before You Use FCOPY	1-2

Chapter 2

Using FCOPY

The Components of an FCOPY Command	2-1
"From" and "To" Files	2-1
FCOPY Functions	2-2
Defining Files	2-4
General Guidelines for FCOPY Commands	2-4
Issuing an FCOPY Command	2-4
Issuing Commands from within the FCOPY Subsystem	2-5
Issuing MPE Commands	2-5
Interrupting FCOPY	2-5
Exiting FCOPY	2-5
Issuing a Command Directly from MPE	2-5
Using FCOPY in a Job	2-5
FCOPY Messages	2-6
Composing a Command for a Particular Purpose	2-6

Chapter 3

FCOPY Applications and Examples

General Methods for Copying Files	3-1
Copying to a New Disc File	3-1
Appending One File to Another	3-1
Extracting Subsets of a File	3-2
Defining Subsets from Character Strings	3-2
Displaying the Contents of a Disc File	3-3
Displaying a Binary File in ASCII Format	3-3
Displaying an ASCII File	3-3
Displaying Hexadecimal and Octal Codes	3-4
Displaying a Subset of a File	3-4
Printing the Contents of a File	3-5
Using your Terminal as the "From" File	3-5
Copying Files with Varying Block Size	3-6
General Methods for Copying Tapes	3-6
Specifying Tape Files	3-6
Ignoring Tape Errors	3-7
Comparing Files	3-7
Verifying Copies to and from Tape	3-7
Changing Tape Blocking Factor	3-7
Combining Disc Files on Magnetic Tape	3-8
Combining Tape Files on Disc	3-9
Methods for Copying Labeled Tapes	3-9
Copying to Labeled Tape	3-9
Sequentially Copying Files to Labeled Tape	3-9

Table of Contents (continued)

Copying to the End of a Labeled Tape	3-10
Copying from Labeled Tape	3-11
Copying Files by Number	3-11
Copying Named Files	3-11
Sequentially Copying Files	3-12
Deblocking Labeled Tapes	3-13
Using Tapes to Copy Files between HP 3000 Systems	3-13
Copying Foreign Tapes	3-14
Remote Copies	3-15
Copying KSAM Files	3-15
Copying a KSAM File to an Existing KSAM File	3-15
Copying a KSAM File to a New KSAM File	3-16
Copying a KSAM File in a Specified Key Sequence	3-16
Copying a KSAM File to a Non-KSAM File	3-16
Copying Inactive Records in Relative I/O and Message Files	3-17
Copying Files in Native Languages	3-17
Using Native Languages with CHAR	3-17
Using Native Languages with EBCDICIN and EBCDICOUT	3-18
Using Native Languages with UPSHIFT	3-18

Chapter 4

FCOPY Syntax

FCOPY Command Syntax	4-1
Syntax of FCOPY Functions	4-2

Chapter 5

FCOPY Functions

BCDICIN/BCDICOUT	5-2
CCTL/NOCTL	5-4
CHAR	5-7
CLEAR	5-10
COMPARE	5-13
COPYACD	5-14A
DEBLOCK	5-15
EBCDICIN/EBCDICOUT	5-17
EBCDIKIN/EBCDIKOUT	5-19
FILES	5-21
HEX	5-25
HEXO	5-27
IGNERR	5-29
KANA	5-30
KEY	5-32
NEW	5-34
NOKSAM	5-35
NOUSERLABELS	5-36
OCTAL	5-37
SKIPEOF	5-40
SUBSET	5-44
UPSHIFT	5-48
VERIFY	5-49

Table of Contents (continued)

Chapter 6

Using FCOPY with Terminal Peripherals

Terminal Peripheral File Designators	6-1
Terminal Settings	6-1
Copying Between Terminal Peripherals	6-2
Examples	6-2

Appendix A

FCOPY Messages

Error Message Display During a Session	A-1
Error Message Display During a Job	A-2
FCOPY Messages	A-2

Appendix B

Default File Definition Values

Appendix C

Code Conversion Tables

Tables

Table 2-1.	FCOPY FUNCTIONS	2-3
Table 4-1.	Syntax of FCOPY Functions	4-2
Table 4-1.	Syntax of FCOPY Functions (Continued)	4-3
Table 5-1.	BCDIC to ASCII Conversion	5-3
Table 5-2.	Copying Files without Using CCTL or NOCCT	5-4
Table 5-3.	Copying Files Using CCTL	5-5
Table 5-4.	Copying Files Using NOCCTL	5-6
Table 6-1.	Terminal Peripheral Designators	6-1
Table A-1.	FCOPY Error, Warning, and Status Messages	A-3
Table B-1.	Default “from” Device Values	B-2
Table B-2.	Default “to” Device Values	B-3
Table C-1.	ASCII and EBCDIC Characters	C-2
Table C-2.	JIS and EBCDIC Characters	C-8
Table C-2.	ASCII and BCDIC Characters	C-14

Figures

Figure 5-1.	Logical Record Length	5-17
Figure 5-2.	Tape Position	5-23
Figure 5-3.	Copying Subsets of Files to Tape	5-24
Figure 5-4.	Copying Subsets of Files to Labeled Tape	5-25
Figure 5-5.	Copying from a Single Labeled Tape File	5-26
Figure 5-6.	Skipping Absolute File Numbers	5-43
Figure 5-7.	SKIPEOF = +3	5-44
Figure 5-8.	SKIPEOF = -3	5-44
Figure 5-9.	SKIPEOF = +0	5-45

Conventions

NOTATION	DESCRIPTION
UPPERCASE	Within syntax statements, characters in uppercase must be entered in exactly the order shown, though you can enter them in either uppercase or lowercase. For example: <pre>SHOWJOB</pre> Valid entries: showjob ShowJob SHOWJOB Invalid entries: shojwob Shojjob SHOW_JOB
<i>italics</i>	Within syntax statements, a word in italics represents a formal parameter or argument that you must replace with an actual value. In the following example, you must replace <i>filename</i> with the name of the file you want to release: <pre>RELEASE <i>filename</i></pre>
punctuation	Within syntax statements, punctuation characters (other than brackets, braces, vertical parallel lines, and ellipses) must be entered exactly as shown.
{ }	Within syntax statements, braces enclose required elements. When several elements within braces are stacked, you must select one. In the following example, you must select ON or OFF: <pre>SETMSG { ON } { OFF }</pre>
[]	Within syntax statements, brackets enclose optional elements. In the following example, brackets around ,TEMP indicate that the parameter and its delimiter are optional: <pre>PURGE <i>filename</i>[,TEMP]</pre> When several elements within brackets are stacked, you can select any one of the elements or none. In the following example, you can select <i>devicename</i> or <i>deviceclass</i> or neither: <pre>SHOWDEV [<i>devicename</i>] [<i>deviceclass</i>]</pre>

Conventions (continued)

NOTATION	DESCRIPTION
[...]	Within syntax statements, a horizontal ellipsis enclosed in brackets indicates that you can repeatedly select elements that appear within the immediately preceding pair of brackets or braces. In the following example, you can select <i>itemname</i> and its delimiter zero or more times. Each instance of <i>itemname</i> must be preceded by a comma: [,<i>itemname</i>][...] If a punctuation character precedes the ellipsis, you must use that character as a delimiter to separate repeated elements. However, if you select only one element, the delimiter is not required. In the following example, the comma cannot precede the first instance of <i>itemname</i>: [<i>itemname</i>][,...]
{...}	Within syntax statements, a horizontal ellipsis enclosed in parallel vertical lines indicates that you can select more than one element that appears within the immediately preceding pair of brackets or braces. However, each element can be selected only one time. In the following example, you must select , A or , B or , A, B or , B, A : { , A } { , B } ... If a punctuation character precedes the ellipsis, you must use that character as a delimiter to separate repeated elements. However, if you select only one element, the delimiter is not required. In the following example, you must select A or B or A, B or B, A . The first element cannot be preceded by a comma: { A } { B } ,...
... :	Within examples, horizontal or vertical ellipses indicate where portions of the example are omitted.
△	Within syntax statements, the space symbol △ shows a required blank. In the following example, you must separate <i>modifier</i> and <i>variable</i> with a blank: SET[(<i>modifier</i>)]△(<i>variable</i>);

Conventions (continued)

NOTATION

DESCRIPTION

shading

Within an example of interactive dialog, shaded characters indicate user input or responses to prompts. In the following example, OMEGA is the user's response to the NEW NAME prompt:

NEW NAME? OMEGA



The symbol  indicates a key on the terminal's keyboard. For example,  indicates the Control key.

 *char*

 *char* indicates a control character. For example,  Y means you have to simultaneously press the Control key and the Y key on the keyboard.

base prefixes

The prefixes %, #, and \$ specify the numerical base of the value that follows:

%num specifies an octal number

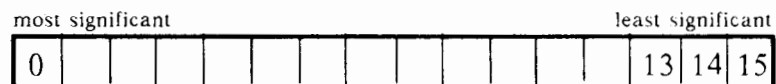
#num specifies a decimal number

\$num specifies a hexadecimal number

When no base is specified, decimal is assumed.

Bit (*bit:length*)

When a parameter contains more than one piece of data within its bit field, the different data fields are described in the format Bit (*bit:length*), where *bit* is the first bit in the field and *length* is the number of consecutive bits in the field. For example, Bits (13:3) indicates bits 13, 14, and 15:



Bit(0:1)

Bits(13:3)

Introduction

This manual describes the HP 3000 file copy utility, FCOPY/3000. FCOPY is a utility program within the HP 3000 Multiprogramming Executive (MPE V and MPE XL) operating systems that lets you copy data from one file to another (one manual is used to describe FCOPY on both operating systems, since FCOPY works the same on both). Because MPE V 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 disc files to move from one HP 3000 to another, or for transferring programs or data from one medium to another such as from magnetic tape to disc.

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 to be used at different computer installations or to incorporate files from magnetic tapes from other computer installations into your HP 3000 system. Another FCOPY function lets you copy only portions of a file. You might use this “subset” function to divide an existing file into several separate files.

How to Use this Manual

This manual documents all of FCOPY's functions and uses. Therefore, it is organized and written to serve a wide range of users, ranging from the occasional MPE user who wants to know how to copy an existing disc file to the experienced System Administrator who needs to know how to copy files to magnetic tape in IBM format. How you use this manual depends on your needs and experience.

The manual is organized into three major parts. Chapters 1 through 3 explain how to use FCOPY. Read these chapters first if you are a new FCOPY user. Chapter 3 contains several examples, so when you have a specific task to perform you might look at Chapter 3 and follow an example first. Chapters 4 through 6 contain FCOPY syntax and descriptions of all of FCOPY's functions. Turn to these chapters first if you need to look up the syntax for an FCOPY function. The Appendices contain FCOPY message descriptions and additional useful information for particular situations.

Before You Use FCOPY

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

- Running jobs and sessions
- The MPE V or MPE XL accounting structure
- File security
- File and device relationships
- Specifying and modifying file characteristics

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, the method of identifying the files that you reference in a command, the procedures for issuing an FCOPY command, and the types of messages you might see in response to a command.

The Components of an FCOPY Command

There are several ways to issue an FCOPY command, but, whichever method you choose, the command consists of the same components. It identifies a *fromfile*, a *tofile*, and one or more FCOPY functions that you want to perform.

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 "to" file is the output file to which you want to copy the data.

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

$$>FROM \left[= \left\{ \begin{array}{l} \text{fromfile} \\ * <file> \\ <empty> \end{array} \right\} \right]$$

The value you assign to FROM can be either an input file name (*fromfile*), an asterisk (*), or nothing at all (<empty>). An asterisk backreferences the fromfile named in the file equation. Leaving FROM empty lets you 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 "input" for an FCOPY command.

```
FROM=input
```

You specify a *tofile* with the TO parameter. TO has the following format:

$$;TO \left[= \left\{ \begin{array}{c} (dfile,kfile) \\ tofile \\ * \\ <empty> \end{array} \right\} \right]$$

The value that you assign to TO can be either the names of data and key files for a new KSAM file (*dfile* and *kfile*), the name of the output file (*tofile*), an asterisk (*), or nothing at all (<empty>). An asterisk backreferences the *tofile* named in the file equation. Leaving TO empty lets you copy files to your terminal (or to a printer during a job).

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

```
;TO=outfile
```

FCOPY Functions

In addition to specifying input and output files in an FCOPY command, you also describe the FCOPY functions that you want to perform. Each function has its own syntax and guidelines for its use. An example of an FCOPY command using the NEW function is:

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

The NEW function, for example, lets you create a new disc file. Chapter 3 contains examples using FCOPY functions, and Chapter 5 describes all FCOPY functions in detail. Table 2-1 lists FCOPY functions and their descriptions in alphabetical order.

Table 2-1. FCOPY FUNCTIONS

Function	Description
BCDICIN	Translates from BCDIC to ASCII
BCDICOUT	Translates from ASCII to BCDIC
CCTL	Designates the first character of each record as a carriage control character in the <i>tofile</i>
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
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
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 key sequence in which to copy a KSAM file
NEW	Creates a new permanent disc file
NOCCTL	Specifies that the first character of each record in the <i>fromfile</i> is not to be a carriage control character
NOKSAM	Copies a KSAM data file to a non-KSAM file
NOUSERLABELS	Omits user labels when copying between disc 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
UPSHIFT	Converts lowercase characters to uppercase
VERIFY	Compares files after copying

Defining Files

If you are copying files from or to devices other than disc, 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 *tofiles* 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 MPE :FILE command, refer to the *MPE V Commands Reference Manual* (32033-90009) or the *MPE XL Commands Reference Manual* (32650-90003).

General Guidelines for FCOPY Commands

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. For example, all three FCOPY commands below are valid.

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

To continue an FCOPY command onto more than one line, use an ampersand (&) at the end of each line except the last. An FCOPY command has no maximum length, however a single line of an FCOPY command cannot be more than 72 characters long. For example:

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

To interrupt FCOPY while it is copying a file, press CONTROL-Y. If you use this procedure to cancel a command, you cannot use an asterisk (*) to backreference files in the next command.

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

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.

Issuing Commands from within the FCOPY Subsystem

Within the FCOPY subsystem, you can issue a single command or a set of commands before returning to MPE. To enter the FCOPY subsystem, type `RUN FCOPY.PUB.SYS` in response to an MPE prompt. For example:

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

Or you can simply type `:FCOPY`. FCOPY prompts you for commands with a greater-than symbol (>). 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=B; NEW
```

Issuing MPE Commands

You can issue any MPE command that does not create another process from the FCOPY subsystem. Type a colon immediately after the FCOPY prompt, and then type the MPE command. This feature of FCOPY is useful when you need to define the files you intend to copy with the MPE `:FILE` command. For example:

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

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.

Exiting FCOPY

To leave the FCOPY subsystem, type `exit` in response to the FCOPY prompt. For example:

```
>exit
```

Issuing a Command Directly 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 within 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 after the command. For example:

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

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 Commands Reference Manual* (32033-90009) or the *MPE XL Commands Reference Manual* (32650-90003). Your job input file can 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
!FCOPY.PUB.SYS
FROM=A; TO=B; NEW
EXIT
!FCOPY FROM=B; TO=C; NEW
!EOJ
#J144
:
```

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 potential 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.

Composing a Command for a Particular Purpose

By combining FCOPY functions and using different forms of functions, you can use FCOPY for many purposes. The examples in Chapter 3 illustrate how to use FCOPY commands for several common situations. When you do not find the task you need to perform in Chapter 3, use one of the other methods described below to find the information you need.

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

- Look up the task you need to perform in the Index.
- Look for instructions for the task you need to perform in Chapter 3.
- Look at Table 2-1 to determine the functions you might use.
- Look up the descriptions of the functions you plan to use in Chapter 5.



FCOPY Applications and Examples

This chapter contains instructions and examples for using FCOPY and its functions for several common purposes. While 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.

General Methods for Copying Files

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

Copying to a New Disc File

To create a new disc file as you copy information to it, use the New function. For example:

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

The new file, in this case the file named NEWFILE, has the default file characteristics described in Appendix B and exists in your current account and group.

Appending One File to Another

Often you need to string several files together. Use FCOPY to accomplish this.

To append a file to another:

1. Use a :FILE command to give the *tofile* APPEND access and set its limit large enough to hold the contents of all the files you are appending to it. Refer to the *MPE File System Reference Manual* (30000-90236) or *Accessing Files Programmers Guide* (32650-90017) for instructions on calculating file space. For example:

```
>:FILE DISC3;REC=-80,10,F,ASCII;ACC=APPEND; disc= 500, 2
```

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-DISC1; TO=*DISC3; NEW  
EOF FOUND IN FROMFILE AFTER RECORD 19  
  
20 RECORDS PROCESSED * * * 0 ERRORS
```

- Copy any subsequent files in the order you want to append them. Use an asterisk alone to backreference the *tofile* in the previous command. For example:

```
>FROM=DISC2; TO=*
EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

Extracting Subsets of a File

Use the SUBSET function to extract subsets from a file. You can define a subset as a set of continuous records or a set of records with a certain pattern of characters or numbers. For example, a file contains information about the employees in your company. Each record in the file describes one employee, and the employee's last name begins in column 0 <or 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 0 <or 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 0 <or 1>.

Defining Subsets from Character Strings

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

<pre>;SUBSET [= "characterstring" [, [column]] [, EXCLUDE]]</pre>
--

Substitute the series of characters that you want to match for *characterstring*, and the beginning column number of the characterstring for *column*. For example, the command below copies the subset of all records with the characters MALE beginning in the fortieth column from the EMPRECS file to the MEN file :

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

2859 RECORDS PROCESSED * * * 0 ERRORS
```

To copy only selected records, use the EXCLUDE parameter. For example, the command below copies all records in EMPRECS except those with the characters "MALE" beginning in position 40 to the file WOMEN.

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

3127 RECORDS PROCESSED * * * 0 ERRORS
```

<pre>;SUBSET [= "characterstring" [, [column]] [, EXCLUDE] = #patternlist # [, [column]] [, EXCLUDE] = (range [; range] [;....])]</pre>
--

Displaying the Contents of a Disc File

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

- CHARACTER
- OCTAL
- HEXADECIMAL

Displaying a Binary File in ASCII Format

To display the contents of a binary disc file at your terminal in ASCII format:

1. Leave the TO parameter empty.
2. Use the CHAR function and if you wish, its NORECNUM parameter. CHAR instructs FCOPY to display the contents of the file as ASCII characters. NORECNUM tells FCOPY not to display record numbers (if they exist).

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

```
>FROM=FILE; TO=; CHAR; NORECNUM
This is a test file.
  It is an example for the CHAR function.
  It contains three lines of text.
  EOF FOUND IN FROMFILE AFTER RECORD 2

  3 RECORDS PROCESSED * * * 0 ERRORS
```

Displaying an ASCII File

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

```
>FROM=ASCIIFILE; TO=;
This is an ASCII file.
It contains two lines of text.
EOF FOUND IN FROMFILE AFTER RECORD 1

2 RECORDS PROCESSED * * * 0 ERRORS
```

Displaying 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 right are text editor line numbers; that start at 1 (00001000):

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

TEXT3.PUB.ACCOUNT RECORD 9 (%11)

000000:	052110	044523	020111	051440	046111	047105	020061	027040	THIS IS LINE 1
000010:	020124	044105	020070	026504	044507	044524	020116	052515	THE 8-DIGIT NUM
000020:	041105	051123	020101	052040	052110	042440	051111	043510	BERS AT THE
000030:	052040	040522	042440	052110	042440	020040	020040	020040	RIGHT ARE THE
000040:	020040	020040	020040	020040	030006	030006	030006	030006	00001000

TEXT3.PUB.ACCOUNT RECORD 10 (%12)

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

Displaying a Subset of a File

To display a subset of a file on your terminal, use the SUBSET function. The example below copies a subset of 5 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.
WIDGET, INC.
ROBERT ALCOTT & ASSOCIATES
WILLIAM LEWIS & SON
SUPER GIZMO OF NEVADA, INC.
```

```
131 PINE STREET SF
43721 OAK STREET DET
752 SURF AVE. CHI
3217 W. MADISON AVENUE NY
431 S. JACKPOT LANE RENO
```

Printing the Contents of a File

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 disc file to save and study it later. To print such a file, use FCOPY to copy it to a printer.

To print an ASCII file:

1. Use an 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=DISC1; TO=*LIST
```

To print a binary file:

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

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

2. Use the CHAR function and its NORECNUM parameter in the FCOPY command. For example:

```
>FROM=DISC1; TO=*LIST; CHAR; NORECNUM
```

Using your Terminal as the "From" File

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 this is easier to do with an editor).

To copy information from your terminal:

1. Leave the FROM parameter empty. If the *tofile* is a new disc 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:

```
RECORD 1 Return
RECORD 2 Return
RECORD 3 Return
RECORD 4 Return
RECORD 5 Return
```

3. Type **CTRL Y** to mark the end of the *fromfile*.

Copying Files with Varying Block Size

A file's block size is determined by multiplying its record size by its block factor.

To copy a file when you do not know the block size or if the block size varies (as on a STORE tape) use undefined-length records for both files.

Specify a record size in the file equations, 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 a STORE tape, specify the number of files stored plus 6. These extra files contain tape header information for the STORE program.

General Methods for Copying Tapes

The general methods for copying tapes described in this chapter apply to unlabeled magnetic tapes, serial discs, 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

A labelled tape must be used if the copy operation will span two reels.

Specifying Tape Files

Use the SKIPEOF function to position an unlabeled tape, cartridge tape, or serial disc at the beginning of the *from* 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 specify the number of end-of-file marks to skip 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.

The example below tells FCOPY to copy the file three files before TAPEA's current position to 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 end-of-file 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 to ignore certain file system errors while copying files from magnetic tape. Ignoring these errors permits recovering 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

With 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 disc file and a tape file or compare 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 disc file DISCA with the fifth file on TAPE1:

```
>:FILE TAPE1; DEV=TAPE
>FROM=DISCA; 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 Copies to and from Tape

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 don't 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 Tape Blocking Factor

You use the DEBLOCK function to change the record blocking factor as you copy a file. DEBLOCK is most useful for copying tapes from other computer systems to the HP 3000. The HP 3000 requires record lengths

to be an even number of bytes, however, many other systems use records consisting of an odd number of bytes. When you copy a foreign file to the HP 3000, you must reassign data to even-byte records. For more information on copying foreign tapes refer to Copying Foreign Tapes, later in this chapter.

To deblock a file with odd-byte records:

1. Use a :FILE command that treats each block of the *fromfile* as a single record and sets the blocking factor to 1. The record length should be the old record length times the old blocking factor.

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

2. In the FCOPY command set DEBLOCK equal to the original record length. Because the HP 3000 requires even-length records, FCOPY adds one byte to each output record.

```
>FROM=*TAPEFILE; TO=DISCFILE; DEBLOCK=-79 EOF FOUND IN fromfile AFTER  
RECORD 0
```

```
1 RECORD PROCESSED * * * 0 ERRORS
```

```
>LISTF,2
```

Notice that the file now has 80 byte records.

Combining Disc Files on Magnetic Tape

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

1. Use a :FILE command to describe the tape. For example:

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

2. Copy the first disc 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 disc files to tape without repositioning the tape drive. Use an asterisk as the *tofile* name.

```
>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 Disc

To combine several tape files, or subsets of tape files, in a single disc file:

1. Use a `:FILE` command to give the disc file APPEND access (`ACC=APPEND`). The disc file must be large enough to hold the contents of all of the tape files, and its record structure must be specified. For example:

```
>:FILE AMALGUM; 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 disc 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=*AMALGUM; 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 *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
```

Methods for Copying Labeled Tapes

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

Copying to Labeled Tape

When copying files to labeled tapes, you can copy files sequentially or to the end of the previously written files.

Sequentially Copying Files to Labeled Tape

To write a series of disc files to the labeled tape:

1. Use the NEXT sequence option in a `:FILE` command describing the labeled tape. For example:

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

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

```
>FROM=FILEA; TO=*LTAPE
?8:20/#S92/17/MOUNT TAPE VOLUME TLABEL
=REPLY 17,7
8:20/#S92/17/VOL UNLABELED MOUNTED ON LDEV# 7
EOF FOUND IN fromfile AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

3. Copy subsequent files to the tape in the same manner. For example:

```
>FROM=FILEB; TO=*LTAPE
EOF FOUND IN fromfile AFTER RECORD 29

30 RECORDS PROCESSED * * * 0 ERRORS
```

4. To combine two disc files on tape, use an asterisk alone as the *tofile* in the command for copying the second disc 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
```

Copying to the End of a Labeled Tape

To add a new file following the existing files on the tape volume:

1. Use the ADDF parameter in a :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
?8:23/#S92/17/MOUNT TAPE VOLUME TLABEL
8:23/12/VOL TLABEL(ANSI) MOUNTED ON LDEV# 7
EOF FOUND IN fromfile AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

Copying from Labeled Tape

Copy a particular file from labeled tape by using its number or its name reference. Copy a series of sequential files using the NEXT parameter of the :FILE command. When reading ANSI labeled tape, it is

usually not necessary to specify the tape file record structure in a file equation because the information is read from the labels.

Copying Files by Number

To copy a file from labeled tape:

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
```

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

```
>FROM=*LT; TO=NEWDISC; NEW
EOF FOUND IN fromfile AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

Copying Named Files

You can also copy a file by name from a labeled tape using its name (if it has one).

To copy a named file from labeled tape:

1. Specify the file name in a :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 your FCOPY command, backreference the :FILE command by preceding the *fromfile* name with an asterisk. For example:

```
>FROM=*LTNAME; TO=FILED
?8:28/#S92/17/MOUNT TAPE VOLUME TLABEL
8:29/12/VOL TLABEL (ANSI) MOUNTED ON LDEV# 7
EOF FOUND IN fromfile AFTER RECORD 9

10 RECORDS PROCESSED * * * 0 ERRORS
```

Sequentially Copying Files

To sequentially copy files from labeled tape:

1. Use the NEXT parameter in a :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 FILES function.

```
>FROM=*LTAPE; TO=CONCAT; FILES=4
78:25/#S92/17/MOUNT TAPE VOLUME TLABEL
8:25/12/VOL TLABEL (ANSI) MOUNTED ON LDEV# 7
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

Labeled tapes with blocked off-byte records can be read and deblocked similarly to unlabeled tapes.

To deblock an odd-byte labeled tape:

In the FCOPY command, set DEBLOCK equal to the original record length. FCOPY adds one byte to each output record. For example:

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

Using Tapes to Copy Files between HP 3000 Systems

You can use magnetic tapes to copy disc files between HP 3000 systems. All files on a single tape must have the same record size and type. First copy the files from the first system to tape. Then copy the tape files onto the second system.

To copy disc files between HP 3000 systems using magnetic tape:

1. Issue a :FILE command describing the tape to which you want to copy the disc files. The record size and structure must be specified. For example:

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

2. Copy each of the disc 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 a :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 disc. 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
```

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

Copying Foreign Tapes

Tapes written on foreign machines often need to be read on HP 3000's. If you don't know the precise 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, character data is probably ASCII, but if the tape is IBM labeled, 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 will read one block from the tape, convert it to ASCII, format it as both octal and characters, and print 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 IBM format 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, disc.) 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 = -132M 20, F, ASCII  
:FILE IBMDATA; REC = -132, F, ASCII; DISC = 12000, 32  
:FCOPY FROM = *LL, TO = *IBMDATA; NEW; EBCDICIN
```

You will need an estimate of the amount of data on the tape. If you underestimate, the disc file will fill up and FCOPY will terminate, leaving some data unread. If you overestimate, you will waste some disc space, but the wastage will be reduced if you specify a large number of extents (32 in the example).

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 by the operator. Next, type `:DSLIN nodename` to open the line. To logon to the remote system, type `:REMOTE HELLO user.account`. Then type the following commands:

```
:FILE output ; DEV = nodename # LP  
:FCOPY FROM = discfile ; 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 ways that you copy regular HP 3000 files. SKIPEOF, IGNERR, and SUBSET, do not apply to KSAM files and KEY and NOKSAM, apply only to KSAM files. Refer to the *KSAM/3000 Reference Manual* (30000-90079) for more information on using KSAM files.

Copying a KSAM File to an Existing KSAM File

Copy a KSAM file to another KSAM file using the FROM and TO parameters. For example:

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

FCOPY copies the file in primary key sequence.

Copying a KSAM File to a New KSAM File

To copy a KSAM file to a new KSAM disc file, include the names of both the new KSAM data and the key files in the TO parameter, as shown in the example below:

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

Copying a KSAM File in a Specified Key Sequence

To copy records from a KSAM file in a particular key sequence, use the KEY function. 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 chronological sequence. For example, the FCOPY command below instructs FCOPY to copy the KSAM file KSAM1 to the KSAM file KSAM2 in the sequence determined by the key beginning in position 21 of KSAM1's data records:

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

To copy a KSAM file in chronological order, specify zero as the character location. For example, the command below copies the file KSAM3 to the file KSAM4 in chronological 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 a KSAM File 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 file KSAM7 to the disc file DATA in 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 file to a non-KSAM file, use the NOKSAM and NOUSERLABELS functions. NOKSAM instructs FCOPY to copy the KSAM data file to a non-KSAM file. NOUSERLABELS instructs FCOPY not to copy the KSAM file's user labels. For example, the command below copies the data file of the KSAM file KSAM8 to the disc file NOKSAM.

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

Copying Inactive Records in Relative I/O and Message Files

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

To exactly copy a RIO or message file:

1. Specify NOBUF and COPY in :FILE commands describing both the *fromfiles* and *tofiles*. For example:

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

2. Backreference the :FILE commands with an asterisk before the *fromfiles* 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 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 more information on using native languages.

Using Native Languages 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 as German characters:

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

Using Native Languages with EBCDICIN and EBCDICOUT

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=MYGEBCL; TO=MYROM8FL; LANG=GERMAN; EBCDICIN  
EOF FOUND IN fromfile AFTER RECORD 29  
  
30 RECORDS PROCESSED * * * 0 ERRORS
```

Using Native Languages with UPSHIFT

UPSHIFT converts lowercase characters to uppercase as you copy them to another file. When you specify a language along 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
```

FCOPY Syntax

This chapter summarizes the syntax of an FCOPY command and the FCOPY functions.

FCOPY Command Syntax

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

where:

- from* is the name of the input file.
- tofile* is the name of the output file.
- (dfile, kfile)* are the names of the data and key files of a new 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.
- <empty>* represents no entry. By entering nothing, you specify the standard job/session device: \$STDIN for the *fromfile*, or \$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 the individual functions is given in Table 4-1.

Syntax of FCOPY Functions

Table 4-1 illustrates the syntax of the different FCOPY functions. 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.

Table 4-1. Syntax of FCOPY Functions

[;NOUSERLABELS]		[;CCTL ;NOCCTL]		[;NEW]	
$\left[; \left\{ \begin{array}{c} \text{CLEAR} \\ \text{KANA} \end{array} \right\} \left[; \left\{ \begin{array}{c} \text{HEX} \\ \text{OCTAL} \\ \text{HEXO} \end{array} \right\} \right] \left[;\text{NORECNUM} \right] \left[;\text{TITLE=} \textit{title} \right] \right]$					
$\left[; \text{CHAR} \left[; \left\{ \begin{array}{c} \text{HEX} \\ \text{OCTAL} \\ \text{HEXO} \end{array} \right\} \right] \left[;\text{NORECNUM} \right] \left[;\text{TITLE=} \textit{title} \right] \left[;\text{LANG=} \textit{language} \right] \right]$					
$\left[; \left\{ \begin{array}{c} \text{HEX} \\ \text{OCTAL} \\ \text{HEXO} \end{array} \right\} \left[; \left\{ \begin{array}{c} \text{CHAR} \\ \text{CLEAR} \\ \text{KANA} \end{array} \right\} \right] \left[;\text{NORECNUM} \right] \left[;\text{TITLE=} \textit{title} \right] \right]$					
[;DEBLOCK= <i>logical-record-length</i>]					
$\left[\begin{array}{l} \text{EBCDICIN} \\ ; \text{EBCDICOUT} \end{array} \right] = \left[\begin{array}{l} \left\{ \begin{array}{l} \textit{field} \\ (\textit{field} \text{ [, } \textit{field} \text{ [, ...]])} \end{array} \right\} \end{array} \right] \left[;\text{EXCLUDE} \right] \left[;\text{LANG=} \textit{language} \right] \right]$					
$\left[\begin{array}{l} \text{BCDICIN} \\ ; \text{BCDICOUT} \\ \text{EBCDIKIN} \\ \text{EBCDIKOUT} \end{array} \right] = \left[\begin{array}{l} \left\{ \begin{array}{l} \textit{field} \\ (\textit{field} \text{ [, } \textit{field} \text{ [, ...]])} \end{array} \right\} \end{array} \right] \left[;\text{EXCLUDE} \right] \right]$					

Table 4-1. Syntax of FCOPY Functions (Continued)

$\left[\text{;FILES= } \left\{ \begin{array}{l} \text{number-of-files} \\ \text{ALL} \end{array} \right\} \right]$	
$\left[\text{;IGNERR[= } \text{number-of-errors} \text{]} \right]$	$\left[\text{;COMPARE[= } \text{number-of-errors} \text{]} \right]$
$\left[\text{;SKIPEOF= } \left[\begin{array}{l} \{+/-\} \text{ from-eofs} \\ \text{from-file-number} \end{array} \right] , \left[\begin{array}{l} \{+/-\} \text{ to-eofs} \\ \text{to-file-number} \end{array} \right] \right]$	
$\left[\text{;SUBSET } \left[\begin{array}{l} = \text{"characterstring"} \text{ [, [column] [,EXCLUDE]} \\ = \text{\#patternlist \#} \text{ [, [column] [,EXCLUDE]} \\ = (\text{range} \text{ [; range] [;...])} \end{array} \right] \right]$	
$[\text{;NOKSAM}]$	
$\left[\text{;KEY [= } \text{character-location} \text{]} \right]$	
$\left[\text{;UPSHIFT [;LANG= } \text{language} \text{]} \right]$	
$\left[\text{;VERIFY [= } \text{number-of-errors} \text{]} \right]$	

FCOPY Functions

This chapter describes each of the FCOPY functions in detail. It presents the functions in alphabetical order, so you can quickly look up information about a function. 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, usage restrictions (if any), instructions for using it, and examples of 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 will be added following the old data.

BCDICIN/BCDICOUT

Function

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

Format

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

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 number 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 copied.

Use

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 is in five character positions, as a packed decimal. The BCDIC fields need to be converted to ASCII, but the packed decimal file should be copied unchanged. Use this command:

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

The *field* and EXCLUDE parameters let you translate specific data within the file. If you do not specify fields or EXCLUDE, the system converts all data within 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 within the fields you specified.

Usage 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 convert 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.

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

Examples

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

```
>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; BCDICIN=(1,5;10:30), EXCLUDE
```

CCTL/NOCCTL

Function

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*.

Format

$\left\{ \begin{array}{l} \text{CCTL} \\ \text{NOCCTL} \end{array} \right\}$
--

Use

The effect of CCTL or NOCCTL depends upon whether or not the file equations, or disc 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 disc label for		DEFAULT EFFECT
<i>fromfile</i>	<i>tofile</i>	
NO	NO	Entire record of <i>fromfile</i> is copied to the <i>tofile</i> . the "to" file.
NO	YES	A single spacing character is supplied; the entire "from" record is copied into the <i>tofile</i> .
YES	NO	The entire "from" record including the car- riage control character is copied into the <i>tofile</i> as data
YES	YES	The entire "from" record is copied into the <i>tofile</i> . The carriage control character from the "from" record becomes the carriage control character for the "to" record.

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

Table 5-3. Copying Files Using CCTL

CCTL specified in the file equation or disc label for		EFFECT
fromfile	tofile	
NO	NO ...	Entire record of <i>fromfile</i> is copied to <i>tofile</i> . No carriage control characters supplied.
NO	YES	Entire record of <i>fromfile</i> copied to <i>tofile</i> . First character of <i>fromfile</i> record interpreted as a carriage control character, not as data.
YES	NO ...	Entire record of <i>fromfile</i> copied as data, including the carriage control character.
YES	YES	Entire record of <i>fromfile</i> copied into <i>tofile</i> , with carriage control character of "from" record used as carriage control character for "to" record.

Table 5-4 shows the effect of using NOCCTL.

Whenever a file equation or disc file label specifies CCTL in a to file and you specify NOCCTL in an FCOPY command, the CCTL specification in the file equation or disc label overrides the FCOPY command (these cases are marked with asterisks in the table).

Table 5-4. Copying Files Using NOCCTL

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

Examples

ALPHA is a disc file, and its label specifies the CCTL option. BETA is a disc file, having the equation:

```
: FILE BETA; NOCCTL
```

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
```

The command below 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:

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

ALPHA is a disc file that has CCTL specified in its label. BETA is a disc file defined by a file equation that specifies NOCCTL.

The FCOPY command below copies each record from ALPHA to BETA including the carriage control character as data:

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

CHAR

Function

Lets you display the contents of a file, record by record, in the form of character symbols in ASCII code. CHAR displays character codes that are not represented by symbols in ASCII code, as decimal points.

Format

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

Where:

OCTAL	displays octal codes. Refer to 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 HEXO description in this section.
NORECNUM	omits file identification information, record numbers, and word offset numbers from the display.
<i>title</i>	is a character string to be used as the title of the display. If the <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters, but may not be broken to be continued on the following line.
<i>language</i>	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. Refer to the <i>Native Language Support Reference Manual</i> (32414-90001).

File Attribute Requirements

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 that of the list device that will ultimately print the data and must be greater than or equal to 60 bytes.

Use

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, for example, when the output device to which you are copying automatically upshifts

lowercase characters, or when you encounter a disc or magnetic tape file with unknown contents. Refer to “Notes” below.

The *tofile* can be a printer, terminal, or an intermediate disc file. If the *tofile* is a disc 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, as 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.

Usage 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 actual (unaltered) character codes in addition to the character symbols.

Example

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

This combined command displays the contents of a file. Control sequences may be chosen that do not affect the display. All escape codes and other non-graphic control characters are shown as dots, allowing the exact position within the file of all the displayable characters to be seen. See also the :CLEAR command.

For example:

:FCOPY FROM=DISPL;TO=,OCTAL;CHAR

HP32212A.03.22 FILE COPIER (C) HEWLETT-PACKARD CO. 1987

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 the a
00010:	064544	020157	063040	072150	062440	070141	071164	074456	id of the party.
00020:	020040	020040	020040	020040	020040	020040	020040	020040	
00030:	SAME: TO 000044-1								

DISPL RECORD 2 (%2, #2)

00000:	015446	062102	052150	064563	020154	064556	062440	064563	.&dBThis line is
00010:	020151	067040	064556	073145	071163	062440	073151	062145	in inverse vide
00020:	067456	015446	062100	020040	020040	020040	020040	020040	o...&d@
00030:	SAME: TO 000044-1								

DISPL RECORD 3 (%3, #3)

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 3

4 RECORDS PROCESSED *** 0 ERRORS

END OF SUBSYSTEM

CLEAR

Function

Lets you display the contents of a file, record by record, in the form of character symbols for all codes in the file. Unlike CHAR, CLEAR does not replace character codes with decimal points. The graphic representation of each code depends upon how the displaying device interprets the code: as a command, displayable character, or non-displayable character. If an output device interprets a code as a command, it displays nothing unless you've entered a "display functions" mode at a terminal that has such a mode.

Format

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

Where:

OCTAL	displays octal codes. Refer to 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 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 <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attribute Requirements

The *fromfile* must have read access. 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 that of the list device that will ultimately print the data, and must be greater than or equal to 60 bytes.

Use

Use CLEAR to display files that contain codes other than ASCII.

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, for example, when 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 disc file. If the *tofile* is a disc file, it 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 as 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.

Usage 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 actual (unaltered) character codes in addition to the character symbols.

Example

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

This combined command 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 such that video enhancements, cursor positioning, and other control functions are made visible. See also the :CHAR command.

For example:

```
:FCOPY FROM=DISPL;TO=;OCTAL;CHAR
```

HP32212A.03.22 FILE COPIER (C) HEWLETT-PACKARD CO. 1987

DISPL RECORD 0 (%0, #0)

```
00000: 047157 073440 085563 020164 064145 020164 064555 062440 Now is the time
00010: 063157 071040 080554 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 the a
00010: 064544 020157 063040 072150 062440 070141 071164 074456 id of the party.
00020: 020040 020040 020040 020040 020040 020040 020040 020040
00030: SAME: TO 000044-1
```

DISPL RECORD 2 (%2, #2)

```
00000: 015446 062102 052150 064563 020154 064556 062440 064563 .&dBThis line is
00010: 020151 067040 064556 073145 071163 062440 073151 062145 in inverse vide
00020: 067456 015446 062100 020040 020040 020040 020040 020040 o..&d@
00030: SAME: TO 000044-1
```

DISPL RECORD 3 (%3, #3)

```
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 3

4 RECORDS PROCESSED *** 0 ERRORS

END OF SUBSYSTEM

COMPARE

Function

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

Format

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

Where:

number-of-errors is an integer designating the number of errors after which FCOPY terminates. Its default value is 1.

File Attribute Requirements

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

Use

The two files may reside on any device, but their record characteristics (record size, record format, blocking factor, 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.

Usage Restrictions

You cannot use COMPARE with any other FCOPY function, except SUBSET, SKIPEOF, and IGNERR.

Examples

In the example below, FCOPY compares the contents of the disc file FILEA against the contents of the disc file DUP1. If it finds an error, 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 against the contents of the disc file DISC2. Each time it finds an error, FCOPY prints a message on the standard listing device. FCOPY terminates the operation after the fifth error.

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



COPYACD

Function

Copies the Access Control Definition (ACD) associated with a file when the file is being copied.

Format

```
;COPYACD
```

Use

If the source file has an ACD and the user is authorized to copy it, the file and the ACD are both copied to the target file.

Usage Restrictions

You cannot use COPYACD with the COMPARE function.

Example

```
> FROM=sourcef; to=targetf; COPYACD
```

NOTE

COPYACD applies only to MPE V Delta 4 and subsequent releases and NOT to MPE XL at this time.

DEBLOCK

Function

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

Format

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

Where:

logical-record-length

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

Use

You use DEBLOCK primarily for copying tapes from other systems to an HP 3000 system. For more information on copying foreign tapes, refer to Chapter 3.

FCOPY assumes 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.

If the *logical-record-length* you specify is an odd number of bytes, FCOPY adds an extra byte to each record in order to facilitate alignment.

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. Note that each output record has 46 bytes. FCOPY added the extra byte to each record after copying it to ensure proper alignment.

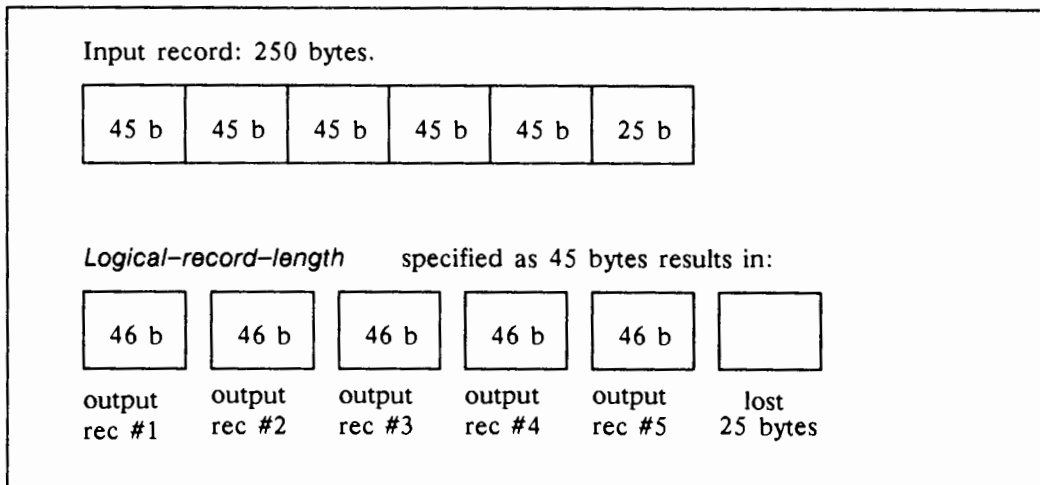


Figure 5-1. Logical Record Length

If you DEBLOCK labeled tapes that have an odd number of bytes in the block, you should 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 disc, grouping records into 80-byte blocks. 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 is 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, grouping records into 79-byte blocks. Since MPE aligns records on word boundaries, FCOPY adds an additional byte to each record in the *fromfile* without changing the contents of the file:

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

EBCDICIN/EBCDICOUT

Function

EBCDICIN translates from EBCDIC to the character code specified in the translation table of the language you select. EBCDICOUT translates 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 from EBCDIC to ASCII, and EBCDICOUT translates from ASCII to EBCDIC.

Format

$$;\left\{ \begin{array}{l} \text{EBCDICIN} \\ \text{EBCDICOUT} \end{array} \right\} \left[= \left\{ \begin{array}{l} \text{field} \\ (\text{field} [, \text{field} [, \dots]]) \end{array} \right\} [;\text{EXCLUDE}] [;\text{LANG= language}] \right]$$

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 number 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 copied.

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).

Use

Use EBCDICIN and EBCDICOUT to translate information from one computer code system to the HP 3000 code system. For example, use EBCDICIN to translate information from IBM codes to HP 3000 codes, but you use 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. Refer to the *Native Language Support Reference Manual* (32414-90001) for more information.

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 *fields* you specified.

Usage 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 *TAPE to the file DISC1, converting characters from EBCDIC to ASCII. It uses the default (NATIVE-3000) translation table:

```
>FROM=*TAPE; TO=DISC1; EBCDICIN=3:7,14:27
```

The second example converts MYGEBFL to MYROM8FL converting from German EBCDIC to Roman8:

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

The example below copies columns 3 through 7 and 14 through 27 of each record in the file DISC1 to the file *TAPE, converting characters from ASCII to EBCDIC. It uses the default (NATIVE-3000) translation table:

```
>FROM=DISC1; TO=*TAPE; EBCDICOUT=3:7,14:27
```

The final example converts MYROM8FL to MYGEBFL, converting from Roman8 to German EBCDIC:

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

EBCDIKIN/EBCDIKOUT

Function

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

Format

$; \left\{ \begin{array}{l} \text{EBCDIKIN} \\ \text{EBCDIKOUT} \end{array} \right\} \left[= \left\{ \begin{array}{l} \text{field} \\ (\text{field } [, \text{field } [, \dots]]) \end{array} \right\} [;\text{EXCLUDE}] \right]$
--

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 number 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 copied.

Use

These functions support the KATAKANA language. Use EBCDIKIN and EBCDIKOUT to translate information to and from EBCDIC and the HP KANA8 character. The mapping between these character sets is defined by the JIS and IBM (refer to the *Native Language Support Reference Manual* (32414-90001)). Use these functions, for example, when 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 translate. 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 *fields* you specified.

Usage 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 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 FILE1 to the file FILE2 converting the columns from EBCDIK to JIS codes.

```
>FROM=FILE1; TO=FILE2; 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

Function

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

Format

```
;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.

Use

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

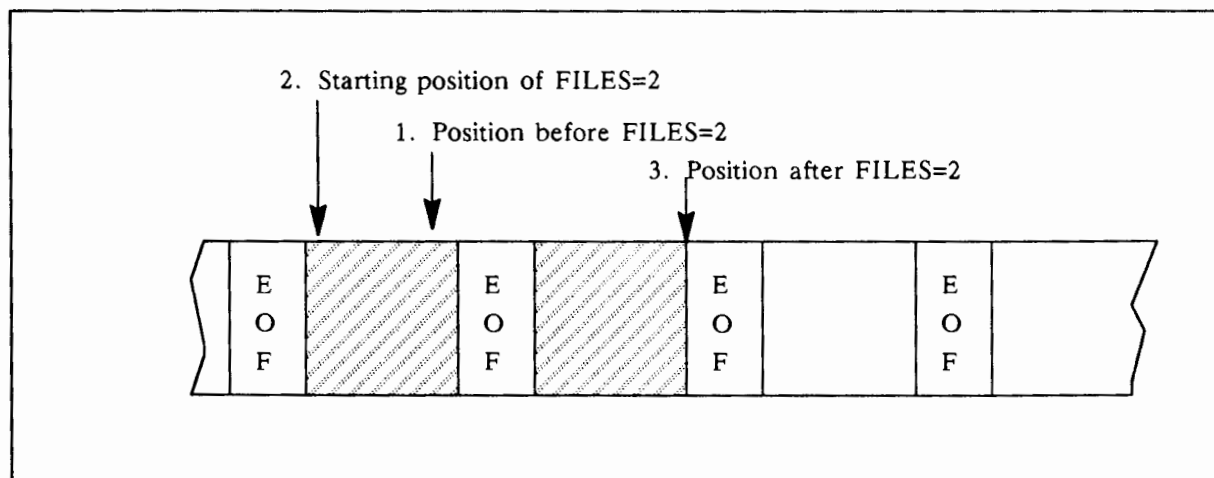


Figure 5-2. Tape Position

If you use FILES 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
```

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

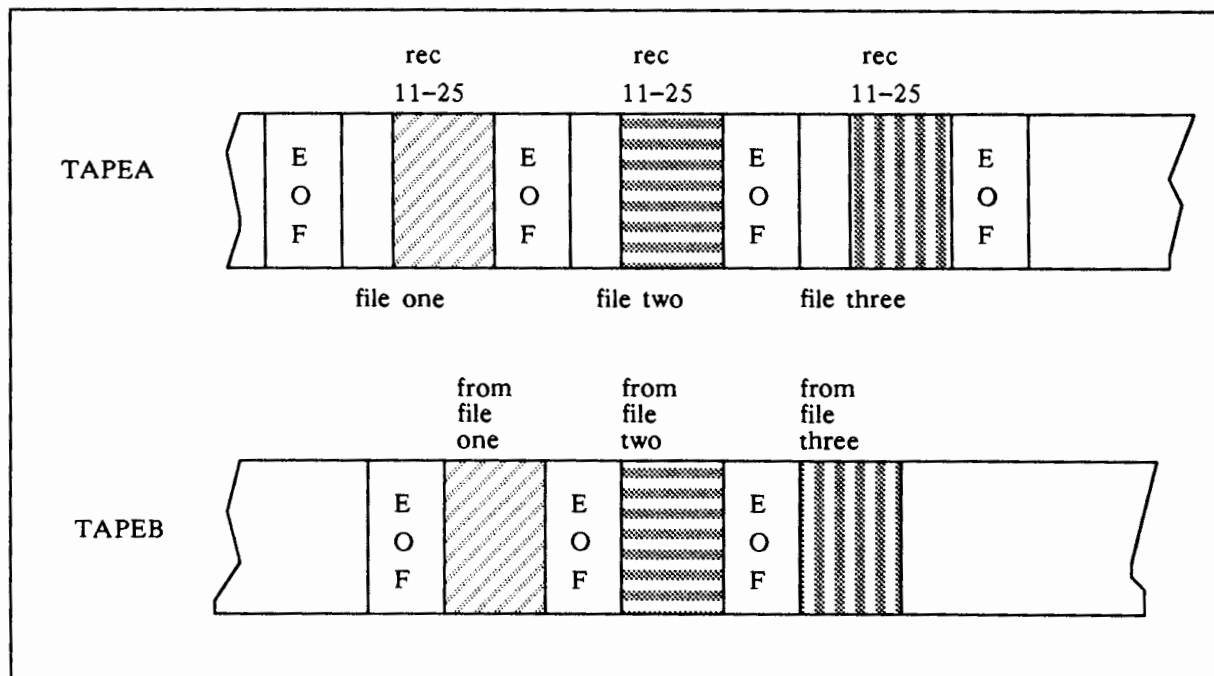


Figure 5-3. Copying Subsets of Files to Tape

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 tape's file equation. For example, the commands below copy subsets of three files from labeled tape TAPEALFA to TAPEBETA as shown in the Figure 5-4.

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

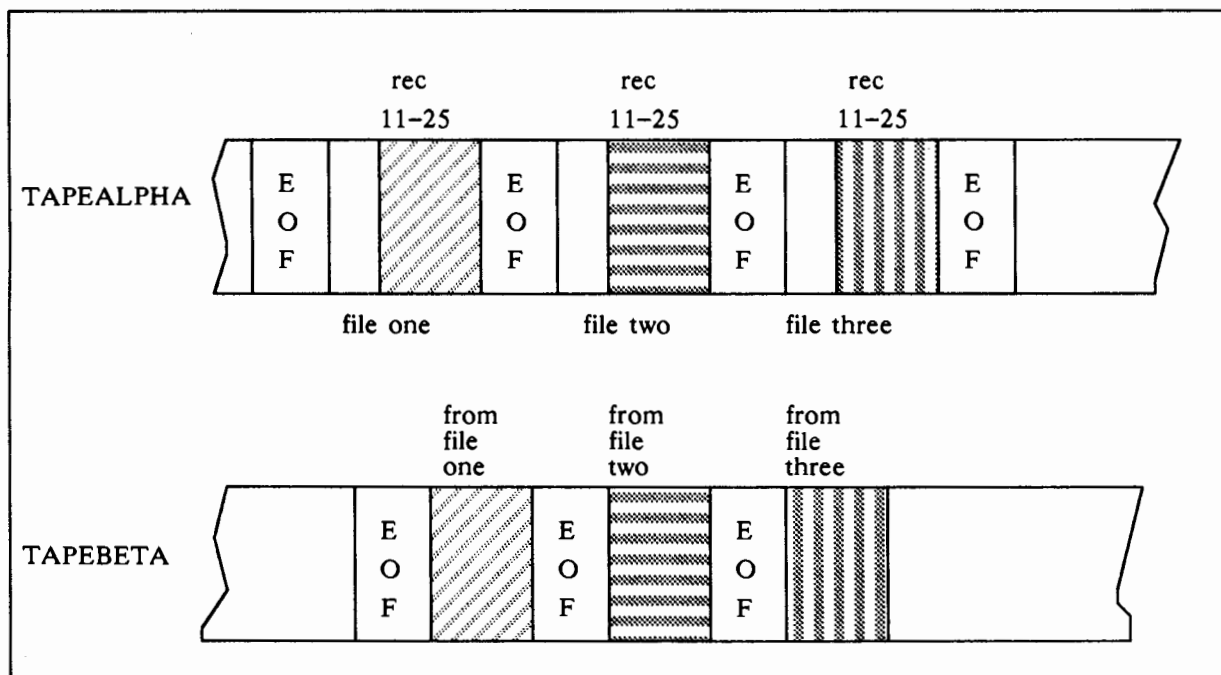


Figure 5-4. Copying Subsets of Files to Labeled Tape

If you do not use the NEXT sequence option for the labeled tape, FCOPY copies from the same file to three new ones. 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
```

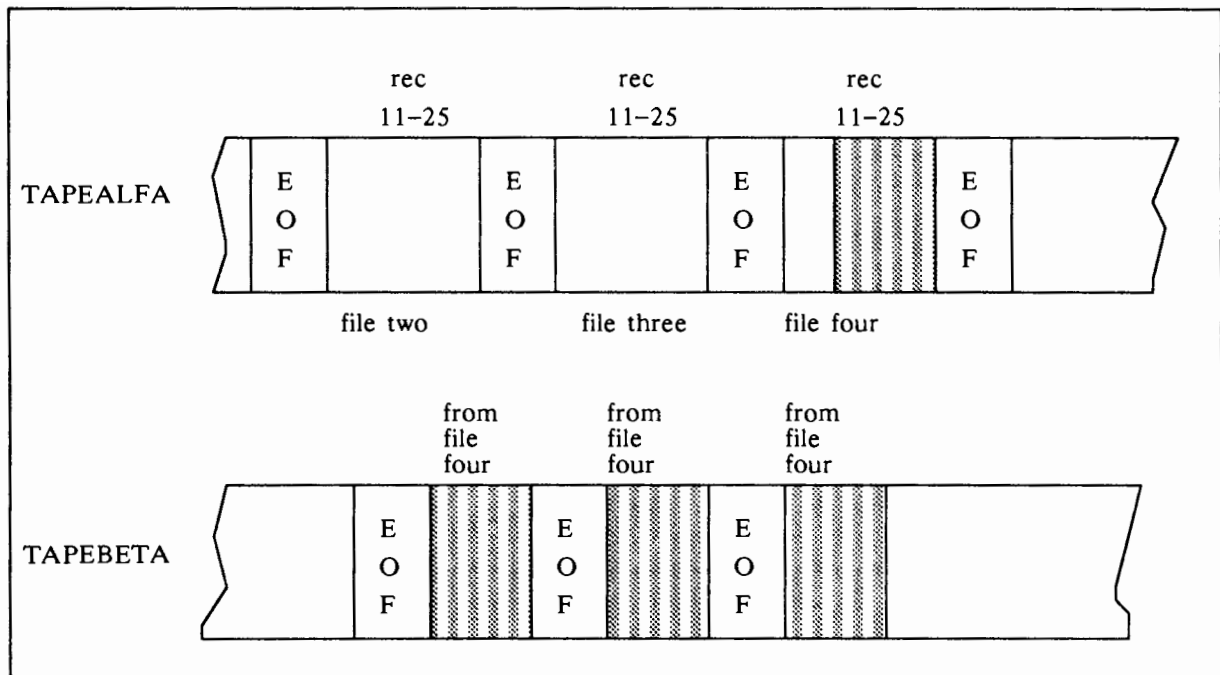


Figure 5-5. Copying from a Single Labeled Tape File

For more information on the sequence option of the :FILE command, refer to the :FILE command in the *MPE Commands Reference Manual* (32033-90006) or *MPE XL Commands Reference Manual* (32650-90003). Refer to Chapter 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 "T1 to magnetic tape "T7:

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

The second example combines the SKIPEOF and FILES functions. FCOPY positions "T5 at its third file, positions "T2 four files forward from its current position, and then copies two files (files three and four) from *T5 to *T2:

```
>FROM=*T5; TO=*T2; 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

Function

Lets you display the contents of a file, record by record, in the form of character code numbers in hexadecimal form.

Format

; HEX		[; { CHAR CLEAR KANA }]		[;NORECNUM]	[;TITLE= <i>title</i>]
-------	--	---------------------------------	--	-------------	-------------------------

Where:

CHAR	displays character symbols. Refer to 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.
<i>title</i>	is a character string to be used as the title of the display. If the <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attribute Requirements

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 that of the list device that will ultimately print the data ,and must be greater than or equal to 60 bytes.

Use

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 disc file. If the *tofile* is a disc file, it must be at least 60 bytes in length.

Example

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 example below 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="TITLE LINE FOR CHAR/HEX DISPLAY EXAMPLE"
```

TITLE LINE FOR CHAR/HEX DISPLAY EXAMPLE

TEXT3.PUB.ACCOUNT	RECORD	O (%O)	
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

Function

Lets you display the contents of a file, record by record, in the form of character code numbers, the data in hexadecimal form, and the record number in octal form.

Format

<code>; HEXO [; { CHAR CLEAR KANA }] [;NORECNUM] [;TITLE= <i>title</i>]</code>

Where

CHAR	displays character symbols. Refer to CHAR description in this chapter.
CLEAR	displays all character symbols. Refer 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.
<i>title</i>	is a character string to be used as the title of the display. If the <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters, but may not be broken to be continued on the following line.

File Attribute Requirements

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 that of the list device that will ultimately print the data and must be greater than or equal to 60 bytes.

Use

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 disc file. If the *tofile* is a disc 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 below 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 line described by the title parameter.

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

TITLE LINE FOR CHAR/HEX DISPLAY EXAMPLE

TEXT3.PUB.ACCOUNT RECORD 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

Function

Bypasses errors in a magnetic tape *fromfile* and reports each ignored error.

Format

```
;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 Attribute Requirements

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

Use

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 a maximum of five errors while copying the first file on the magnetic tape unit *TAPE to the disc file FILE3. The operation terminates if FCOPY encounters a sixth error:

```
>FROM=*TAPE, TO=FILE3; SUBSET; IGNERR=5
```

KANA

Function

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.

Format

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

Where:

OCTAL	displays octal codes. Refer to 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 number in octal. Refer to HEXO description in this chapter.
NORECNUM	omits file identification information, record numbers, and word offset numbers.
<i>title</i>	is a character string to be used as the title of the display. If the <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters but may not be broken to be continued on the following line.

File Attribute Requirements

The *fromfile* must have read access, whereas 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 that of the list device that ultimately prints the data and must be greater than or equal to 60 bytes (30 two-byte words).

Use

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, for example, when 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 disc file. If the *tofile* is a disc file, it 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 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.

KEY

Function

KEY lets you choose a key sequence in which to copy KSAM files. The KEY function works only with KSAM *fromfiles*.

Format

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

Where:

character location

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

File Attribute Requirements

The *fromfile* must be a KSAM file.

Use

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 chronological sequence (the sequence in which the records were originally written to the file) rather than key sequence.

Usage Restrictions

KEY cannot be combined 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).

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 final example below copies the KSAM1 file in chronological sequence, the sequence in which the records were written to the file:

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

NEW

Function

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

Format

```
;NEW
```

Use

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

File Characteristics

The characteristics of the new disc file are summarized in Appendix B. If you want to predefine the characteristics of the new file, define them with an MPE :FILE command and then backreference that file in an 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 CONTROL-Y, FCOPY saves the new file.

Usage Restrictions

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

Examples

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

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

NOKSAM

Function

Copies the data file of a KSAM file into another, non-KSAM file.

Format

```
;NOKSAM
```

File Attribute Requirements

The *fromfile* must be a KSAM file. The *tofile* cannot be a KSAM file.

Use

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

Usage Restrictions

You cannot use NOKSAM with the KEY function.

Notes

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

For a complete description of KSAM files and how they operate with FCOPY, refer to the *KSAM 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 (records with the octal number 377 in column 1) in the KSAM file KSAMONE on the standard list device.

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

NOUSERLABELS

Function

NOUSERLABELS lets you omit user labels when copying from a tape or disc file to another file.

Format

```
;NOUSERLABELS
```

Use

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

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

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 disc file named DISC1 without copying the user label of the tape file.

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

The command below copies the second file on TAPEA, including user labels, to the disc file DISC2. The last 176 bytes of DISC2's user label are zeros.

```
>FROM=*; TO=DISC2; SKIPEOF=2
```

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

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

OCTAL

Function

Lets you display the contents of a file, record by record, in the form of character code numbers in octal form.

Format

<code>; OCTAL</code>	<code>[</code>	<code>{</code>	<code>CHAR</code>	<code>}</code>	<code>]</code>	<code>[;NORECNUM]</code>	<code>[;TITLE=</code>	<code>title</code>	<code>]</code>
----------------------	----------------	----------------	-------------------	----------------	----------------	--------------------------	-----------------------	--------------------	----------------

Where:

CHAR	displays character symbols. Refer to 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.
<i>title</i>	is a character string to be used as the title of the display. If the <i>tofile</i> 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 <i>tofile</i> . The title may contain up to 62 characters but may not be broken to be continued on the following line.

File Attribute Requirements

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 that of the list device that ultimately prints the data and must be greater than or equal to 60 bytes.

Use

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 disc file. If the *tofile* is a disc 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.

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

The command above produces the following display on a line printer:

```
TEXT3.PUB.ACCOUNT RECORD 9 (%11)
```

```
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 10 (%12)
```

```
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 11 (%13)
```

```
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="TITLE LINE FOR CHAR/OCTAL DISPLAY EXAMPLE"
```

The previous command produces the following display. The numbers at the right are text editor line number that start at 1.

TITLE LINE FOR CHAR/OCTAL DISPLAY EXAMPLE

TEXT3.PUB.ACCOUNT RECORD 9 (%11)

00000:	052110	044523	020111	051440	046111	047105	020081	027040	THIS IS LINE 1
00010:	020124	044105	020070	026504	044507	044524	020118	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	030080	030080	030480	030080	00001000

TEXT3.PUB.ACCOUNT RECORD 10 (%12)

00000:	052105	054124	020105	042111	052117	051040	046111	047105	TEXT EDITOR LINE
00010:	020118	052515	041105	051123	027040	020118	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	030080	030080	031080	030080	00002000

TEXT3.PUB.ACCOUNT RECORD 11 (%13)

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 E
00030:	042111	052117	051040	046111	047105	020040	020040	020040	DITOR LINE
00040:	020040	020040	020040	020040	030080	030080	031480	030080	00003000

SKIPEOF

Function

SKIPEOF instructs FCOPY to skip end-of-file markers on a serial storage device, in order to position the device at the desired file before copying.

Format

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

Where:

<i>+/-</i>	specifies forward (+) or reverse (-) movement.
<i>from-eofs</i>	is a positive integer specifying how many files to skip on the "from" device.
<i>from-file-number</i>	is an integer specifying an absolute file number on the "from" device. Absolute file numbers begin with 1.
<i>to-eofs</i>	is a positive integer specifying how many files to skip on the "to" device.
<i>to-file-number</i>	is an integer specifying an absolute file number on the "to" device.

File Attribute Requirements

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

Use

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

Absolute File Numbers

In the syntax description of 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 disc. Tape and serial disc 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*.

When you specify a *from-file-number* or a *to-file-number*, SKIPEOF 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.

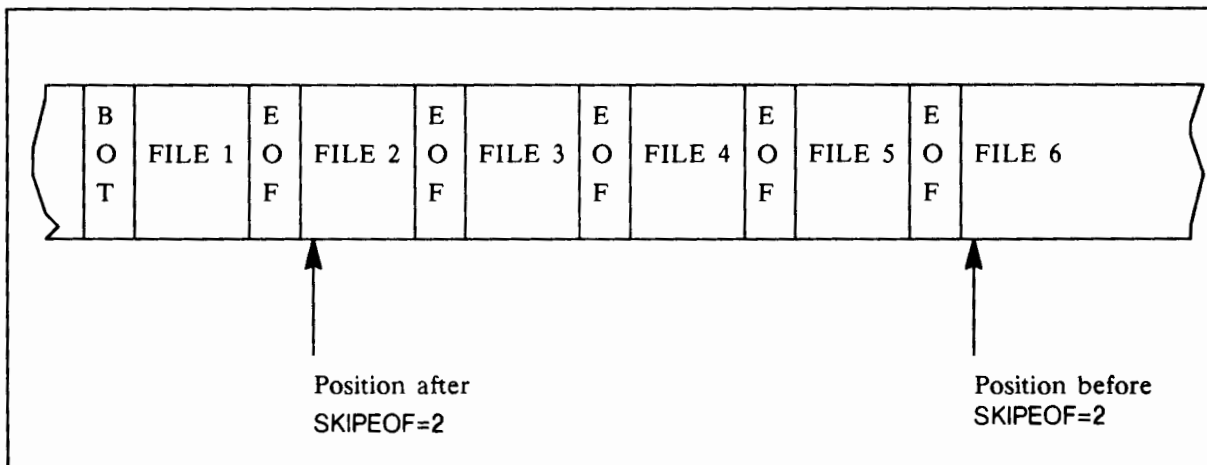


Figure 5-6. Skipping Absolute File Numbers

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 disc.

Number of Files to Skip

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, SKIPEOF skips forward over the specified number of end-of-file (EOF) marks. Figure 5-7 shows the effect of specifying SKIPEOF=+3.

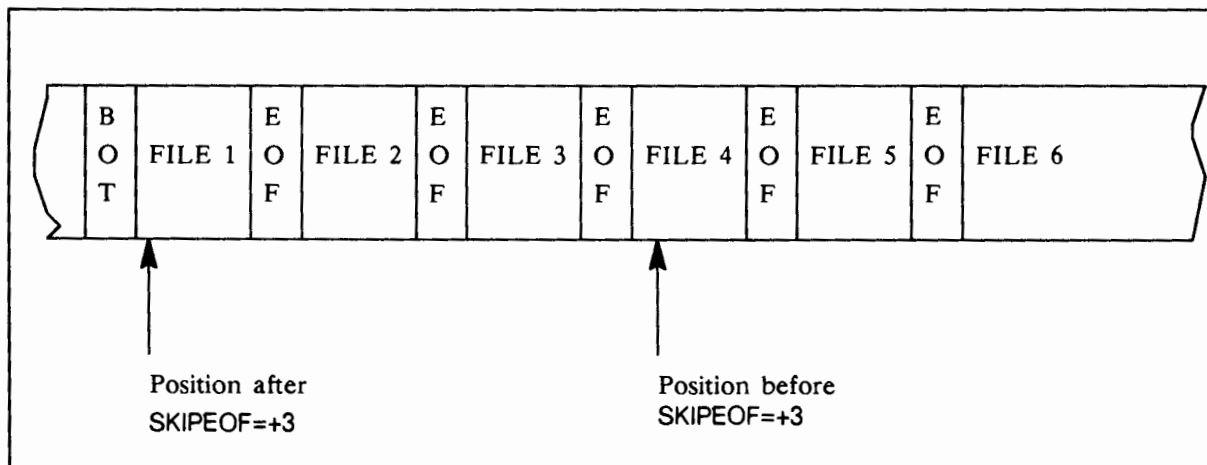


Figure 5-7. SKIPEOF = +3

If you use a minus sign (-) to specify backward movement, SKIPEOF skips backward over the specified end-of-file marks plus 1, and then skips forward 1 end-of-file mark. This method 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.

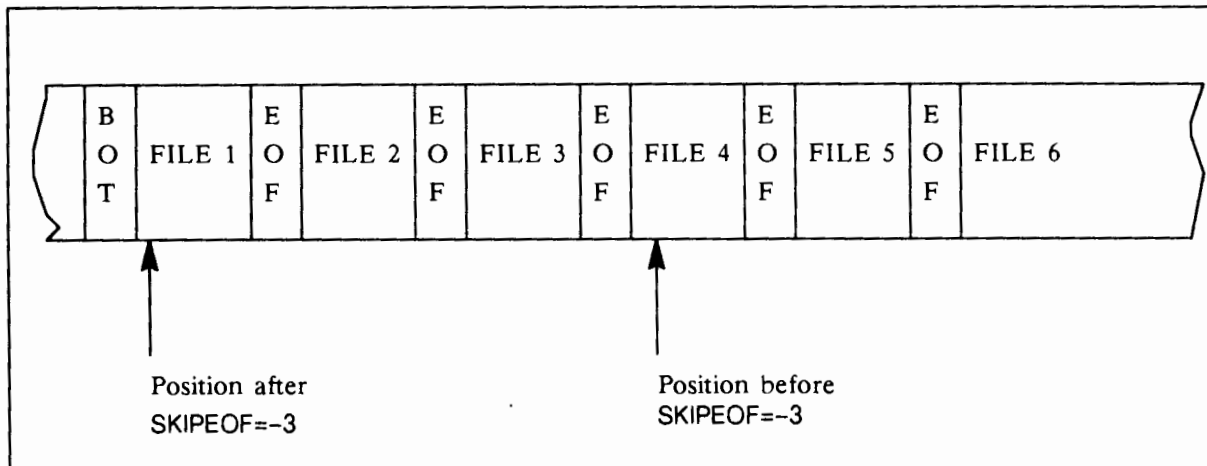


Figure 5-8. SKIPEOF = -3

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.

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
```

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, *TAPES's current position is at the second file mark, FCOPY begins to copy from the sixth file.

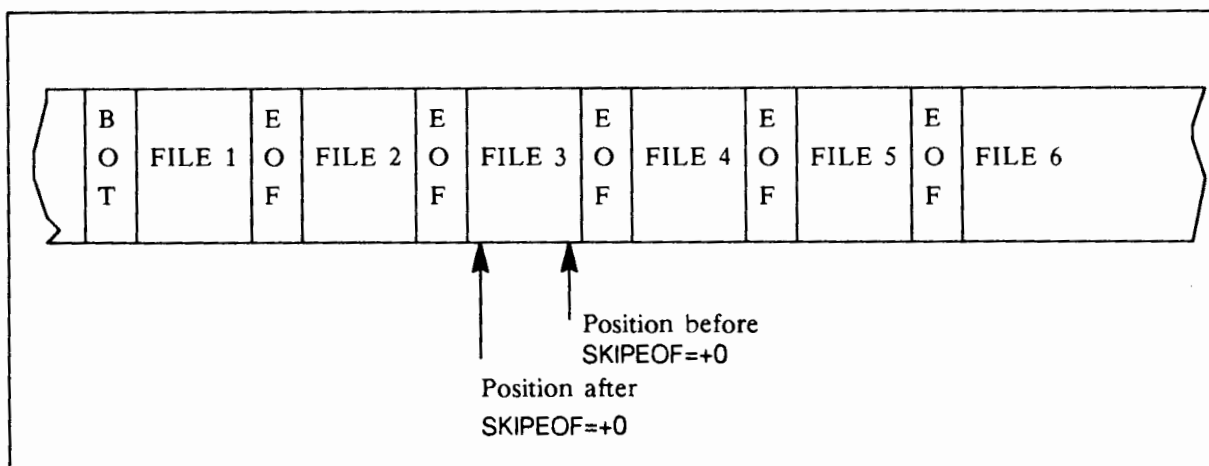


Figure 5-9. SKIPEOF = +0

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 five 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 two of *TAPES, the from device in the previous example, to file eight 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

Function

SUBSET lets you copy only a specific portion (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 of continuous records.

Format

<code>;SUBSET</code>	$\left[\begin{array}{l} = " \textit{characterstring} " [, [\textit{column}] [, \textit{EXCLUDE}] \\ = \# \textit{patternlist} \# [, [\textit{column}] [, \textit{EXCLUDE}] \\ = (\textit{range} [; \textit{range}] [; \dots]) \end{array} \right]$
----------------------	---

Where:

<i>characterstring</i>	is a string of ASCII characters.
<i>patternlist</i>	is a string of decimal and/or octal integers, each representing the bit pattern of one byte.
<i>column</i>	is an integer specifying at which character position within each record the particular <i>characterstring</i> or <i>patternlist</i> is expected to begin. If omitted, FCOPY assumes the default value, 1 (first character position).
EXCLUDE	specifies that the designated subset is to be excluded from the FCOPY operation (all of the file, except for the subset, is to be copied).
<i>range</i>	is either of the form: [<i>starting-record-number</i>] [, <i>number-of-record</i>] or [<i>starting-record-number</i>] [: <i>last-record-number</i>]
<i>starting-record-number</i>	is an integer record number specifying the start of the subset. If omitted, the default value 0 (first record) is assumed.

NOTE

If translating EBCDIC to ASCII "Packed Fields (comp-3)" should NOT be translated.

number-of-records

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

last-record-number

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

Use

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 of continuous 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. 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 character string or pattern list specified does not exist entirely within a particular record in the *fromfile*, the record is considered not 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 listed them. They must be in numerical order, and they must not overlap.

Using Subset Alone

You can use SUBSET alone to copy a disc file to a duplicate that uses less disc space. If you use SUBSET to copy from a disc file that does not fill the space allocated to it to a new disc file, the resulting *tofile* is only as large as actual contents of the *fromfile*. FCOPY does not copy the unused space. For example, the command below copies only 20 records from DISC1:

```
>FROM=DISC1; TO=DISC2; NEW; SUBSET
EOF FOUND IN FROMFILE AFTER RECORD 19

20 RECORDS PROCESSED * * * 0 ERRORS
```

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

```
:LISTF DISC@,2
ACCOUNT=  SUBSYS      GROUP=EDITOR

FILENAME CODE-----LOGICAL RECORD-----      ----SPACE-----
          SIZE    TYPE          EOF    LIMIT  R/B    SECTORS #X  MX
DISC1      80B     FA           20     1023   1      128     1   8
DISC2      80B     FA           20      20    1       21     7   7
```

Copying Only User Labels

Use SUBSET=0,0 to copy only user labels from a disc file. For example:

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

The command above copies the user labels from the disc file LABELDISC to the magnetic tape *TAPE.

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 end-of-file (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 disc file MASTER that contain the character string MALE starting in character position 17, to the disc file MEN:

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

The following example copies all records in the file MASTER that do not contain the character string MALE in position 17 to the disc file WOMEN:

```
>FROM=MASTER; TO=WOMEN; SUBSET="MALE",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 disc file DATA that contain the lowercase alphabetic characters "data" (octal codes 144, 141, 164, and 141) in character positions 77 through 80 to the disc file SUBFILE3:

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

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

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

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

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

The next example copies records 500 through 1499 of the disc file MASTER into the disc file SUB2:

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

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

```
>FROM=MASTER; 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;30,75)
```

In the example below, the ranges are incorrectly specified. The first is not numerically ordered, and the second overlaps the first:

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

UPSHIFT

Function

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

Format

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

Where:

language the name or number of a supported language. FCOPY uses the translation table associated with that language. The default is NATIVE-3000. Refer to the *Native Language Support Reference Manual* (32414-90001) for more information.

Use

You use the *language* parameter to describe the language conventions you want to use in converting characters. You can use any language installed on your system. If you do not specify a language, FCOPY uses the default, 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 FCOPY UPSHIFT function. If you are copying to a device that prints uppercase characters only and does not automatically upshift, then use UPSHIFT.

Examples

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

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

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

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

VERIFY

Function

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 disc or magnetic tape and must have identical record characteristics; record size, record format (fixed, variable, or undefined), blocking factor, and record type (ASCII or binary).

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

Format

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

Where:

number-of-errors

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

File Attribute Requirements

Both the *fromfiles* and *tofiles* must be disc files or magnetic tape files. The *fromfile* must have read access and the *tofile* must have read/write access.

Use

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.

Usage Restrictions

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

Examples

In the example below, FCOPY copies the disc file OLDDISC to the disc file COPY 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=OLDDISC; TO=COPY; 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
```



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 disc 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.

Terminal Peripheral File Designators

You specify a terminal peripheral 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. Terminal Peripheral Designators

Peripheral Type	HP 3000 Terminals	File Designator
Left cartridge tape unit	HP 2640 HP 2641 HP 2645	\$CTUL
Right cartridge tape unit	HP 2647A HP 2648 HP 2649	\$CTUR \$HARD
Line printer	HP 2621, HP 2622 HP 2626	
Floppy disc drive	HP 2644 HP 2647F	

For example, use the following command to copy a disc 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've specified in a command that references terminal peripherals. If your terminal type is not set correctly, FCOPY prompts you to log on again and set the terminal type with the TERM parameter. Refer to Table 7-3 in the *System Operation and Resources Management Reference Manual* (32033-90005) for a complete list of terminal types. 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 Peripherals

Although you can use FCOPY to copy from one terminal peripheral 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 peripherals.

The command below copies the first file on the right cartridge tape unit of an HP 2645 terminal to the disc file named DISC1:

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

The command in the second example, displays the disc file named DISC4 on the HP 2644A terminal line printer in both octal and character formats:

```
>FROM=DISC4; 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
```



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 to WARN. Status messages supply you with information while FCOPY is copying a file.

Error Message Display During a Session

During a session, FCOPY displays a message and its reference number on your terminal. For example:

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

If the error involves a file, FCOPY asks you whether you want to display 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 , or answer NO by pressing N and . If you answer YES, FCOPY displays the corresponding file system error message and file information. Answering YES in the example above, results in the display shown below:

```
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 Intrinsic Reference Manual* (32033-90007) or *MPE XL Intrinsic Reference Manual* (32650-90028) for a complete description of the file information display.

Error Message Display During a Job

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.

FCOPY Messages

Table A-1 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.

Errors cause FCOPY to terminate in a job stream, however, only the command is terminated in a session.

Warnings do not cause an FCOPY command to terminate. In a job, they set the JCW (Job Control Word) to WARN.

Status messages do not indicate errors, but simply provide information about what is happening during the copying operation.

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

Table A-1. FCOPY Error, Warning, and Status Messages

MESSAGE	<CONTROL-Y>	
CAUSE	Acknowledges receipt of a CONTROL-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:	Re-submit the job.
	In a session:	Re-enter the command.
MESSAGE	WRITE ERROR TO COMMAND LISTFILE	
CAUSE	An error occurred while writing an FCOPY message to \$STDLIST.	
ACTION	More than likely nothing serious has occurred and all FCOPY operations have been performed successfully. If you want to be sure, however, do the following:	
	In a job:	Re-submit the job.
	In a session:	Re-enter 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.	
MESSAGE	SYNTAX ERROR: IN SUBSET OPTION	
CAUSE	The subset function was not specified properly.	
ACTION	In a job:	Correct the command and re-submit the job.
	In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter the command using the correct format.
9	MESSAGE	SNYNTAX ERROR: IN NEW OPTION	
	CAUSE	The new file function was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
10	MESSAGE	SYNTAX ERROR: IN HEX OPTION	
	CAUSE	The display hexadecimal function was not specified.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.

11	MESSAGE	SYNTAX ERROR: IN EBCDICOUT OPTION	
	CAUSE	The EBCICOUT character translate function was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter the command using the correct format.
15	MESSAGE	SYNTAX ERROR: IN BCDICIN OPTION	
	CAUSE	The BCDICIN character translate function was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter the command using the correct format.

17	MESSAGE	SYNTAX ERROR: IN EBCDICIN OPTION	
	CAUSE	The EBCDICIN character translate function was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
18	MESSAGE	SYNTAX ERROR: IN BCDICOUT OPTION	
	CAUSE	The BCDICOUT character translate function was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter 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 <i>characterstring</i> specified for the subset function is not valid.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
52	MESSAGE	SYNTAX ERROR: IN BIT PATTERN	
	CAUSE	The <i>patternlist</i> specified for the subset function is not valid.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
53	MESSAGE	SYNTAX ERROR: INVALID INTEGER	
	CAUSE	An integer specified is outside the range allowed for the particular FCOPY function.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter the command using the correct format.
55	MESSAGE	SYNTAX ERROR: IN FROMFILE SPECIFIER	
	CAUSE	The <i>fromfile</i> was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
56	MESSAGE	SYNTAX ERROR: INTOFILE SPECIFIER	
	CAUSE	The <i>tofile</i> was not specified properly.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.
57	MESSAGE	SYNTAX ERROR: ILLEGAL COMBINATION OF OPTIONS	
	CAUSE	Two or more <i>functionlist</i> entries conflict with one another.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter 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 re-submit the job.
		In a session:	Re-enter the command using the correct format.
59	MESSAGE	SYNTAX ERROR: ILLEGAL USE OF NEW	
	CAUSE	The context used to specify a new <i>tofile</i> is not valid.	
	ACTION	In a job:	Correct the command and re-submit the job.
		In a session:	Re-enter the command using the correct format.

60	MESSAGE	SYNTAX ERROR: ILLEGAL USE OF*
	CAUSE	The context used to specify * as a <i>fromfile</i> or <i>tofile</i> is not valid.
	ACTION	In a job: Correct the command and re-submit the job. In a session: Re-enter the command using the correct format.
62	MESSAGE	SYNTAX ERROR: FILE NAME TOO LONG
	CAUSE	The <i>fromfile</i> or <i>tofile</i> name specified is longer than the 35 characters allowed in a fully qualified file name with lockword.
	ACTION	In a job: Correct the command and re-submit the job. In a session: Re-enter the command using the correct format.
102	MESSAGE	CANNOT CLOSE FROMFILE
	CAUSE	MPE cannot close the <i>fromfile</i> . 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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.
103	MESSAGE	CANNOT SAVE NEW TOFILE
	CAUSE	MPE cannot close the <i>tofile</i> . 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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.
104	MESSAGE	CANNOT SAVE NEW TOFILE
	CAUSE	MPE cannot close the <i>tofile</i> 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 <i>fromfile</i> . 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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.

106	MESSAGE	CANNOT OPEN TOFILE
	CAUSE	MPE cannot open the <i>tofile</i> . 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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.
107	MESSAGE	VERIFY OR COMPARE OPTION: Cannot GET READ ACCESS TO TOFILE
	CAUSE	MPE cannot get read access to the <i>tofile</i> for a VERIFY or COMPARE operation. The read access specified in the file label has been overridden by an MPE :FILE command containing ACC=APPEND, ACC=OUT, or ACC=OUTKEEP
	ACTION	Reset the particular :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 <i>fromfile</i> '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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.
109	MESSAGE	ERROR IN CALLING FGETINFO FOR TOFILE
	CAUSE	An error prevented MPE from obtaining information from the <i>tofile</i> '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 <i>MPE Intrinsic Reference Manual</i> (30000-90010) and act accordingly.
110	MESSAGE	IGNERR OPTION: FROMFILE NOT TAPE
	CAUSE	The <i>fromfile</i> '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 <i>fromfile</i> . 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 particular :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 <i>tofile</i> . The write access specified in the file label 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 <i>fromfile</i> , and the <i>fromfile</i> device is not a magnetic tape unit, serial disc, or cartridge tape unit.
	ACTION	If the intended <i>fromfile</i> or <i>tofile</i> is on magnetic tape or serial disc, 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 <i>tofile</i> , and the <i>tofile</i> device is not a magnetic tape unit, serial disc, or cartridge tape unit.
	ACTION	If the intended <i>fromfile</i> or <i>tofile</i> is on magnetic tape or serial disc, check the associated MPE : FILE command and its backreference.
115	MESSAGE	SUBSET OPTION: STRING FALLS OUTSIDE OF FROMFILE RECIZE
	CAUSE	The <i>characterstring</i> or <i>patternlist</i> specified is greater than the record size of the <i>fromfile</i> . No such subset can exist in the specified <i>fromfile</i> .
	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 requested operation. 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 will accommodate your record size in the maximum allowed data segment size.

117	MESSAGE	SKIPEOF OPTION: ERROR WHILE SKIPPING IN FROMFILE
	CAUSE	An error occurred while skipping end-of-file marks in the <i>fromfile</i> .
	ACTION	Retry the operation.
118	MESSAGE	SKIPEOF OPTION: ERROR WHILE SKIPPING IN TOFILE
	CAUSE	An error occurred while skipping end-of-file marks in the <i>tofile</i> .
	ACTION	Retry the operation.
119	MESSAGE	SUBSET OPTION: ERROR WHILE SPACING IN FROMFILE
	CAUSE	An error occurred while skipping end-of-file marks in the <i>fromfile</i> .
	ACTION	Retry the operation.
120	MESSAGE	SUBSET OPTION: SUBSET STARTS OVER EOF BOUNDARY
	CAUSE	The subset specified 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 <i>fromfile</i> , but the device for that file is not a disc or magnetic tape.
	ACTION	Check the MPE : FILE command associated with the <i>fromfile</i> and its backreference.
124	MESSAGE	READ ERROR IN FROMFILE AT RECORD <i>recnum</i>
	CAUSE	An error occurred while spacing through the <i>fromfile</i> in search of the start of a subset.
	ACTION	Retry the operation.
125	MESSAGE	SUBSET OPTION: NUMERIC SUBSET IS EMPTY
	CAUSE	A subset specified by <i>starting-record-number</i> , <i>numer-of-records</i> , and/or <i>last-record-number</i> 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 <i>fromfile</i> at the start of a verify operation.
	ACTION	Retry the operation.
127	MESSAGE	VERIFY OPTION: ERROR WHILE REWINDING TOFILE
	CAUSE	An error occurred while spacing backward to the beginning of the <i>tofile</i> at the start of a verify operation.
	ACTION	Retry the operation.
128	MESSAGE	EOF FOUND WHILE SPACING IN FROMFILE
	CAUSE	An end-of-file mark was encountered while spacing through the <i>fromfile</i> in search of the start of a subset.
	ACTION	Make sure that the subset starting record number 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 <i>tofile</i> in search of the start of a subset during a compare or verify operation.
	ACTION	Compare operation: The <i>fromfile</i> and <i>tofiles</i> are not identical. Display the <i>tofile</i> to determine what it actually contains. 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 <i>tofile</i> .
	ACTION	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 <i>fromfile</i> during a verify operation.
	ACTION	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 <i>tofile</i> during a verify operation.
	ACTION	Retry the operation.
134	MESSAGE	FOUND EOF IN TOFILE
	CAUSE	FCOPY has performed the specified operation but has filled the <i>tofile</i> before completing the operation.
	ACTION	The <i>tofile</i> was not large enough. Use the MPE :LISTF <i>filename</i> ,2 command to determine the <i>tofile</i> 's size and then increase its size (using the MPE :PURGE and :BUILD commands) and retry the operation. To copy from a file to a smaller file, use SUBSET=.
135	MESSAGE	WRITE ERROR TO TOFILE
	CAUSE	An error occurred while writing to the <i>tofile</i> .
	ACTION	Retry the operation.
136	MESSAGE	READ ERROR FROM TOFILE
	CAUSE	An error occurred while reading from the <i>tofile</i> during a compare or verify operation.
	ACTION	Compare operation: Retry the operation. If the error persists, you must try to recreate the <i>tofile</i> . Verify operation: Retry the operation.

137†	MESSAGE	WARNING: AN UNLABELED TAPE OPERATION ENDS ON AN ERROR
	CAUSE	An operation involving a magnetic tape <i>fromfile</i> 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= <i>n</i> , where <i>n</i> 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 extended over more than one line (record).
	ACTION	In a job: Correct the command and re-submit the job. In a session: Re-enter 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 > 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 "to" and "from" files are not identical.
	ACTION	Compare operation: None. The compare operation revealed that the fixed record sizes of the two files are not identical. Verify operation: Change the record size of the "to" file (using the MPE :PURGE and :BUILD commands) so that it is the same as that of the "from" file and then retry the operation.
141‡	MESSAGE	COMPARE BEGINS
	CAUSE	The comparison phase of a verify operation has begun.
	ACTION	None.

† WARNING
‡ STATUS

143†	MESSAGE	WARNING: FROMFILE IS EMPTY
	CAUSE	The "from" file contained no data. Nothing was copied or compared.
	ACTION	None. You may have accidentally specified the wrong file as the "from" file.
144	MESSAGE	NEW OPTION: FILE ALREADY EXISTS
	CAUSE	The "to" file named for the new file function already exists in the specified (or implied) group and account.
	ACTION	Change the name of the "to" file and try the operation again.
145	MESSAGE	BACKSPACE ERROR IN FROMFILE
	CAUSE	An error occurred while spacing backward to the beginning of the "from" file or a subset within it.
	ACTION	Retry the operation.
146	MESSAGE	CANNOT OPEN TEMPORARY LABEL FILE
	CAUSE	Possibly 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	Retry the operation.
148	MESSAGE	READ ERROR IN TEMPORARY LABEL FILE
	CAUSE	An error occurred while reading from the "from" file.
	ACTION	Retry the operation.
149	MESSAGE	OVERFLOW IN TEMPORARY LABEL FILE
	CAUSE	There are more labels in the "from" file than can be copied.
	ACTION	Use the no user labels function to copy files without labels.

† WARNING

150	MESSAGE	READ ERROR IN FROMFILE LABELS
	CAUSE	An error occurred while reading labels in the "from" file.
	ACTION	Retry the operation.

151	MESSAGE	WRITE ERROR IN TOFILE LABELS
	CAUSE	An error occurred while reading labels in the "to" file.
	ACTION	Retry the operation.

152	MESSAGE	OVERFLOW IN TOFILE LABELS
	CAUSE	There are too many labels in the "from" file to be copied to the "to" file.
	ACTION	If the "to" file 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 FROMFILEBOT
	CAUSE	The negative number specified in a skip end-of-file function for the "from" file is too large.
	ACTION	Change the number to a smaller one to avoid skipping too many files back.

154	MESSAGE	ATTEMPT TO BACKSPACE OVER TOFILEBOT
	CAUSE	The negative number specified in a skip end-of-file function for the "to" file 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 physical record length of "from" file, or 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 specified bytes.

200†	MESSAGE	WARNING: FROMFILE RECSIZE IS <i>number type</i> , TOFILE RECSIZE IS <i>number type</i>
	CAUSE	The record sizes of the “from” and “to” files are not identical.
	ACTION	In a job: FCOPY performs the specified operation despite the conflict. In a session: You are given the choice whether or not to continue the operation. Note that if the “from” record size is larger than the “to” record size, the “from” records would be truncated. If the “to” record size is larger than the “from” record size, the content of the excess byte positions in the “to” records is unpredictable.

201†	MESSAGE	WARNING: FROMFILE IS ASCII, TOFILE IS BINARY OR WARNING: FROMFILE IS BINARY, TOFILE IS ASCII
	CAUSE	The data formats of the “from” and “to” files are not identical.
	ACTION	In a job: FCOPY performs the specified operation despite the conflict. In a session: You are given the choice whether or not to continue the operation.

301	MESSAGE	READ ERROR IN FROMFILE AT RECORD <i>recnum</i>
	CAUSE	An error occurred while reading from the “from” file at the record number displayed (<i>recnum</i>).
	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 <i>recnum</i>
	CAUSE	The verify function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (<i>recnum</i>).
	ACTION	Retry the operation specifying a larger <i>number-of-errors</i> parameter.

† WARNING

303	MESSAGE	RAN OUT OF IGNERRS AT FROMFILESREC
	CAUSE	More errors found in the "from" file than are permitted by IGNERRS option.
	ACTION	
304	MESSAGE	COMPARE OPTION: RAN OUT OF COMPARE ERRORS AT FROMFILE RECORD <i>recnum</i>
	CAUSE	The compare function was terminated because the specified maximum number of errors has been exceeded at the record number displayed (<i>recnum</i>).
	ACTION	Retry the operation specifying a larger <i>number-of-errors</i> 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
		$\left\{ \begin{array}{l} \text{AT} \\ \text{AFTER} \\ \text{BEFORE} \end{array} \right\} \text{ RECORD X}$
	CAUSE	The "to" file is larger than the "from" file.
	ACTION	Purge the "to" file and retry the operation using the NEW option if you are copying to the "to" file. Alternatively, purge the "to" file and build the "to" file with the same attributes as the "from" file and retry the operation.

† WARNING

801†

MESSAGE

EOF IN TOFILE BUT NOT IN FROMFILE

$\left\{ \begin{array}{c} \text{AT} \\ \text{AFTER} \\ \text{BEFORE} \end{array} \right\}$ RECORD X

CAUSE

The *tofile* is smaller than the *fromfile*.

ACTION

Purge the *tofile* and retry the operation using the NEW option if you are copying to the *tofile*. Alternatively, purge the *tofile* and build the *tofile* with the same attributes as the *fromfile* and retry the operation.

802†

MESSAGE

READ ERROR IN FROMFILE

$\left\{ \begin{array}{c} \text{AT} \\ \text{AFTER} \\ \text{BEFORE} \end{array} \right\}$ RECORD X

CAUSE

Read error while verifying or comparing.

ACTION

Retry the operation and check the MPE file error.

803†

MESSAGE

READ ERROR IN TOFILE

$\left\{ \begin{array}{c} \text{AT} \\ \text{AFTER} \\ \text{BEFORE} \end{array} \right\}$ RECORD X

CAUSE

Read error while verifying or comparing.

ACTION

Retry the operation and check the MPE file error.

† WARNING

901	MESSAGE	KSAM FROMFILE BOUNDARY (EOF OR BOF)
	CAUSE	The beginning or end of the <i>fromfile</i> is reached during the FCOPY operation.
	ACTION	Try again.
902	MESSAGE	KSAM FROMFILE POSITIONING ERROR
	CAUSE	Could not position to desired place.
	ACTION	Try again or check the length of the <i>fromfile</i> .
903	MESSAGE	ERROR: WRONG CONDITIONS FOR OPENING NEW KSAM FILE
	CAUSE	The <i>fromfile</i> 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.
910	MESSAGE	DELETE ACD FAILS (SOURCE ACD)
	CAUSE	Internal error.
	ACTION	Notify your System Manager.

911	MESSAGE	DELETE ACD FAILS (NO SOURCE ACD)
	CAUSE	Internal error.
	ACTION	Notify your System Manager.
912	MESSAGE	DELETE ACD FAILS (NO COPYACD)
	CAUSE	Internal error.
	ACTION	Notify your System Manager.
913	MESSAGE	COPY ACD TO OLD FILE: ERROR
	CAUSE	Internal error.
	ACTION	Notify your System Manager.
914	MESSAGE	COPY ACD TO NEW FILE: ERROR
	CAUSE	Internal error.
	ACTION	Notify your System Manager.
915†	MESSAGE	WARNING: NO SOURCE ACD
	CAUSE	COPYACD option specified but the <i>fromfile</i> doesn't have an ACD.
	ACTION	Quit or continue to copy only the data <i>fromfile</i> .
916†	MESSAGE	WARNING: COPY ACD NOT RELEVANT
	CAUSE	COPYACD option should be specified only if the <i>tofile</i> is a local disc file.
	ACTION	Quit or continue to copy only the data <i>fromfile</i> .
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 EBCDIC.
	ACTION	Do not use EBCDIC.
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:

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



Default File Definition Values

Appendix B summarizes the default file definition values that FCOPY uses to open a “from” or “to” file. It contains two tables; Table B-1 shows default values for different types of “from” files, and Table B-2 shows default values for different types of “to” files. The information stored in the file label of an existing disc file always describes its characteristics.

Shaded items in the tables are those most likely to cause problems when copying from one device to another. Refer to the *File System Reference Manual* (30000-90236) or *Accessing Files Programmer's Guide* (32650-90017) for information about domains, buffers, and record formats.

You can override a file's default definition, by describing its characteristics with an MPE :FILE command before issuing an FCOPY command, and then using 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 Disc File	\$STDIN \$STDINX
Formal File Designator	(empty)	"from" file	"from" file	(empty)
Default File Designator	(empty)	"from" file	"from" file	(empty)
Domain	new	new	old perm. or temp.	old
RecordType	see note 1	binary	as specified in file label	ASCII
RecordFormat	fixed	undefined		fixed
Access Type	I/O	read/write	read/write	read only
Carriage Control	no	no	as specified in 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
RecordSize	256 bytes	256 bytes	as specified in file label	80 bytes
Device	ldev number of terminal (left- justified ASCII number)	byte array containing ldev number or device class name from :FILE command		session or batch input device
Userlabels	0	0		0
Blocking Factor	1	1		1
Buffers	2	2	2	2
File Size	N/A	N/A	as specified in file label	N/A
Initial Allocation	N/A	N/A		N/A
File Code	0	N/A		
Allow : FILE Command	no	yes	yes	no

Table B-2. Default "to" Device Values

"To" Device	\$CTUL \$CTUR	\$HARD see note 5	Line Printer	Magnetic Tape	Named Disc File
Formal File Designator	(empty)	HARD	"to" file	"to" file	"to" file
Default File Designator	(empty)	HARD	"to" file	"to" file	"to" file
Domain	new	new	new	new	see note 2
RecordType	see note 1	ASCII	ASCII	binary	as specified in file label
RecordFormat	fixed	fixed	undefined	undefined	
Access Type	I/O	I/O	write only	read/write	see note 3
Carriage Control	no	yes	no	no	as specified in file label
Multirecord Access	yes	yes	no	no	no
Dynamic Locking	no	no	no	no	no
Exclusive Access	yes	yes	yes	yes	see note 4
Inhibit Buffering	yes	yes	no	no	no
RecordSize	256 bytes	86 bytes	132 bytes	256 bytes	
Device	ldev number of terminal (left- justified ASCII number)	ldev number of term. (left- justified ASCII number)	byte array containing ldev # or device class name from :FILE command	byte array containing ldev number or device class name from :FILE command	as specified in file label
Userlabels	0	0	N/A	0	
Blocking Factor	1	1	1	1	
Buffers	2	2	2	2	2
File Size	N/A	N/A	N/A	N/A	as specified in file label
Initial Allocation	1	N/A	N/A	N/A	
File Code	0	N/A	N/A	N/A	
Allow :FILE Command	no	yes	yes	yes	yes

Table B-2. Default "to" Device Values (Continued)

"To" Device	\$STDLIST
Formal File Designator	(empty)
Default File Designator	(empty)
Domain	new
RecordType	ASCII
RecordFormat	undefined
Access Type	WRITEONLY
Carriage Control	no
Multirecord Access	no
Dynamic Locking	no
Exclusive Access	no
Inhibit Buffering	no
RecordSize	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

NOTES

1. \$CTUL or \$CTUR domain is ASCII (JIS for the HP 2645K terminal) or BINARY depending on the "to" file.
2. A named disc file's domain is old permanent for all functions except CHAR, OCTAL, HEX , and HEXO. For these three functions the domain can be old permanent or temporary.
3. A named disc file's access type is as follows:
For COMPARE: READONLY
For VERIFY: input/output
For all other functions: APPEND
4. A named disc file has Exclusive Access for all functions except COMPARE.

FROM=magnetic tape

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

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

Domain: new

Record Type (ASCII/binary): binary

Record Format: undefined

Access Type: read/write

Carriage Control? no

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

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

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

Userlabels: 0

Blocking Factor: 1

Number of Buffers: 2

File Size: (not applicable)

Initial Allocation: (not applicable)

File Code: (not applicable)

Allow :FILE Command? yes

FROM=named disc file

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

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

Domain: old permanent or temporary

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

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

Access Type: read/write

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

Multirecord Access? no

Dynamic Locking? no

Exclusive Access? yes

Inhibit Buffering? no

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

Device: (as specified when the file was created)

Userlabels: (as specified when the file was created)

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

Number of Buffers: 2

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

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

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

Allow :FILE Command? yes

Code Conversion Tables

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

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



How To Use This Table

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

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

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
16	020	10	DLE	020	12 11 1 8 9	DLE	020
17	021	11	DC1	021	11 1 9	DC1	021
18	022	12	DC2	022	11 2 9	DC2	022
19	023	13	DC3	023	11 3 9	TM	023
20	024	14	DC4	074	4 8 9	RES	235
21	025	15	NAK	075	5 8 9	NL	205
22	026	16	SYN	062	2 9	BS	010
23	027	17	ETB	046	0 6 9	IL	207
24	030	18	CAN	030	11 8 9	CAN	030
25	031	19	EM	031	11 1 8 9	EM	031
26	032	1A	SUB	077	7 8 9	CC	222
27	033	1B	ESC	047	0 7 9	CU1	217
28	034	1C	FS	034	11 4 8 9	IFS	034
29	035	1D	GS	035	11 5 8 9	IGS	035
30	036	1E	RS	036	11 6 8 9	IRS	036
31	037	1F	US	037	11 7 8 9	IUS	037
32	040	20	SP 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	12	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
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A	:	172	2 8		232
59	073	3B	;	136	11 6 8	CU3	233
60	074	3C	<	114	12 4 8	DC4	024
61	075	3D	=	176	6 8	NAK	025
62	076	3E	>	156	0 6 8		236
63	077	3F	?	157	0 7 8	SUB	032

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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	<	074
77	115	4D	M	324	11 4	(	050
78	116	4E	N	325	11 5	+	053
79	117	4F	O	326	11 6		041
80	120	50	P	327	11 7	&	046
81	121	51	Q	330	11 8		251
82	122	52	R	331	11 9		252
83	123	53	S	342	0 2		253
84	124	54	T	343	0 3		254
85	125	55	U	344	0 4		255
86	126	56	V	345	0 5		256
87	127	57	W	346	0 6		257
88	130	58	X	347	0 7		260
89	131	59	Y	350	0 8		261
90	132	5A	Z	351	0 9	!	135
91	133	5B	[	112	12 2 8	\$	044
92	134	5C	\	340	0 2 8	*	052
93	135	5D	]	132	11 2 8	)	051
94	136	5E	^	137	11 7 8	;	073
95	137	5F	_ unlin	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

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
112	160	70	p	227	12 11 7		272
113	161	71	q	230	12 11 8		273
114	162	72	r	231	12 11 9		274
115	163	73	s	242	11 0 2		275
116	164	74	t	243	11 0 3		276
117	165	75	u	244	11 0 4		277
118	166	76	v	245	11 0 5		300
119	167	77	w	246	11 0 6		301
120	170	78	x	247	11 0 7		302
121	171	79	y	250	11 0 8	'	140
122	172	7A	z	251	11 0 9	:	072
123	173	7B	{	300	12 0	#	043
124	174	7C		152	12 11	@	100
125	175	7D	}	320	11 0	'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
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

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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
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
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

Character Code			ASCII			EBCDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
208	320	D0		237	12 11 7 8	}	175
209	321	D1		240	11 0 1 8	J	112
210	322	D2		252	11 0 2 8	K	113
211	323	D3		253	11 0 3 8	L	114
212	324	D4		254	11 0 4 8	M	115
213	325	D5		255	11 0 5 8	N	116
214	326	D6		256	11 0 6 8	O	117
215	327	D7		257	11 0 7 8	P	120
216	330	D8		260	12 11 0 1 8	Q	121
217	331	D9		261	12 11 0 1	R	122
218	332	DA		262	12 11 0 2		356
219	333	DB		263	12 11 0 3		357
220	334	DC		264	12 11 0 4		360
221	335	DD		265	12 11 0 5		361
222	336	DE		266	12 11 0 6		362
223	337	DF		267	12 11 0 7		363
224	340	E0		270	12 11 0 8	\	134
225	341	E1		271	12 11 0 9		237
226	342	E2		272	12 11 0 2 8	S	123
227	343	E3		273	12 11 0 3 8	T	124
228	344	E4		274	12 11 0 4 8	U	125
229	345	E5		275	12 11 0 5 8	V	126
230	346	E6		276	12 11 0 6 8	W	127
231	347	E7		277	12 11 0 7 8	X	130
232	350	E8		312	12 0 2 8 9	Y	131
233	351	E9		313	12 0 3 8 9	Z	132
234	352	EA		314	12 0 4 8 9		364
235	353	EB		315	12 0 5 8 9		365
236	354	EC		316	12 0 6 8 9	⌈	366
237	355	ED		317	12 0 7 8 9		367
238	356	EE		332	12 11 2 8 9		370
239	357	EF		333	12 11 3 8 9		371
240	360	F0		334	12 11 4 8 9	0	060
241	361	F1		335	12 11 5 8 9	1	061
242	362	F2		336	12 11 6 8 9	2	062
243	363	F3		337	12 11 7 8 9	3	063
244	364	F4		352	11 0 2 8 9	4	064
245	365	F5		353	11 0 3 8 9	5	065
246	366	F6		354	11 0 4 8 9	6	066
247	367	F7		355	11 0 5 8 9	7	067
248	370	F8		356	11 0 6 8 9	8	070
249	371	F9		357	11 0 7 8 9	9	071
250	372	FA		372	12 11 0 2 8 9		372
251	373	FB		373	12 11 0 3 8 9		373
252	374	FC		374	12 11 0 4 8 9		374
253	375	FD		375	12 11 0 5 8 9		375
254	376	FE		376	12 11 0 6 8 9		376
255	377	FF		377	12 11 0 7 8 9		377

JIS/EBCDIK CONVERSION TABLE

- The table is sorted by character code, each code being represented by its decimal, octal, and hexadecimal equivalent.
- Each row of the table gives the JIS and EBCDIK meaning of the character code, the JIS ↔ EBCDIK conversion code, and the Hollerith representation (punched card code) for the JIS and EBCDIK character. Note that 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 the table on p. C-2.

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

Table C-2. JIS and EBCDIK Characters

Character Code			JIS			EBDIC	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (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		227
9	011	09	HT	005	12 5 9		215
10	012	0A	LF	045	0 5 9	SMM	216
11	013	0B	VT	013	12 3 8 9	VT	013
12	014	0C	FF	014	12 4 8 9	FF	014
13	015	0D	CR	015	12 5 8 9	CR	015
14	016	0E	SO	016	12 6 8 9	SO	016
15	017	0F	SI	017	12 7 8 9	SI	017

Character Code			JIS			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (Octal)
16	020	10	DLE	020	12 11 1 8 9	DLE	020
17	021	11	DC1	021	11 1 9	DC1	021
18	022	12	DC2	022	11 2 9	DC2	022
19	023	13	DC3	023	11 3 9	TM	023
20	024	14	DC4	074	4 8 9	RES	235
21	025	15	NAK	075	5 8 9	NL	205
22	026	16	SYN	062	2 9	BS	010
23	027	17	ETB	046	0 6 9	IL	207
24	030	18	CAN	030	11 8 9	CAN	030
25	031	19	EM	031	11 1 8 9	EM	031
26	032	1A	SUB	077	7 8 9	CC	222
27	033	1B	ESC	047	0 7 9	CU1	217
28	034	1C	FS	034	11 4 8 9	IFS	034
29	035	1D	GS	035	11 5 8 9	IGS	035
30	036	1E	RS	036	11 6 8 9	IRS	036
31	037	1F	US	037	11 7 8 9	IUS	037
32	040	20	SP 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	12	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
55	067	37	7	367	7	EOT	004
56	070	38	8	370	8		230
57	071	39	9	371	9		231
58	072	3A	:	172	2 8		232
59	073	3B	;	136	11 6 8	CU3	233
60	074	3C	<	114	12 4 8	DC4	024
61	075	3D	=	176	6 8	NAK	025
62	076	3E	>	156	0 6 8		236
63	077	3F	?	157	0 7 8	SUB	032

Character Code			JIS			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (Octal)
64	100	40	@	174	4 8	SP	040
65	101	41	A	301	12 1	\	241
66	102	42	B	302	12 2	┌	242
67	103	43	C	303	12 3	└	243
68	104	44	D	304	12 4	\	244
69	105	45	E	305	12 5	•	245
70	106	46	F	306	12 6	•	246
71	107	47	G	307	12 7	•	247
72	110	48	H	310	12 8	•	250
73	111	49	I	311	12 9	-	251
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	:	252
82	122	52	R	331	11 9		253
83	123	53	S	342	0 2		254
84	124	54	T	343	0 3	:	255
85	125	55	U	344	0 4	=	256
86	126	56	V	345	0 5	•	257
87	127	57	W	346	0 6		240
88	130	58	X	347	0 7	-	260
89	131	59	Y	350	0 8		141
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	_ unlin	155	0 5 8	~	136
96	140	60	` grave	171	1 8	- hyphen	055
97	141	61	a	131	11 1 8	/	057
98	142	62	b	142	11 0 29		142
99	143	63	c	143	11 0 39		143
100	144	64	d	144	11 0 49		144
101	145	65	e	145	11 0 59		145
102	146	66	f	146	11 0 69		146
103	147	67	g	147	11 0 79		147
104	150	68	h	150	11 0 89		150
105	151	69	i	151	0 1 8		151
106	152	6A	j	166	12 11 0 69		174
107	153	6B	k	167	12 11 0 79	, comma	054
108	154	6C	l	170	12 11 0 89	%	045
109	155	6D	m	200	12 0 1 8	_ unlin	137
110	156	6E	n	160	12 11 0	>	076
111	157	6F	o	213	12 3 38	?	077

Character Code			JIS			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (Octal)
112	160	70	p	161	12 11 0 1 9		156
113	161	71	q	162	12 11 0 2 9		160
114	162	72	r	163	12 11 0 3 9		161
115	163	73	s	234	11 11 4 8		162
116	164	74	t	165	12 11 0 5 9		165
117	165	75	u	164	12 11 0 4 9		164
118	166	76	v	233	12 11 3 8		152
119	167	77	w	240	11 0 1 8		153
120	170	78	x	261	12 11 0 1		154
121	171	79	y	253	11 0 3 8		140
122	172	7A	z	260	12 11 0 1 8	:	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		155
129	201	81		041	0 1 9		261
130	202	82		042	0 2 9		262
131	203	83		043	0 3 9		263
132	204	84		044	0 4 9		264
133	205	85		025	11 5 9		265
134	206	86		006	12 6 9		266
135	207	87		027	11 7 9		267
136	210	88		050	0 8 9		270
137	211	89		051	0 1 8 9		271
138	212	8A		052	0 2 8 9		272
139	213	8B		053	0 3 8 9		157
140	214	8C		054	0 4 8 9		273
141	215	8D		011	12 1 8 9		274
142	216	8E		012	12 2 8 9		275
143	217	8F		033	11 3 8 9		276
144	220	90		060	12 11 0 1 8 9		277
145	221	91		061	1 9		300
146	222	92		032	11 2 8 9		301
147	223	93		063	3 9		302
148	224	94		064	4 9		303
149	225	95		065	5 9		304
150	226	96		066	6 9		305
151	227	97		010	12 8 9		306
152	230	98		070	8 9		307
153	231	99		071	1 8 9		310
154	232	9A		072	2 8 9		311
155	233	9B		073	3 8 9		166
156	234	9C		004	12 4 9		163
157	235	9D		024	11 4 9		312
158	236	9E		076	6 8 9		313
159	237	9F		341	11 0 1 9		314

Character Code			JIS			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (Octal)
160	240	A0		127	12 11 7 9		167
161	241	A1		101	12 0 1 9		176
162	242	A2		102	12 0 2 9		315
163	243	A3		103	12 0 3 9		316
164	244	A4		104	12 0 4 9		317
165	245	A5		105	12 0 5 9		320
166	246	A6		106	12 0 6 9		321
167	247	A7		107	12 0 7 9		322
168	250	A8		110	12 0 8 9		323
169	251	A9		111	12 1 8		324
170	252	AA		121	12 11 1 9		325
171	253	AB		122	12 11 2 9		171
172	254	AC		123	12 11 3 9		326
173	255	AD		124	12 11 4 9		327
174	256	AE		125	12 11 5 9		330
175	257	AF		126	12 11 6 9		331
176	260	B0		130	12 11 8 9		172
177	261	B1		201	12 0 1		170
178	262	B2		202	11 0 2		342
179	263	B3		203	11 0 3		343
180	264	B4		204	11 0 4		344
181	265	B5		205	11 0 5		345
182	266	B6		206	11 0 6		346
183	267	B7		207	11 0 7		347
184	270	B8		210	11 0 8		340
185	271	B9		211	12 0 9		341
186	272	BA		212	12 0 2 8		332
187	273	BB		214	12 0 4 8		333
188	274	BC		215	12 0 5 8		334
189	275	BD		216	12 0 6 8		335
190	276	BE		217	12 0 7 8		336
191	277	BF		220	12 11 1 8		337
192	300	C0		221	12 11 1		173
193	301	C1		222	12 11 2	A	101
194	302	C2		223	12 11 3	B	102
195	303	C3		224	12 11 4	C	103
196	304	C4		225	12 11 5	D	104
197	305	C5		226	12 11 6	E	105
198	306	C6		227	12 11 7	F	106
199	307	C7		230	12 11 8	G	107
200	310	C8		231	12 11 9	H	110
201	311	C9		232	12 11 2 8	I	111
202	312	CA		235	12 11 5 8		350
203	313	CB		236	12 11 6 8		351
204	314	CC		237	12 11 7 8		352
205	315	CD		242	11 0 2		353
206	316	CE		243	11 0 3		354
207	317	CF		244	11 0 4		355

Character Code			JIS			EBCDIK	
Decimal	Octal	Hex	Control/ Graphic	To EBCDIK (Octal)	YHP Hollerith	Control/ Graphic	To JIS (Octal)
208	320	D0		245	11 0 5		175
209	321	D1		246	11 0 6	J	112
210	322	D2		267	11 0 7	K	113
211	323	D3		250	11 0 8	L	114
212	324	D4		251	11 0 9	M	115
213	325	D5		252	11 0 2 8	N	116
214	326	D6		254	11 0 4 8	O	117
215	327	D7		255	11 0 5 8	P	120
216	330	D8		256	11 0 6 8	Q	121
217	331	D9		257	11 0 7 8	R	122
218	332	DA		272	12 11 0 2 8		356
219	333	DB		273	12 11 0 3 8		357
220	334	DC		274	12 11 0 4 8		360
221	335	DD		275	12 11 0 5 8		361
222	336	DE		276	12 11 0 6 8		362
223	337	DF		277	12 11 0 7 8		363
224	340	E0		270	12 11 0 8	\$	134
225	341	E1		271	12 11 0 9		237
226	342	E2		262	12 11 0 2	S	123
227	343	E3		263	12 11 0 3	T	124
228	344	E4		264	12 11 0 4	U	125
229	345	E5		265	12 11 0 5	V	126
230	346	E6		266	12 11 0 6	W	127
231	347	E7		267	12 11 0 7	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

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

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	SOH	001	12 1 9	SOH	001
2	002	02	STX	002	12 2 9	STX	002
3	003	03	ETX	003	12 3 9	ETX	003
4	004	04	EOT	067	7 9	PF	234
5	005	05	ENQ	055	0 5 8 9	HT	011
6	006	06	ACK	056	0 6 8 9	LC	206
7	007	07	BEL	057	0 7 8 9	DEL	177
8	010	08	BS	026	11 6 9		227
9	011	09	HT	005	12 5 9		215
10	012	0A	LF	045	0 5 9	SMM	216
11	013	0B	VT	013	12 3 8 9	VT	013
12	014	0C	FF	014	12 4 8 9	FF	014
13	015	0D	CR	015	12 5 8 9	CR	015
14	016	0E	SO	016	12 6 8 9	SO	016
15	017	0F	SI	017	12 7 8 9	SI	017

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
16	020	10	DLE	020	12 11 1 8 9	DLE	020
17	021	11	DC1	021	11 1 9	DC1	021
18	022	12	DC2	022	11 2 9	DC2	022
19	023	13	DC3	023	11 3 9	TM	023
20	024	14	DC4	074	4 8 9	RES	235
21	025	15	NAK	075	5 8 9	NL	205
22	026	16	SYN	062	2 9	BS	010
23	027	17	ETB	046	0 6 9	IL	207
24	030	18	CAN	030	11 8 9	CAN	030
25	031	19	EM	031	11 1 8 9	EM	031
26	032	1A	SUB	077	7 8 9	CC	222
27	033	1B	ESC	047	0 7 9	CU1	217
28	034	1C	FS	034	11 4 8 9	IFS	034
29	035	1D	GS	035	11 5 8 9	IGS	035
30	036	1E	RS	036	11 6 8 9	IRS	036
31	037	1F	US	037	11 7 8 9	IUS	037
32	040	20	SP 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	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
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 6 8	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

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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
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	_ unlin	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

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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
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

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
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
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 0 4 8	F	106
199	307	C7		215	12 0 5 8	G	107
200	310	C8		216	12 0 6 8	H	110
201	311	C9		217	12 0 7 8	I	111
202	312	CA		220	12 11 1 8		350
203	313	CB		232	12 11 2 8		351
204	314	CC		233	12 11 3 8		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

Character Code			ASCII			BCDIC	
Decimal	Octal	Hex	Control/ Graphic	To BCDIC (Octal)	Hollerith	Control/ Graphic	To ASCII (Octal)
208	320	D0		237	12 11 7 8	I	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
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



Symbols

:FCOPY, 2-5
:FILE Commands, 2-4
:RUN, 2-5
\$STDIN, B-2
\$STDINX, B-2
\$STDLIST, 5-7

A

ADDF, 3-10
Ampersand, 2-4
APPEND, 3-1, 3-9, 5-1
Appending Files, 3-1
ASCII Code, 3-4
ASCII/BCDIC Conversion, 5-2
ASCII/EBCDIC Conversion, C-2, C-14
Asterisk, 2-1, 2-2, 2-4, B-1

B

BCDIC/ASCII Conversion, 5-2
BCDICIN, 2-3, 5-2
BCDICOUT, 2-3, 5-2
Block Size, 3-6
BREAK Key, 2-5

C

Carriage Control, 5-4
Cartridge Tapes, 3-6
CCTL, 2-3, 5-4
CHAR, 2-3, 5-7
Characterstring, 5-44
CLEAR, 2-3, 5-10
COMPARE, 2-3, 3-7, 5-13
CONTROL-Y, 2-4

COPY, 3-17
Column, 5-44
Combining Files
 combining disc files on disc, 3-9
 combining disc files on magnetic tape, 3-8
Combining Files, See also APPEND, 3-8
Comparing Files, 5-13
Components of an FCOPY Command, 2-1
Composing a Command, 2-6
Conversion Codes, 5-3, C-1
Copying
 appending files, 3-1
 between HP 3000 Systems, 3-13
 extracting Subsets, 3-2
 files by number, 3-11
 files in native languages, 3-17
 foreign tapes, 3-14
 from labeled tape, 3-10
 inactive records, 3-17
 KSAM files, 3-15
 named files, 3-11
 remote copies, 3-15
 sequentially, 3-11
 to a new disc file, 3-1
 to labeled tape, 3-9

D

DEBLOCK, 2-3, 3-7, 3-13, 5-15
Default File Definition Values, B-1
Disc, 3-8, 3-9, 5-34, B-1
Displaying
 ASCII file, 3-3
 binary file, 3-3
 file contents, 3-3
 hexadecimal and octal codes, 3-4
 subset of a file, 3-4

E

EBCDICIN, 2-3, 3-18, 5-17
EBCDICOUT, 2-3, 3-18, 5-17
EBCDIK, 5-19

Index (continued)

EBCDIKIN, 2-3, 5-19
EBCDIKOUT, 2-3, 5-19
EOF, 5-41
Error Messages, 2-6, A-1, A-2
EXCLUDE, 5-2, 5-17, 5-19, 5-44

F

FCOPY, 2-1, 2-2
FILES, 2-3, 5-21
File Definition Values, Default, B-1
File Designators, 6-1
Files, Defining, 2-4
Foreign Tapes, Copying, 3-14
FROM, 2-1
Fromfile, 2-1
Functions, FCOPY, 2-2, 2-3
 BCDICIN/BCDICOUT, 5-2
 CCTL/NOCTL, 5-4
 CHAR, 5-7
 CLEAR, 5-10
 COMPARE, 5-13
 DEBLOCK, 5-15
 EBCDICIN/EBCDICOUT, 5-17
 EBCDIKIN/EBCDIKOUT, 5-19
 FILES, 5-21
 HEX, 5-25
 HEXO, 5-27
 IGNERR, 5-29
 KANA, 5-30
 KEY, 5-32
 NEW, 5-34
 NOKSAM, 5-35
 NOUSERLABELS, 5-36
 OCTAL, 5-37
 SKIPEOF, 5-40
 SUBSET, 5-44
 UPSHIFT, 5-48
 VERIFY, 5-49
field, 5-2, 5-17, 5-19

G

Guidelines for FCOPY Commands, 2-4

H

HEX, 2-3, 5-7, 5-10, 5-25
HEXO, 2-3, 5-7, 5-10, 5-27
Hexadecimal Code, 3-4

I

IGNERR, 2-3, 3-7, 5-29
Issuing a Command
 existing FCOPY, 2-5
 interrupting FCOPY, 2-5
 issuing a command directly from MPE, 2-5
 issuing MPE commands, 2-5
 using FCOPY in a job, 2-5

J

JIS, 5-19, 5-30
Jobs, 2-6
Jobs, Streaming, 2-6

K

KANA, 2-3, 5-30
KATAKANA, 5-19
KEY, 2-3, 3-16, 5-32
KSAM, 2-3, 5-32
KSAM Files, 3-16
 copying a KSAM file to a non-KSAM file,
 3-16
 copying a KSAM file to a Specified Key Se-
 quence, 3-16
 copying a KSAM file to an existing KSAM
 file, 3-15

Index (continued)

L

LABEL, 5-22
LANG, 3-17
Labeled Tape, 3-10, 3-11, 3-13, 5-22
 copying files by number, 3-11
 copying from labeled tape, 3-10
 copying named files, 3-11
 copying to labeled tape, 3-9
 deblocking labeled tapes, 3-13
 sequentially copying files, 3-11
Language, 5-7, 5-17
Lowercase, 5-48

M

Magnetic Tape, 3-6, 3-8, 3-13
MPE Commands, Issuing, 2-5
magnetic tapes, 3-6

N

Native Languages, 3-18
 using Native Languages with CHAR, 3-17
 using Native Languages with EBCDIC and
 EBCDICOUT, 3-17, 3-18
 using Native Languages with UPSHIFT, 3-18
NEW, 2-3, 5-34
NEXT, 3-9, 3-10
NOBUF, 3-17
NOCCTL, 2-3, 5-4
NOKSAM, 2-3, 3-16, 5-35
NORECNUM, 3-3, 5-7, 5-8, 5-10
NOUSERLABELS, 2-3, 3-16, 5-35, 5-36

O

OCTAL, 2-3, 5-7, 5-10, 5-37
Octal Code, 3-4

P

Patternlist, 5-44
Peripherals, 6-1
Printing the Contents of a File, 3-5

R

Range, 5-44
Relative I/O (RIO), 3-17
Remote Copies, 3-15
RIO, 3-17
Running FCOPY, 2-5

S

Serial Discs, 3-6
Settings, 6-2
SKIPEOF, 2-3, 3-6, 3-9, 3-13, 5-22, 5-40
Status, 2-6
Status Messages, A-1
Streaming Jobs, 2-6
SUBSET, 2-3, 3-2, 3-9, 5-44
Subset, 5-3, 5-22
Syntax, FCOPY Command, 4-1

T

TERM Parameter, 6-2
Terminal, 3-5
 control sequences, 5-12
Title, 5-7, 5-10
TO, 2-2
Tofile, 2-2
Translate Function, 5-3

Index (continued)

U

Unlabeled Tapes, 3-6

UPSHIFT, 2-3, 3-18, 5-48

Uppercase, 5-48

V

VERIFY, 2-3, 3-7, 5-49

Part Number 03000-90064
Printed in U.S.A. 11/87
U0788

