WANG

VS

System Utilities Reference

Updated by 800-1303-04.01

VS
System Utilities Reference

4th Edition — August 1983

800-1303-04

WANG LABORATORIES, INC., ONE INDUSTRIAL AVENUE, LOWELL, MA 01851 • TEL. (617) 459-5000, TWX 710-343-6769, Telex 94-7421

HISTORY OF THE FOURTH EDITION

800-1303-04 (fourth edition) -- August, 1983
800-1303-04.01 (first addendum) -- January, 1984

ADDENDUM GRAPHICS KEY

Changes or Additions (|)	Vertical bar in outside margin
Deletions (⇄)	Arrow pointing to outer margin of page

ADDENDUM PAGES

Page Number	Addendum Date Code	Page Number	Addendum Date Code	Page Number	Addendum Date Code
iii	1/84	6-1	1/84	12-1	1/84
iv	1/84	6-2	1/84	12-2	1/84
v	1/84	6-3	1/84	12-15	1/84
vi	1/84	6-4	1/84	12-16	1/84
vii	1/84	6-5	1/84	12-16.1	1/84
viii	1/84	6-6	1/84	14-1	1/84
ix	1/84	6-7	1/84	14-2	1/84
x	1/84	6-8	1/84	14-3	1/84
xi	1/84	6-9	1/84	14-4	1/84
xii	1/84	6-10	1/84	14-5	1/84
xiii	1/84	6-11	1/84	14-6	1/84
xiv	1/84	6-12	1/84	14-7	1/84
1-1	1/84	7-1	1/84	14-8	1/84
1-2	1/84	7-2	1/84	14-8.1	1/84
1-3	1/84	7-3	1/84	14-13	1/84
1-4	1/84	7-4	1/84	14-14	1/84
2-1	1/84	7-5	1/84	19-1	1/84
2-2	1/84	7-6	1/84	19-2	1/84
2-3	1/84	7-7	1/84	19-3	1/84
2-3.1	1/84	7-8	1/84	20-1	1/84
2-3.2	1/84	7-9	1/84	20-2	1/84
2-4	1/84	7-10	1/84	20-3	1/84

ADDENDUM PAGES (continued)

Page Number	Addendum Date Code	Page Number	Addendum Date Code	Page Number	Addendum Date Code
20-4	1/84	23-3	1/84	A-36	1/84
20-5	1/84	23-4	1/84	D-1	1/84
20-6	1/84	23-5	1/84	D-2	1/84
20-7	1/84	23-6	1/84	DH-1	1/84
20-8	1/84	23-7	1/84	DH-2	1/84
21-1	1/84	23-8	1/84	Index-1	1/84
21-2	1/84	A-2.1	1/84	Index-2	1/84
22-1	1/84	A-3	1/84	Index-3	1/84
22-2	1/84	A-4	1/84	Index-4	1/84
22-3	1/84	A-19	1/84	Index-5	1/84
22-4	1/84	A-20	1/84	Index-6	1/84
22-5	1/84	A-20.1	1/84	Index-7	1/84
22-6	1/84	A-21	1/84	Index-8	1/84
22-7	1/84	A-22	1/84	Index-9	1/84
22-8	1/84	A-32.1	1/84	Index-10	1/84
22-9	1/84	A-33	1/84		
22-10	1/84	A-34	1/84		
22-11	1/84	A-35	1/84		
22-12	1/84	A-35.1	1/84		
22-13	1/84	A-35.2	1/84		
22-14	1/84	A-35.3	1/84		
22-15	1/84	A-35.4	1/84		
23-1	1/84	A-35.5	1/84		
23-2	1/84	A-35.6	1/84		

SUMMARY OF CHANGES

Topic	Description	Pages
Compress-In-Place (CIP)	Compress-In-Place (CIP), a new utility, consolidates free extents on nonsystem volumes.	19-1 through 19-3
COPY	Files to be copied can be opened in Input or Shared mode. Relative files are supported.	2-1 through 2-4
DISKINIT	A page block (for the VS50 and VS80) or page pool (for the VS25, VS45, VS85, VS90, and VS100) can be allocated to a volume through the INITIALIZE, REFORMAT, or RELABEL functions.	6-1 through 6-12
DISPLAY	Files to be displayed can be opened in Input or Shared mode. Relative files are supported.	7-1 through 7-10
IOELOG	The I/O error log now records machine check errors as a type of Nonstandard error.	12-2, 12-15, 12-16
IOTRACE	IOTRACE, a new utility, monitors the I/O trace table and records I/O information for a range of devices, based on criteria you define.	20-1 through 20-8
POOLSTAT	POOLSTAT, a new utility, monitors the utilization of page pools on the system.	21-1, 21-2
SORT	SORT supports consecutive and indexed files in Shared mode. Technical corrections were made.	14-4 to 14-8, 14-13, 14-14
SORTINT	SORTINT, a new utility, enables you to sort files according to standard ASCII or an external collating sequence. It also allows you to reformat the output record.	22-1 through 22-15, D-2

(continued)

Topic	Description	Pages
STABLEMT	STABLEMT, a new utility, enables you to create or modify files that define external collating sequences.	23-1 through 23-8
GETPARMs	The GETPARMS have been updated.	A-2.1, A-3, A-19 through A-21, A-32.1, A-34 through A-35.6
Document History	The Document History has been included.	DH-1, DH-2
Index	The Index has been updated.	Index-1 through Index-10

Disclaimer of Warranties and Limitation of Liabilities

The staff of Wang Laboratories, Inc., has taken due care in preparing this manual; however, nothing contained herein modifies or alters in any way the standard terms and conditions of the Wang purchase, lease, or license agreement by which this software package was acquired, nor increases in any way Wang's liability to the customer. In no event shall Wang Laboratories, Inc., or its subsidiaries be liable for incidental or consequential damages in connection with or arising from the use of the software package, the accompanying manual, or any related materials.

NOTICE:

All Wang Program Products are licensed to customers in accordance with the terms and conditions of the Wang Laboratories, Inc. Standard Program Products License; no ownership of Wang Software is transferred and any use beyond the terms of the aforesaid License, without the written authorization of Wang Laboratories, Inc., is prohibited.

This manual has been updated by Addendum 800-1303-04.01. For a list of Addendum Changes, refer to the Summary of Changes page.

PREFACE

Wang Laboratories, Inc., provides with the VS Operating System a variety of programs that perform routine data processing operations. These programs, referred to as utilities, extend the Operating System support of the programming environment. The VS utilities are documented in a set of five manuals, each describing a different phase of the data processing environment. You can use any one of the manuals separately or in combination with other manuals.

VS Program Development Tools Reference (800-1307)
Documents the program development tools provided with the VS Operating System. The VS Operating System includes a program development Editor that enables you to enter, modify, compile, and/or run an application program. Compiled or assembled program modules can then be linked through the LINKER utility. A Symbolic Debugger allows you to monitor program processing.

VS File Management Utilities Reference (800-1308)
Documents the utilities that facilitate the data entry and retrieval process. You can construct, modify, and report on data files through the VS File Management Utilities.

VS System Administrator's Reference (800-1144)
Documents the utilities that enable a system administrator to create configuration files and to protect system resources against unauthorized use. It also describes the Stand-Alone Utility System, which includes the COLDSTART and Compress-In-Place (CIP) utilities.

VS System Operator's Reference (800-1102)
Documents the BACKUP utility, which enables you to back up or restore files, libraries, or volumes. It also describes other system operation functions.

VS System Utilities Reference (800-1303)
Documents the utilities that perform the support tasks involved in program and file processing. System utilities can copy files, sort files, display files, print files, and initialize or analyze system storage media.

The utilities in the VS System Utilities Reference are alphabetized. Five new utilities, Compress-In-Place (CIP), IOTRACE, POOLSTAT, SORTINT, and STABLEMT, follow the utilities that are described in alphabetical order. These utilities include: Copy Support utilities, the Display Support utility, Program Support utilities, and System Support utilities. Introductory concepts and brief descriptions of all System Utilities are included in Chapter 1.

This manual is intended for all levels of programmers. You should be familiar with the VS environment, as described in the VS Programmer's Introduction (800-1101). In addition, topics treated in the following manuals are helpful to the discussions of individual System Utilities:

VS Operating System Services Reference	(800-1107)
VS Data Management System (DMS) Reference	(800-1124)
VS Principles of Operation	(800-1100)
VS Procedure Language Reference	(800-1205)
VS COBOL Reference	(800-1201)
VS RPG II Language Reference	(800-1203)
VS/IIS Operator's Guide	(800-1109)
VS Programmer's Guide to VS/IIS	(800-1304)
VS PL/I Language Reference	(800-1209)
VS FORTRAN Reference	(800-1208)
VS BASIC Language Reference	(800-1202)
OIS BASIC Language Reference	(800-5650)
VS DMS/TX Reference	(800-1128)
VS Release 6.20 Software Bulletin	(800-3114)

CONTENTS

APPENDICES

FIGURES

TABLES

TABLES

CHAPTER 1
INTRODUCTION TO THE VS SYSTEM UTILITIES

1.1 THE VS SYSTEM UTILITIES

The VS System Utilities perform many of the routine tasks in the programming cycle. The System Utilities perform four classes of essential functions on the VS: Copy Support, Program Support, Display Support, and System Support. The System Utilities are classified as follows:

Copy Support	Program Support	Display Support	System Support
COPY	EZFORMAT	DISPLAY	Compress-In-Place (CIP)
COPYOIS	SORT		DISKINIT
COPY2200	SORTINT		FORMCTL
COPYWP	STABLEMT		IOELOG
FLOPYDUP			IOTRACE
IBMCOPY			LISTVTOC
TAPECOPY			POOLSTAT
TRANSL			TAPEINIT
			VERIFY

1.1.1 The Copy Support Utilities

The Copy Support utilities transfer files, libraries, and/or volumes between one device and another. Some utilities also convert data to and from the format of another system, so that you can transfer data easily from one type of system to another. The Copy Support utilities are summarized as follows:

COPY — Creates a copy of an input file, a library, or a volume on an output disk or diskette. Optionally, COPY also modifies the file organization, or, for indexed files, rebuilds the index structure when a file is copied. (Refer to Chapter 2 for more information.)

COPYOIS — Converts the contents of OIS diskettes to VS files and copies OIS files from the VS to the OIS. (Refer to Chapter 3 for more information.)

COPY2200 Converts the contents of 2200 disks or diskettes to VS files and converts VS files to 2200 format for transfer to a Wang 2200 System. (Refer to Chapter 4 for more information.)

COPYWP Converts VS word processing documents to VS files and converts VS files to VS word processing documents. This function enables you to transfer information between the VS and Word Processing Systems. In addition, COPYWP performs filing functions for VS word processing documents and libraries. (Refer to Chapter 5 for more information.)

FLOPYDUP Duplicates diskettes, creates diskette image files (files containing byte-by-byte copies of entire diskettes), and transfers diskette image files to diskettes. (Refer to Chapter 9 for more information.)

IBMCOPY Copies and converts IBM format diskette files to and from the VS file format, with optional translation to and from the ASCII and EBCDIC character sets. IBMCOPY requires that you use a diskette drive that accepts soft-sectored diskettes. (Refer to Chapter 11 for more information.)

TAPECOPY Copies input disk or tape files to output disk or tape files. TAPECOPY is most appropriate for copying that involves a tape volume. (Refer to Chapter 15 for more information.)

TRANSL Translates files to and from the ASCII and EBCDIC character sets. TRANSL also translates files according to a translation table you specify. (Refer to Chapter 17 for more information.)

1.1.2 The Program Support Utilities

The Program Support utilities facilitate programming tasks, either by generating source code sections or by performing a common programming task. The Program Support utilities are summarized as follows:

EZFORMAT Generates source code that reproduces a screen format you designed, or creates a complete menu program that executes a menu you specified. (Refer to Chapter 8 for more information.)

SORT Sorts records in a data file(s) according to key values, or merges two or more sorted files. (Refer to Chapter 14 for more information.)

SORTINT Sorts up to 20 files into a single, ordered output file according to standard ASCII or an external collating sequence. SORTINT also provides the option to reformat the output record. (Refer to Chapter 22 for more information.)

STABLEMT Creates or modifies files that define an external collating sequence. (Refer to Chapter 23 for more information.)

1.1.3 The Display Support Utility

The Display Support utility enables you to view a file either at the workstation or on a printout. The Display Support utility is summarized as follows:

DISPLAY Displays the contents of any file on the workstation screen, and enables you to locate, view, and print data. (Refer to Chapter 7 for more information.)

1.1.4 The System Support Utilities

The System Support utilities initialize, analyze, and manipulate VS disks, diskettes, and tapes. The System Support utilities are summarized as follows:

Compress-In-Place (CIP) Consolidates free extents on nonsystem disks without the need to do a full volume backup and volume restore. (Refer to Chapter 19 for more information.)

DISKINIT Initializes (or reinitializes) a disk or a diskette so that you can use it for storing data on the VS. DISKINIT optionally creates an identifying volume label and a Volume Table of Contents (VTOC). DISKINIT also performs read and write verification, reformatting, relabelling, and bad block replacement. (Refer to Chapter 6 for more information.)

FORMCNTL Creates a form control definition, which is placed in a Forms Definition file, for VS serial forms-loadable printers. The forms control definition consists of information to generate formatted printed output. (Refer to Chapter 10 for more information.)

IOELOG Examines the contents of an I/O error log file. You can use this utility to evaluate the history and condition of equipment configured to a system. (Refer to Chapter 12 for more information.)

IOTRACE Monitors the I/O trace table and records I/O information for a range of devices, based on criteria you define. (Refer to Chapter 20 for more information.)

LISTVTOC Analyzes the contents and accuracy of a disk volume's Volume Table of Contents (VTOC). The analysis can include a dump of the VTOC control blocks. (Refer to Chapter 13 for more information.)

POOLSTAT Monitors the utilization of up to three page pools on VS25, VS45, VS85, VS90, and VS100 systems. (Refer to Chapter 21 for more information.)

TAPEINIT Initializes (or reinitializes) a tape by writing one of three types of labels and an end-of-tape marker, so that you can use the tape to record data. (Refer to Chapter 16 for more information.)

VERIFY Tests the primary index, the alternate indices, and the data chain of indexed files to disclose file structure problems. (Refer to Chapter 18 for more information.)

1.2 UTILITIES IN THE VS ENVIRONMENT

All utilities reside in the System Program library (@SYSTEM@) on the System Program volume. The file name of each utility is the name of that utility. The COPYWP utility, for example, is located in the file COPYWP in the @SYSTEM@ library on the System Program volume.

1.2.1 Running Utilities

You can run utilities either directly through the Command Processor or under procedure control. If the utility is run through the Command Processor, enter the utility name in the FILE parameter. You can omit the library and volume names (or retain the same default values) because the VS always searches the @SYSTEM@ library on the System Program volume if the specified file name cannot be located in the indicated library and volume location. Erase your default library if it contains a file with the same name as the utility. If the utility is run under procedure control, specify only the utility name following the Procedure language RUN statement.

1.2.2 Utilities and the VS Procedure Language

VS System Utilities are designed so that workstation interaction can be controlled through the VS Procedure language. Most utility information requests are processed through GETPARMs, which are the VS interface to the Procedure language. A brief description of GETPARMs and the VS Procedure language is found in Appendix A. The VS Procedure Language Reference provides a complete description of procedure processing requirements.

If a utility is used regularly with the same input and/or output options and parameters, you may find it helpful to write a procedure to automatically provide the repeated parameters. The description of each utility in this manual includes a sample procedure to help you write a customized procedure.

CHAPTER 2
THE COPY UTILITY

2.1 INTRODUCTION

The COPY utility performs two functions: it copies files, libraries, and volumes from one location on the system to another; it also manipulates file organization. When a single file is copied, an output file is created to contain the copied information. When a library or a volume is copied, either a new output library or volume is created, or the copied information is added to an existing library or volume. The COPY utility leaves the input file, library, and volume intact. However, when a single file is selected for COPY processing, you can reorganize the output file in the following ways:

- Select consecutive, indexed, or relative file organization
- Specify fixed-length or variable-length records
- Allow the utility to compress records
- Select the key length and key position of indexed files
- Change the packing density for index blocks and data blocks
- Rebuild indexed files by copying files one record at a time instead of one block at a time
- Create a copy of DMS/TX files with or without the recovery blocks
- Convert an indexed file to a DMS/TX file by creating recovery blocks
- Attach the copy of a DMS/TX file to the same data base that the original file is attached to

You can only copy information to which you have security access rights. Information copied through the COPY utility acquires your access rights. You also become the owner of the copied files.

An overview of COPY processing is provided in Figure 2-1.

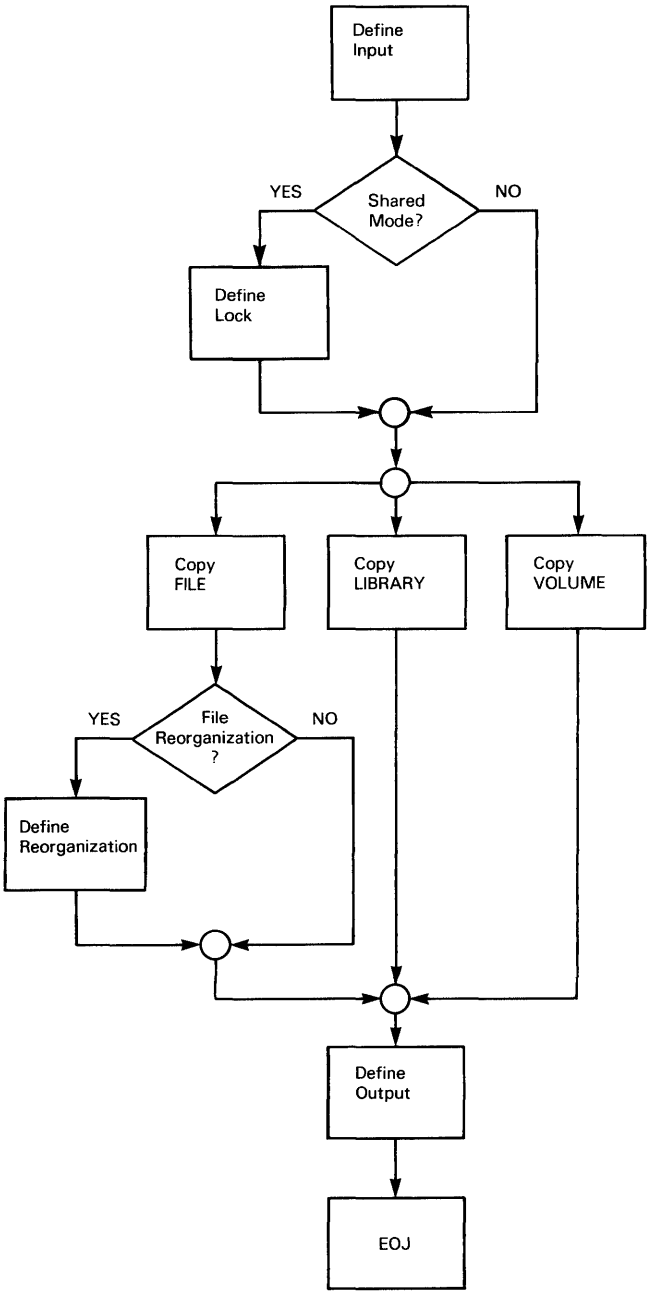


Figure 2-1. COPY Processing

NOTE

The COPY utility should not be used for system backup purposes. If the data being copied is a system file, or other file with special access rights, the file's rights can be inadvertently changed. (The BACKUP utility is more appropriate for backup purposes because it copies file characteristics exactly as they are on the input version. Refer to the VS System Operator's Reference for more information on the BACKUP utility.)

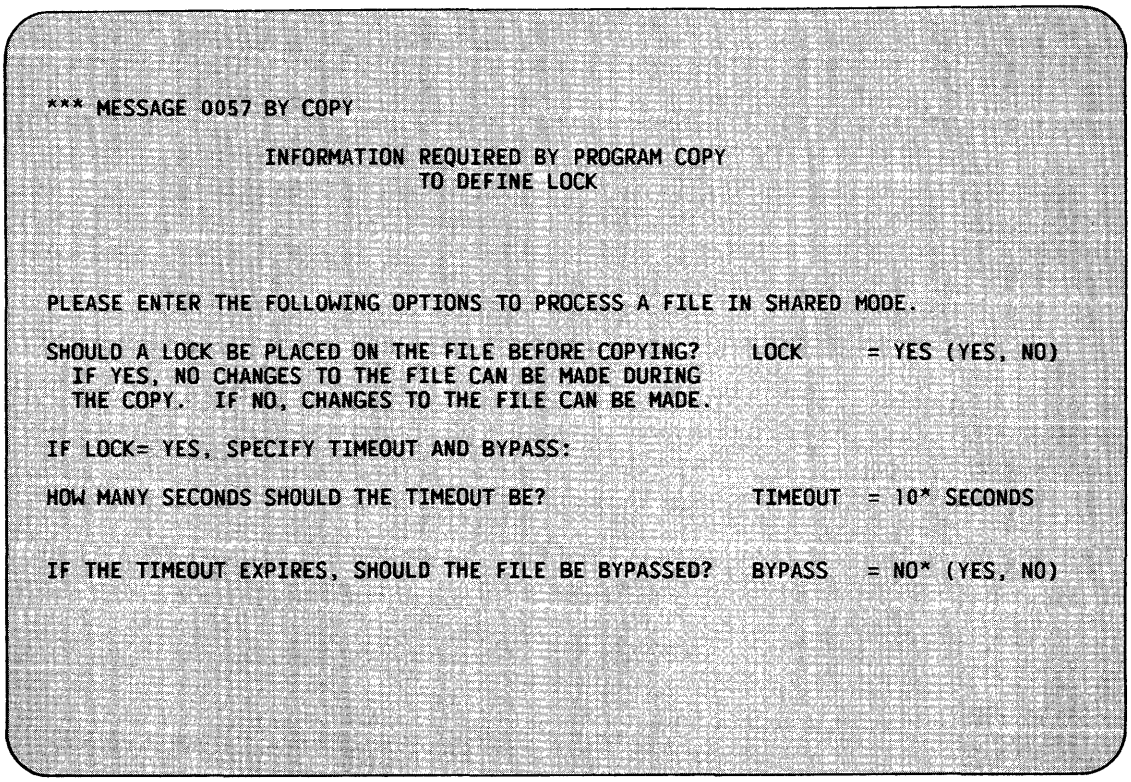


Figure 2-2. The Lock Screen

If a file is held for update by another user, the TIMEOUT field specifies the length of time that COPY waits to open the file in Shared mode with a lock. You can specify a timeout for a file if LOCK is equal to YES. Enter a value from 0 to 255 seconds; the default is 10 seconds.

The BYPASS field allows you to specify whether a file should be skipped when a timeout expires. This feature is especially useful for background tasks. Values for BYPASS are YES and NO; the default is NO.

If BYPASS is NO during a background task and the timeout expires, that copy is cancelled. If BYPASS is NO during a foreground task and the timeout expires, the Lock screen reappears with the message

FILE XXXXXXXX IN XXXXXXXX ON XXXXXX IS HELD BY USER XXX.

You can then redefine the LOCK, TIMEOUT, and BYPASS options and press ENTER to continue with the copy operation. You can also press PF1 to skip the particular file on which the timeout occurred, or you can press PF16 to terminate the copy operation if you are copying a library or volume.

If BYPASS is YES for foreground or background tasks, the copy operation proceeds, but the output will not contain a copy of the file for which the timeout occurred.

The Record Access Method (RAM) is always used to copy shared files. For more information about RAM, refer to the VS Data Management System (DMS) Reference.

2.3 COPYING A FILE

The COPY utility can copy a file from one library to another or copies a file within a given library. The output library can reside on the same volume as the original library or on a different volume. During COPY processing, you have the option of changing the output file's name and organization. When COPY processing is complete, the original file remains intact.

To operate the COPY utility, you must complete the Input Definition screen. If the input file is a print file or a program, Copy displays the Output Definitions screen next (refer to Subsection 2.3.2); otherwise, the Options screen is displayed (refer to Subsection 2.3.1).

2.3.1 Defining Options

The file organization of the output file is the same as the input file, unless you change the output file's organization through the Options screen. Figure 2-3 is a sample Options screen.

To operate the COPY utility, perform the following steps:

1. Specify the input file, library, and volume you want to copy.

2. Specify any changes you want in a file's organization and recovery capability.

2.2 DEFINING THE INPUT

The initial screen of the COPY utility is the Input Definition screen. You must specify the entity to be copied: a single file (FILE), all files in a library (LIBRARY), or all libraries in a volume (VOLUME). You must also specify whether COPY will open indexed and consecutive files in Input or Shared mode.

Enter the required information as follows:

COPY — The input unit to be copied. Three options are available: FILE, LIBRARY, or VOLUME. You do not need to specify the whole word for each option; the initial letter is sufficient. COPY defaults to FILE.

FILE — The name of the input file to be copied. Specify a file name if you specify FILE to the COPY field above. If COPY=LIBRARY or VOLUME, no entry is made for this parameter. There is no default for FILE.

LIBRARY — The name of the input library to be copied. Specify the library name if COPY=FILE or LIBRARY. The library name remains blank if you specify COPY=VOLUME. LIBRARY defaults to the INLIB value previously set through the SET USAGE CONSTANTS function of the Command Processor.

VOLUME — The name of the input volume to be copied. VOLUME defaults to the INVOL previously set through the SET USAGE CONSTANTS function of the Command Processor.

MODE — The mode in which COPY will open indexed and consecutive files. Specify INPUT or SHARED. MODE defaults to INPUT. If you specify Shared mode for a library or volume copy, COPY opens all indexed and consecutive files in Shared mode; it opens all other files in Input mode.

2.2.1 Copying Files in Shared Mode

If you specify Shared mode, COPY displays the Lock screen (refer to Figure 2-2). The LOCK field enables you to request that the file be locked during the copy operation. Enter YES or NO; the default is YES. If you lock a file, no changes to the file can occur while you are copying it. If you specify NO, no lock is placed on the file, and there is no need to specify the TIMEOUT and BYPASS options.

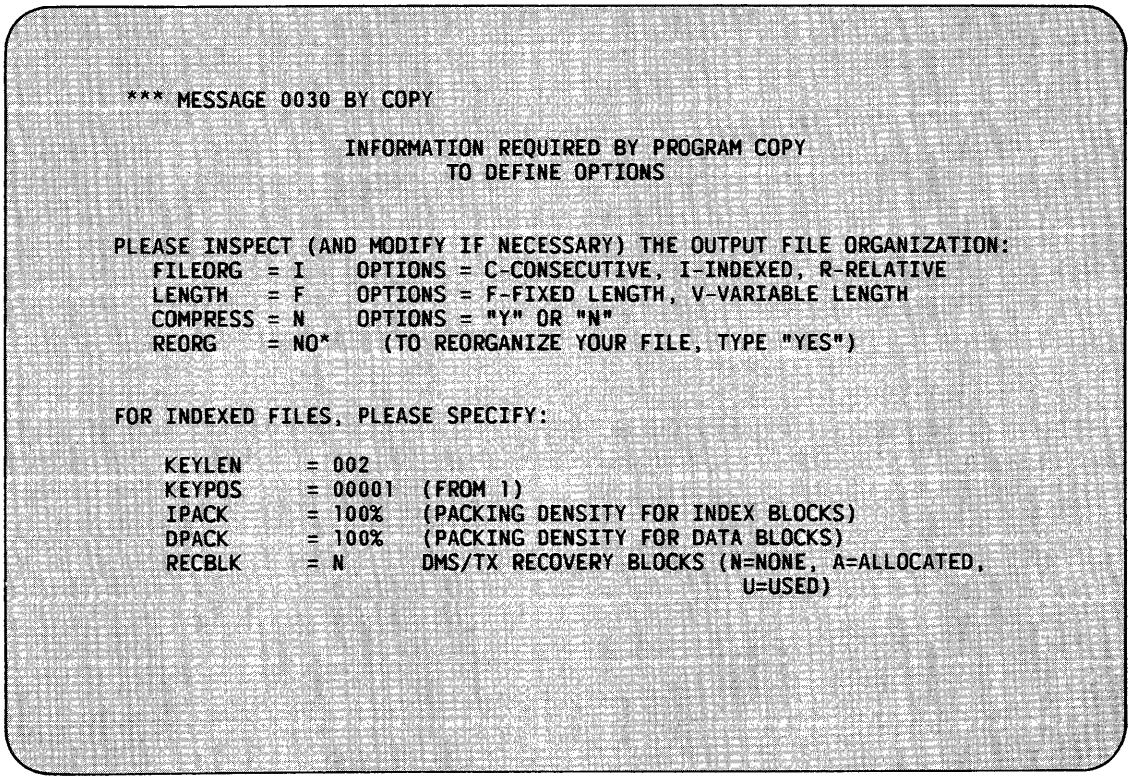


Figure 2-3. A Sample Options Screen

The following options for file reorganization are available:

FILEORG — The organization of the output file. You can specify one of three organizations: C (Consecutive), I (Indexed), or R (Relative). By using this option, you can supply consecutive files with an index and supply indexed files with a new primary key. You can also convert a relative file to a consecutive or indexed file, or you can convert a consecutive or indexed file to a relative file.

LENGTH — The type of record length. The options are F for Fixed-length records and V for Variable-length records. If data compression is used, you must specify the length as variable. If you change the length from variable to fixed, a pad character is required. The pad display appears after the Output Definition screen is displayed, enabling you to select either a blank character or a hexadecimal zero as the pad character.

COMPRESS — Indicates whether or not data compression is required. If you enter YES, the data is compressed. If you enter NO, the data is not compressed. Data compression can produce more efficient use of storage. When data is compressed, you must set the length to variable (V).

REORG — Specifies whether or not the file is copied in block mode or record mode. You can use the REORG function to rebuild an indexed file that has errors in its index structure. You can also use it to improve file access time. This option is meaningful when an indexed file is to be copied with no change to the file attributes. If you enter YES for this field, the system copies the file in record mode. In record mode, only the input file's data blocks are read, new index blocks are generated, and the data blocks are consolidated. Thus, the file assumes its most optimum form: high density data blocks and high density, single-level, primary index blocks. A DMS/TX file in an inconsistent state causes the REORG field to default to YES. This results in the file being restored to its original, pre-crash state. (For a discussion of DMS/TX protocol, refer to the VS DMS/TX Reference.)

If the output file is indexed, the following five specifications are available:

KEYLEN — The length in bytes of the index key. KEYLEN defaults to the primary index key length of the input file.

KEYPOS — The position number of the first byte in the index key field. KEYPOS defaults to the position number of the first byte in the input file's primary index key field.

IPACK — The density at which index blocks are packed. IPACK defaults to 100.

DPACK — The density at which data blocks are packed. DPACK defaults to 100.

RECBLK — The DMS/TX recovery option. Your choices are N, A, and U. This field defaults to N for non-DMS/TX files. You can choose A to allocate recovery blocks in the output file, even if the input file is a non-DMS/TX, Indexed file. The U option causes COPY to attach the output file to the same data base that the input file is attached to. Refer to the VS DMS/TX Reference for more information.

The system can pack index or data blocks at any percentage density. If you set the packing density to 100 percent, each block is completely filled, unless there is insufficient data. If you set the packing density to 50 percent, each block is half filled. Twice as many blocks are required when the packing density is 100 percent. Setting the packing density to 50 percent allows space for file growth.

2.3.2 Defining the Output

The Output Definition screen is displayed after you define the input and/or output reorganization options. Enter the required information as follows:

FILE — The name you assign to the output file.

LIBRARY — The library name in which the new file will reside.

VOLUME — The volume name on which the library resides.

RETAIN — The length of time that the file is protected from deletion. A file can be protected for a maximum of 999 days and a minimum of zero days. RETAIN defaults to the input file's retention period.

RELEASE — Indicates whether or not unused storage extents, previously allocated for data, can be released for use by other files. A value of YES releases the unused extents, a value of NO does not. RELEASE defaults to the input file specification.

FILECLAS — The protection class of the output file. The protection class determines which users have access to the file. The following file protection classes are available: #, $, @, blank, and A to Z. (For more information on fileclass options, consult the VS Programmer's Introduction.)

DEVICE — The device to which the output file is copied. When a file is copied to diskette or hard disk, you set DEVICE to DISK. If the output file is a print file, you indicate PRINTER as the DEVICE. (The COPY utility does not copy to tape. The TAPECOPY utility, described in Chapter 15, is provided for this purpose.) DEVICE defaults to DISK.

If the input file is a print file, the following options are also available:

PRTCLASS — The print class of the file. Print classes are identified with a one-letter code from A to Z. Because each printer on a system is assigned a list of print classes, what you assign to this field determines the printer on which the file will be printed.

FORM# — The form type to be used for printing a file. The default form number can range from 0 to 255.

COPIES — The number of copies of the file you want printed. The default value is 1.

A naming conflict occurs if you assign the output file a name identical to an existing file name in the same library. The Output Definition screen cannot be processed until you resolve the naming conflict.

If a naming conflict occurs, the Output Definition screen requests that you perform one of two actions. First, you can press PF3 to scratch the existing file from the output library (allowing the input file to be copied to that location and assume the specified name). Second, you can indicate a different output file, library, or volume name for the output file. File naming conflicts that occur during library and volume copying are handled differently. (Refer to Subsection 2.4.1.)

If you change a file's organization from variable- to fixed-length records on the Options screen, the Pad screen is displayed when the Output Definition screen is processed. The Pad screen requests that you select a blank character or a hexadecimal zero to pad short records to the specified fixed length.

When the COPY activity is complete, the End-of-Job (EOJ) screen is displayed. This screen contains a message indicating the name of the output file, library, and volume, as well as the number of records contained in the file. You can terminate COPY by pressing PF16 or restart the program by pressing PF1. Pressing PF1 returns you to the Input Definition screen.

2.4 COPYING A LIBRARY

To copy a library, specify an input library and a volume on the Input Definition screen. Pressing ENTER processes the Input Definition screen and displays the Output Definition screen. Indicate the output library and the volume on the Output Definition screen. When the Output Definition screen is processed, the COPY utility copies files from the named input library to the designated output library. The output library can reside on the same volume as the input library or on another volume. If the specified output library exists, new files are added to it. If it does not exist, the specified output library is created by the COPY utility. Files already stored in the output library are not affected by the copy operation, nor is the input library destroyed. File organization cannot be changed by the COPY utility when you copy a library.

If the utility encounters a file from the input library that has the same file name as a file that exists in the output library, the copy operation halts. This is called a duplicate file name conflict. You must resolve the conflict before copy processing can continue. Duplicate file name conflicts are discussed in Subsection 2.4.1.

When the input library is successfully copied to its assigned output library, the End-of-Job screen is displayed. You can terminate COPY by pressing PF16 or restart the program by pressing PF1. Pressing PF1 returns you to the Input Definition screen.

2.4.1 Resolving Duplicate File Names

When the COPY utility encounters a duplicate file name while copying a library or a volume, the program halts and a screen is displayed informing you of the conflict. This screen identifies the name of the first duplicate file name found in both the input and output libraries. You can resolve the conflict by selecting one of the following options:

N The input file is not copied, but the copy process continues for the rest of the library, or volume, until another duplicate file is encountered or the copy process is complete.

S The file with the duplicate name in the output library is scratched, enabling the file of the same name from the input library to be copied into the corresponding output library.

R The file that already exists in the output library is renamed to the name specified in the NEWNAME field. The input file is then copied into the output library without conflict.

C The input file is copied into the output library with the file name you specified in the NEWNAME field. The input file itself is not renamed.

2.5 COPYING A VOLUME

To copy a volume, specify an input volume on the Input Definition screen. Press ENTER to process the Input Definition screen and to display the Output Definition screen. Specify the output volume on the Output Definition screen. When the Output Definition screen is processed, files from the named input volume are copied to a designated output volume. Files from each library on the input volume are copied into the corresponding libraries of the output volume. When no corresponding library appears on the output volume, the COPY utility creates the library and adds it to the volume.

If a file and a library on the input volume have the same names as a file and a library on the output volume, a duplicate file name conflict occurs. The copy operation does not proceed until you resolve any such conflict. Duplicate file name conflicts are resolved in the same way that library name conflicts are. Refer to Subsection 2.4.1 for a discussion of conflict resolution. With the exception of any file name changed because of conflict, a volume's data is unaffected by the copy operation. File organization cannot be changed by the COPY utility when performing a volume copy.

When the input volume is successfully copied to its assigned output volume, the End-of-Job screen is displayed. You can terminate COPY by pressing PF16 or restart the program by pressing PF1. Pressing PF1 returns you to the Input Definition Screen.

2.6 A SAMPLE COPY PROCEDURE

You can control the COPY process through the VS Procedure language. The COPY utility acquires all necessary information through GETPARM requests. A complete list of COPY GETPARMs is given in Appendix A. Consult the VS Procedure Language Reference for details concerning procedure syntax.

The following is a sample procedure that performs the periodic reorganization of a frequently modified indexed file. The procedure specifies the input file and selects the reorganization option with new packing densities of 50 percent. The procedure then provides the file parameters of the output file. Copy does not make the Options function available to you when you copy a single file. As a result, you should scratch any existing versions of the output file before running the procedure.

```
PROCEDURE
RUN COPY
ENTER INPUT COPY=FILE, FILE=PLEASE, LIBRARY=REORGME, VOLUME=SYSTEM
ENTER OPTIONS REORG=YES, IPACK=50, DPACK=50
ENTER OUTPUT FILE=THANKS, LIBRARY=REORGME, VOLUME=SYSTEM
ENTER EOJ 16
RETURN
```

CHAPTER 3
THE COPYOIS UTILITY

3.1 INTRODUCTION

The COPYOIS utility transfers OIS files to and from the VS, using OIS hard-sectored, standard label diskettes. This utility allows you to convert OIS BASIC source programs to VS BASIC source programs and OIS data files to VS data file format. COPYOIS enables you to implement integrated data processing and word processing applications in the VS and/or OIS environment.

COPYOIS either runs interactively or through a procedure or an Assembly language program.

> **NOTE**
>
> The OIS does not support soft-sectored diskettes. You must, therefore, have an archive diskette drive when you run COPYOIS on the VS25 or the VS45.

COPYOIS supports the following functions:

- Initialize a diskette with an OIS standard label
- Copy single or multiple OIS list processing, data base files, or BASIC source and data files from the VS to an OIS diskette
- Copy single or multiple OIS files from an OIS diskette to the VS, and/or convert them to VS files
- Create and display and/or print the diskette catalog listing of OIS files on an OIS diskette
- Rename an OIS file on the OIS diskette
- Delete an OIS file from the diskette
- Assign a password to the OIS diskette volume and/or any file node name residing on the diskette
- Convert an OIS file already residing on the VS (through a previous copy or a telecommunications operation) to a VS file

An overview of COPYOIS processing is provided in Figure 3-1.

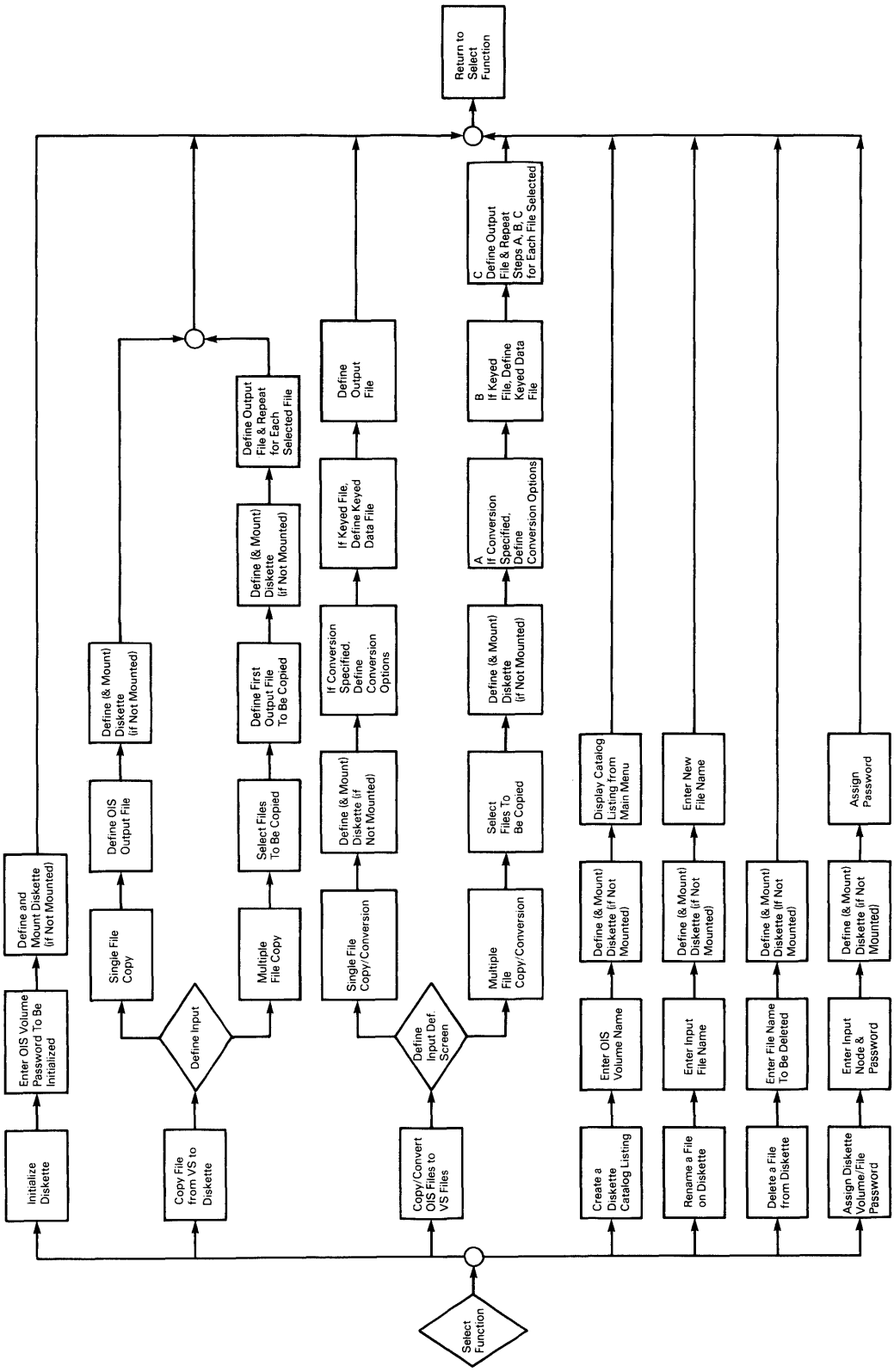


Figure 3-1. COPYOIS Processing

3.2 FUNCTION SELECTION

When COPYOIS processing begins, the main menu is displayed. You can select any one of the following functions:

PF Key	Function
1	Initialize Diskette
2	Copy File from VS
4	Copy/Convert OIS File
5	Create Diskette Catalog Listing
7	Rename File on Diskette
8	Delete File from Diskette
9	Assign Diskette Volume/File Password
16	Terminate Processing

3.3 MOUNTING VOLUMES

You can mount any VS or OIS volume from within a COPYOIS function. Whenever you specify the name of an unmounted volume, COPYOIS displays the Mount screen. The Mount screen uses the same parameters for VS and OIS volumes, but the parameters are interpreted differently in each case. The parameters for mounting a VS volume are defined as follows:

VOLUME — The VS name of the volume you mount.

DEVICE — Specifies whether the volume is a disk or diskette. You can specify DISK or DISKETTE; DEVICE defaults to DISK for VS volumes. Note that a diskette must be a VS standard label (SL) diskette.

UNIT — The device number of the drive.

PASSWORD — A password is not relevant for VS volumes, and you cannot modify this field.

The parameters for mounting an OIS volume are defined as follows:

VOLUME — The VS name of the OIS volume you are mounting.

DEVICE — The type of volume. Because COPYOIS can only access OIS diskettes, DEVICE defaults to DISKETTE. You cannot modify this field.

UNIT — The device number of the drive.

PASSWORD — This field is optional, unless you initialize an OIS diskette that already has a password. If you plan to initialize a diskette within the COPYOIS session, enter the correct password. The Mount screen is displayed if you attempt to initialize the diskette and did not enter the password. Also, you must enter the correct password if you are initializing a diskette that was not previously mounted.

COPYOIS recognizes an OIS volume by the OIS volume name you use, and not by the VS volume name you use for the physically mounted OIS diskette. If you enter an OIS volume name (for input, output, or initialization) different from the name on the OIS diskette, a Mount screen appears.

3.4 INITIALIZING DISKETTES

COPYOIS enables you to initialize a hard-sectored diskette to the OIS standard label format when you press PF1. This diskette can subsequently be used by COPYOIS as an output medium to transfer OIS files to an OIS, and is identical to a standard label diskette initialized on an OIS. COPYOIS performs the same integrity checking as the DISKINIT utility discussed in Chapter 6.

When you press PF1, the OIS Volume Definition screen is displayed with the following parameters. You can return to the main menu without initializing the diskette by pressing PF1.

VOLUME — The name of the OIS volume you want to initialize.

PASSWORD — The password you assign to the OIS volume. This field is optional.

If you have not already mounted the diskette, the Mount screen requests you to enter parameter values as described in Section 3.3.

After you enter the parameters and mount the diskette, COPYOIS initializes the volume. When the diskette is initialized, COPYPOIS displays the main menu, with a completion message indicating the OIS and VS volume names.

3.5 COPYING OIS FILES FROM VS TO DISKETTE

COPYOIS enables you to copy unconverted OIS files residing on the VS to an OIS diskette when you press PF2. The OIS file can be an unconverted list processing data base file, or an unconverted OIS BASIC source file or data file that has been previously transferred to the VS by COPYOIS, the Remote WangNet facility, or TCCOPY. You can copy a single file or a group of files selected from a VS library containing OIS files. When you press PF2, the Input Definition screen appears with the following parameters:

FILE — The name of the OIS file on the VS you want to copy. Supply a value for this field only when copying a single file.

LIBRARY — The VS library in which the OIS file(s) reside.

VOLUME — The VS volume on which the OIS file(s) or library reside.

Press PF1 to terminate the file copy. COPYOIS performs a single file copy when you press ENTER. COPYOIS performs a multiple file copy when you press PF9.

3.5.1 Copying a Single File

When you identify the file you want to copy on the Input Definition screen, press ENTER to identify a single file copy. COPYOIS then requests you to enter the OIS output file and volume names on the OIS Output File Definition screen.

VOLUME — The name of the OIS volume on which the OIS file is to reside.

FILE — The name of the OIS file you want to copy. A default value is generated automatically, using the VS library name as the first node and the file name as the second node. However, you can change this value to any valid OIS name. OIS BASIC file names are known as file-identifiers. A file-identifier contains a file name, which consists of one to six node names (each containing one to eight letters or digits) concatenated by a period (e.g., basic.srce.pjk.test).

Continue processing by pressing ENTER or respecify the input by pressing PF1. If you have not already mounted the diskette, the Mount screen requests you to enter parameter values as described in Section 3.3.

After you mount the output OIS volume, the file is copied to the diskette, and the main menu is displayed with a completion message.

3.5.2 Copying Multiple Files

You can perform a multiple file copy by entering the names of the library and volume containing the files and pressing PF9 from the Input Definition screen. COPYOIS then displays a menu that lists all the files in the specified library. You select the files you want to copy by typing a nonblank character in front of each file name. You can also press PF1 to respecify the input on the Input Definition screen. When you have selected all the files to be copied, pressing ENTER continues processing.

You must then define the output OIS file parameters through a series of OIS Output File Definition screens. The OIS Output File Definition screen requests values for the following parameters:

VOLUME — The name of the OIS volume on which the files are to reside.

FILE — The name of the OIS file you want to copy. A default value is generated automatically if you use the VS library as the first node, followed by a period, with the file name as the second node.

If the specified output OIS diskette is not mounted, the Mount screen is displayed. Enter the parameters required to mount the OIS diskette, as defined in Section 3.3, and mount the diskette. COPYOIS displays one OIS Output File Definition screen for each file you selected. This procedure allows you to change the default file name. When you enter the name of the last file you want to copy and the files is copied to the diskette, the main menu is displayed with a completion message.

3.6 COPYING/CONVERTING OIS FILES TO VS

COPYOIS enables you to copy and/or to convert a single or group of OIS source and data files to the VS. The OIS file can reside on an OIS diskette or on a VS volume. If the file resides on an OIS diskette, you can copy and convert the files in a single operation, or simply transfer the OIS files to the VS for conversion at a later date. If the OIS files reside on a VS volume, you can copy them to another VS volume or convert them to the VS file format. Note that even if the file resides on the VS, you must enter the file name in OIS format (LIBRARY.FILE).

COPYOIS enables you to copy and/or convert OIS files when you press PF4 from the main menu. The OIS file can be a list processing data base file, or an OIS BASIC source or data file. OIS files that already reside on the VS have been previously transferred to the VS by COPYOIS, the Remote WangNet facility, or TCCOPY. You can copy a single file or a group of files. When you press PF4 from the main menu, the Input Definition screen is displayed with the following parameters:

VOLUME — The volume name on which the file(s) you want to copy and/or convert resides.

FILE — The OIS name of the file(s) you want to copy and/or convert. This field can be omitted for a multiple file copy and/or conversion from an OIS diskette. For multiple file operations, you can also specify a partial file name. If you specify a partial file name, COPYOIS displays for selection only those files whose names begin with the partial file name.

DEVICE — The type of device on which the file(s) you want to copy and/or convert resides. The files can reside on a VS or an OIS volume. DEVICE defaults to VS.

CONVERT — Determines whether or not the file is to be converted. If you enter YES, the OIS files are converted to VS files. If you enter NO, the file is copied, but not converted.

Pressing ENTER indicates a single file operation; pressing PF9 specifies a multiple file copy and/or conversion. You can also return to the main menu by pressing PF1. If you specify an unmounted volume, COPYOIS allows you to mount the volume, as described in Section 3.3.

NOTE

When you copy multiple OIS files that reside on the VS, enter a library name and a device. When you copy multiple OIS files that reside on diskette, enter a volume name and a device. The largest classification of data that can be copied from the VS is the library. The largest classification of data that can be copied from the OIS is the volume.

3.6.1 Copying and/or Converting a Single File

COPYOIS can transfer a single file with or without conversion when you identify the file on the Input Definition screen and press ENTER. When you are copying a single file without any conversion, you need to identify only the output VS file location through the following parameters:

FILE — The VS file name of file you want to produce.

LIBRARY — The VS library in which the VS file is to reside.

VOLUME — The VS volume on which the file is to reside.

However, if you are converting the input OIS file, you may have to first specify conversion options that depend on the type (BASIC source, data, or keyed) of OIS file. Because there are no conversion options for BASIC source files, you need only identify the output VS file, as described above. Refer to Section 3.6.3 for a description of the type of conversion performed on an OIS BASIC source file.

If you are converting a BASIC data file, COPYOIS requests the following conversion options before you define the VS output file. (The field definitions can be accessed during processing by pressing PF13 from the Output Format Definition screen.)

FORMAT — The format of the BASIC data file record (FIXED, VARIABLE, and KEYED). Consecutive files can have FIXED or VARIABLE format; indexed files have FORMAT = KEYED.

LENGTH — The keyed or fixed record length, or the maximum length for the variable record length data file. The actual record length is required for keyed and fixed-length record files. The maximum record size is required for variable-length record files. The maximum fixed record length is 2048. The maximum variable record length is 2024. The maximum keyed record length is 2040.

NUMERICS — For fixed, variable, and keyed files, you must specify whether or not any numeric conversion is necessary. You must also indicate if all numerics encountered in the data record are to be converted in the output data record, that is, either to 4-byte integers, 8-byte floating-point numbers, or 8-byte packed decimal numbers. All numerics encountered are converted to the same type. If any number is outside the range of the VS numeric representation, that number is converted to zero. Because an OIS numeric occupies 9 bytes in the data record (for fixed or keyed files), any trailing spaces are output as zeros in the data record. For variable length records, the exact number of bytes required by the output number (that is, 4 or 8 bytes) is output in the data record. You do not have to specify numeric conversion if this field does not contain numeric data.

DECIMALS — The number of decimal positions for the output packed number.

KEYFIELD If the data file is keyed, specify whether or not an exact image of the key value in the key file is also present in the actual data record. If the key is embedded, you must also specify its starting location in KEYSTART.

KEYSTART The position of the key field in the data record if the key is embedded. If the key is not embedded, specify the position in the data record to insert the key. If the key is not embedded, the record size is increased automatically to account for the addition of the key field in the output record.

For keyed file conversions, if the key file allows duplicates, the occurrence (OCC) number, converted to character format of length five (5), is appended to the key field. The record size and key field length is increased automatically to account for the addition of the OCC number in the output data record.

When you convert a keyed file, you must also provide the keyed data file location on a following screen in the following parameters: VOLUME, FILE, and DEVICE. This screen is identical to the Input Definition screen except the CONVERT parameter is not included.

Once you have specified the conversion options for data and keyed files, COPYOIS requests the file location of the output VS file, as described above. When the file has been copied and/or converted, the main menu is displayed with an appropriate completion message.

A conversion statistics report is generated for all data files converted to VS format. You can display or print the report by pressing PF12 from the main menu. The statistics report summarizes the options you requested for data file conversion. The report contains the number of records in the file, the maximum or the actual record size for the records in the data file, the type of output file generated, and the type of numeric data conversion performed.

3.6.2 Copying and/or Converting Multiple Files

When you are performing a multiple file operation, COPYOIS displays a menu listing all the files in the specified library. When copying and/or converting from a VS volume, only OIS files residing in the VS libaray appear for selection. When copying and/or converting from the OIS volume, if you do not specify a file node in the FILE field of the input definition screen, files on the volume are displayed. If you enter a file node name in the Input Definition screen FILE field, all files whose node names begin with the file node(s) entered are displayed for selection.

Select the files you want to copy by typing a nonblank character in front of the file name. You can also specify whether or not you want to convert the file if you have not yet done so by entering YES or NO in the CONVERT field. The default conversion option is displayed using the value you supplied in the Input Definition screen. Press ENTER to perform the selected file copy, or press PF1 to respecify the input. When you convert a BASIC data file, the Output Format Specification screen described in Section 3.6.1 is displayed. If the file is keyed, you must also specify the keyed data file location, as described in Section 3.6.1.

After you have specified any conversion options required for the file, you indicate the output VS file location for the first file through the FILE, LIBRARY, and VOLUME parameters. The first file entered is copied, and then you specify the conversion options and output file location for the next OIS file. This process enables you to change the file name and specify conversion options individually for each file. When all files are copied and/or converted, the main menu is displayed.

A conversion statistics report is generated for all data files converted to VS format. You can press PF12 from the main menu to display and/or print the report. The statistics report summarizes the options you requested for data file conversion. The report contains the number of records in the file, the maximum or the actual record size for the records in the data file, the type of output file generated, and the type of numeric data conversion performed.

3.6.3 Source File Conversion

When you enter the source file name you want to transfer and/or copy to the VS, the conversion option should be specified with the copy. COPYOIS checks the input file to verify that the file is actually an unprotected, OIS BASIC source file. If the file cannot be converted to a VS BASIC source program, it is assumed to be an OIS BASIC data file, and you are requested to enter the parameters of the the data files structure.

The output source program is generated so that COPYOIS performs the conversion as follows:

- All file SELECT statements are placed before any executable code.
- Lowercase letters, except alphanumeric-literal-string constants and comment lines, are converted to uppercase letters, as VS BASIC requires.
- Output statements take any VS BASIC spacing requirements into account.
- The original statement indentation is preserved.
- Statements incompatible with VS BASIC are preceded by an asterisk (*) to convert the statement to a comment.
- Messages generated during conversion are flagged as comments by an asterisk * to convert the statements to comments.
- All compound statements in an OIS source line are generated as separate source statement lines.

COPYOIS separates OIS BASIC source lines longer than 66 characters (excluding the line number) in the following ways:

- If multiple statements follow the OIS BASIC ERROR statement on the same line, all the statements following ERROR, on the current input line, are converted to * (asterisk) comment statements.

- Variable names and BASIC keywords are not separated from one line to the next.

- If the line being separated is a REM statement, an additional REM statement is generated to contain the remaining text.

- If the characters you want to separate are actually an alpha-literal-string, the string continues to Line 71. An exclamation sign (!) follows in Column 72, and the remaining string starts in Column 7 of the next line.

For example, consider the following OIS BASIC input statement:

```
0200    If str(a$,2,1) = "*" then input"Please specify the number of items to be
 included in the list ? ",s:ifs=0thenprint "Zero items cannot be specified":goto
200
```

COPYOIS converts this input statement to the following VS BASIC lines:

```
000200    IF STR(A$,2,1) = "*" THEN INPUT "Please specify the number of!
000202 items to be included in the list ? ",S
000204    IF S = 0 THEN PRINT "Zero items cannot be specified"
000206    GOTO 200
```

NOTE

Because incompatible BASIC statements are converted to comments, you should not renumber the VS BASIC source: line number references can be contained in a commented line. In this case, you may have difficulties trying to track the line reference after renumbering.

All numeric and alphanumeric expressions are examined for incompatible functions (for example, OCC # or ERRNO). If incompatible functions occur in the expression, the statement is flagged; that is, output as a * (comment) statement. In addition, the expressions are examined for functions that must be converted to equivalent VS BASIC statements (for example, CONVERT). The expression is translated so that its execution in VS BASIC is as close as possible to its execution in OIS BASIC. Refer to the VS BASIC Language Reference and the OIS BASIC Reference for more information about the BASIC language.

3.7 CREATING A DISKETTE CATALOG LISTING

COPYOIS allows you to create a diskette catalog listing of your OIS files on the OIS diskette by pressing PF5. When you select this option, COPYOIS requests you to identify the OIS input volume name.

If you specify the name of an unmounted diskette, you can mount it as described in Section 3.3. When COPYOIS has created the listing, the main menu is displayed. You can then view the listing by pressing PF14 from the main menu.

3.8 RENAMING A FILE ON DISKETTE

COPYOIS allows you to rename a file or files on diskette. If you press PF7 from the main menu, the following parameters appear:

VOLUME — The OIS volume name on which the file to be renamed resides.

FILE — The file name to be renamed.

After the input file has been identified, if you did not mount the diskette, the utility requests you to enter the parameters (refer to Section 3.3). When the diskette is mounted, COPYOIS requests you to enter the new name of the file. After you assign a new file name, the main menu is displayed.

3.9 DELETING A FILE FROM DISKETTE

COPYOIS allows you to delete an OIS file from the OIS diskette. If you press PF8 from the main menu, the following parameters appear:

VOLUME — The OIS name of the volume on which the file to be deleted resides.

FILE — The OIS file name of the file you want to delete.

If you did not mount the diskette, identify the file you want to delete, enter the parameters, and mount the diskette. After you mount the diskette, the file is deleted and the main menu is displayed.

3.10 ASSIGNING DISKETTE VOLUME/FILE PASSWORD

COPYOIS allows you to assign or to reassign a password to a volume or a file on a diskette. If you press PF9 from the main menu, the following parameters appear:

VOLUME — The OIS name of the volume on which the name you assign resides.

NAME — The node name to which you want the password assigned. If you want to assign a password to a volume that already has a password, enter the password in the form password=:. If the volume does not contain a password, this field is left blank. If you assign a password to a file node, enter a name to the file node name that will be assigned a password. If the node has already been assigned a password, you must enter the old password or any password of a higher level. If the volume has a password, enter password=: file name node.

PASSWORD — The password you assign to the volume or file node.

If you specify the name of an unmounted diskette, you can mount it as described in Section 3.3. When you mount the diskette, assign or reassign the password. After you assign the password, the main menu is displayed.

3.11 A SAMPLE COPYOIS PROCEDURE

You can control COPYOIS processing through the VS Procedure language. A complete list of COPYOIS GETPARMs is provided in Appendix A; refer to the VS Procedure Language Reference for details about procedure syntax.

The following procedure mounts an OIS diskette, copies an OIS data file from the diskette, and then converts it to a VS file. Once the file is copied, the procedure displays the statistics report and dismounts the diskette. VS Procedure language automatically converts lowercase to uppercase. For this reason, the colon (:), equals sign (=), period (.) and any OIS volume names and file names that are lowercase must be enclosed in quotes. Any names that contain a period or a colon in the string must be enclosed in quotes.

```
PROCEDURE
RUN COPYOIS
ENTER MENU 4
ENTER INPUT VOLUME=OISVOL, FILE="Keyed.Reclen.409",DEVICE=OIS
ENTER MOUNT VOLUME=VOL, UNIT=023
ENTER OPTIONS FORMAT=KEYED, LENGTH=409, NUMERICS=INTEGER, DECIMALS=,
              KEYFIELD=YES, KEYSTART=1
ENTER DATAFILE VOLUME=OISVOL, FILE="pjadata.datak11", DEVICE=OIS
ENTER OUTPUT FILE=OISKEYED, LIBRARY=PJADATA, VOLUME=ZENITH
ENTER MENU 12
ENTER MENU 16
DISMOUNT DISK VOL
RETURN
```

CHAPTER 4
THE COPY2200 UTILITY

4.1 INTRODUCTION

The COPY2200 utility provides a convenient mechanism for information exchange between the VS and the Wang 2200 Series systems. COPY2200 transfers data to and from 2200 diskettes and converts the files to and from the 2200 file formats. The VS can access the information on the following 2200 diskettes. You must mount all 2200 diskettes on the VS as nonlabeled (NL) diskettes.

- Single-sided, single-density, hard-sectored diskettes (Wang White Label) used by the 2200VP, 2200LVP, and 2200MVP systems
- Double-sided, double-density, soft-sectored diskettes (Wang Red Label) used by the 2200LVP and 2200SVP systems

NOTE

To access 2200 soft-sectored diskettes, COPY2200 requires a diskette drive that supports soft-sectored diskettes.

COPY2200 can also generate and access 2200 and VS nonlabeled (NL) image files. An image file, described in detail in Section 4.2, is a byte-by-byte copy of an entire diskette or disk. An image file enables you to separate the copying and conversion processes.

COPY2200 performs the following functions. Function selection is described in Section 4.3. COPY2200 processing can also be accomplished through the VS Procedure language, as described in Section 4.8.

- Creates VS files from 2200 files residing on a 2200 diskette or image file
- Converts VS files to the 2200 format and then transfers the files to a 2200 diskette or image file
- Creates a 2200 image file from the contents of one or more 2200 diskettes
- Generates a 2200 diskette(s) by copying a 2200 image file residing on the VS to a diskette(s)

An overview of COPY2200 processing is provided in Figure 4-1.

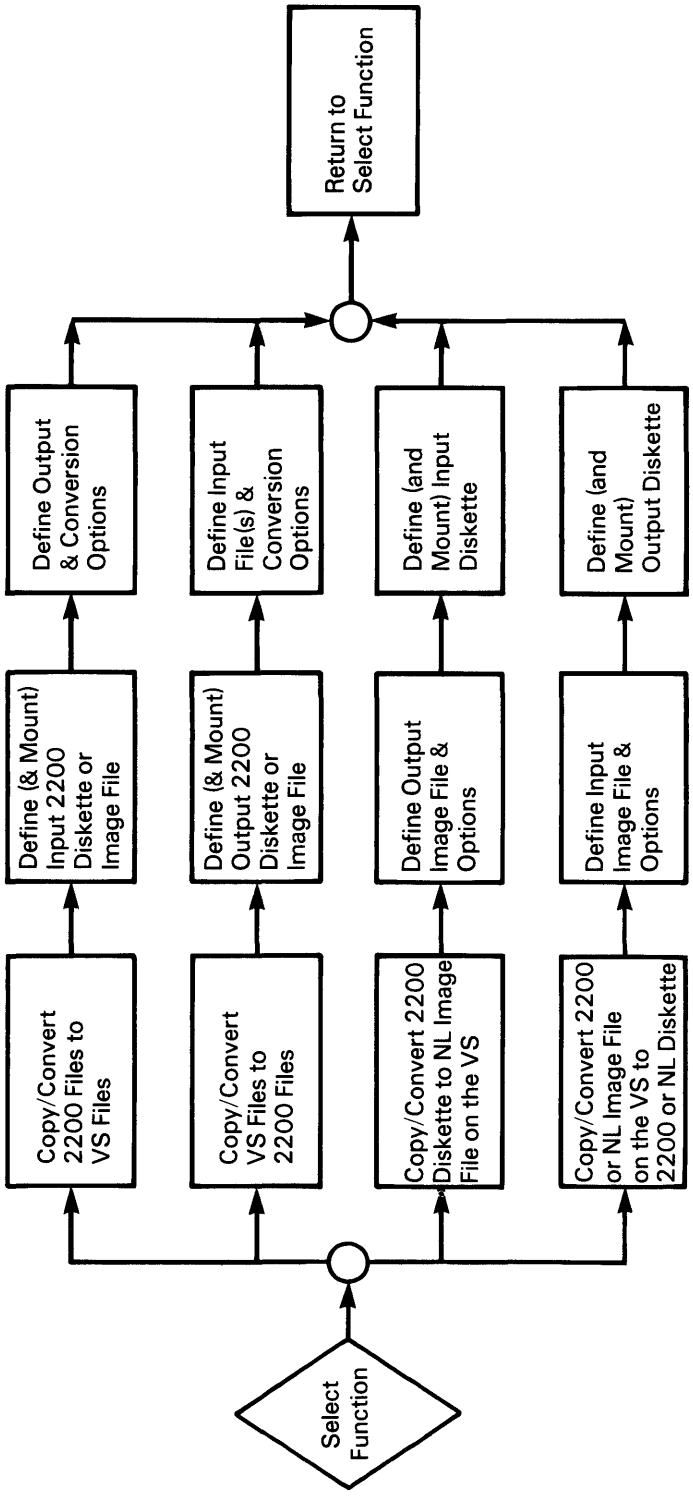


Figure 4-1. COPY2200 Processing

4.2 IMAGE FILES

An image file is an exact image of a file on a disk or diskette, and consists of consecutive 256-byte records. You can create an image file in two ways. First, you can copy the entire contents of a diskette(s) (which can include many diskette files) into an image file, with the contents undergoing no reformatting. Each sector of the diskette becomes a 256-byte record in the image file. Second, you can convert the contents of a VS file into the diskette format and then copy the contents into an image file, where each 256-byte record represents a diskette sector.

You can use image files to store the contents of 2200 or NL diskettes or to make copies of these diskettes. The files can also be used to temporarily save diskette contents until they are converted into VS format (freeing the diskette drive) and to store converted VS files until the diskette drive is available.

4.3 SELECTING A FUNCTION

When processing begins, COPY2200 displays the main menu. Select one of the following functions:

PF Key	Function
1	Create VS Files from a 2200 Diskette or Image File
2	Create a 2200 Diskette or Image File from VS Files
9	Create a VS Image File from One or More 2200 Diskettes
10	Create One or More 2200 Diskettes from a VS Image File
16	Exit Program

After you select a function, processing continues. Sections 4.4 through 4.7 describe the subsequent processing for each function.

4.4 COPYING TO THE VS FROM THE 2200

If you press PF1 from the main menu, COPY2200 can create VS files from program files and certain types of data files; the files can reside on 2200 diskettes or in 2200 image files. When you press PF1, the Input Definition screen requests you to supply the following input information and to mount the input 2200 diskette or the volume on which the image file is mounted (image volume), if required. If you mount the 2200 diskette through the Command Processor rather than through COPY2200, you must mount the diskette as an NL diskette. You can also return to the main menu from the Input Definition screen by pressing PF1.

2200 DISKETTE or FILE, LIBRARY, VOLUME	The source of the input information. The source can be the name of a 2200 diskette or the file name, library, and volume of a 2200 image file.

STARTER Precedes the first character of any 2200 file name when that character is invalid on the VS. Valid 2200 characters are described following this description of parameters. A STARTER value must be an alphabetic or numeric character @ or $. The default response is V.

FILLER Replaces all other invalid characters within the 2200 file name. Valid 2200 characters are described following this description of parameters. A FILLER value can be any allowable VS file name character (A to Z, 0 to 9, #, @, or $). The default response is #.

DEVICE# The device number of the drive on which the input volume is mounted. By specifying a value for DEVICE# and pressing PF4, you can mount the input volume through COPY2200. COPY2200 automatically mounts the diskette as a NL volume. You can omit DEVICE# if the input volume is already mounted.

You must specify the STARTER and FILLER values because of the differences in valid file names on the 2200 and the VS. File naming conventions on the VS are more restrictive, with only alphabetic, numeric, and certain special characters (#,@, and $) allowed for a file name. File names on 2200 Series systems can consist of any combination of alphanumeric characters (all alphabetic, numeric, and special characters). The differences create the need to convert file name characters that are valid on the 2200, but invalid on the VS, into valid VS format values. Therefore, you must specify STARTER and FILLER values for both diskette and image file conversion.

After you define the input, the Copy Mode screen requests you to select the 2200 to VS copy mode from among the following options. Program and/or data files can be copied. Data files include several types of files, described in Subsection 4.4.1.

A Copy all 2200 files to the VS (data and program files). All files on the 2200 diskette or image file are copied and converted into the VS format.

D Copy all 2200 data files to the VS (no program files). Only Data files are copied and converted into VS format.

P Copy all 2200 program files to the VS (no data files). BASIC program files are copied and converted into the VS format. BASIC-2 files can be copied, but BASIC-2 syntax cannot be correctly converted or flagged.

L Specify a list of files (any type) to be copied. This option displays a listing of all files on the diskette or in the image file, and indicates which are data files and which are program files. You then select individual files you want to copy and convert into VS format from this list and press ENTER.

Two separate screens, one for data files and one for program files, request output information. For option A, both screens require information. For option D, only the Data Files Output screen requires information. For option P, only the Program Files Output screen requires information. If you select option L, one screen or both screens appear. If you select option A, the Data Files Output screen is displayed before the Program Files Output screen. The following sections describe the parameters required on the output screens.

4.4.1 Data Files Output Screen

The Data Files Output screen requests you to enter the following information to define the VS output data file:

LIBRARY
VOLUME — The location of the output file(s) to be created by the copy operation. The file name of the output file is the same as the 2200 file, modified as necessary by the previously defined STARTER and FILLER values.

FILECLAS — The file class of the output files to be created by the copy operation. The default response is obtained from your default values (assigned through the SET function of the Command Processor or through the VS Procedure language SET statement).

TYPE — The type of conversion performed. Seven conversion types are described. You must specify a RECSIZE value for types F, V, T, and X. You must specify the TRANSL values for types T and X.

- F Creates an output data file with consecutive, fixed-length records.
- V Creates an output data file with consecutive, variable-length records.
- T Standard Telecommunications (TC) copy. A VS data file is created from an input 2200 standard TC format data file (4 by 62 logical record size). One VS record is generated from each TC record.
- X Extended TC copy. A data file is created from an input 2200 extended TC format data file (4 by 62 logical record size). One or more TC records is used for each VS record, with or without a header.
- E Source copy. 2200/VS EDITOR format, as created by COPY2200 (on the VS through image files) or by the 2200/VS Source Editor (a user aid on the 2200).
- S Sector copy. Copies all sectors of the files exactly as catalogued, from first to last, including any end-of-file (EOF) and/or control records. You are responsible for ensuring that the files are in VS format.
- N None. You must specify the conversion type separately for each file.

COMPRESS Determines whether or not the output file(s) will be compressed. The default response is YES.

RECSIZE The size of the records in the files being copied. You must specify RECSIZE for TYPE values F, V, T, and X. Types F and V produce multiple output records if an input record is too long. All four TYPE options are padded with blanks or truncation, where required. The valid values of RECSIZE for each file organization type follow:

- F Any valid VS record size or 0. If the record size is 0, RECSIZE is assumed to be that of the first record in the 2200 file.
- V The maximum VS record size for the variable-length file. The size must be a legal nonzero value.
- T A valid VS record size to be created from each TC record or 0. If the record size is 0, the output file is variable length, with a maximum length of 192 bytes.
- X Either a valid VS record size or 0. If you specify a nonzero record size, each output record is created from as many TC records as needed, assuming each record determines an 80-byte piece of the logical (output) record. If the record size is 0, the first TC record is assumed to be the 2-byte binary RECSIZE; the output records are then constructed as described previously.

TRANSL Translates 2200 information from EBCDIC into ASCII during the copy or process.

If you specify the file type as F, V, or N, the Numeric Data Format screen follows the Data Files Output screen. The Numeric Data Format screen requests the output format for the 2200 numeric values through the following parameters:

TYPE The output format of the 2200 numeric values for all the specified data files. The output format consists of the following types:

- D 2200 (decimal) floating point (i.e., no change in format).
- F VS (hexadecimal) floating point.
- B VS (fullword) binary.
- H VS (halfword) binary.
- P VS packed decimal. You must specify the LENGTH and PLACES parameters.
- N None. Indicate the output format separately for each file. You need only specify the format if the file contains numeric data.

LENGTH — Field length in bytes for packed decimal values. Lengths of 1 through 16 are allowed.

PLACES — Implied decimal places for packed decimal values. Responses of 0 through 99 are allowed.

After you specify all parameters, the utility copies the file(s). The main menu is displayed when the copying is complete. You can then dismount the 2200 diskette by pressing PF14.

4.4.2 Program Files Output Screen

The Program Files Output screen requests the following information, which is similar to that required on the Data Files Output screen:

LIBRARY VOLUME — The location of the output file to be created by the copy procedure. You must specify the library and volume names.

COMPRESS — Determines whether or not the output files will be compressed. The default response, YES, compresses the data. If you enter NO next to COMPRESS, the data is not compressed.

FILECLAS — The file class of the output files to be created by the copy procedure. The default response is obtained from your default values.

CONVERT — Determines whether or not the 2200 BASIC syntax is converted to VS BASIC syntax during the copy procedure. The default response, YES, performs the conversion; a NO response leaves the file in the input 2200 BASIC or BASIC-2 syntax.

After you specify the parameters, the utility copies the file(s). The main menu is displayed when the copying is complete. You can then dismount the 2200 diskette by selecting PF14 from the main menu.

4.5 COPYING TO THE 2200 FROM THE VS

When you press PF2, you can create Wang 2200-format diskettes or image files from certain types of VS files. The Output Definition screen requests the following information and allows you to mount the output 2200 diskette, if required. If you mount the 2200 diskette through the Command Processor, rather than through COPY2200, you must mount the diskette as a NL diskette.

2200 DISKETTE or FILE, LIBRARY, VOLUME — The location of the output information. The location can be the name of a 2200 diskette, or the file name, library, and volume names of the 2200 image file to be created.

FILECLAS — The file class of the image file to be created by the copy procedure. The default response is obtained from your default values.

COMPRESS Determines whether or not the image file will be compressed. The default response, YES, compresses the file. If COMPRESS = NO, the data is not compressed.

SECTORS The maximum sectors of information to be created on the 2200 diskette or in the image file. For diskette output, the maximum sectors is 1232 for hard-sectored diskettes and 3874 for soft-sectored diskettes. For diskette output, you can also specify ALL to allow the utility to use as many sectors as are available on the type of diskette mounted. SECTORS defaults to 1024. If the VS file being copied requires more sectors than 1232 (for hard-sectored diskettes) or 3874 (for soft-sectored diskettes), you must convert the file into an image file, which has a maximum of 32,767 sectors, and then copy the file to a 2200 disk through multiple diskettes.

INDEX The number of sectors to be allocated for the Catalog Index on the output 2200 diskette or diskette image file. Index sectors, except the first sector, can contain cataloguing information for 16 files; the first sector can contain information for only 15 files. The default value for the index sector count is 16. The maximum number of index sectors is 255.

DEVICE# The device number of the drive on which the output 2200 diskette or image volume is mounted. By specifying a value for DEVICE# and pressing PF4, you can mount the output diskette or the image volume through COPY2200. COPY2200 automatically mounts a 2200 diskette as a nonlabeled volume. You can omit DEVICE# if you have already mounted the output volume.

If you use a diskette as an output volume, another screen informs you that all data currently residing on the diskette will be destroyed by the copy operation. You can then continue the copy operation by entering YES, or terminate the function by entering anything else.

The Input Definition screen requests you to enter the file, library, and volume names of the first file to be copied, and allows you to mount an input volume by pressing PF4.

After you define the first input file, the Conversion Type Definition screen requests you to enter the output type for the file being created. There are four possible output types, described as follows.

T Standard TC copy. The VS data file is copied and converted to the 2200 TC format (4 by 62) logical record size). This is the standard format consisting of one VS record for each TC record.

X Extended TC copy. The VS data file is copied to a 2200 TC format file (4 by 62 logical record size). This format assumes one or more TC records for each VS record, with or without a header.

E — Source Copy. 2200/VS Editor format files are used with the 2200/VS source Editor (a user aid on the 2200). The input file must have 80-byte records. If a VS BASIC source file is copied, the resulting 2200 file can only be edited by the 2200/VS Editor user aid.

S — Sector Copy. The VS data file is copied directly to the physical sectors of the output 2200 file, with no additional trailer or other control information. (Trailer information can include information at the end of the sector.) The input file must have 256-byte records. You are responsible for including the 2200 file control information within the 256 byte record.

The Conversion Type Definition screen also requests RECSIZE values for output types T and X, and a TRANSL value for output type X.

RECSIZE — Indicates the size of the records in the files being copied. RECSIZE is required for output types T and X. Legal RECISIZE values are discussed below.

- T — Either a valid TC record size (1 to 192) or 0. If the record size is nonzero, the output records will be padded or truncated to the specified record size. If the record size is 0, the output records will be variable-length (1 to 192), with blank-padding and/or truncation of characters beyond Position 192.

- X — Either a valid VS record size (1 to 2048) or 0. If the record size is nonzero, each VS record will produce one or more TC records, each of which is an 80-byte piece (with trailing blanks truncated) of the specified record size. If the record size is 0, the output records will be constructed as a nonzero record length file, with a record size equal to that of the VS file. Also, the output records will contain an additional 2-byte initial record that contains this record size.

TRANSL — Translates VS information during the copy operation from ASCII to EBCDIC.

You can copy additional VS files; a Conversion Type Definition screen is supplied for each file. After you specify all the files you want to copy, press PF16 from an Input Definition screen to initiate copying the specified files. Upon completion, the main menu is displayed. You can then dismount the 2200 diskette by pressing PF14.

4.6 CREATING IMAGE FILES FROM DISKETTES

When you press PF9 from the main menu, VS image files are created from 2200 diskettes. The Output Definition screen requests the following information and allows you to mount the volume on which the image file is to reside. You can also return to the main menu by pressing PF1. The output volume is mounted as a VS standard label (SL) volume by COPY2200. Thus, if you mount the volume through the Command Processor, rather than through COPY2200, you must mount it as a SL volume.

FILE, LIBRARY, VOLUME — The location of the image file to be created.

FILECLAS — The file class of the output file to be created by the copy procedure. The default response is obtained from your default values.

COMPRESS — Determines whether or not the image file will be compressed. The default response, YES, compresses the data. If you enter NO, the data is not compressed.

TYPE — The input diskette type. A response of T indicates a 2200-format diskette; a response of N indicates a VS-supported NL diskette.

MULTIPLE — Indicates whether or not multiple diskettes are to be used in the copy. Enter YES or NO.

SECTORS — The total number of sectors on the input diskette to be copied. The image file cannot exceed the size determined by SECTORS. To copy only those sectors actually used by the 2200, you can enter USED for the number of sectors. You can also supply a specific number. If MULTIPLE = NO, you can specify ALL to copy the entire diskette. SECTORS defaults to ALL.

DEVICE# — The device number of the drive on which the output volume for the image file is to reside. By specifying a value for DEVICE# and pressing PF4, you can mount the output volume through COPY2200. You can omit DEVICE# if the output volume is already mounted.

After you define the output image file, the Input Definition screen requests the following information and allows you to mount the input 2200 diskette. If you mount the 2200 diskette through the Command Processor, rather than through COPY2200, you must mount the diskette as a NL diskette. If MULTIPLE = YES, the Input Definition screen is repeated until you press PF16 or reach the specified SECTORS value.

DISKETTE — The name of the input diskette volume.

SECTORS — The number of sectors to be copied from the specified diskette if MULTIPLE = YES. You can specify ALL or a specific number; SECTORS defaults to ALL.

DEVICE# The device number of the drive on which you mount the input 2200 diskette. By specifying a value for DEVICE# and pressing PF4, you can mount the input diskette through COPY2200. COPY2200 automatically mounts the 2200 diskette as a NL volume. You can omit DEVICE# if you have already mounted the input volume.

After all diskettes are copied, the main menu is displayed. You can then dismount the most recently mounted diskette by selecting PF14 from the main menu.

4.7 CREATING DISKETTES FROM IMAGE FILES

If you press PF10 from the main menu, 2200 diskettes can be generated from VS image files. The Input Definition screen requests the following information and allows you to mount the volume on which the diskette image file resides. You return to the main menu by pressing PF1. If you mount the input volume through the Command Processor instead of COPY2200, you must mount the volume as a SL volume.

FILE, LIBRARY, VOLUME — The location of the input VS image file.

TYPE — The output diskette type. A response of T indicates a 2200-format diskette; a response of N indicates a VS-supported NL diskette(s).

MULTIPLE — Indicates whether or not multiple diskettes are to be created. Enter YES or NO.

After you specify the input image file, the Output Definition screen requests the following information and allows you to mount the diskette. You can also return to the Input Definition screen by pressing PF1. If you mount the diskette through the Command Processor instead of COPY2200, you must mount the diskette as a NL volume. If you enter YES, the Output Definition screen is repeated until you press PF16 or until you have copied all the records from the diskette image file.

DISKETTE — The name of the output diskette volume.

SECTORS — The number of sectors to be created on the specified diskette if MULTIPLE = YES. You can specify a specific number or ALL; SECTORS defaults to ALL.

DEVICE# — The device number of the drive on which you mount the output 2200 diskette. By specifying a value for DEVICE# and pressing PF4, you can mount the output diskette through COPY2200. COPY2200 automatically mounts the 2200 diskette as a NL volume. You can omit DEVICE# if you have already mounted the output diskette.

The utility prompts you to enter the number of sectors to be copied and the 2200 diskette name. It also informs you that the copy function will destroy all information currently stored on the diskette. You can continue the copy function by entering YES or terminate the operation by entering anything else. After the diskette(s) have been created, the main menu is displayed. You can then dismount the most recently mounted diskette by pressing PF14 from the main menu.

4.8 A SAMPLE COPY2200 PROCEDURE

You can control COPY2200 processing through the VS Procedure language. You can specify all COPY2200 operations through a procedure, including diskette mounting and dismounting. You can embed the mount and dismount operations in the RUN, ENTER (or DISPLAY) sequence, providing that you use the COPY2200 mount and dismount operations instead of the Procedure language MOUNT and DISMOUNT statements. A list of COPY2200 GETPARMs is provided in Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

The following procedure copies all 2200 data files on VL2200 to the variable-length files in the 2200LIB library on the VS SYSTEM volume. Numeric data is converted to VS 4-byte packed decimal data. Invalid characters in 2200 file names are converted to valid VS file names by using a STARTER value of M and a FILLER value of L. The procedure logically mounts and dismounts the diskette before exiting the utility. You (or the system operator if the procedure is run in background mode) need only physically mount and dismount the 2200 diskette.

```
PROCEDURE
RUN COPY2200
ENTER ACTION 1
ENTER INPUT 4, DISKETTE=VL2200, STARTER=M, FILLER=L, DEVICE#=023
ENTER COPYMODE MODE=D
ENTER DATAOUT LIBRARY=2200LIB, VOLUME=SYSTEM, TYPE=V, RECSIZE=60
ENTER NUMERICS TYPE=P, LENGTH=4, PLACES=0
ENTER ACTION 14
ENTER ACTION 16
RETURN
```

CHAPTER 5
THE COPYWP UTILITY

5.1 INTRODUCTION

The COPYWP utility provides a variety of functions for VS word processing document filing and conversion. Document filing functions permit you to manipulate VS word processing documents and libraries in the following ways:

- Copy Document or Library -- Copies an individual document from one document ID to another, or copies all the documents in a library from one library to another.
- Delete Document or Library -- Deletes an individual document or all documents in a library.
- Rename Document or Library -- Renames an individual document or all documents in a library, and, optionally, moves the document(s) to a different library.
- Reorganize Document or Library -- Releases unused space in a document or in all documents in a library.
- Merge Documents -- Merges two input documents into a single output document.

NOTE

COPYWP does not generate or read archive (NL) diskettes. Document archiving is supported only by VS Word Processing.

The document conversion functions of COPYWP convert a VS word processing document to a VS data file or a VS data file to a VS word processing document. COPYWP supports the following conversions:

- Convert Document to File -- Converts a VS word processing document to a VS data, source, print, or 2780 Telecommunications (TC) file.
- Convert File to Document -- Converts a VS source, print, image, 2780 TC, or consecutive data file to a VS word processing document.

COPYWP also supports VS Procedure language control of COPYWP workstation interaction. (Refer to Section 5.6 for details.)

An overview of COPYWP processing is provided in Figure 5-1.

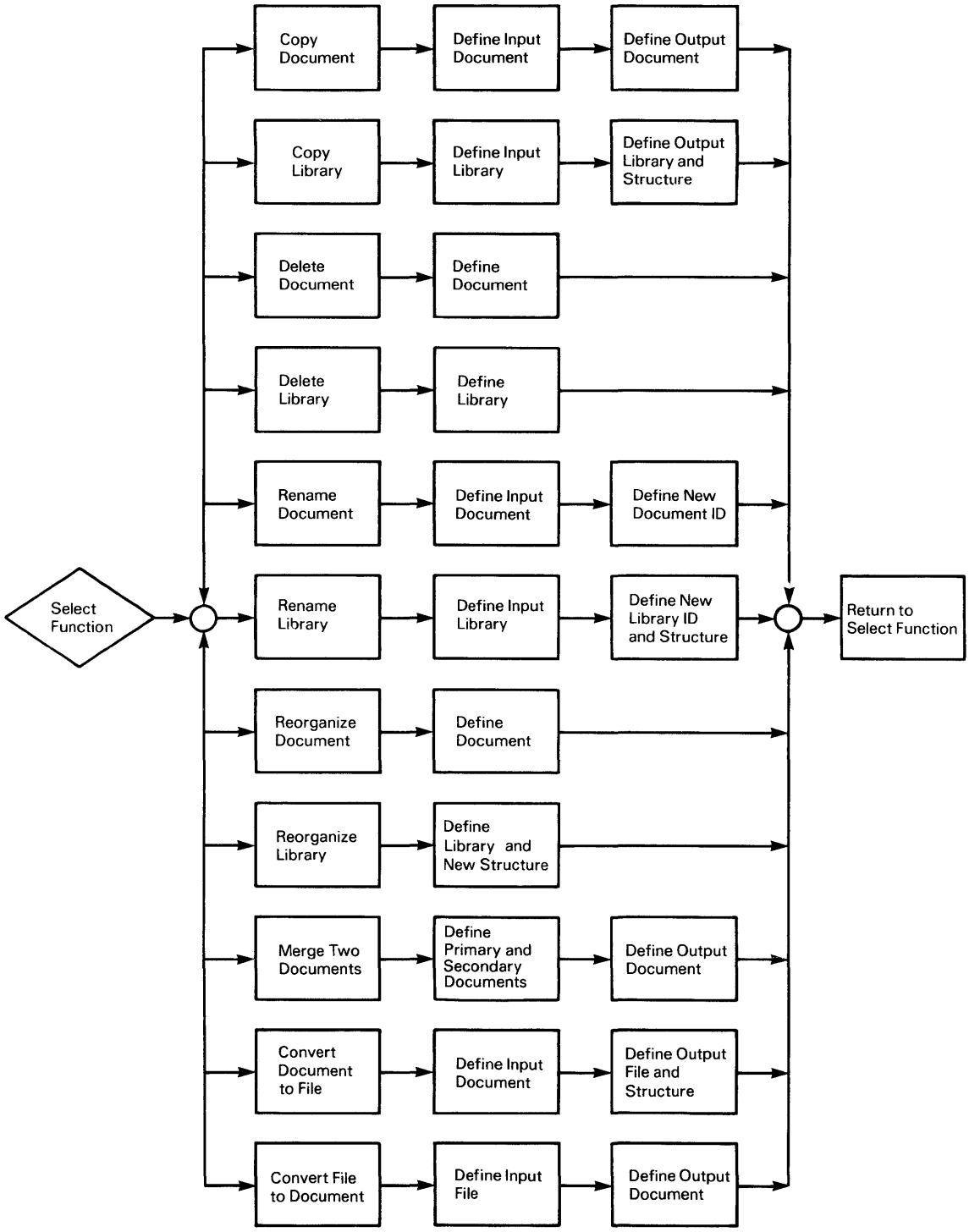


Figure 5-1. COPYWP Processing

5.2 REFERENCING VS WORD PROCESSING DOCUMENTS

The file parameters of a VS word processing document are represented differently from those of a data processing file. A data processing file is referred to by file, library, and volume names; a VS word processing document is referred to by a document ID and a volume name. A document ID is composed of a 4-digit number followed by an uppercase or lowercase letter. The volume name is the name of the VS volume on which the document resides. VS Word Processing translates the document ID to a VS file and library name; the document ID and the volume name, however, are the only information COPYWP and VS Word Processing need to identify a VS word processing document. For a description of the document ID and other components of the word processing document summary, refer to the VS/IIS Word Processing Operator's Guide.

5.3 RUNNING COPYWP

You can invoke the COPYWP utility in both data processing and word processing environments. In the data processing environment, you run COPYWP like any other utility, through the Command Processor RUN option or a Procedure language RUN statement. In the word processing environment, you run COPYWP by first selecting the Utilities option from the Word Processing main menu, and then selecting the Document Filing and Conversion option from the VS Word Processing Utilities menu.

COPYWP incorporates International options that are transparent to traditional processing. These options only have meaning for conversion operations and are described in Section 5.3.1. If you do not need to respecify any International options, processing begins with selecting a function, described in Section 5.3.2.

5.3.1 International Options

When you run COPYWP through a procedure, you can optionally specify International options for conversion operations through a "hidden" GETPARM, identified by the INTERNAT prname. Hidden GETPARMs are not displayed automatically when a program runs, but can be specified through a procedure. Because COPYWP supplies default values for the International options, you need only specify the INTERNAT GETPARM if you wish to change the default responses.

The INTERNAT GETPARM allows you to specify the following options for conversion operations:

DATE — Indicates whether dates in the document summary are converted in American date format (mm/dd/yy) or in European format (dd/mm/yy). A value of A indicates American format; a value of E indicates European format. DATE defaults to the date format defined for your VS system through GENEDIT.

DECALIGN — Indicates whether the American or European decimal alignment character should be used. The decimal alignment character can be either "." or ","; DECALIGN defaults to "." if your system is generated with American dates and to "," if your system is generated with European dates.

CURRENCY Indicates the currency symbol that is used. CURRENCY defaults to $.

REQSPACE Indicates what ASCII character should be used to generate a required space character. You can specify any ASCII character; REQSPACE defaults to 5C (\).

DEVCHARS Indicates whether device-dependent characters in the file or document should be converted. A response of YES converts the characters; a response of NO indicates that the characters should not be converted. DEVCHARS defaults to YES.

5.3.2 Selecting a COPYWP Function

Unless you elect to specify International options, the first screen COPYWP displays is the main menu, shown in Figure 5-2. From this menu, you can select the desired function by pressing the corresponding PF key. The document filing functions are explained in Section 5.4; Section 5.5 discusses converting files to documents and documents to files.

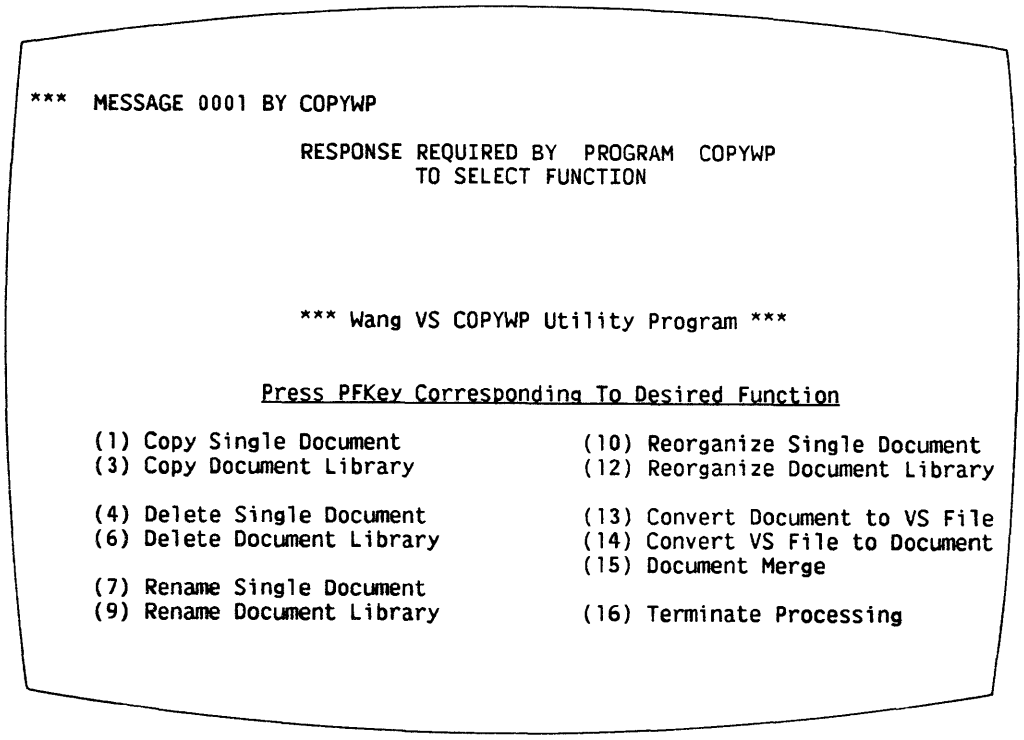


Figure 5-2. The Main Menu

To perform any COPYWP function, you must have appropriate access to the document or library file class. The document, or all documents in the library (for library functions), must reside on mounted, VS standard label (SL) volumes. Table 5-1 summarizes the level of access needed for each function. A COPYWP function that requires Read Only access can operate on any document that is not in use or is only being read by another program. Those functions that require Write access can only operate on documents that are not currently in use.

Table 5-1. Access Rights

Access Rights	Functions					
	Copy	Delete	Rename	Reorganize	Merge	Convert
Read	x				x	x
Write		x	x	x		

Although COPYWP performs many individual functions, the utility is designed to request document and/or library information in the same way for each function. COPYWP can accept as input or produce as output both VS word processing documents and libraries. Subsections 5.3.3 through 5.3.6 explain the method of specifying a particular input or output range for all functions, and are referenced by later discussions of each function in Sections 5.4 and 5.5. The errors that can occur when COPYWP accesses VS word processing documents and libraries are summarized in Subsection 5.3.7.

5.3.3 Single Document Input

When you select a single document function, COPYWP requests you to enter the document ID and volume name of the input document on the Input Document Specification screen shown in Figure 5-3. You do not have to supply the volume name if the document's library resides on the volume associated with the library letter through VS Word Processing. If VS Word Processing is installed on your system, your workstation's default library letter is retrieved as the default library for the first input request in a COPYWP session. After the first input request, or if VS Word Processing is not installed, the input library defaults to the most recently specified input library (if any) in the current COPYWP session. You can override the default library by supplying a different library letter.

If the input document is password-protected, COPYWP requests you to enter the password on a subsequent screen. You can return to the main menu from the Input Document Specification screen or the Password Specification screen by pressing PF1. Otherwise, the selected function begins if you have entered all information. If you do not specify the correct document ID or volume name, a screen is displayed indicating the error and allowing you to respecify the information.

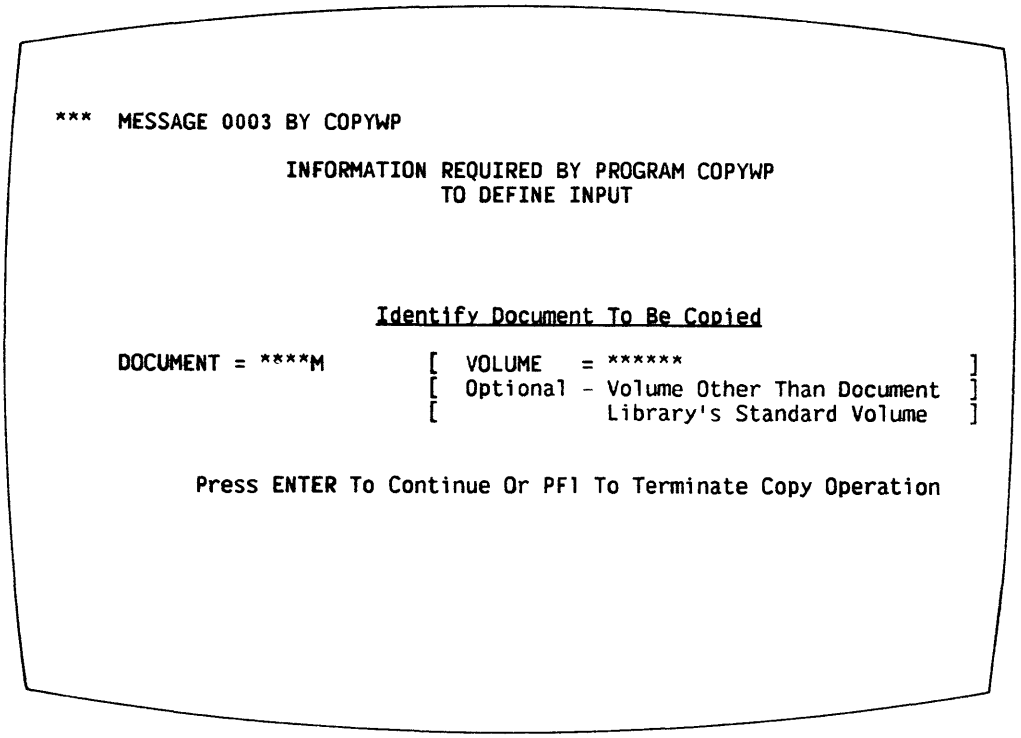


Figure 5-3. An Input Document Specification Screen

5.3.4 Document Library Input

When you select a library function, COPYWP requests you to enter the library letter and volume name of the input library on the Input Library Specification screen shown in Figure 5-4. You do not have to supply the volume name if the library resides on the volume associated with the library letter through VS Word Processing. If VS Word Processing is installed on your system, your workstation's default library letter is retrieved as the default library for the first input request in a COPYWP session. After the first input request, or if VS Word Processing is not installed, the input library defaults to the most recently specified input library (if any) in the current COPYWP session. You can override the default library by supplying a different library letter.

For all functions, except library reorganization, COPYWP requests you to enter passwords if any documents in the library are password-protected. Passwords are not requested for library reorganizations to facilitate performing this function in background mode. For all other functions, COPYWP displays a Password Specification screen when it encounters the first password-protected document and whenever it encounters a document with a different password. For example, if all documents in the library have the same password, COPYWP requests the password only once. If you do not know the password, you can cancel the selected function for the entire library and return control to the main menu by pressing PF1 from the Password Specification screen. Alternatively, you can skip the individual document if you press PF2. If you press PF1 from the Input Library Specification screen, the utility cancels the particular function and returns control to the main menu. If you specify an incorrect library or volume name, a screen is displayed indicating the error and allowing you to respecify the information.

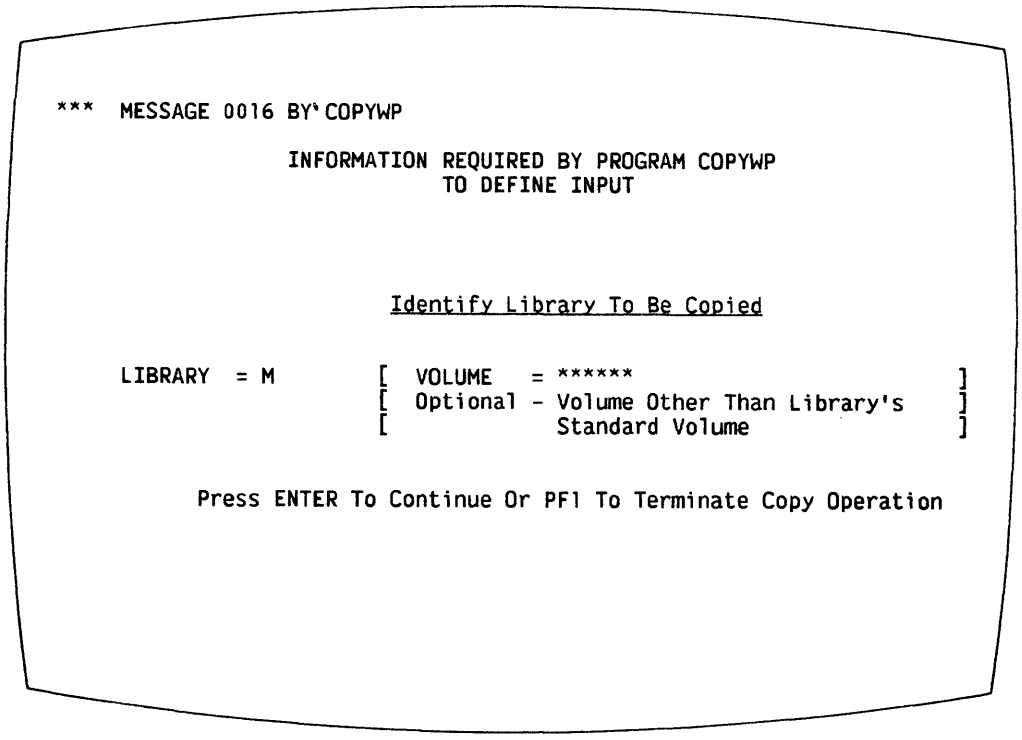


Figure 5-4. An Input Library Specification Screen

5.3.5 Single Document Output

For file-to-document conversion or single document functions, COPYWP requests you to enter the document ID and the volume name of the output document on the Output Document Specification screen shown in Figure 5-5. You do not have to supply the volume name if the document's library resides on the volume associated with the library letter through VS Word Processing. COPYWP retrieves the most recently supplied output document library as a default value for an output document ID. You do not need to assign a specific 4-digit value in the document ID for an output document, because you can specify next instead of a numeric value. (You can also specify next in uppercase and lowercase.) If you specify next, COPYWP obtains the next available document number in the library in a manner analogous to that used by the Create Document function of VS Word Processing. Note that unless you concatenate a library letter to next, an output document is placed in the next available document in the glossary library.

The specified output document must not already exist. If you specify an existing document ID, you must respecify the document ID or delete the existing document by pressing PF3 (you must have Write access and know the password of the existing document). If you do not specify the correct document ID or volume name, a screen is displayed indicating the error and allowing you to respecify the information.

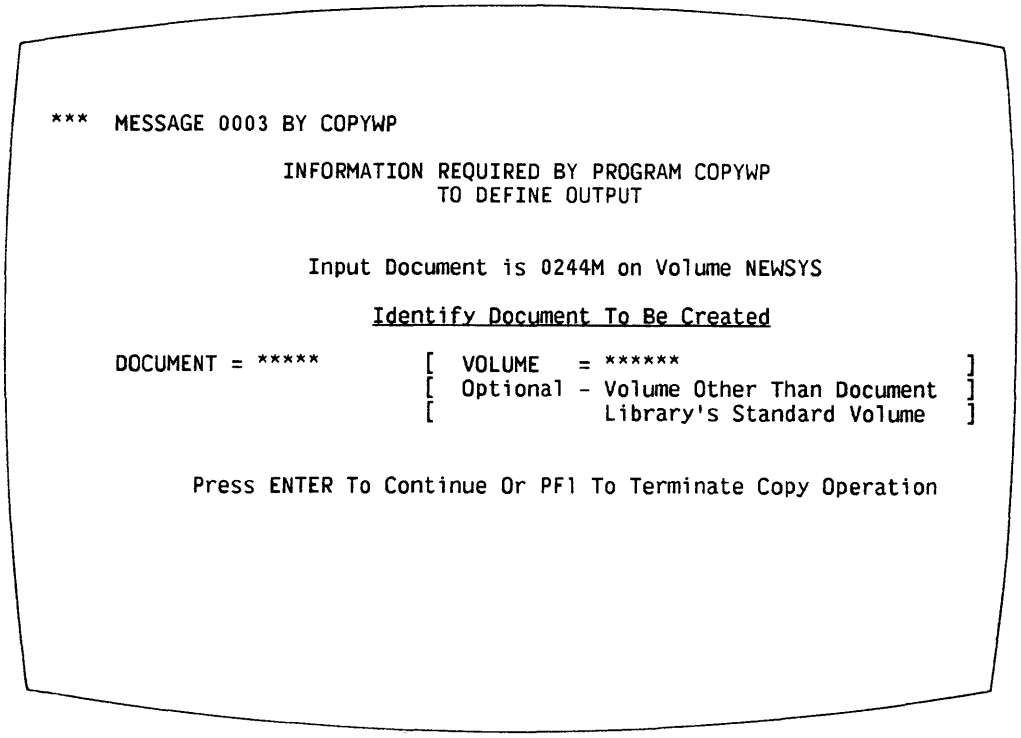


Figure 5-5. An Output Document Specification Screen

5.3.6 Document Library Output

For document library functions, the Output Library Specification screen, shown in Figure 5-6, requests you to enter the library letter and the volume name of the output library. You do not have to supply the volume name if the library resides on the volume associated with the library letter through VS Word Processing. COPYWP retrieves the most recently supplied output document library as a default value for the library. Two other options, RENUMBER and DUPFILES, are provided to allow you more control over the documents in the output library.

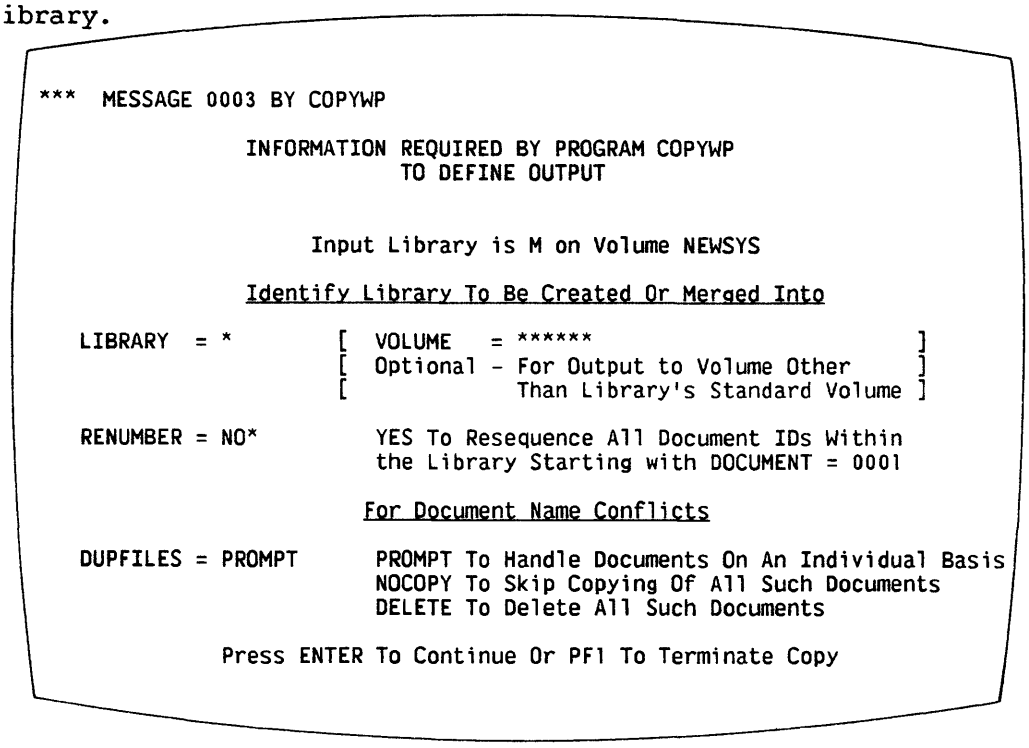


Figure 5-6. An Output Library Specification Screen

The Output Library Specification screen requests values for the following parameters:

RENUMBER — Renumbers the documents in the output library starting with a number of 0001 or greater. Renumbering documents in the library through COPYWP does not affect the number that VS Word Processing uses to create the next document. You can only change the starting number for document creation through the VS Word Processing Create Library Supervisory function. The default response, NO, uses the input document number for the output document. Specifying YES renumbers the output document library, beginning with the number that you specified. You can override the default starting number of 0001.

If the output library does not exist before the renumbering operation and if RENUMBER=YES, COPYWP places any input prototype document in document 0000 in the output library. The other input documents are placed in output documents numbered from the starting number that you specified.

If the output library already exists and RENUMBER=YES, documents already residing in the output library are renumbered from the starting number before the input documents are transferred. An existing prototype document, or any document to which you do not have Write access, is not renumbered. The renumbering operation always transfers an input prototype document to document 0000. If the output library already contains a prototype document, the conflict is resolved by the DUPFILES option.

DUPFILES Determines the method of resolving conflicts that occur when the input document ID number duplicates the number of an existing output document. If you select resequencing, a conflict can only occur with prototype documents. The three methods of resolving conflicts are: NOCOPY, DELETE, and PROMPT. The default response is PROMPT. Each method is described as follows:

NOCOPY Automatically skips the input document (i.e., the function is not carried out on the input document) without displaying any message at the workstation.

DELETE Automatically scratches the existing document in the output library and processes the input document, without displaying any message at the workstation. You need Write access to the existing document, and the expiration date must have passed.

PROMPT Prompts you each time a document ID conflict occurs. In each case, a special screen provides you with the following options for manually resolving the conflict:

N Do not process this document.

D Delete the existing document in the output library and process the input document. You need Write access to the existing document, and the expiration date must have passed.

R Rename the existing document in the output library as specified in NEWDOCID and process the input document. You need Write access to the existing document, and the expiration date must have passed.

C Process the input document using a new document ID as specified in NEWDOCID. If you specify next, a subsequent conflict can occur.

5.3.7 COPYWP Error Conditions

The following two error conditions can be detected by COPYWP during a single document operation:

- Document Damaged -- If COPYWP determines that the input document is damaged, it terminates the copy and displays an error message. A damaged document is usually recovered when the document is edited within VS Word Processing.

- I/O Error -- An I/O error message is displayed if COPYWP detects an error with the disk drive or storage medium.

Two other types of errors can also be detected during a document library operation. The following errors terminate the operation on the current document and cause an error message to appear.

- Current Document Error -- If only the current document is damaged, COPYWP allows two responses. You can press PF1 to terminate the entire library operation, or you can press PF2 to skip the current document and continue the operation with the next document.

- Fatal Error -- If more serious errors occur, COPYWP only allows you to terminate the entire library operation by pressing PF1.

If COPYWP runs as a noninteractive procedure, a current document error automatically skips over the damaged document and continues the operation with the next document. A fatal error causes cancellation of the COPYWP program.

5.4 DOCUMENT FILING FUNCTIONS

The document filing functions perform maintenance operations on VS word processing documents and libraries. You can copy, delete, rename, and reorganize documents and libraries. You can also merge two documents.

NOTE

The document filing functions do not include an archiving facility. The archiving facility is supported as part of the Document Filing function of VS Word Processing. If you use a diskette as an input or an output volume for COPYWP, it must be formatted as a VS SL-diskette and mounted before you select an operation. Thus, COPYWP can only use standard VS volumes or diskettes and does not support archive diskettes.

5.4.1 Copying a Single Document

The Copy Single Document function, selected by pressing PF1 from the main menu, enables you to make a copy of a specified document. All unused space in the input document (caused by frequent edits) is eliminated from the output document. Any glossary verification information is transferred to the output document.

To perform the copy, you need only specify the input and output documents as described in Subsections 5.3.3 and 5.3.5.

5.4.2 Copying a Document Library

The Copy Document Library function, selected by pressing PF3 from the main menu, copies all documents in a specified input library to an output library, and, optionally, resequences the documents in the output library. The output library can be a new or existing library. All unused space in the input documents is removed from the output documents. The copy operation preserves any glossary verification information contained in library documents.

After selecting the Copy Document Library function, you specify the input library, as described in Subsection 5.3.4, and the output library and organization, as explained in Subsection 5.3.6.

5.4.3 Deleting a Single Document

The Delete Single Document function, selected by pressing PF4 from the main menu, deletes a single document from a volume, freeing the disk space it occupied. The document expiration date must have passed. You must also have Write access to the document and must know any assigned password.

COPYWP provides a Security Erase feature that destroys the data on the disk as well as the document's VTOC entry. Intended for the deletion of confidential documents, the Security Erase Feature overwrites with binary ones all disk blocks spanned by the document before deleting the VTOC entry. Files and documents are usually deleted by eliminating the associated entries in the VTOC. However, unless you set ERASE to YES, the data remains on the disk until it is overwritten by another file or document.

When you select the Delete Single Document function, COPYWP requests the input document parameters, as described in Subsection 5.3.3. The Delete Single Document Input screen also allows you to select the Security Erase feature. Because this feature involves more time and system resources than normal document deletion, the default value of the ERASE parameter is NO. You can select the Security Erase feature by changing this value to YES. COPYWP deletes the document without prompting you for confirmation.

5.4.4 Deleting a Document Library

The Delete Document Library function, selected by pressing PF6 from the main menu, deletes an entire document library from a disk volume, freeing the disk space it occupies. The expiration dates of all documents in the library must have passed, you need Write access to each document, and you must know any assigned passwords.

You can also select the Security Erase feature described in Subsection 5.4.3. When you select Security Erase, all blocks spanned by all documents in the library are overwritten with binary ones before the VTOC is updated.

When you select the Delete Document Library function, COPYWP requests you to enter the input library parameters as described in Section 5.3.4. The Delete Document Library Input screen also allows you to select the Security Erase feature. Because this feature consumes more time and resources than normal library deletion, the default value of the ERASE parameter is NO. If you change the default value to YES, all documents in the library are deleted with the Security Erase feature. COPYWP deletes the library without prompting you for confirmation.

5.4.5 Renaming a Single Document

The Rename Single Document function, selected by pressing PF7 from the main menu, changes the ID of a single document. You must have Write access to the document and must know any assigned password. You can use the Rename function to change both the document ID and the library ID. If you change the library ID, the document is assigned to the new library, which need not be on the same volume. If the library of the new document ID is on a different volume from the library of the original document ID, the document is moved to the new volume. When the document is moved, it is copied and the original document is deleted.

When you select the Rename Single Document function, COPYWP requests you to enter the input document parameters, as described in Subsection 5.3.3, and then requests the output document parameters, as described in Subsection 5.3.5.

5.4.6 Renaming a Document Library

The Rename Document Library function, selected by pressing PF8 from the main menu, changes the library IDs of all the documents in a library, implicitly assigning them to the new library. The new library may or may not already exist, and can be on a different volume. Optionally, you can resequence the documents in the output library. To rename a library, you need Write access to all the documents and must know any assigned passwords. If the new library is on a different volume from the original library, all documents in the original library are moved to the new volume. When the documents are moved, they are copied and the original documents are deleted.

To rename a document library, you first define the input library, as described in Section 5.3.4. COPYWP then requests the output library parameters and structure options, as discussed in Section 5.3.6.

5.4.7 Reorganizing a Single Document

The Reorganize Single Document function, selected by pressing PF10 from the main menu, eliminates all unused disk space within a document and consolidates the document's file to a single disk extent. You need Write access to the document and must know any assigned password.

To reorganize a document, you need only specify the input document, as described in Subsection 5.3.3.

5.4.8 Reorganizing a Document Library

The Reorganize Document Library function, selected by pressing PF12 from the main menu, eliminates unused disk space within all documents in a library and consolidates each document's file to a single disk extent. Optionally, this function can also resequence the documents within a library. To reorganize a library, you only need Write access to all the documents in the library.

To reorganize a document library, you need only supply the information requested on the Reorganize Document Library Input screen, shown in Figure 5-7. Library specification is described in Subsection 5.3.4, and the Renumber option is identical to that described in Subsection 5.3.6.

```
***  MESSAGE 0016 BY COPYWP

                INFORMATION REQUIRED BY PROGRAM COPYWP
                          TO DEFINE INPUT


                     Identify Library To Be Reorganized

     LIBRARY  = M        [  VOLUME   = ******                        ]
                         [  Optional - Volume Other Than Library's   ]
                         [             Standard Volume               ]

     RENUMBER = NO*         YES To Resequence All Document IDs Within
                            the Library Starting with DOCUMENT = 0001

           Press ENTER To Continue Or PF1 To Terminate Operation
```

Figure 5-7. The Reorganize Document Library Screen

5.4.9 Merging Two Documents

The Document Merge function, selected by pressing PF15 from the main menu, merges a secondary input document with a primary input document. You can then subsequently edit and print the resulting output document. The merge operation takes place according to the rules of the Word Processing Merge Print function. The merge process copies the Header, Footer, and Work pages from the primary document into the output document. If either or both primary or secondary documents contain a glossary, the glossary(s) is not copied into the output document.

When you select the Document Merge function, COPYWP first requests you to enter the document parameters of the primary and secondary documents, as described in Subsection 5.3.3. After you specify the input documents, the output document parameters are requested, as discussed in Subsection 5.3.5. If the resulting document exceeds 120 pages or becomes too large to archive, you are prompted for an additional document ID. However, if you specified next as the original output document ID, the merge process continues uninterrupted with the next available document ID in the specified library.

5.5 DOCUMENT CONVERSION FUNCTIONS

The document conversion functions convert VS word processing documents to text data records within certain forms of conventional VS data files. They also convert text within certain forms of conventional VS data files into VS word processing documents.

> NOTE
>
> With the exception of conversions to and from 2780 TC data file format, the conversion functions are not symmetric. Use of the conversion functions in a circular fashion does not yield documents or files equivalent to the original files or documents.

The document conversion functions convert an entire file or document. COPYWP does not provide field or record selection capabilities. If you need to update documents or perform extensive text string searches from within programs, you should use the VS Document Access Subroutines discussed in the VS Programmer's Guide to VS/IIS. The subroutines are more efficient and may require less time and resources than the conversions described here. They also preserve all document formatting characters.

The conversion process is performed subject to the International options selected (or defaulted) for the current COPYWP session through the INTERNAT GETPARM. All conversion operations performed during the session must use the same International options. Refer to Section 5.3.1 for details.

The COPYWP conversion process is only approximate. Documents have formatting codes and conventions that do not have analogies in the conventional VS data file, and vice versa. These incompatibilities are discussed in detail in Subsections 5.5.1 and 5.5.2.

5.5.1 Document-to-File Conversion

The Document-to-File Conversion function, selected by pressing PF13 from the main menu, transforms a VS word processing document into a specified type of VS data processing file. You need Read access to the input document and, if the document is password-protected, you must know the password.

When you select the Document-to-File Conversion function, COPYWP requests you to enter the document parameters of the input document, as described in Subsection 5.3.3. After you identify the input document, COPYWP prompts you for the file, library, and volume names of the output file, as well as for the file structure in which the output file is to be constructed. The library and volume names default to your OUTLIB and OUTVOL values, respectively. You can convert the input VS word processing document to a consecutive data file, a print file, a program source file, or a TC file. When you specify one of the values listed below in the TYPE parameter, the indicated conversion occurs. If subsequent document-to-file conversions are processed, TYPE defaults to the previously specified value. If you supply the output file parameters and conversion type, the document is converted. Alternatively, you can return to the main menu by pressing PF1.

DATA Conversion of a VS word processing document to the DATA format yields a file that is suitable for access by a user program. The resulting file is consecutive and compressed, with a 628-byte maximum record length. The DATA format is unique on the VS system and is organized for easy access by a user program. The DATA format contains two record types, which are detailed in Appendix B. The first record type describes the format of the text; the second contains the actual text of the document, as well as information describing the current record. Refer to Table 5-2 for a listing of those document elements that cannot be transferred to the data processing file.

NOTE

You cannot convert files in DATA format back to VS word processing documents.

PRINT Conversion of a VS word processing document to the PRINT format yields a VS print file that provides a line-formatted representation of the original document. Depending upon your PRNTMODE, the resulting print file can be automatically queued to print. You can specify certain formatting options, comparable to those provided on the Word Processing Document Print menu, on the Print File Options screen shown in Figure 5-8. Default information is obtained from the document, but the document is not updated with any new information. Refer to Table 5-2 for a listing of those document elements that cannot be transferred to the data processing file.

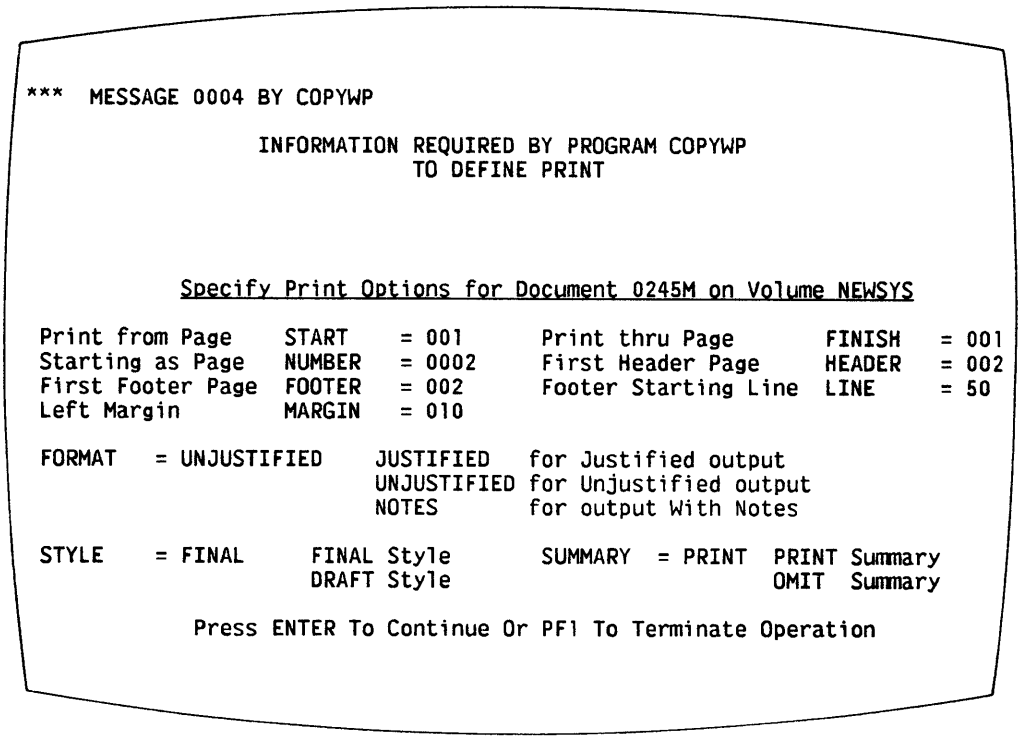


Figure 5-8. The Print File Options Screen

SOURCE Conversion of a VS word processing document to the SOURCE format yields a file that can be accessed by the EDITOR and VS language compilers. Documents to be converted must not have any format lines exceeding 81 characters. If any line in the document is 81 characters long, character 81 must be a RETURN. You must provide the proper line sequence numbers within the document or leave space for such sequence numbering by the EDITOR. Refer to Table 5-2 for a listing of those document elements that cannot be transferred to the data processing file.

TC Conversion of a VS word processing document to the TC format yields a file that can be used as input to the VS TCCOPY utility for 2780-type transmission to another Wang Word Processing System, Office Information System (OIS), or VS System. This conversion preserves all of the original document's formatting information.

VS word processing documents have some formatting codes and conventions that do not have analogies in conventional VS data files. Table 5-2 represents the restrictions in conversions from VS word processing documents to VS data files.

Table 5-2. Document-to-File Conversion Restrictions

Document Feature	Print Conversion	Source Conversion	Data Conversion
Document summary	Optional	Ignored	Ignored
Headers/footers	Implemented	Ignored	Ignored
Single underscore	Implemented	Ignored	Ignored
Double underscore	Single underscore	Ignored	Ignored
1/4 Spacing	Single spacing	Single spacing	Described by format record
1/2 Spacing	Single spacing	Single spacing	Described by format record
1 1/2 Spacing	Double spacing	Double spacing	Described by format record
Superscripts	Ignored	Ignored	Ignored
Subscripts	Ignored	Ignored	Ignored
Emphasis printing	Implemented	Ignored	Ignored
Notes	Optional	Implemented	Implemented
Stop codes	Ignored	Ignored	Ignored
Merge codes	Ignored	Ignored	Ignored
Don't Merge codes	Ignored	Ignored	Ignored

5.5.2 File-to-Document Conversion

The File-to-Document Conversion function, selected by pressing PF14 from the main menu, transforms VS Print, Source, Image (text), and TC data processing files into VS word processing documents. No other type of file, including the output of a DATA type document-to-file conversion, can be converted to a document. Thus, COPYWP cannot convert indexed, program, or WP files to VS word processing documents.

If you select the File-to-Document Conversion function, COPYWP requests you to enter the file, library, and volume names of an input data processing file. If you do not correctly specify the file, library, or volume names, COPYWP prompts you to respecify the input file parameters. You must also respecify the parameters if you do not have Read access to the file or if the utility encounters a possession conflict.

After you identify the input data processing file, you must provide the output document parameters on the Output Document Specification screen. COPYWP requests you to enter the document ID and volume name of the output document and allows you to specify document summary information (title, operator, author, and comments). COPYWP determines the input file's format based on the file label's attribute, so you need not specify the input file type for print and TC file conversions. Consecutive data files with record lengths greater or less than 80 bytes are converted to documents as Image files. However, because 80-byte consecutive files can be converted as Source or Image files, you must select the conversion type for 80-byte record files. Depending on the type of input file, COPYWP can also provide two additional conversion options: Lines per Page and Automatic Insertion of Tabs. These options are described in the context of the particular file conversion process.

Print, Source, Image, and TC input files and the associated conversion processes are described as follows:

PRINT

A PRINT file is a consecutive, compressed file with the print attribute. To be acceptable as input to COPYWP, the maximum record length (line length) of a print line cannot exceed 160 characters, including the print control characters.

Each page of the print file (based upon skips to Channel 1 carriage control characters) translates into a page of the document. If the print file exceeds 120 pages or requires too much space to be archived, the utility prompts you for an additional document ID. However, if you specified next as the original output document ID, the conversion process continues uninterrupted with the next available document ID in the specified library.

All occurrences of carriage control skips to channels other than Channel 1 (top of form) are translated into an extra RETURN (◀) and an Operator Note (!!) line indicating the location of the original channel skip.

Unless the International options indicate otherwise, the following hexadecimal codes in the input print file are converted as indicated:

Input Code	Output Code
00-0D	5C (\)
80-8D	5C (\)
5E	5C (\)
0E(↑)	5E
0F(↓)	5F

If you set the International option REQSPACE to a character other than 5C, the input codes that convert to 5C are converted to the current value of REQSPACE. If you set DEVCHARS to NO, hexadecimal 0E and 0F are not converted.

Because the print file automatically determines page breaks, COPYWP does not request a value for the Lines per Page option on the Output Document screen.

COPYWP allows you to select the Automatic Insertion of Tabs option for PRINT file conversion. If you specify TABS=YES (the default response), COPYWP determines optimal tab settings on a page-by-page basis and uses TAB (▶) and DEC TAB (␣) characters where appropriate. Depending upon the contents and format of the text within the PRINT file, use of the Automatic Insertion of Tabs feature can substantially reduce the document's disk storage requirements and ease subsequent manual editing of the document. If TABS=NO, no TAB or DEC TAB characters appear in the resulting document. If you intend to typeset the document, set TABS = NO.

SOURCE

A SOURCE file is a consecutive file with 80-byte fixed-length records, or with compressed records with a maximum length of 80 bytes. The input file must be acceptable as input to the VS EDITOR or the product of the VS EDITOR, but need not have line sequence numbers.

Files eligible for Source file conversion can also be converted to documents as Image files. Thus, the Output Document Specification screen for Source file conversion allows you to switch to Image conversion. Consecutive input files with an 80-byte record length default to Source file conversion. If Source file conversion is appropriate, the other parameters on the Output Document Specification screen must be defined as described below. COPYWP invokes the Image file conversion process if you set the TYPE parameter value to IMAGE.

Each record of the input file translates into a line of the document. If the resulting document exceeds 120 pages or requires too much space to be archived, the utility prompts you for an additional document ID. However, if you specified next as the original output document ID, the conversion process continues uninterrupted with the next available document ID in the specified library.

Unless the International options indicate otherwise, the following hexadecimal codes in the input SOURCE file are converted as indicated:

Input Code	Output Code
00-0D	5C (\)
80-8D	5C (\)
5E	5C (\)
0E(↑)	5E
0F(↓)	5F

If you set the International option REQSPACE to a character other than 5C, the input codes that convert to 5C are converted to the current value of REQSPACE. If you set DEVCHARS to NO, hexadecimal 0E and 0F are not converted.

Because no pagination is contained in the input file, COPYWP enables you to set the number of lines per page through the Lines per Page option on the Output Document Specification screen. The utility retrieves a default value for Lines per Page from your Default Lines per Page parameter, which can be set by PF2 of the Command Processor, a procedure, or a program.

COPYWP allows you to select the Automatic Insertion of Tabs option for Source file conversion. If you specify TABS=YES (the default response), COPYWP determines optimal tab settings on a page-by-page basis, and uses TAB and DEC TAB characters where appropriate. Depending on the contents and format of the text within the Source file, use of the Automatic Insertion of Tabs feature can substantially reduce the document's disk storage requirements and ease subsequent manual editing of the document. If TABS=NO, no TAB or DEC TAB characters appear in the resulting document. If you intend to typeset the document, specify a value of NO for TABS.

IMAGE

An IMAGE file is any consecutive data file that does not qualify for TC or PRINT conversion. Image file conversion is automatically invoked for eligible files with record lengths other than 80 bytes; for 80-byte record files, a different Output Document Specification screen allows you to specify the conversion TYPE, as well as the document summary parameters. You should ignore the Automatic Insertion of Tabs and Lines per Page parameters that appear on the Output Document Specification screen for 80-byte record files, because these options do not operate for the Image file conversion process.

While Source file conversion provides automatic conversion, Image file conversion allows you to exercise more direct control over the output document's format. Pagination in the document is indicated by a slash (/) in the first position of a record in the data file; COPYWP provides no automatic pagination. Automatic document overflow handling is not available for Image file conversions; you must convert the input file to a single document.

With the exception of FORMAT (|) and PAGE (|) characters, which are represented by hexadecimal 06 and 86, respectively, all legal document control characters in the input file pass directly into the output document. This preservation of document control characters is especially useful in the case of MERGE and DON'T MERGE control characters. Table 5-3 relates each document control character to its hexadecimal code and function; each function is described in the VS/IIS Word Processing Operator's Guide.

Table 5-3. Document Control Character Hexadecimal Codes

Document Control Character	Hexadecimal Code	Function
◆	01	CENTER
▶	02	TAB
◀	03	RETURN
→	04	INDENT
⌐	05	DEC TAB
|	06	FORMAT
■	0B	STOP
!!	0C	NOTE
↕	0D	MERGE
↑	0E	SUPERSCRIPT
↓	0F	SUBSCRIPT
|̲	86	PAGE
↕̲	8D	DON'T MERGE

Characters that are illegal in a VS word processing document or inappropriate in the IMAGE file are converted as follows:

Input Code	Output Code
00	20 (' ')
06 (|)	2A ('*')
80	A0 ('_')
81	01
82	02
83	03
84	04
85	05
86 (|̲)	AA ('*̲')

After you specify the output document parameters (and select Image file conversion, if necessary), COPYWP displays the Image Conversion Options screen shown in Figure 5-9.

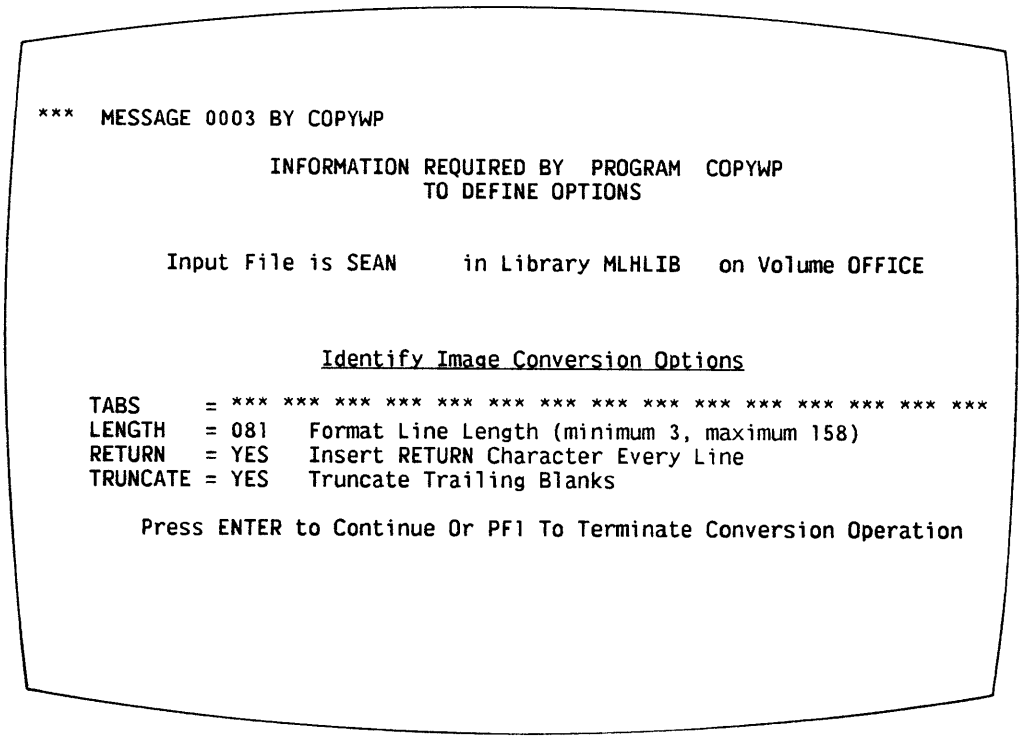


Figure 5-9. The Image Conversion Options Screen

The Image Conversion Options screen allows you to specify the following document format options:

TABS — Converts blanks in the input file to tabs, if the blanks coincide with the selected column settings. Tab selection can save a significant amount of storage space. You can specify up to 15 tab positions. If you do not specify any tab positions (the default action), no tab characters are placed in the format line, and tabs cannot replace spaces within the text. A tab can be located in any column between column 3 and the column location that equals the LENGTH parameter value minus one. You should not set tab positions if the output document will be typeset.

LENGTH — Determines the length of the format line in the resulting document. LENGTH defaults to one greater than the record length of the input file or to 158, whichever is smaller.

RETURN — Places a RETURN at the end of each record in the input file. Because an input file record generally corresponds to a document line, the default value of RETURN is YES. However, RETURN=NO is appropriate for custom-designed VS files that contain their own carriage returns and/or other control characters.

TRUNCATE — Truncates trailing spaces from the end of each line, saving space in the output document. The default response, YES, performs the truncation.

TC — A TC file is the product of the VS TCCOPY utility. It results from a 2780-type transmission from another Wang Word Processing System, Office Information System (OIS), or VS System. TC conversion preserves all of the original document's formatting information.

You are not prompted for the Automatic Insertion of Tabs option or for a value for Lines per Page because pagination, format, and tab information are contained in the TC file.

5.6 A SAMPLE COPYWP PROCEDURE

You can control COPYWP workstation interaction through the VS Procedure language. For example, document filing or conversion operations that you regularly perform can be automated by writing a procedure that specifies all functions, input, and output. Appendix A provides a complete list of all COPYWP GETPARMs. Consult the VS Procedure Language Reference for information about procedure syntax.

COPYWP also allows you to select International options for document conversion operations through a hidden GETPARM. The hidden GETPARM is identified by the INTERNAT prname, and allows you to specify the date format, decimal alignment character, currency symbol, required space character, and determine whether device-dependent characters are converted. Because COPYWP supplies default values for the date format and the decimal alignment character based on the date format specified for your system through GENEDIT, you need only specify the INTERNAT GETPARM if you wish to override the system defaults or change the default values for other International options.

The responses specified through the INTERNAT GETPARM remain in effect during an entire COPYWP session. Thus, all conversion operations performed during the session must use the same International options because you cannot respecify the GETPARM for individual operations. The keywords, meanings, and default values for the INTERNAT GETPARM are listed in Appendix A.

The following procedure performs periodic maintenance on two VS word processing libraries. The procedure reorganizes the M document library and renumbers the resulting library, beginning with document 1001. The procedure then copies document 2011a to another volume before deleting the "a" library and exiting from COPYWP. You must enclose lowercase library letters, passwords containing lowercase letters, or document IDs with lowercase library letters in matching single or double quotes to prevent the Procedure Interpreter from converting the value to uppercase.

```
PROCEDURE
RUN COPYWP
ENTER FUNCTION 12
ENTER INPUT LIBRARY=M, VOLUME=SYSTEM, RENUMBER=YES, DOCUMENT=1001
ENTER FUNCTION 1
ENTER INPUT DOCUMENT='2011a', VOLUME=SYSTEM
ENTER PASSWORD PASSWORD=MOLLY
ENTER OUTPUT DOCUMENT='2011a', VOLUME=ALTSYS
ENTER FUNCTION 6
ENTER INPUT LIBRARY='a', VOLUME=SYSTEM, ERASE=NO
ENTER PASSWORD PASSWORD=MOLLY
ENTER FUNCTION 16
RETURN
```

CHAPTER 6
THE DISKINIT UTILITY

6.1 INTRODUCTION

Before you can store programs or data on a disk volume, you must initialize the disk volume. The initialization process verifies that all sectors can be addressed and read. You use the DISKINIT utility to perform this process. DISKINIT records the addresses of any bad sectors on the volume to prevent the system from using them. DISKINIT also writes control information that enables the Operating System to record blocks of information on the volume.

DISKINIT also provides four related functions for initialized disk volumes. The functions provided by DISKINIT are as follows:

- The INITIALIZE function initializes a new disk volume. You also have the options to create a Volume Table of Contents (VTOC), to allocate a dump file, and to allocate a page block or page pool. Any data existing on the volume is destroyed.

- The REFORMAT function reformats a disk volume by creating a new volume label and VTOC for a previously initialized volume. You can optionally use this function to allocate a dump file and a page block or page pool. Any data existing on the volume is destroyed.

- The RELABEL function relabels a disk volume by changing the volume name. You have the option to allocate a page block or page pool. For a VS25 or VS45, RELABEL also calculates a pointer for the bootstrap file, if the file is present. Data on the volume is not affected.

- The VERIFY function verifies the accessibility of blocks on an initialized volume. Data on the volume is not affected.

- The REMOVE function removes a bad block from use. This function is only valid for the 2220, 2265V1, 2265V2, 2265V3, 2280V1, 2280V2, 2280V3, and Q2040 disks.

DISKINIT operates on all types of disks, including hard- and soft-sectored diskettes. An overview of DISKINIT processing is provided in Figure 6-1.

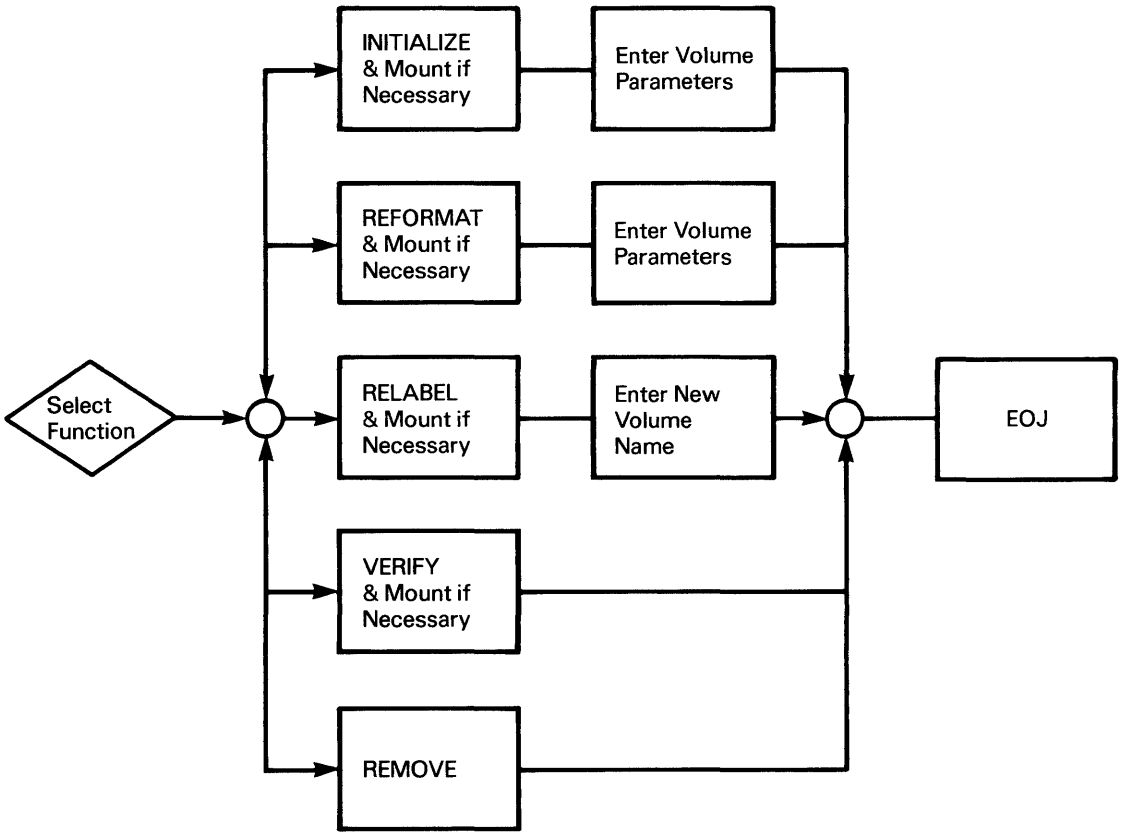


Figure 6-1. DISKINIT Processing

The Function Selection menu appears first on your screen when you run DISKINIT. Enter the function name in the FUNCTION parameter and the volume name in the VOLUME parameter. If you are initializing a new volume, enter the new volume name.

6.2 MOUNTING A VOLUME

After you enter the function and volume name, DISKINIT begins processing. If you have not yet mounted the named volume, DISKINIT displays a message at the top of the screen requesting you to perform a mount operation. The cursor is positioned beneath the volume name, indicating that no volume with this name is currently mounted. Press PF5 to invoke the Mount Disk Volume routine.

Mount Disk Volume requests you to enter the unit address of the disk unit on which the volume is to be mounted. A Wang Customer Engineer assigns peripheral device addresses at installation time; the address should be clearly marked on the unit. You can also get the device number from the Manage Devices function (PF6) of the Command Processor. If no volume is mounted at the specified address, you can mount the new volume.

If another volume is mounted at the specified address, you must dismount it before you mount the new volume. The program displays a message instructing you to dismount the volume, unless the volume is currently in use or mounted for exclusive access by another workstation. A volume is in use if one or more of its files is currently open. The Mount Disk Volume routine does not allow you to dismount a volume with open files. A file mounted for exclusive use can be dismounted only from the workstation at which it was mounted.

After the volume is successfully mounted, the disk unit automatically informs the system, which resumes processing of the selected DISKINIT function.

NOTE

You can mount new disks or diskettes for initialization only through DISKINIT. You cannot mount them through the Operator's Console, through the Command Processor, or through a procedure.

6.3 NONLABELED VOLUMES

Nonlabeled (NL) diskette volumes are used to transfer data between different manufacturer's computers and the Wang VS System. NL diskettes are used by VS Word Processing, as well as by the FLOPYDUP and COPY2200 utilities. NL diskettes can also be used to contain Control mode dumps. If the mounted volume is nonlabeled, the program alerts you and asks whether it can proceed with the selected function. If you do not wish to proceed, press PF1 to terminate processing.

A mounted volume is nonlabeled if

- It has not been previously initialized by DISKINIT
- It was specified as nonlabeled (NL) when previously initialized
- It was created on another manufacturer's computer system, a Wang Word Processing System, or a Wang 2200 System

6.4 THE INITIALIZE FUNCTION

The INITIALIZE function initializes a new volume with sector control information and verifies that you can read and write to the disk. Optionally, it creates a label and VTOC, allocates a dump file, and allocates a page block or page pool.

6.4.1 Formats

DISKINIT initializes both single- and double-sided, soft-sectored diskettes to Wang standard formats. Because all Wang standard soft-sectored diskette formats are double-density, you need not select a format. Thus, soft-sectored diskette initialization does not differ externally from other disk initializations. If you mount a double-sided, soft-sectored diskette, DISKINIT initializes it as a double-sided, double-density diskette. Single-sided, soft-sectored diskettes are initialized to single-sided, double-density.

6.4.2 Fault Tolerance

DISKINIT allows you to format a volume's VTOC with one of three levels of fault tolerance. Fault tolerance is a feature that protects a volume's VTOC from damage by duplicating the physical representation of the VTOC blocks. The first level is actually not tolerant: that is, no duplicates of the VTOC blocks are made. The second level makes two copies of each VTOC block, and the third level makes four copies of each VTOC block.

The second level is Crash tolerance and provides protection against VTOC damage caused by system failure in the middle of a VTOC operation. Each block within the VTOC is allocated two physical blocks on the volume. The blocks in these pairs are updated alternately. Alternate updating results in one of the two blocks serving as a backup copy. A bit map block at the head of the VTOC indicates which block is currently the update version. A copy of the bit map block is maintained in an adjacent block.

The third level is Media tolerance and provides protection against mechanical disk failures, such as power failures or defective disk media. Mechanical failures affect entire cylinders. Each block within the VTOC is allocated four physical blocks on the volume, giving you two complete crash tolerant VTOCs located on separate cylinders. The volume label, bad block list, and bit map block are also duplicated.

The first version of the VTOC begins at block 0, cylinder 0, occupying all of cylinder 0. If necessary, it also occupies cylinder 2, and as many more even-numbered cylinders as needed. The second version begins with a volume label in block 0, cylinder 1, and continues on as many odd-numbered cylinders as needed. Thus, for example, if the VTOC requires three cylinders, the first version of the VTOC occupies block 0, cylinders 0, 2, and 4; the second version of the VTOC occupies block 0, cylinders 1, 3, and 5. A media-tolerant VTOC occupies at least two complete cylinders.

Media tolerance virtually eliminates the possibility of losing files due to VTOC errors. It requires more than four times the disk space and four times the number of VTOC write operations as a VTOC without fault tolerance. In addition, a bit map read operation is required when the VTOC is first accessed; the bit map block, once read, is retained in memory. Media tolerance is recommended whenever maximizing disk storage space is not an overriding consideration.

You can specify fault tolerance through the INITIALIZE or REFORMAT functions.

6.4.3 Dump File

DISKINIT allows you to allocate a dump file through the INITIALIZE or REFORMAT options. This file is used in conjunction with the Control Mode Dump facility. When present on a disk volume, the dump file receives the contents of main memory after the system has entered Control mode. Control mode occurs when the system encounters a problem that prevents it from proceeding with normal operations. Refer to the VS System Operator's Reference for a complete explanation of Control Mode Dump.

DISKINIT allocates the dump file in a single extent and places a pointer in the volume label. It is created as a consecutive file with 2048-byte records. You set the size of the file when you initialize or reformat the disk. The dump file must be large enough to contain the main memory of any CPU for which you intend to use it. DISKINIT names the file @CMDUMP@ in the library @SYSDUMP. You cannot rename or scratch the file @CMDUMP@ or the library @SYSDUMP. The RELABEL function of DISKINIT preserves the pointer to the dump file, if one exists. If the dump file does not exist, DISKINIT stores ASCII blanks in the field.

6.4.4 Page Block and Page Pool

DISKINIT allows you to allocate a page block (for the VS50 and VS80) or page pool (for the VS25, VS45, VS85, VS90, and VS100) through the INITIALIZE, REFORMAT, or RELABEL functions. The page block or pool has the file name @POOL@, and it is located in the library @SYSPOOL.

The use of a page block or page pool keeps the Segment 2 paging files contiguous. Through DISKINIT, you can locate the page block or page pool near the most active areas of the volume. The contiguity of paging areas and the ability to control their placement can reduce seek times and thus significantly improve system throughput.

For the VS50 and VS80, if a page block is available, the system allocates paging files within the page block as each task is initiated. If there is more than one volume with a page block, the paging files are evenly distributed among them. If none of the active page blocks have sufficient contiguous space, or if there are no volumes enabled with page blocks, the system spreads the paging files evenly among the eligible volumes that have available space.

For the VS25, VS45, VS85, VS90, and VS100, the page pool, while similar to the page block, only assigns pages as needed. Thus, in most cases you can reserve considerably less space for the page pool than discrete paging files require. If a page pool is available, tasks are assigned to the page pool until it is fully committed (i.e., until the total Segment 2 sizes of the tasks assigned to the pool exceed the pool's physical size multiplied by the page pool commitment ratio). If there is more than one volume with a page pool, the system assigns a task to the pool with the fewest tasks. After a task is assigned to a page pool, the system allocates from the pool all Segment 2 pages required by the task. If all of the page pools are fully committed, or if there are no volumes enabled with page pools, the system spreads paging files among the eligible volumes that have available space.

For more information about the page block and page pool, refer to the VS Release 6.20 Software Bulletin.

6.4.5 Entering Data

Specify the following parameters to initialize a volume:

NEWVOL — The name you assign to the new volume. The default is the value of the VOLUME parameter on the Function Selection menu.

LABEL — The type of label you want placed on the disk. A volume can contain a standard label and a VTOC, or no label and no VTOC. The default is standard label (SL); all volumes used in standard processing must have standard labels. You can use a NL diskette volume to exchange data between the VS and other computer systems or VS Word Processing Systems, or to receive FLOPYDUP output.

TOLERATE — The level of fault tolerance you wish the volume's VTOC to have. Enter NONE, CRASH, or MEDIA. The default is NONE.

VTOCSIZE — The size of the Volume Table of Contents. The minimum size you can assign to any disk is four blocks. The default size of the VTOC can be overridden if you expect to make an unusually large number of entries to it. For example, you should enlarge the VTOC size if many relatively small files are to be created on the volume. The default size of the VTOC is computed by the system and assumes an average combination of average-sized program and text files. The averages used are based on empirical tests, and may not be appropriate for all applications. You are encouraged to modify the VTOC size, based on your experience with your applications.

OWNER — The volume owner's name. The OWNER field defaults to lowercase. To enter uppercase, use the Shift key.

PASSES — The depth of analysis performed. The INITIALIZE function automatically verifies the accessibility of each block on the disk surface. This verification process can be done briefly (BRIEF) if the volume is not expected to contain especially critical information, or it can be done more thoroughly (NORMAL) if the integrity of the information to be stored on the volume is a matter of critical concern. Because you initialize new disk volumes, NORMAL is the recommended option.

DUMPFILE Enter YES to allocate a permanent dump file. The default is NO.

PAGEPOOL Enter YES to allocate a page block or page pool. The default is NO.

Accept the default value for each parameter or specify a new value. After you enter all the parameters, the INITIALIZE function begins initialization. After initialization is complete, you can terminate DISKINIT by pressing PF16 or restart the program by pressing PF1, which returns you to the Function Selection menu.

6.5 THE REFORMAT FUNCTION

The REFORMAT function creates a new label and VTOC on a previously initialized disk volume. Reformatting destroys any data existing on the volume.

After you select the REFORMAT function and enter the volume name to the Function selection screen, you must supply the following information:

NEWVOL The volume name. The volume name defaults to the volume name specified on the Function Selection menu (i.e., the current name of the volume). You can change this name.

TOLERATE The level of fault tolerance you wish the volume's VTOC to have. Enter NONE, CRASH, or MEDIA. The default is the tolerance level of the existing VTOC.

VTOCSIZE The size of the VTOC. VTOCSIZE defaults to an optimum size determined by the system. Refer to Subsection 6.4.5.

OWNER The name of the volume owner to be placed in the volume label. The OWNER field defaults to lowercase; to enter uppercase, use the Shift key.

DUMPFILE Enter YES to allocate a dump file. The default is NO, unless a dump file is already present, in which case the default is YES.

PAGEPOOL Enter YES to allocate a page block or page pool. The default is YES if you are reformatting a volume that already contains a page block or page pool; otherwise, the default is NO.

After you enter all parameters, the program begins the reformatting procedure. After reformatting is complete, you can terminate DISKINIT by pressing PF16 or restart the program by pressing PF1. Pressing PF1 returns you to the Function Selection menu.

6.6 THE RELABEL FUNCTION

The RELABEL function renames an initialized disk volume and gives you the option to allocate a page block or page pool. For a VS25 and a VS45, RELABEL calculates a pointer to the bootstrap file, @MCBOOT@, in library @SYSTEM@ if the file currently resides on the volume. You need this file to perform an Initial Program Load (IPL). The pointer is necessary because the VS25 and VS45 IPL software cannot read the volume's VTOC. The software uses the pointer to gain access to the bootstrap file, which can read the VTOC. RELABEL does not reformat the volume and does not destroy the information on the volume.

After you select the RELABEL function from the Function Selection menu, the current volume name and device address are displayed. To rename the volume, enter the new name in the NEWVOL field. To allocate a page block or pool, enter YES in the PAGEPOOL field. The default is YES if the volume already contains a page block or pool; otherwise, the default is NO. You can terminate the RELABEL function without renaming the volume by pressing PF1. After relabelling is complete, you can terminate DISKINIT by pressing PF16 or restart the program by pressing PF1. Pressing PF1 returns you to the Function Selection menu.

6.7 THE VERIFY FUNCTION

The VERIFY function reads a disk volume to ensure that all blocks on the disk surface can be addressed and read. VERIFY does not alter or destroy any information on the volume. Unlike the INITIALIZE function, which reads each block and rewrites it, the VERIFY function only reads. It is therefore possible for a block to be bad for a write operation and not be reported as such by this function.

Any block that cannot be read is a bad block. The REMOVE function, detailed in Section 6.8, enables you to replace a bad block with a reserve block from the replacement pool. The replacement pool is a reserve of blocks that you can allocate as the need arises. Once the pool is depleted, the volume is unreliable.

After you select the VERIFY function from the Function Selection menu, the name and device address of the volume are displayed. Press ENTER to begin the verification process. The program checks the availability of all blocks and counts the number of bad blocks. DISKINIT compares the number of bad blocks to the number of available blocks remaining in the replacement pool. If the number of bad blocks exceeds those available in the pool, DISKINIT informs you that the volume is unreliable.

If the volume contains enough replacement blocks, the number of each bad block and, if it occurs in a file, the file name, are displayed. The VERIFY function does not replace a bad block; it only informs you that one exists.

After the VERIFY is complete, you can terminate DISKINIT by pressing PF16 or restart the program by pressing PF1, which returns you to the Function Selection menu.

6.8 THE REMOVE FUNCTION

The REMOVE function allows you to remove a bad block from the volume and replace it with a reserve block, if one is available. You can use the REMOVE function on 2220, 2265V1, 2265V2, 2265V3, 2280V1, 2280V2, 2280V3, or Q2040 disks.

The INITIALIZE function validates all blocks on a volume, flagging any bad ones. Sometimes an initially validated block shows Input/Output (I/O) errors after a period of use. If the I/O Error Log shows repeated errors in a particular block, or if the VERIFY function discloses the bad block, the block can be flagged as unreliable and can be removed from the volume by the REMOVE function. A number of extra blocks are reserved on each volume to serve as replacements for bad blocks. If a reserved block is available, REMOVE replaces the removed block with a reserved block.

The location of a bad block determines how you should replace it. If a bad block is located in the VTOC, you should back up the data, reinitialize the volume, and restore the data. If a bad block is listed as an available block, you can replace it with one from the reserve pool. If a block is located within a file, you should back up the file, scratch the original, replace the block, and restore the file.

After you select the REMOVE function and enter the volume name to the Function Selection menu, the program displays the Define Input screen. This screen displays the function, volume name, and device number. The screen also displays a cautionary note reminding you that this function permanently removes blocks from the volume. The program allows you to terminate the function before any blocks are removed by pressing PF1.

To continue the REMOVE function, enter the block number in the BLOCK field and press ENTER. After the REMOVE function is complete, you can terminate DISKINIT by pressing PF16 or restart the program by pressing PF1, which returns you to the Function Selection menu.

6.9 THE DUMP FILE OPTION

If you enter YES in the DUMPFILE field on the INITIALIZE or REFORMAT screen, DISKINIT displays the Dump File screen. This screen requests you to specify in kilobytes the size of the dump file. The dump file must be large enough to contain the main memory of the CPU for which you intend to use it. The maximum size of the dump file is 65,536 kilobytes. If you specify a size of zero, no dump file is allocated. During a REFORMAT operation on a volume already containing a dump file, the size defaults to the previously established file.

Upon execution, DISKINIT creates the file as a consecutive file with 2048-byte records. If the size you specify is an odd value, DISKINIT rounds it up to the next 2K value. The file is named @CMDUMP@ and is stored in library @SYSDUMP.

6.10 THE PAGE BLOCK AND PAGE POOL OPTIONS

If you specify YES in the PAGEPOOL field on the INITIALIZE, REFORMAT, or RELABEL screen, the Specify Page Pool screen prompts you to specify the size and location of the page block or pool.

The size of the page block can range from 1 to 65,536 kilobytes. The size of the page pool can range from 1 to 32,767 kilobytes. If the volume already contains a page block or pool, the default is the size of the current page block or pool. If you specify a size of zero, DISKINIT does not create a page block or pool.

> **NOTE**
>
> It is better to overestimate the size of the page block or pool than to underestimate it. You can easily decrease, but not increase, the size of the page block or pool at a later time.

Enter an integer between 0 and 9 for the location of the page block or pool relative to the VTOC. A value of 0 indicates a location nearest to the VTOC; a value of 9 indicates a location farthest from the VTOC. If a volume already contains a page block or pool, the default is the location of the current page block or pool. You should locate the page block or pool near the most active areas of the volume. For example, if the VTOC is the most active part of the volume, specify 0 as the relative location.

DISKINIT allocates the page block or pool in a single extent. If there is not sufficient space to do this, DISKINIT displays a message and allocates as large a page block or pool as possible. Because DISKINIT looks for an extent in the specified location, you may want to specify another location relative to the VTOC in order to allocate a page block or pool of the requested size.

6.10.1 Estimating the Page Block Size

You can use the following formula to estimate the initial size of the page block (in Mb):

PAGE BLOCK SIZE = [3 + (AVERAGESEG2 * MAXUSERS)] / VOLUMES

Note that 1 Mb is equal to 1024 Kb. Enter the following values for the variables in the formula:

Variable	Value
AVERAGESEG2	The average Segment 2 size (in Mb) currently used in your system. Use the default Segment 2 size specified in GENEDIT or, if adjustments have been made for specific tasks or User IDs, calculate the actual average. When in doubt, overestimate the value.

Variable	Value
MAXUSERS	The number of tasks (interactive and non-interactive) that are supported during a peak period.
VOLUMES	The number of volumes to be enabled with page blocks.

The formula includes 3 Mb to accommodate system tasks.

6.10.2 Estimating the Page Pool Size

To estimate the initial size of a page pool (in Mb), you can use the following formula:

```
PAGE POOL SIZE = [5 + (1/2 * AVERAGESEG2 * MAXUSERS)] / VOLUMES
```

Note that 1 Mb is equal to 1024 Kb. The variables AVERAGESEG2, MAXUSERS, and VOLUMES have the same values as in Subsection 6.10.1. The formula includes 5 Mb to accommodate system tasks and to provide a small margin of safety.

For example, if most tasks use 512K of Segment 2, if there are 60 tasks at peak usage, and if one volume is enabled for paging, the formula is as follows:

```
PAGE POOL SIZE = [5 + (1/2 * .5 Mb * 60)] / 1 = 20 Mb page pool
```

After you establish a page pool and monitor its activity, you can reduce its size through the RELABEL function to save additional space if actual use is light, or you can increase the size of the page pool to provide a satisfactory margin of safety if actual use nears the pool's capacity. You can use the POOLSTAT utility to monitor the utilization of a system's page pools; refer to Chapter 21 for more information.

6.11 A SAMPLE DISKINIT PROCEDURE

You can control DISKINIT processing through the VS Procedure language. The DISKINIT utility acquires information through GETPARM requests. The VS Procedure language MOUNT and DISMOUNT statements can provide the logical requirements of any necessary mounting or dismounting of initialized volumes. Because you cannot embed the Procedure language MOUNT and DISMOUNT statements in a RUN, ENTER (or DISPLAY) sequence, all Procedure language MOUNT and DISMOUNT operations must be performed prior to or following automated DISKINIT processing. You must use the DISKINIT MOUNT prname (parameter reference name) to mount uninitialized volumes. A complete list of DISKINIT GETPARMs is provided in Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

The following procedure relabels a disk volume. When you run the procedure, DISKINIT relabels the volume and allocates a page pool. The page pool has a size of 32,567 Kb, and its location is 0 (i.e., nearest to the VTOC).

```
PROCEDURE
RUN DISKINIT
ENTER FUNCTION FUNCTION = RELABEL, VOLUME = ZENITH
ENTER INPUT PAGEPOOL = YES
ENTER PAGEPOOL SIZE = 32567, LOCATION = 0
ENTER EOJ 16
RETURN
```

CHAPTER 7
THE DISPLAY UTILITY

7.1 INTRODUCTION

The DISPLAY utility allows you to examine the contents of any disk file by retrieving the file and displaying it at your workstation. The utility automatically provides information about the type of file organization and the number and size of records in the file. For indexed files, DISPLAY also indicates the location of the index key and its size and the number of alternate indices.

A file can be displayed in block mode or record mode. Block mode displays an exact image of the file layout as it exists. The file is displayed in 2K (2048)-byte blocks in physical block sequence. Data blocks and index blocks are displayed for an indexed file.

Record mode displays the file in records (as opposed to blocks) in two different formats: report-oriented format and record-oriented format. Report-oriented format displays a consecutive or relative file line by line; that is, each record is displayed on one line. DISPLAY cannot present indexed files in report-oriented format. Record-oriented format displays a consecutive, relative, or indexed file record by record.

The DISPLAY utility performs the following functions:

- Displays a file in the ASCII or hexadecimal representation
- Finds and displays an indexed record by key value
- Lists the index descriptions for an alternate indexed file and enables you to change the access path
- Finds and displays a record in a consecutive file by record, line, or block number
- Finds and displays a record in a relative file by record, line, or block number
- Finds and displays a record by text string
- Produces a printed copy of all or part of a file in record or block mode

To operate the DISPLAY utility, perform the following steps:

1. Specify the input file.
2. Select the display options.

An overview of DISPLAY processing is provided in Figure 7-1.

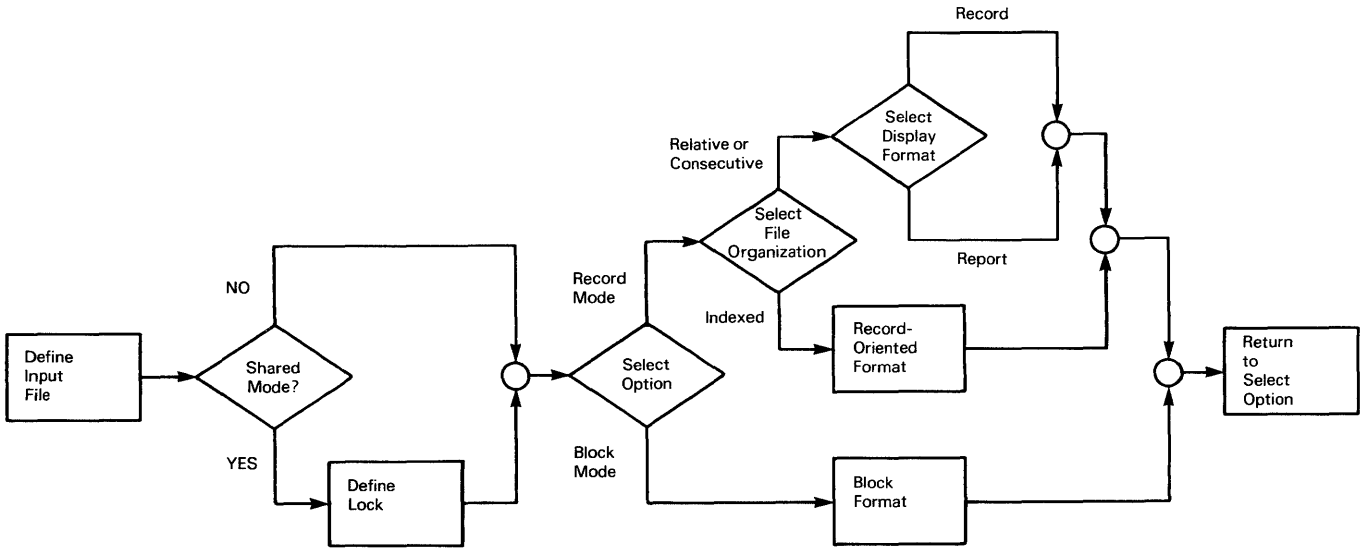


Figure 7-1. DISPLAY Processing

7.2 DEFINING THE INPUT

When DISPLAY processing begins, the Input Definition screen is displayed. The Input Definition screen allows you to define the input file and select a block mode or record mode display. The input parameters are defined as follows:

FILE — The name of the input file you want to display or print.

LIBRARY — The name of the library in which the input file resides. The default value for LIBRARY is your INLIB, as set through SET Usage Constants of the Command Processor or through the Procedure Language SET statement.

VOLUME — The name of the volume on which the input file resides. The default value for VOLUME is your INVOL, as set through SET Usage Constants of the Command Processor or through the Procedure Language SET statement.

ACCESS Allows you to specify the mode in which the input data is displayed or printed. If you select ACCESS=RECORD, the input file is displayed in logical records. If the input file is consecutive or relative, you can format the records record by record or line by line. You must format indexed files record by record.

If you select ACCESS=BLOCK, the input file is displayed in 2K (2048)-byte physical records. Specifying BLOCK allows you to view an exact image of the file layout as it exists on disk. You cannot display or print shared files in block mode.

If the input file is consecutive or relative, you can also specify ACCESS=PRINT. ACCESS=PRINT displays the file in report-oriented format without first displaying the Display Options menu. ACCESS=PRINT does not display a file in record-oriented format. ACCESS defaults to RECORD. For more information on display formats, refer to Section 7.3.

MODE Allows you to specify the mode in which DISPLAY will open an indexed or consecutive file. The options are INPUT and SHARED; the default is INPUT. To display a relative file, you must specify Input mode.

7.2.1 Displaying Files in Shared Mode

If you specify Shared mode, the Lock screen prompts you to indicate whether you want the file to be locked. Enter YES or NO in the LOCK field; the default is NO. If you lock a file, no changes to the file can occur while you are displaying it. If you specify NO, no lock is placed on the file, and there is no need to specify the TIMEOUT and BYPASS options.

If a file is held for update by another user, the TIMEOUT field specifies the length of time that DISPLAY waits to open the file in Shared mode with a lock. You can specify a timeout for a file if LOCK is equal to YES. Enter a value from 0 to 255 seconds; the default is 10 seconds.

The BYPASS field allows you to specify whether the file should be skipped if the timeout expires. Values for BYPASS are YES and NO; the default is NO.

If BYPASS is YES and the timeout expires, DISPLAY skips the file. If BYPASS is NO and the timeout expires, the Lock screen reappears with the message

FILE XXXXXXXX IN XXXXXXXX ON XXXXXX IS HELD BY USER XXX.

You can then redefine the LOCK, TIMEOUT, and BYPASS options and press ENTER to continue with the display operation. You can also press PF1 to skip the file on which the timeout occurred.

The Record Access Method (RAM) is always used to display shared files. For more information about RAM, refer to the VS Data Management System (DMS) Reference.

7.3 DISPLAY OPTIONS

After you specify the input file and the access type (RECORD or BLOCK), the Display Options menu appears. Six different display options menus are available; the utility displays the menu appropriate to the type of access and the organization of the input file. Included at the top of each menu is the input file name, the file organization, the maximum number of bytes in each record or block, and the number of records or blocks in the file. For indexed files, key position, key size, and alternate indices information is included. The six display options are described as follows.

- Record mode, consecutive file, and report-oriented format -- The input file is displayed line by line; that is, each record is displayed on one line. The options for manipulating the display operate on a line basis. Horizontal and vertical scrolling is allowed. You can print a hard copy in this format for all or part of the file.

  The following example illustrates a consecutive file in report-oriented format:

```
13217GRAY, DIANE      587 COLFAX ST          TEWKSBURY        MA01873
14281JONES, ARNOLD    514 GREENWILLOW RD     TEWKSBURY        MA01874
14359EVANS, ARNOLD    692 DOUGLAS AV         LOS ANGELES      CA90041
15692GREEN, HELEN     440 JENKINS ST         SAN FRANCISCO    CA94112
```

- Record mode, consecutive file, and record-oriented format -- The input file is displayed record by record. Records are displayed in the order in which they appear in the file. You can use PF10 to display the record in the ASCII or hexadecimal representation. Pressing PF10 converts the record display to the opposite of the current representation. You can print a hard copy in this format for all or part of the file.

  The following example illustrates a consecutive file in record-oriented format:

```
RECORD 1
   0   13217GRAY, DIANE      587 COLFAX ST       TEWKSB
 64   URY       MA01873
RECORD 2
   0   14281JONES, ARNOLD    514 GREENWILLOW RD  TEWKSB
 64   URY       MA01874
RECORD 3
   0   14359EVANS, ARNOLD    692 DOUGLAS AV      LOS AN
 64   GELES     CA90041
RECORD 4
   0   15692GREEN, HELEN     440 JENKINS ST      SAN FR
 64   ANCISCO   CA94112
```

- Record mode, relative file, and report-oriented format -- The input file is displayed line by line; that is, each record is displayed on one line. The phrase "<MISSING RECORD>" indicates an empty record slot; the phrase "<EMPTY RECORD>" indicates a zero-length record (a record that is present but contains no data). The options for manipulating the display operate on a line basis. Horizontal and vertical scrolling is allowed. You can print a hard copy in this format for all or part of the file.

  The following example illustrates a relative file in report-oriented format:

```
13217GRAY, DIANE      587 COLFAX ST         TEWKSBURY       MA01873
<MISSING RECORD>
14281JONES, ARNOLD    514 GREENWILLOW RD    TEWKSBURY       MA01874
14359EVANS, ARNOLD    692 DOUGLAS AV        LOS ANGELES     CA90041
<EMPTY RECORD>
15692GREEN, HELEN     440 JENKINS ST        SAN FRANCISCO   CA94112
```

- Record mode, relative file, and record-oriented format -- The input file is displayed record by record. Records are displayed in the order in which they appear in the file. Empty record slots are not displayed. Zero-length records, however, are displayed; the phrase "<empty>" appears beneath the record number to indicate that the record contains no data. You can use PF10 to display the record in the ASCII or hexadecimal representation. Pressing PF10 converts the record display to the opposite of the current representation. You can print a hard copy in this format for all or part of the file.

  The following example illustrates a relative file in record-oriented format:

```
RECORD 1
   0   13217GRAY, DIANE      587 COLFAX ST        TEWKSB
  64   URY       MA01873
RECORD 3
   0   14281JONES, ARNOLD    514 GREENWILLOW RD   TEWKSB
  64   URY       MA01874
RECORD 4
   0   14359EVANS, ARNOLD    692 DOUGLAS AV       LOS AN
  64   GELES     CA90041
RECORD 5
<empty>
RECORD 6
   0   15692GREEN, HELEN     440 JENKINS ST       SAN FR
  64   ANCISCO   CA94112
```

In the example, because Record 2 represents an empty record slot, it is not displayed. Record 6 is a zero-length record.

- Record mode, indexed file, and record-oriented format -- The input file is displayed record by record in primary key sequence. You can change the access path of indexed files with alternate indices. When you change the access path or select a record by key or text string, you should always select the first record in the file you want to display. To ensure that the entire file is read, select the first record. An indexed file always displays the next available record; it does not go back automatically to the beginning of the file. You can use PF10 to display the record in the ASCII or hexadecimal representation. Pressing PF10 converts the record display to the opposite of the current representation. You can print a hard copy in this format for all or part of the file.

  The following example illustrates an indexed file in record-oriented format:

```
KEY = 132
   0   13217GRAY, DIANE      587 COLFAX ST       TEWKSB
 64   URY       MA01873
KEY = 142
   0   14281JONES, ARNOLD    514 GREENWILLOW RD  TEWKSB
 64   URY       MA01874
KEY = 143
   0   14359EVANS, ARNOLD    692 DOUGLAS AV      LOS AN
 64   GELES    CA90041
KEY = 156
   0   15692GREEN, HELEN     440 JENKINS ST      SAN FR
 64   ANCISCO  CA94112
```

- Block mode -- Records are displayed in 2K (2048)-byte blocks in physical block sequence, starting with the first block in the first extent belonging to the input file. Because block mode displays an exact image of each block of the file as it is recorded on disk, control characters are displayed as special characters. You can use PF10 to display the record in the ASCII or hexadecimal representation. Pressing PF10 converts the file display to the opposite of the current representation. Data blocks and index blocks are displayed for an indexed file. You can print a dump in this format for all or a part of the file.

In record and block access modes, you can change the file display on the screen by pressing the PF key associated with the desired option. Combinations of the following options are available on the various display options menus:

DISPLAY Resumes displaying the input file in the most recently specified mode.

FIRST Displays the first record, group of lines, or block of data in the file. If you enter RECORD next to access mode and the access mode is record, the first record(s) in the file is displayed. If access mode is record and the format is report-oriented, the first 20 lines in the file are displayed. If the access mode is block, the first block of data in the file is displayed.

POSITION Displays the line or record number and column position of the cursor for consecutive or relative files in report-oriented format.

INDICES Lists the index descriptions for an alternate indexed file and enables you to change the access path.

PREVIOUS Displays the previous record, group of lines, or block of data in the file. If the access mode is record and the format is record-oriented, the previous record in the file is displayed. If the access mode is record and the format is report-oriented, the previous 15 lines in the file are displayed. If the access mode is block, the previous block in the file is displayed.

NEXT Displays the next record(s), group of lines, or block of data in the file. If the access mode is record and the format is record-oriented, the next record(s) in the file is displayed. If the access mode is record and the format is report-oriented, the next 15 lines in the file are displayed. If the access mode is block, the next block is displayed.

DOWN Moves the display back (down) one record or one line. If the access mode is record and the format is record-oriented, the display moves back one record. If the access mode is record and the format is report-oriented, the display moves back one line. If the access mode is block, the display moves back one line within the block.

UP Moves the display forward (up) one record or one line. If the access mode is record and the format is record-oriented, the display moves forward one record. If the access mode is record and the format is report-oriented, the display moves forward one line. If the access mode is block, the display moves forward one line within the block.

FIND Finds a record by the key value, record number, line number, text string, or block number you enter. DISPLAY locates indexed file records by key value. In a consecutive or relative file, if the access mode is record and the format is record-oriented, a record is found by record number. In a consecutive or relative file, if the access mode is record and the format is report-oriented, a line or text string can be found. If the access mode is block, a block in the file is found by block number. (You must delimit hexadecimal values by single quotes.)

TEXT — Finds the next occurrence of a text string. If you indicated the access mode is block, or if the access mode is record and the format is record-oriented, the next occurrence of a text string is found by DISPLAY.

L.MARGIN — Displays the first 80 columns of each line. If the access mode is record and the format is report-oriented, DISPLAY presents the first 80 columns of each line to your screen.

R.MARGIN — Displays the last 80 columns of each line. If the access mode is record and the format is report-oriented, DISPLAY presents the last 80 columns of each line to your screen.

MODE — Changes the display to hexadecimal mode if currently in ASCII, or to ASCII if currently in hexadecimal. This option is available only if the access mode is block, or if the access mode is record and the format is record-oriented.

REPORT — Displays the file in a report-oriented format, that is, by line rather than by record. This option is available only for consecutive or relative files if the access mode is record and the format is record-oriented.

LEFT 15 — Moves the display 15 columns to the left, that is, the display window is moved to the left, not the text. This option is available only for consecutive or relative files with record lengths greater than 80 bytes, if the access mode is record and the format is report-oriented.

RIGHT 15 — Moves the display 15 columns to the right, that is, the display window is moved to the right, not the text. This option is available only for consecutive or relative files with record lengths greater than 80 bytes, if the access mode is record and the format is report-oriented.

LEFT 1 — Moves the display 1 column to the left, that is, the display window is moved left, not the text. This option is available only for consecutive or relative files with record lengths greater than 80 bytes, if the access mode is record and the format is report-oriented.

RIGHT 1 — Moves the display 1 column to the right, that is, the display window is moved to the right, not the text. This option is available only for consecutive or relative files with record lengths greater than 80 bytes, if the access mode is record and the format is report-oriented.

PRINT — Prints all or part of the file in record or block mode, as appropriate. If the file is a print file in report-oriented format, and the starting value ALL is selected, the file is immediately spooled for printing. Otherwise, the file is printed according to the PRNTMODE default, as set through SET Usage Constants of the Command Processor. For additional information on printing a file, refer to Subsection 7.3.1.

EXIT — Returns the file to record-oriented format: that is, the file is to be displayed by record rather than by line. This option is available only for consecutive or relative files if the access mode is record and the format is report-oriented.

EOJ — Ends display processing of the file and generates a screen that gives the option of displaying another file or terminating the DISPLAY utility.

7.3.1 Printing a File

To print all or part of a file, select the PRINT option by pressing PF15 from the Display Options menu. The utility then displays the Print Options screen. This screen enables you to select the starting and ending record, the key, line, or block numbers, and the number of lines per page to be printed.

The printout for a file in block mode is a dump. If the file is a print file in report-oriented format, and the starting value ALL is selected, the file is immediately spooled for printing. Otherwise, the file is printed according to the PRNTMODE default, as set through SET Usage Constants of the Command Processor. The print parameters are defined as follows:

STARTING VALUE — The location at which the print file is to begin. For a consecutive or relative file, the STARTING VALUE is a record number if the format is record-oriented; the STARTING VALUE is a line number if the format is report-oriented. For an indexed file, the STARTING VALUE is a key number if the access mode is record. The STARTING VALUE is a block number if the access mode is block; if you specify the block number in hexadecimal, you must enclose the number in single quotes. In addition to record, line, key, or block number, FIRST is also a valid delimiter. STARTING VALUE defaults to ALL.

ENDING VALUE — The location at which the print file ends. For a consecutive or relative file, the ENDING VALUE is a record number if the format is record-oriented; the ENDING VALUE is a line number if the format is report-oriented. For an indexed file, the ENDING VALUE is a key number if the access mode is record. The ENDING VALUE is a block number if the access mode is block. If you specify the block number in hexadecimal, enclose the number in single quotes. In addition to record, line, key, or block number, you can use LAST as a valid delimiter. No default for ENDING VALUE exists.

LINES PER PAGE — The number of lines you want printed on each page. LINES PER PAGE defaults to the LINES previously set through SET Usage Constants of the Command Processor.

7.4 A SAMPLE DISPLAY PROCEDURE

You can control DISPLAY processing through the VS Procedure language. While you can specify the input to DISPLAY, you cannot specify the menu options as there are no GETPARM requests. For a complete list of DISPLAY GETPARMs refer to Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

The following example shows a procedure to run DISPLAY. Using this procedure you can display, manipulate, and print a hard copy of the file in record mode. If the input file is consecutive or relative, you can specify ACCESS=PRINT. ACCESS=PRINT displays the file in report-oriented format and does not display the Display Options menu first. ACCESS=PRINT does not allow a file to be displayed in record-oriented format. The end-of-job option terminates the DISPLAY utility.

```
PROCEDURE
RUN DISPLAY
ENTER INPUT FILE=SAVE3, LIBRARY=DJDDATA, VOLUME=NEWSYS, ACCESS=RECORD
ENTER EOJ 16
RETURN
```

CHAPTER 8
THE EZFORMAT UTILITY

8.1 INTRODUCTION

The EZFORMAT utility provides the following three functions that facilitate interactive program development:

- Enables you to dynamically design a workstation screen format and to automatically generate Assembler, BASIC, COBOL, or RPG II source code that reproduces the screen format. You can specify messages and input requests on the screen. You can also copy the source code EZFORMAT generates into your program through the external copy feature of the EDITOR during program development. EZFORMAT enables you to experiment with several screen formats before it generates the source code for the final screen format.

- Creates a complete Assembler or RPG II menu program that associates specified programs with PF keys. You can design the workstation screen format that describes this menu.

- Creates a COBOL or RPG II data entry program from a screen format that you design and a control file corresponding to the file to be accessed. This function provides an alternative to the DATENTRY utility, and is discussed in the VS File Management Utilities Reference.

EZFORMAT processing involves the following steps. Processing can also be controlled through the VS Procedure language, as described in Section 8.7.

1. Select a language or Menu option.

2. For the Menu option, define the functions to be performed by the menu program.

3. Define the screen format.

4. Define the output files. For the Menu option, define the language to be used to create the menu program.

An overview of the EZFORMAT processing described in this chapter is provided in Figure 8-1.

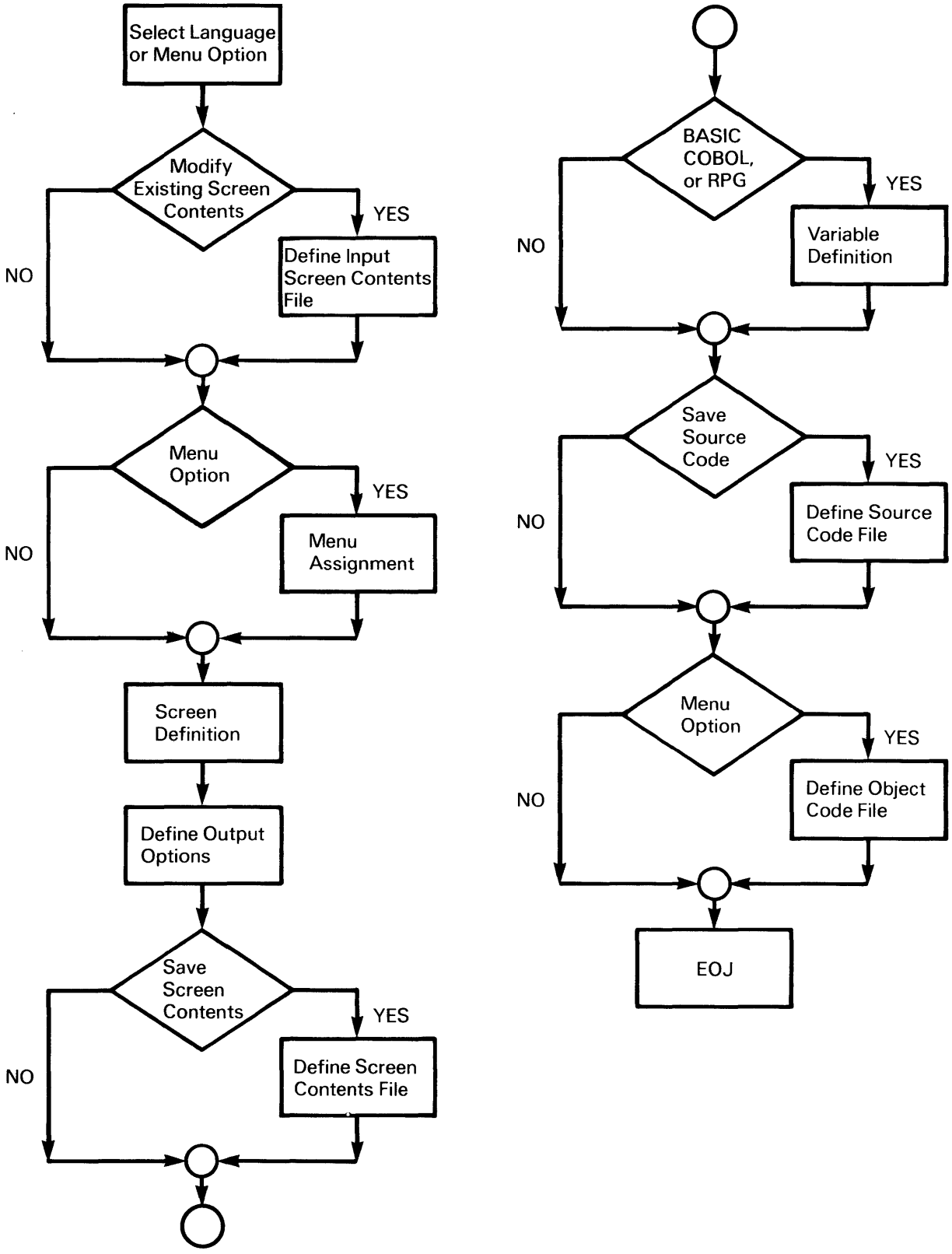


Figure 8-1. EZFORMAT Processing

8.2 RUNNING EZFORMAT

You can run the EZFORMAT utility in two ways: through the Command Processor and from the CONTROL utility. Refer to the VS File Management Utilities Reference for details on CONTROL. EZFORMAT displays the Function Selection screen when processing begins. You can select the desired EZFORMAT function and indicate if you want to create a new screen format or modify an existing screen format. To select the function, you type ASSEMBLER or A, BASIC or B, COBOL or C, MENU or M, or RPG or R in the OPTION field on the Function Selection screen.

Through the following PF keys, you can create a new format, modify an existing format, or exit the utility:

PF Key	Function
2	Create New Format Definition
3	Use Previously Saved Screen Contents As Input for Format Definition
16	Exit Without Generating Format Definition

If you select the Menu option, EZFORMAT processing continues as described in Section 8.3. If you select a language option, EZFORMAT processing continues with screen definition, presented in Section 8.4. Section 8.5 describes the process of altering an existing screen format.

8.3 CREATING A MENU

If you choose to create a menu program, you must associate the PF keys to be used in the menu with the desired programs or functions before you define the screen format. You do not need to select the menu language until you save the generated output (refer to Subsection 8.6.4). An Assembly language or RPG II menu program can perform any or all of the following functions:

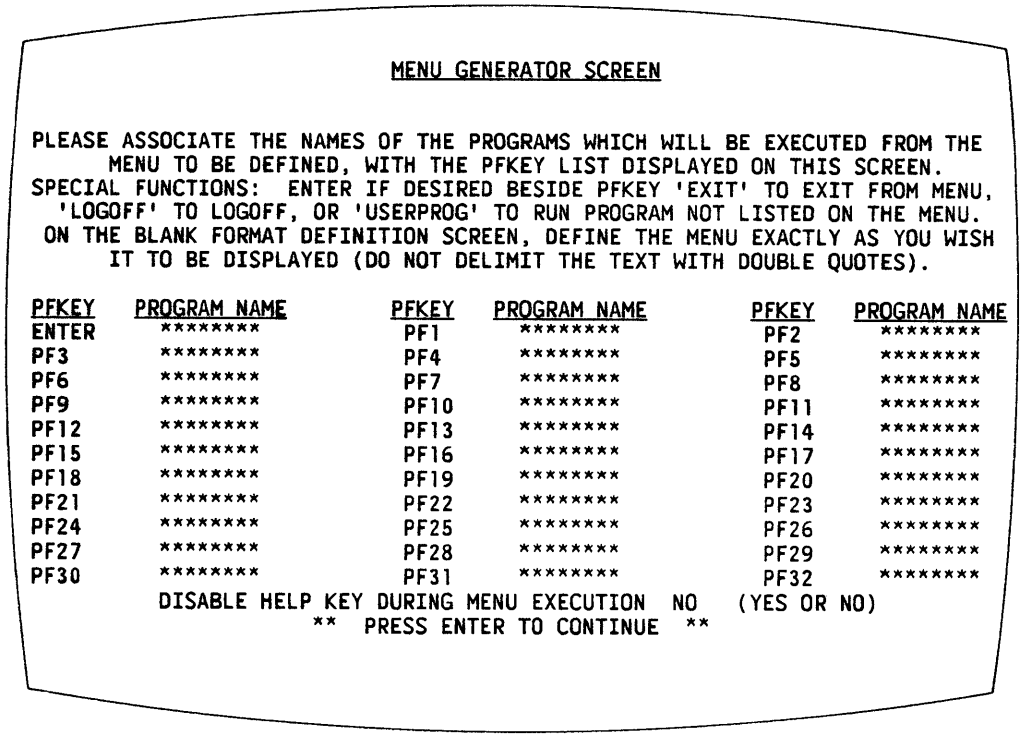
- Execute a specific program
- Allow the menu user to run any program
- Exit the menu
- Log the menu user off the system

You assign the menu functions to PF keys on the Menu Assignment screen, shown in Figure 8-2. EZFORMAT displays the Menu Assignment screen when you select the Menu option. The defined menu performs only those programs or functions that you associate with workstation PF keys on the Menu Assignment screen.

You can specify an action for the ENTER key and the PF1 through PF32 keys through the Menu Assignment screen. You can also disable the HELP key during menu processing. Any program or procedure that resides in the same library as the menu object code or in the System Library (@SYSTEM@) can be directly associated with a PF key. To associate a program or procedure with a PF key, you type the file name in the field corresponding to the PF key.

You can run any program from the resulting menu if you specify USERPROG for a key function. When the key designated with USERPROG is selected during menu execution, the menu requests the file, library, and volume names of the program or procedure to be run.

A PF key can terminate menu processing if you associate EXIT with ENTER or a PF key on the Menu Assignment screen. A workstation key can log the menu user off the system if you assign it a value of LOGOFF. A LOGOFF returns control to the process one link level up. For example, if the menu is running under control of a link from another program when the menu LOGOFF function is performed, the menu user is not logged off, but instead is returned to the calling program. HELP is disabled during menu processing if you modify the default response of NO to YES.



MENU GENERATOR SCREEN

PLEASE ASSOCIATE THE NAMES OF THE PROGRAMS WHICH WILL BE EXECUTED FROM THE MENU TO BE DEFINED, WITH THE PFKEY LIST DISPLAYED ON THIS SCREEN.
SPECIAL FUNCTIONS: ENTER IF DESIRED BESIDE PFKEY 'EXIT' TO EXIT FROM MENU, 'LOGOFF' TO LOGOFF, OR 'USERPROG' TO RUN PROGRAM NOT LISTED ON THE MENU.
ON THE BLANK FORMAT DEFINITION SCREEN, DEFINE THE MENU EXACTLY AS YOU WISH IT TO BE DISPLAYED (DO NOT DELIMIT THE TEXT WITH DOUBLE QUOTES).

PFKEY	PROGRAM NAME	PFKEY	PROGRAM NAME	PFKEY	PROGRAM NAME
ENTER	********	PF1	********	PF2	********
PF3	********	PF4	********	PF5	********
PF6	********	PF7	********	PF8	********
PF9	********	PF10	********	PF11	********
PF12	********	PF13	********	PF14	********
PF15	********	PF16	********	PF17	********
PF18	********	PF19	********	PF20	********
PF21	********	PF22	********	PF23	********
PF24	********	PF25	********	PF26	********
PF27	********	PF28	********	PF29	********
PF30	********	PF31	********	PF32	********

DISABLE HELP KEY DURING MENU EXECUTION NO (YES OR NO)

** PRESS ENTER TO CONTINUE **

Figure 8-2. The Menu Assignment Screen

Once you define the use of the menu keys, you can define the screen format that describes the menu by pressing ENTER. Screen definition is described in Section 8.4. An executable menu program using the PF key values will be generated even if you do not define a screen, but allow EZFORMAT to generate output. However, if you do not define a screen, the menu program will display a blank screen from which the defined menu functions operate.

8.4 CREATING A SCREEN DEFINITION

After you select the EZFORMAT function and make menu assignments (if necessary), EZFORMAT processing continues with screen format definition. For the language options, EZFORMAT generates source code to reproduce the screen format that you design at this phase of processing. For the Menu option, EZFORMAT incorporates the screen you design as the actual menu display for the generated menu program.

When designing the screen format, you can dynamically arrange fields of data and/or message text on the screen. EZFORMAT provides functions that allow you to manipulate the screen contents. The screen format is defined in terms of fields, which are groups of characters of less than one line long. The Menu and language options support different types of fields. The Menu option supports displayed text only; the language options also permit modifiable alphanumeric and numeric fields.

After you select a language option, EZFORMAT displays the Format Options screen, which describes the types of fields available for designing a screen. An equivalent screen is not displayed for the Menu option because only the text field type is available. The following field types are available for language option screen design.

Text Field	Displays nonmodifiable text at the indicated position on the workstation screen. No input is accepted in a text field when the screen is displayed. Text specified on the screen is considered a text field if the text is enclosed in double quotation marks. The quotation marks occupy a screen position, but are not displayed on the generated screen. For the Menu option, you can only define text fields on the screen format. You therefore do not need to delimit text fields with quotes when you define a screen for the Menu option.
Numeric Modifiable Field	Accepts only numeric input when the screen is presented in your program. The field type is identified by a 9, +, -, or any numeric character as the initial character in the field. The field is ended by a space, the end of the line, or the double quote indicator of a text field. For the COBOL and RPG II options, an initial plus (+) sign generates a field that converts any entered negative data to positive values. For example, if you enter -44 into a field specified as +99, ␣44 (a space followed by 44) is stored. You can enter negative, positive, or unsigned values into a field with an initial specification of -.

If the field specification consists of an initial 9, a plus sign (+) followed by a 9, or a minus sign (-) followed by a 9, pseudoblanks are displayed in the resulting field. If you specify the field with an initial numeric character other than 9, or if you type the initial + or - followed by a number other than 9, the specified number is displayed as a default value when the screen is displayed. The initial + or - is displayed as a pseudoblank, regardless of whether it is followed by a default value or pseudoblanks.

Alphanumeric Modifiable Field

Accepts alphanumeric input when the screen you design is presented in your program. The field type is identified by an initial character other than a number, +, or -. The field is ended by a space, the end of the line, or the double quote indicator of a text field. For the BASIC and COBOL options, an initial asterisk (*) determines that the field cannot be left blank when the screen is processed.

If the field specification contains an initial X, or an * followed by an X, the program displays pseudoblanks in the resulting field. If you specify the field with an initial character other than X, or if you type the asterisk (*) followed by a letter other than X, the specified character string is displayed as a default value. An initial asterisk (*) is always displayed as a pseudoblank.

The Manipulation Options screen is displayed after the Format Options screen (for the language options) or after menu assignment (for the Menu option). The Manipulation Options screen describes the options that enable you to position fields on the screen during the screen definition process. The screen is designed on the Screen Definition display, which you can access by pressing ENTER from the Manipulation Options screen. You can return to the Manipulation Options screen, during screen definition, for a description of the formatting options. For the language options, you can also return to the Format Options screen for a description of the available field types. You can also create a printed copy of the formatting and/or field type instructions by pressing PF13 from the Manipulation Options screen. Thus, you can refer to any information and return to the same stage of screen design.

You design the screen format on the Screen Definition display by typing field specifications in the desired screen locations. You can define Rows 1 through 24 of the workstation screen for the Assembler, BASIC, COBOL, and RPG options. For the Menu option, EZFORMAT reserves Row 24 for menu messages. Each row on the screen begins with a nondisplayed Field Attribute Character (FAC) to designate the row's default field type. As a result, only 79 columns of the workstation screen are available for design. For the language options, the Screen Definition display initially shows all pseudoblanks. For the Menu option, the initial display is blank.

The cursor control, new line, home, tab, back tab, back space, insert, and delete keys all function to facilitate screen design on the 24 rows (23 rows for the Menu option) and 79 columns on the Screen Definition display. In addition, the EZFORMAT Screen Manipulation options provide more flexible functions for screen design. The functions referenced by ENTER, PF1 through PF9, PF12, and PF14 operate directly from the Screen Definition display. PF1 and PF12 through PF16 only function directly from the Manipulation Options screen. PF1 is not enabled for Menu screen formatting. The available functions are described on the Manipulation Options screen and as follows:

PF Key	Function	Description
ENTER	Display Formatting Information	Displays the Manipulation Options screen. The available EZFORMAT screen manipulation functions are described on the resulting display.
1	Display Field Type Information	Displays the Format Options screen. The available field types are described on the resulting display. This function is not implemented for the Menu option because you can define only text fields.
2	Column	Displays the current column position of the cursor in the upper right portion of the screen, temporarily overwriting any field defined in that location.
3	Move Up	Moves the entire contents of the row indicated by the cursor (target row) to equivalent positions in the previous row. The contents of the previous row are destroyed. The target row is replaced by pseudoblanks (or blanks, for the Menu option). No action occurs if you attempt to move Row 1. The function is illustrated as follows; Row 6 is the target row in the example:
4	Move Down	Moves the entire contents of the row indicated by the cursor (target row) to equivalent positions in the following row. Any fields you defined in the following row are destroyed. The target row is replaced by pseudoblanks (or blanks, for the Menu option). No action occurs if you attempt to move Row 24 (Row 23 for the Menu option).

Row Number	Before	After
5	AAA	BBB
6	BBB	(pseudo)blanks
7	CCC	CCC
8	DDD	DDD

PF Key	Function	Description
5	Copy Up	Copies the contents of the row indicated by the cursor (target row) to equivalent positions in the previous row. The contents of the target row are unchanged; the contents of the previous row are destroyed. Thus, the target row and the previous row are identical. No action occurs if you attempt to copy Row 1. The function is illustrated as follows; Row 6 is the target row in the example:

Row Number	Before	After
5	AAA	BBB
6	BBB	BBB
7	CCC	CCC

PF Key	Function	Description
6	Copy Down	Copies the contents of the row indicated by the cursor (target row) to equivalent positions in the following row. The contents of the target row are unchanged; the contents of the following row are destroyed. Thus, the target row and following row are identical. No action occurs if you attempt to copy Row 24 (Row 23 for the Menu option).
7	Roll Up	Moves the contents of each row beginning with the row below the one indicated by the cursor (target row) and ending with Row 24 (Row 23 for the Menu option) to the row above (roll up). The contents of the target row are destroyed. The resulting Row 24 (Row 23 for the Menu option) is filled with pseudoblanks (or blanks for the Menu option). For example, if the cursor is located at Row 21, Row 22 is moved into Row 21 (destroying the contents of Row 21), Row 23 is moved into Row 22, Row 24 is moved into Row 23, and Row 24 becomes empty. No action occurs if you attempt a roll-up with Row 24 (Row 23 for the Menu option) as the target row. The roll-up function is illustrated as follows; Row 6 is the target row in the example for a language option.

Row Number	Before	After
5	AAA	AAA
6	BBB	CCC
7	CCC	DDD
.	.	.
.	.	.
23	.	TTT
24	TTT	(pseudo)blanks

PF Key	Function	Description
8	Roll Down	Moves the contents of each row, beginning with the row indicated by the cursor (target row) and ending with Row 24 (Row 23 for the Menu option), to the row below (roll down). The contents of Row 24 (Row 23 for the Menu option) are destroyed. The resulting target row is filled with pseudoblanks (or blanks for the Menu option). For example, if the cursor is located at Row 21, Row 21 is moved into Row 22, Row 22 is moved into Row 23, Row 23 is moved into Row 24, and the previous contents of Row 24 are lost. The contents of Row 24 (Row 23 for the Menu option) are destroyed if a roll down operation is attempted with Row 24 (Row 23 for the Menu option) as the target row.
9	Center	Centers horizontally the contents of the row indicated by the cursor (target row).
12	Tabs and Uplow	Displays a screen that enables you to set a maximum of ten tabs and to allow uppercase and lowercase input for screen design. You set the tabs by typing a non-blank character underneath the desired column position. Note, however, that the tab positions are calculated from column one of the workstation, which is a nondisplayed Field Attribute Character. You can enter lowercase and uppercase input if you change the MODE value to "UPLOW". Changing the value of MODE to UPLOW only affects your input during the screen definition process and not during the actual program operation. Thus, text fields and default values for modifiable fields can have lowercase values, but lowercase input is not accepted in modifiable fields during program execution.
13	Print Information	Creates a print file containing the information given on the Format Options and Manipulation Options screens.

PF Key	Function	Description
14	Display Finished Screen or	For the language options, displays the screen as it will appear in your program. If you press ENTER from the final screen you return to either the Screen Definition display or the Manipulation Options screen. The screen that you return to depends on which screen is displayed when you press PF14.
	Review Menu Functions	For the Menu option, displays the selections you made on the Menu Assignment screen. If you press ENTER, you return to either the Screen Definition display or the Manipulation Options screen. The screen that you return to depends on which screen is displayed when you press PF14.
16	Exit	Ends the screen definition process.

When screen design is completed, you accept the format by pressing PF16 from the Manipulation Options screen. EZFORMAT then creates output files according to your instructions. The output file options for each EZFORMAT function are described in Section 8.6.

8.5 MODIFYING A PREVIOUSLY DESIGNED SCREEN

With EZFORMAT, you can modify a previously defined screen format, providing that the screen contents have been saved in a separate file. (Refer to Section 8.6 for details.) The screen contents file for the Menu option also contains previous menu assignments; therefore, you can also modify the menu functions. You can modify existing screen contents by pressing PF3 from the Function Selection screen when processing begins. EZFORMAT then requests you to enter the file, library, and volume names of the file containing the screen contents and, for the Menu option, menu assignments. The input library name defaults to your User ID concatenated with SAVE; you can override the default vale if it is inappropriate. You can then modify the screen contents and menu assignments, following the procedures detailed in Sections 8.3 and 8.4, and save whatever files are desired.

8.6 DEFINING EZFORMAT OUTPUT

After you design a screen format, EZFORMAT can save the screen contents and generate source code that reproduces the screen. For the Menu option, object code that executes the constructed menu can also be generated. After you press PF16 from the Manipulation Options screen, and accept the screen format, the File Creation Options menu is displayed. The File Creation Options menu offers the following options:

PF Key	Function
2	Save Generated Output Only
3	Save Screen Contents Only
4	Save Screen Contents and Generated Output
16	Exit Without Saving Any Files

The screen contents file contains the row-by-row data for the workstation screen. For the Menu option, the screen contents file also includes the workstation key assignments, as described in Section 8.3. If you save the screen contents, you can modify the screen format at a later date. (Refer to Section 8.5 for details.) The generated output differs according to the selected EZFORMAT function, but in general is the source and/or object code corresponding to the specified screen format. You can choose to save only the screen contents, only the generated output, or both the screen contents and the generated output. You can also choose not to save any files by terminating EZFORMAT processing immediately.

If you elect to save the screen contents, either alone or in combination with the generated output, EZFORMAT immediately requests the file, library, and volume names of the output file for the screen contents. EZFORMAT supplies a default library name, constructed by concatenating your User ID with SAVE, but you can override this value if it is inappropriate. The screen contents file is used as input to EZFORMAT if an existing screen format is modified. (Refer to Section 8.5 for details.) If you save only the screen contents, EZFORMAT processing terminates after you name the file containing the screen contents.

If you save the generated output, subsequent processing differs for each EZFORMAT function. The following sections describe the generated output and required user action in each case.

8.6.1 Assembler Generated Output

If you save the generated output, EZFORMAT creates an Assembler source file that contains the screen format you defined in EZFORMAT. You can copy this source file into an Assembler program through the XCOPY function of the EDITOR. The source file consists entirely of Define Constant (DC) statements; you must code any corresponding WRITE or READ statements in the main Assembler program.

EZFORMAT requests you to enter the file, library, and volume names of the screen format Assembler source file. The library name defaults to the value obtained by concatenating your User ID with COPY; you can override this value if it is inappropriate. Processing terminates after the Assembler source file is created. EZFORMAT return codes are listed in Appendix D.

8.6.2 BASIC Generated Output

If you save the generated output, EZFORMAT constructs BASIC source code that displays the screen you designed and reads any data entered into modifiable fields. EZFORMAT also defines and dimensions variables of the appropriate data type corresponding to the modifiable fields on the screen. The resulting BASIC source file contains appropriate DISPLAY, ACCEPT, and DIM statements. While the EZFORMAT-generated source code will run when compiled, the screen displays only once. Data entered on the displayed screen format is lost unless you modify the source code before compilation.

EZFORMAT allows you to change the generated variable names to more meaningful values, and to set upper and lower bounds for acceptable input. For these purposes, EZFORMAT displays a BASIC Variable Definition screen on which you can alter the default generated variable definitions, prior to creating the source file. The BASIC Variable Definition screen displays the field position, data name, and default input range for each modifiable field defined in the screen format. While you cannot modify the field position, you can change the data name and range to match variables used in the program into which the EZFORMAT screen format is to be copied.

You can view the screen format during data name and range definition by pressing PF14 from the BASIC Variable Definition screen. Variable definition resumes when you press ENTER from the resulting screen format display. If you defined more variables on the screen format than fit on the first BASIC Variable Definition screen, you can define the remaining variables on subsequent screens. You can reach subsequent BASIC Variable Definition screens by pressing ENTER from the current screen. You can return to the first BASIC Variable Definition screen to alter any previous variable definitions by pressing PF1 from the current screen. The modifiable parameters on the BASIC Variable Definition screen(s) are described as follows:

DATA NAME — The name by which the generated source code will reference the modifiable field, i.e., the variable name. The data name for an alphanumeric field defaults to Xn$, where n varies from 0 to 9, for each X varying from A to Z. Thus, EZFORMAT generates up to 260 (A0$, A1$, ..., Z8$, Z9$) unique default character names. If you create more than 260 alphanumeric variables, you must change the corresponding data names for the excess variables to avoid overwriting earlier variables. In a similar way, integer and fractional variables default to data names of A0% through Z9%. Because the default numeric variable name contains a % (indicating an integer variable), you must change the data name for fractional numeric data.

NOTE

While VS BASIC supports 64 character data names, data names that you assign through EZFORMAT cannot exceed 14 characters.

RANGE — The upper and lower bounds that determine acceptable input values. Default range values are displayed for those variables defined in the screen format with an * to ensure nonblank input. These range restrictions can be made more stringent (or eliminated), and/or upper and lower bounds can be applied to other variables, merely by supplying values in the TO and FROM range parameters. You can define only one set of range parameters for each variable.

When all definitions are complete, you can continue EZFORMAT processing by pressing PF16. EZFORMAT requests you to enter the file, library, and volume names for the screen format source file. While the library name defaults to the value obtained by concatenating your User ID with COPY, you can override this value. After you name the output file, EZFORMAT creates the BASIC source file corresponding to the screen format and your variable definitions. Then, processing terminates. EZFORMAT return codes are listed in Appendix D.

8.6.3 COBOL Generated Output

If you save the generated output, EZFORMAT constructs COBOL source code that contains the Data Division definition of the workstation record. The record contains data names, data types, and initial values corresponding to the screen format. EZFORMAT does not generate Procedure Division statements to process this record; you must write such code in the main COBOL program into which the EZFORMAT-generated code is copied.

EZFORMAT generates default field, source, and object names, as well as default range parameters for each modifiable field you defined on the screen format. You can modify these default values to correspond to variables used in the main program on the COBOL Variable Definition screen. You can view the screen format during variable definition by pressing PF14 from the COBOL Variable Definition screen. Variable definition resumes when you press ENTER from the resulting screen format display. If you defined more variables on the screen format than fit on the first COBOL Variable Definition screen, you define remaining variables on subsequent screens. You can reach each successive screen by pressing ENTER from the current screen. You can return to the first COBOL Variable Definition screen to alter any previous variable definitions by pressing PF1 from the current screen. Each variable definition involves four parameters, detailed as follows:

FIELD NAME — The name used by the EZFORMAT-generated COBOL source code to refer to the screen location. EZFORMAT automatically generates a name of the form ROWXX-COLYY, where XX and YY represent the 2-digit row and column positions, respectively. You can change this value, providing the value does not duplicate a source name, an object name, or a COBOL reserved word. You do not need to change the EZFORMAT-generated field name.

SOURCE — The field from which FIELD NAME obtains its initial value when the screen is displayed. EZFORMAT supplies names that reflect the data type and the field occurrence. For example, ALF-OBJ001 reflects an alphanumeric modifiable field, which is the first input request. NUM-OBJ005 reflects the fifth input request, which is a numeric modifiable field.

If you have supplied a default value for the modifiable field during screen definition, neither you nor EZFORMAT can assign a SOURCE name. The default value specified during screen definition is always displayed by the COBOL program.

OBJECT — The field to which the value of FIELD NAME is moved when the ENTER (or any AID key not in an ON clause of a DISPLAY AND READ statement) has been pressed. EZFORMAT supplies a default name constructed in the same manner as the SOURCE default name.

RANGE — The upper and lower bounds of values to be accepted as input by the COBOL program. If you defined an alphanumeric modifiable field with an initial *, EZFORMAT supplies default upper and lower bounds to ensure that a blank value cannot be entered. The COBOL collating sequence determines the interpretation of the upper and lower bounds; consult the VS COBOL Reference for details. RANGE values are optional. You can define only one set of range parameters for each variable.

When you complete all definitions, you continue processing by pressing PF16. EZFORMAT requests you to enter the file, library, and volume names for the screen format source file. The library name defaults to the value obtained by concatenating your User ID with COPY. You can override this value if it is inappropriate. After you name the output file, EZFORMAT creates the COBOL source file corresponding to the screen format and your variable definitions, and processing terminates. EZFORMAT return codes are listed in Appendix D.

8.6.4 Menu Generated Output

If you save the generated output, EZFORMAT constructs an executable program that corresponds to the menu described by the screen format and the Menu Assignment screen selections. EZFORMAT also creates an Assembler or RPG II source file for the menu program and a source listing print file resulting from the menu assembly or compilation. You can further customize the menu program by editing the generated source file.

After you have chosen to save the generated output, EZFORMAT requests you to identify the language in which EZFORMAT should generate the menu program on the Source Code Selection screen. If you press PF2, EZFORMAT generates the menu in Assembly language; if you press PF3, EZFORMAT generates the menu in RPG II. Menu programs created in RPG II can perform the same functions as menus created in Assembly language.

EZFORMAT then requests you to enter the file, library, and volume names for the Assembler or RPG II source file to be generated. The library name defaults to the value obtained by concatenating your User ID with COPY; you can override this value if it is inappropriate. Next, EZFORMAT requests you to enter the file parameters for the menu object code. If user programs are associated with PF keys during menu assignment, place the menu program in the same library as the user programs. For RPG II menus, note that the CEXIT and LINK user subroutines must reside in the USERSUBS library on the System Volume so EZFORMAT can link them to the menu program. The print file is automatically assigned the same file name as the source file, and is placed in your print library. Processing ends once the menu program is created. EZFORMAT return codes are listed in Appendix D.

Running the Menu Program

The screen format you created through EZFORMAT is displayed when the menu program is run. The functions, programs, and procedures are executed by the menu when the user presses the associated PF keys. Control returns to the menu when a program completes. The USERPROG function does not retrieve the menu user's default RUNLIB and RUNVOL values. The USERPROG function supplies the file parameters of the most recently run USERPROG as a default value. A message is displayed in Row 24 of the workstation screen that indicates the status of the most recently completed menu operation. The message states whether the program was cancelled or completed normally. An error is indicated if the program specified by the menu cannot be located. Menu processing ends if the user selects a workstation key associated with the EXIT or LOGOFF functions.

8.6.5 RPG Generated Output

If you save the generated output, EZFORMAT constructs an RPG II source file that displays the screen that you designed, and accepts any data entered into a modifiable field. In a subroutine, EZFORMAT defines and assigns initial values to variables corresponding to the modifiable fields defined on the screen format. The RPG II source code also produces calculations to enable the ENTER and PF16 keys and to terminate the program when PF16 is pressed.

EZFORMAT immediately requests you to enter the file, library, and volume names of the screen format RPG II source file to be generated. While the library name defaults to the value obtained by concatenating your User ID with COPY, you can override this value if it is inappropriate.

After you name the output file, EZFORMAT displays the RPG II Variable Definition screen. The RPG II Variable Definition screen enables you to modify the default source and object field names to correspond to variable names used in the main program. You can view the screen format during variable definition by pressing PF14 from the RPG II Variable Definition screen. Variable definition resumes when you press ENTER from the resulting screen format display. If more variables are defined on the screen format than fit on the first RPG II Variable Definition screen, you can define the remaining variables on subsequent screens, accessed by pressing ENTER from the current screen. You can return to the first RPG II Variable Definition screen to alter any previous variable definitions by pressing PF1 from the current screen. Each variable definition involves two parameters, described as follows:

Source — The variable from which the field's initial value is obtained when the screen is displayed. You cannot define a source field name if the screen format specifies an initial value for the field. EZFORMAT supplies a default value that reflects the data type and modifiable field occurrence. For example, ALF001 reflects an alphanumeric field, which is the first input request on the screen. NUM005 indicates the screen's fifth input request, which is a numeric modifiable field.

Object — The variable that receives the value entered in the screen format field. EZFORMAT supplies a default name constructed in the same manner as the Source default name.

When you complete all definitions, pressing PF16 continues EZFORMAT processing. EZFORMAT requests you to enter the file, library, and volume names for the screen format source file. EZFORMAT creates the RPG II source file corresponding to the screen format and your variable definitions, and terminates processing. EZFORMAT return codes are listed in Appendix D.

8.7 A SAMPLE EZFORMAT PROCEDURE

Although screen definition in EZFORMAT is by nature an interactive process, a significant amount of workstation interaction in EZFORMAT processing can be eliminated or controlled. You can select the function, file creation options, and output file locations through a procedure. If the screen has been previously defined, all workstation interaction, except BASIC, COBOL, or RPG II variable definition, can be eliminated or controlled. A complete listing of the EZFORMAT GETPARM requirements is given in Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

Although the Function Selection screen does not request information through a GETPARM, the utility contains a hidden GETPARM, identified by the OPTIONS prname, that requests the same information as the Function Selection screen. The EZFORMAT function is selected through the LANGUAGE keyword, PF2 effects the creation of a new screen format, and PF3 indicates that previously created screen contents are to be used as input to EZFORMAT. Without procedure control, you must enter the screen definition process even if the input screen format is in final form. The hidden GETPARM also allows you to bypass screen definition completely through the PROCEDUR keyword. The value of the PROCEDUR keyword is only considered when EZFORMAT processes previously created screen contents. The default value, YES, bypasses the screen definition phase of EZFORMAT and continues EZFORMAT processing with the File Creation Options screen. However, if the Menu option is selected, the Menu Assignment screen is not bypassed and is displayed immediately before the File Creation Options screen.

A sample procedure follows that runs the EZFORMAT utility, selects the COBOL function, and chooses to create a new screen format. The procedure saves the screen contents and generated COBOL source file as output, and names the files containing the screen contents and COBOL source (accepting default library names). All workstation interaction is eliminated, except for the actual screen definition and COBOL variable definition processes.

```
PROCEDURE
RUN EZFORMAT
ENTER OPTIONS 2, LANGUAGE=COBOL
ENTER FUNCTION 4
ENTER SCREEN FILE=DATA, VOLUME=SYSTEM
ENTER COPYLIBR FILE=DATA, VOLUME=SYSTEM
RETURN
```

CHAPTER 9
THE FLOPYDUP UTILITY

9.1 INTRODUCTION

The FLOPYDUP utility duplicates diskettes, creates diskette image files (files containing byte-by-byte copies of entire diskettes), and transfers diskette image files to diskettes. You can use FLOPYDUP to create several copies of a single diskette or to back up a diskette to hard disk. Because FLOPYDUP simply transfers diskette image files to diskettes on a block-by-block basis, it can also generate non-VS diskettes. However, FLOPYDUP creates and reads only single-sided, single-density diskettes.

Only diskettes previously initialized as nonlabeled (NL) can be used to receive FLOPYDUP diskette output. Any data previously residing on an output NL diskette (such as word processing documents) is destroyed when FLOPYDUP transfers output data to the diskette.

FLOPYDUP supports the following functions:

- Copies the contents of a diskette to another diskette through the DUPLICATE function
- Copies the contents of a diskette to a diskette image file through the COPY function
- Copies the contents of a diskette image file to a diskette through the GENERATE function
- Supports VS Procedure language control of workstation interaction

Figure 9-1 provides an overview of FLOPYDUP processing.

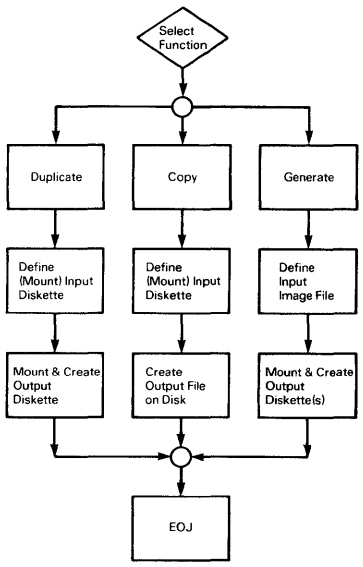


Figure 9-1. FLOPYDUP Processing

9.2 SELECTING A FUNCTION

The first screen FLOPYDUP displays at your workstation is the Function Selection screen. Select the DUPLICATE function by entering D, the COPY function by entering C, or the GENERATE function by entering G. Alternatively, you can cancel processing by pressing PF16.

9.3 THE DUPLICATE FUNCTION

After you specify the DUPLICATE function, FLOPYDUP requests you to supply the name and the label type of the diskette you want to duplicate. You also must enter the device number of the drive on which the diskette is mounted. If you have not mounted the diskette, FLOPYDUP automatically invokes the mount routine. The diskette can be mounted with shared or exclusive usage mode.

The program first creates a complete image of the diskette in a work file on the hard disk. The image is always the size of a full diskette (315,392 bytes), even if the original diskette is not completely full. The entire diskette, including the Volume Table of Contents (VTOC) (if present) and all libraries and files, is duplicated within a single work file.

After you copy the diskette into a work file, the program requests you to enter the name of the output NL diskette and the device number of the drive on which the diskette is mounted. Because the input diskette is no longer required, you can use the same drive that held the input diskette to hold the output diskette. If you have not mounted the output diskette, FLOPYDUP allows you to do so. Any diskette already mounted in the specified drive is automatically dismounted by the program.

After you mount the new diskette, the program copies the diskette image file from the work file onto the diskette. When the copying process is finished, the End-of-Job screen informs you of the details of the operation. The End-of-Job screen then allows you to end or restart FLOPYDUP processing, or to create another duplicate diskette with the same contents. Press PF16 to terminate FLOPYDUP processing. Press PF1 to restart the utility. Press PF2 to create another duplicate diskette. All output diskettes contain exact, duplicate images of the input diskette.

NOTE

FLOPYDUP does not dismount the newly created diskette unless you create a duplicate diskette. If you create a duplicate diskette, FLOPYDUP does not dismount the last duplicate diskette. Because the diskette is originally mounted as an NL diskette, the output diskette cannot be accessed by any utilities requiring VTOC information until the diskette is remounted as a standard label (SL) diskette.

9.4 THE COPY FUNCTION

The COPY function performs the first part of the DUPLICATE process by copying an image of the entire diskette to one file on the disk. Because the diskette image file the COPY function creates is a permanent disk file rather than a work file, you can use the COPY function to back up a diskette to the disk. Subsequently, the GENERATE function can use the diskette image file to create a duplicate of the original diskette.

The COPY function asks you to supply the name and the label type of the diskette you want copied. Also, you must enter the device number of the drive on which the diskette is mounted. You can mount the diskette with shared or exclusive usage mode. If the specified diskette is not already mounted, FLOPYDUP invokes the mount routine. When you mount the input diskette, FLOPYDUP requests you to enter the file, library, and volume names of the disk file that is to contain the diskette image. Upon completion of the COPY function, a message is displayed informing you of the status of the operation. You can exit or restart FLOPYDUP processing. Press PF16 to terminate FLOPYDUP processing or press PF1 to restart the utility.

9.5 THE GENERATE FUNCTION

The GENERATE function creates a new diskette by copying a diskette image file from the hard disk to the output NL diskette. This function is not restricted to image files originally created by any one means. The image file can be created by the COPY function of FLOPYDUP, COPY2200, or by a user program. GENERATE can create new diskettes in formats not standard on the VS.

When generating diskette contents from a disk file, FLOPYDUP transfers the file in blocks of 2048 bytes; it does nothing else to the data. For example, the utility does not decompress a compressed file before wiritting it to the diskette. Therefore, the file must contain an exact image of the contents of the diskette according to requirements for the output. If the image file is created by the COPY function, the diskette image file is automatically the correct size. If you attempt to generate a diskette with a file that is too large, a message is displayed when the diskette is filled. You cannot continue the diskette generation on another volume.

FLOPYDUP first requests you to enter the file parameters of the disk file that contains the diskette image. FLOPYDUP then requests you to enter the name of the output NL diskette and the device number of the drive on which it is mounted. If you have not mounted the output diskette, you can mount the output diskette through FLOPYDUP. Any diskette already mounted in the specified drive is dismounted automatically. When the diskette is generated, a message informs you of the status of the operation, and allows you to end or rerun FLOPYDUP processing. It also allows you to create another diskette with the same contents. Press PF1 to restart FLOPYDUP processing. Press PF16 to terminate processing. To generate another duplicate diskette press PF2. The original diskette image file remains unchanged.

9.6 A SAMPLE FLOPYDUP PROCEDURE

You can control FLOPYDUP processing through the VS Procedure language. You can specify all FLOPYDUP options through a procedure. However, because the VS Procedure language MOUNT and DISMOUNT statements cannot be embedded in a RUN, ENTER (or DISPLAY) sequence, all Procedure language MOUNT or DISMOUNT operations must be performed prior to or following automated FLOPYDUP processing. This restriction particularly affects the DUPLICATE function, which requires that the output diskette be mounted during the function's processing. A complete list of FLOPYDUP GETPARMs is given in Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

The following procedure copies the contents of a diskette to a diskette image file and then generates a diskette from the contents of the diskette image file. All mount specifications are supplied by the procedure; you need only place the diskettes in the drive when the mount message is displayed. The procedure exits and reenters FLOPYDUP each time a MOUNT operation is processed.

```
PROCEDURE
MOUNT DISK SLOPPY ON 039 WITH STANDARD LABEL FOR EXCLUSIVE USAGE
RUN FLOPYDUP
ENTER OPTIONS OPTION=C
ENTER INPUT VOLSER=SLOPPY, LABEL=SL, DEVICE=39
ENTER OUTPUT FILE=SLOPPY, LIBRARY=ZEBRALIB, VOLUME=SYSTEM
ENTER EOJ 16
MOUNT DISK FLOPPY ON 039 WITH NO LABEL FOR EXCLUSIVE USAGE
RUN FLOPYDUP
ENTER OPTIONS OPTION=G
ENTER INPUT FILE=SLOPPY, LIBRARY=ZEBRALIB, VOLUME=SYSTEM
ENTER OUTPUT VOLSER=FLOPPY, DEVICE=039
ENTER EOJ 16
DISMOUNT FLOPPY
RETURN
```

CHAPTER 10
THE FORMCNTL UTILITY

10.1 INTRODUCTION

The Forms Control utility, FORMCNTL, allows you to create and maintain a file of electronic vertical form control definitions for the VS serial printers listed in Table 10-1. A form control definition controls the format of a printing paper form. A form control definition consists of the following information:

- Identification number
- Form length
- Vertical line spacing
- Horizontal characters spacing (pitch)
- Vertical line skip (channel assignment)
- Top-of-form (channel assignment)
- Font selection (where applicable)

Printer Description	Supported Length of Form	Vertical Spacing (LPI)	Horizontal Spacing	Font ** Selection
o Model 6581W - Daisywheel Printer				
o Model 6581WC - Wide Carriage Daisywheel Printer				
o Model 5573 - 300 LPM Band Printer				
o Model 5574 - 600 LPM Band Printer				
o Model 2273V1 - 250 LPM Remote Band Printer				
o Model 5575 - Hi-Speed Band Printer				
o Model 2233 - Remote Matrix Printer, 100 LPM	6 - 144	6, 8	10, 12	--
o Model 2233K - Remote Katakana Matrix Printer, 100 LPM	6 - 144	6, 8	10, 12	--
o Model 2235 - Remote Matrix Printer, 180 LPM	6 - 144	6, 8	10, 12	--
o Model 2235K - Remote Katakana Matrix Printer, 180 LPM	6 - 144	6, 8	10, 12	--
o Model 2273V1 - Remote Band Printer, 250 LPM	6 - 126	6, 8		--
o Model 5533 - Matrix Printer, 100 LPM	6 - 144	6, 8	10, 12	01 (Primary) or 02 (Secondary
o Model 5535 - Matrix Printer, 180 LPM	6 - 144	6, 8	10, 12	01, 02
o Model 5573 - Band Printer, 300 LPM	6 - 126	6, 8		--
o Model 5574 - Band Printer, 600 LPM	6 - 126	6, 8		--
o Model 5575 - High Speed Band Printer	6 - 143	6, 8		--
o Model 5577 - High Speed Matrix Printer	6 - 144	6, 8	10, 12	01, 02
o Model 6581W - Daisywheel Printer, 40 CPS	6 - 144	6, 8, 3, 4	10, 12, 15	--
o Model 6581WC - Wide Carriage Daisywheel Printer, 40CPS	6 - 144	6, 8, 3, 4	10, 12, 15	--
o Model DWOS-20 - Daisywheel Printer, 20 CPS	6 - 144	6, 8, 3, 4	10, 12, 15	--
o Model DWR-20 - Remote Daisywheel Printer, 20 CPS	6 - 144	6, 8, 3, 4	10, 12, 15	--
o Model OK555 - Okidata Matrix Printer	6 - 144			--

The print Speed option is no longer supported.

** New Topic in Documentation

FORMCNTL processing involves the following steps. You cannot run the FORMCNTL utility through a procedure because it does not use GETPARM requests.

- Select the function to review or to add printer forms control definitions. You can also print a listing of existing printer forms control definitions.

- Define printer form length, vertical spacing, horizontal spacing, font selection, and vertical forms control channels when you add a printer form control definition to the forms definition file.

- Delete or modify a printer forms control definition while you review it.

10.2 SELECTING A FUNCTION

The first screen of the FORMCNTL utility is the FORMCNTL main menu. You can select one of the following PF key functions:

PF Key	Function
1	Review printer forms control definition
2	Add a new printer forms control definition to the system
13	Instructions
15	Print listing of printer forms control definitions
16	Terminate processing

10.3 ADDING A NEW FORMS CONTROL DEFINITION

The ADD function (selected by pressing PF2 from the main menu) adds new forms control definition information to the forms definition file (FORMDFFN). After you select the function, you specify a FORM# and PRINTER TYPE on the ADD Forms Control Definition screen, and press ENTER to display the Forms Control Definition screen.

The Forms Control Definition screen allows you to select the length of the form (in lines), the vertical spacing (optional), the horizontal spacing (optional), and the font selection.

An overview of FORMCNTL processing is provided in Figure 10-1.

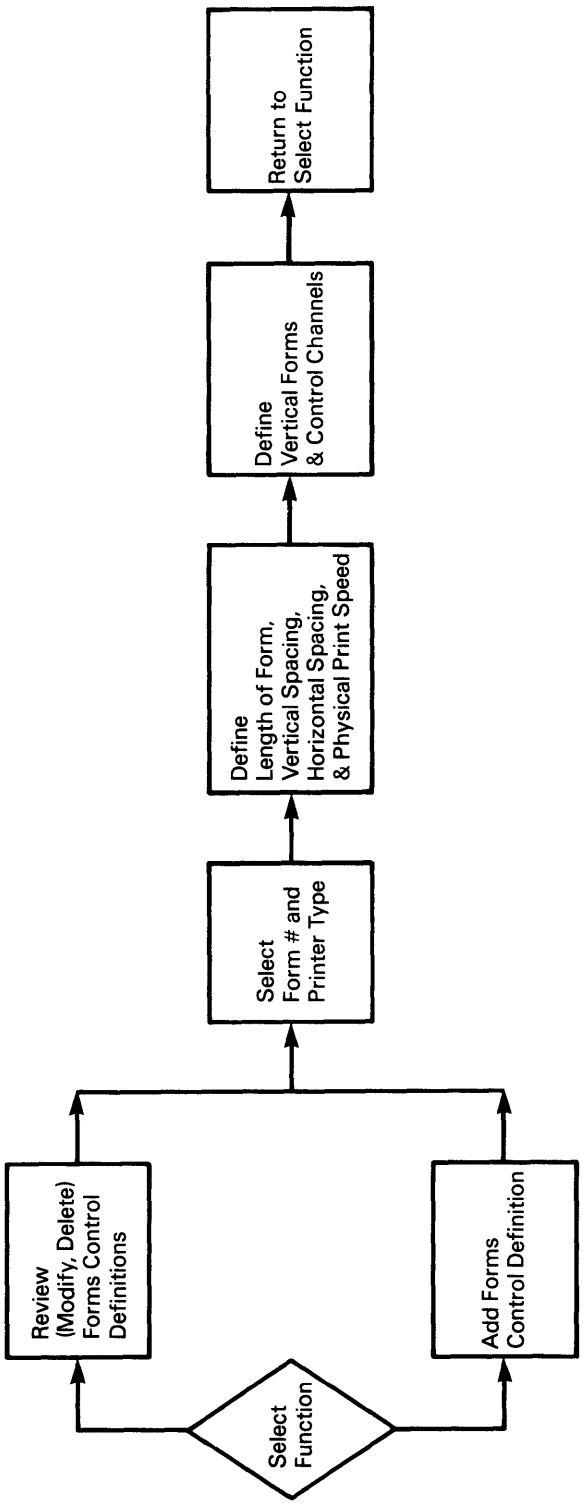


Figure 10-1. FORMCNTL Processing

You can specify the following on the Forms Control Definition screen (refer to Table 10-1 for the correct values):

LENGTH OF FORM	The number of lines that can be printed on each physical page.
VERTICAL SPACING (OPTIONAL)	The number of lines to be printed on each vertical inch of the form. Allowable values are dependent upon the printer type.
HORIZONTAL SPACING (OPTIONAL)	The number of characters to be printed per inch along a line (pitch).
FONT SELECTION	The options are primary and secondary fonts. The default response is primary. The character set that will be used when the form is printed.

You make the appropriate entries and press ENTER to process the Forms Control Definition screen. The Vertical Forms Control Channel Definition screen, discussed in Subsection 10.3.1, is then displayed.

10.3.1 Defining Vertical Forms Control Channels

The Vertical Forms Control Channel Definition screen enables you to select the line (or lines) where vertical tab commands are executed. (Refer to Figure 10-2.) Vertical formatting determines the number of lines to be printed and the amount of space to be left blank between lines of a page of any type of output.

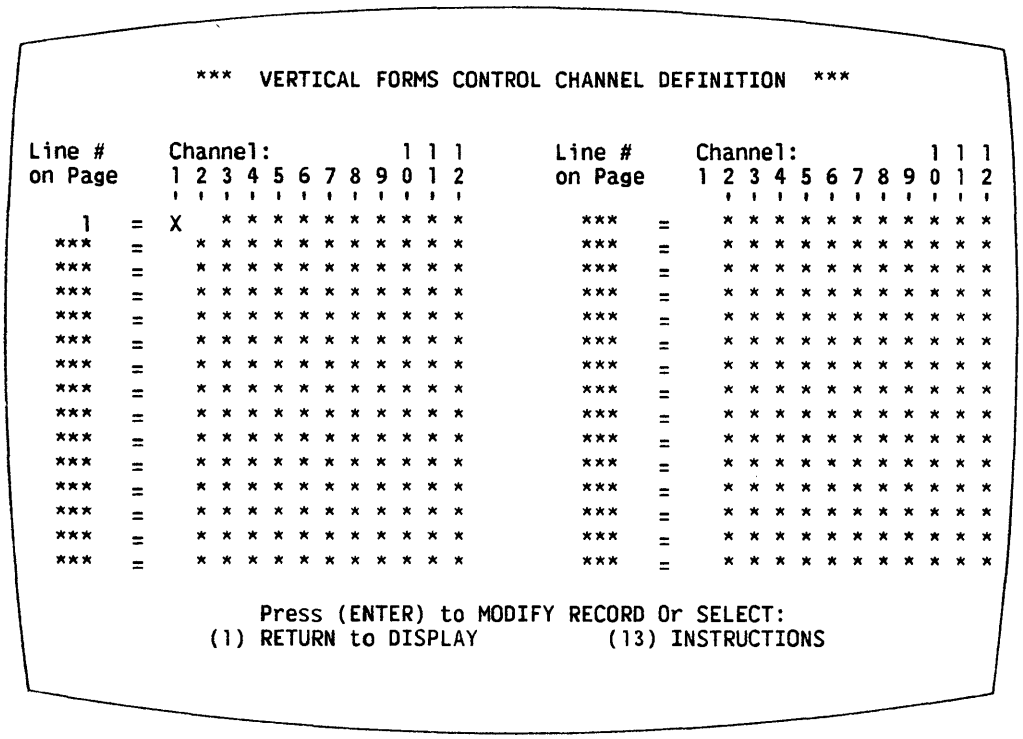


Figure 10-2. The Vertical Forms Control Channel Definition Screen

The following fields appear on the Vertical Forms Control Definition screen:

LINE# ON PAGE — The lines on the page for which a channel entry is to be made. Values of Line # on Page cannot exceed the value of Length of Form specified in the Forms Control Definition screen. Line numbers need not be in ascending order nor in any particular location on the screen.

Line 1 is always defined for Channel 1 (with an X) because this represents the Top-of-Form channel.

CHANNEL — The channel (or vertical tab) for each line number specified on this screen. A channel is defined by entering a nonblank character in the corresponding column. Channel 1 is always defined for Line 1 (with an X) to represent the Top-of-Form channel. Channel 1 cannot be defined on any other line.

10.4 REVIEWING A FORMS CONTROL DEFINITION

You can select the Review Forms Definition screen from the main menu by pressing PF1. This screen allows you to review and/or modify the forms definition records in FORMDFFN, the forms control definition file created by FORMCNTL. Form definition records in the file are identified by a concatenation of the FORM # and PRINTER TYPE. A request to review a Forms Definition will do one of the following:

- Display the existing definition.
- Display the same printer type with another FORM # (searching from 0 to 254).
- Report that no Forms Definitions are defined for this printer type.

For example, review the Forms Definitions starting at FORM #200 and Type 6581W. If that forms definition record does not exist, the system searches for the next combination of the printer number 6581W and form number (from 0 to 254); otherwise, an error message appears.

The form number must be in the range of 0 to 254, and the printer type must be a valid model name. If you do not make any entries, the first definition in the file is assumed. The utility indicates if the Forms Definition file is currently empty (as it is on system initialization). Use the ADD function to enter a forms definition.

The Review Forms Definition screen displays the length of the form, the vertical spacing value (optional), the horizontal spacing value (optional), and the physical print speed (Model 5575 only) of the Form# and printer type indicated on the main menu. The following options are available:

FIRST — Displays the first Forms Control Definition in the file.

NEXT — Positions to and displays the next sequential Forms Control Definition in the file. A message is displayed if the last definition in the file is currently being displayed.

FIND DEFINITION — Positions to and displays the specific Forms Control Definition of the Form# and Printer Type combination you enter.

MODIFY DEFINITION — Allows you to change the Forms Control Definition currently displayed.

DELETE DEFINITION — Allows you to remove the Forms Control Definition currently displayed. If you delete the last definition, then the display is positioned to the first definition. An appropriate message is diplayed when the last record (defn) is deleted and the first definition is displayed. If you delete all definitions, FORMCNTL returns to the main menu.

MENU — Returns to the main menu.

10.4.1 Modifying a Vertical Forms Control

When you press ENTER from the Forms Control Definition screen, a section of the Vertical Forms Control Channel Definition screen, described in Subsection 10.3.1, is displayed. The Vertical Forms Control Channel Definition screen corresponds to the particular Form# and Printer Type combination indicated on the main menu and Forms Control Definition screens.

An X in the channel column indicates that the channel is defined for that line. Note that Channel 1 is always defined for Line 1 and that Channel 1 cannot be defined for any other line.

MODIFY — Allows you to change the channel definition of the forms control definition currently displayed. When you press PF9, the entire Vertical Forms Control Channel Definition screen, discussed in Subsection 10.3.1, is displayed.

To change the forms control definition information, you must use the Forms Control Definition screen.

NOTE

When a VS system is delivered, a definition for Form 000 does not exist. The system default accommodates the 14 by 11 form size. If you do not request a special form, entries in the Print Queue are assigned Form 000. To establish a default Forms Definition for Form 000, you must execute FORMCNTL, and the system administrator must add a definition for Form 000, not modify an existing one.

CHAPTER 11
THE IBMCOPY UTILITY

11.1 INTRODUCTION

The IBMCOPY utility can transfer information between a Wang VS and any system using IBM format, soft-sectored diskettes. IBMCOPY copies a single file, selected files, or a complete volume from an IBM Data Exchange format diskette to VS disk storage, and copies a file, many files, or a library from VS disk storage to IBM Data Exchange format diskettes. IBMCOPY converts the files to and from the IBM and VS file formats. Optionally, IBMCOPY converts files containing only character data to and from the ASCII and EBCDIC character sets. Only IBMCOPY reads and writes IBM Data Exchange format diskettes. The structure of IBM Data Exchange format diskettes is discussed in Appendix C.

> NOTE
>
> You must use the TAPECOPY utility to perform IBM and VS tape transfers (refer to Chapter 15).

IBMCOPY accepts the following types of soft-sectored diskettes:

- Single-sided, single-density (SSSD) diskettes in the IBM Basic Data Exchange format (IBM Diskette 1, Wang Green Label)
- Double-sided, single-density (DSSD) diskettes in the IBM Basic Data Exchange format (IBM Diskette 2, Wang Red Label)
- Double-sided, double-density (DSDD) diskettes in the IBM Type H Data Exchange format (IBM Diskette 2D, Wang Red Label)

> NOTE
>
> You cannot use hard-sectored diskettes for input or output of IBM Data Exchange formatted data. Due to this restriction, IBMCOPY requires you to use a diskette drive that can access soft-sectored diskettes.

IBMCOPY supports the following functions:

- Mounts an IBM Data Exchange format diskette
- Initializes a soft-sectored diskette to an IBM Data Exchange format
- Displays or prints an IBM diskette directory
- Transfers data from an IBM Data Exchange format diskette to a VS disk
- Transfers data from a VS disk to an IBM Data Exchange format diskette
- Supports VS Procedure language control of IBMCOPY workstation interaction

An overview of IBMCOPY processing is provided in Figure 11-1.

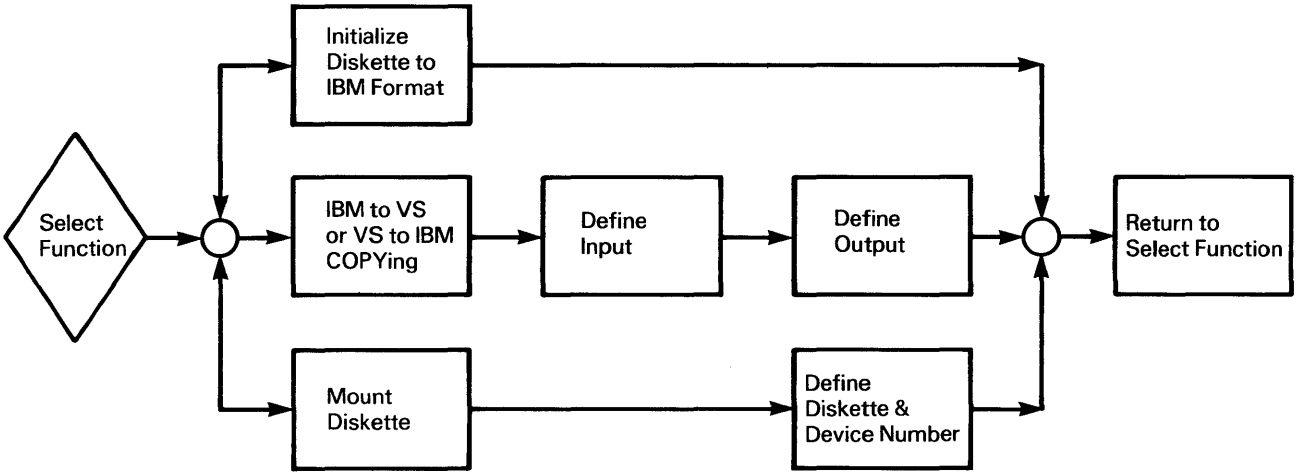


Figure 11-1. IBMCOPY Processing

When IBMCOPY processing begins, the main menu is displayed. From the main menu, you can select one of the following functions associated with the indicated PF keys:

PF Key	Function
1	Copy Files from an IBM Format Diskette to VS Disk Files
2	Copy VS Disk Files to an IBM Format Diskette
3	Initialize the Diskette using an IBM Format
4	Mount an IBM Format Diskette
5	Display an IBM Format Diskette Directory
16	End Processing

11.2 MOUNTING IBM DATA EXCHANGE FORMAT DISKETTES

The VS mount procedure (PF6 from the Command Processor) cannot mount IBM Data Exchange format diskettes. Thus, you must mount all IBM diskettes through IBMCOPY. IBMCOPY mounts diskettes either directly from the main menu or through a processing function when you specify the name of an unmounted IBM diskette as an input or output volume. You can dismount the IBM diskette through the Procedure language DISMOUNT statement; refer to Section 11.7 for details.

If you press PF4 from the main menu, a Mount Definition screen is displayed that requests you to enter the values of two parameters: VOLUME and DEVICE. When you have specified the volume name of an unmounted IBM diskette as input to another IBMCOPY function, the resulting Mount Definition screen requests only a value for DEVICE. In addition, the latter Mount Definition screen allows you to return to the previous screen by pressing PF1 to respecify the volume name. VOLUME and DEVICE are described as follows:

VOLUME — The name of the diskette to be mounted. You must specify a legal Wang volume name, regardless of the IBM volume name of the diskette. VOLUME defaults to blanks. However, if IBMCOPY has previously mounted an IBM diskette, VOLUME defaults to the name of the most recently mounted volume.

DEVICE — The device number of the soft-sectored diskette drive. If associated with an archiving workstation, the drive's device number is generally one greater than the workstation number. DEVICE defaults to blanks, or if you are running IBMCOPY at an archiving workstation, to the device number of the associated diskette drive. However, if IBMCOPY has previously mounted an IBM diskette, DEVICE defaults to the number of the most recently used diskette drive.

After you specify the parameters on the Mount Definition screen, mount the diskette. After you mount the diskette, control returns to the main menu or to the processing function, depending on which mode initiated the mount.

11.3 INITIALIZING IBM DATA EXCHANGE FORMAT DISKETTES

IBM Data Exchange format diskettes differ in format from VS soft-sectored diskettes. Thus, the DISKINIT utility cannot initialize an IBM diskette. However, IBMCOPY can initialize soft-sectored diskettes to IBM Data Exchange formats. IBMCOPY initializes the following types of diskettes to the indicated formats:

Diskette Type	Data Exchange Format	Bytes Per Sector
Single-sided, single density	Basic	128
Double-sided, single density	Basic	128
Double-sided, double-density	Type H	256

When you press PF3 from the main menu, the Define Volume screen is displayed. Specify the name of the diskette to be initialized on the Define Volume screen. The specified name also becomes the IBM volume name. If the specified volume is not mounted, IBMCOPY allows you to mount the diskette, as described in Section 11.2.

The drive detects whether the diskette is single- or double-sided; subsequent IBMCOPY processing differs for each diskette type.

11.3.1 Initializing IBM Single-Sided Diskettes

When the drive detects a single-sided diskette, IBMCOPY displays a screen that informs you that the diskette is single-sided and that the initialization process destroys any data on the diskette. You can press ENTER to acknowledge the message and continue the initialization or press PF1 to return to the Define Volume screen, without initializing or destroying the contents of the diskette.

If no further input is required, IBMCOPY initializes the diskette to the IBM Basic Data Exchange format. When the initialization is complete, the End-of-Job menu is displayed. The End-of-Job menu allows you to initialize another diskette by pressing PF1, to return to the main menu by pressing PF2, or to exit IBMCOPY by pressing PF16.

11.3.2 Initializing IBM Double-Sided Diskettes

When the drive detects a double-sided diskette, IBMCOPY displays a screen that informs you that the diskette is double-sided and that the initialization process destroys any data existing on the diskette.

IBMCOPY formats double-sided diskettes to single- or double- density; therefore, you also must select the desired format through the FORMAT parameter. If FORMAT is BASIC (the default response), the utility initializes the diskette as an IBM Basic Data Exchange format diskette in single-density. If FORMAT is H, the utility initializes the diskette as an IBM Type H Data Exchange format diskette in double-density. Alternatively, you can return to the Define Volume screen, without initializing or destroying the contents of the diskette, by pressing PF1.

If no further input is required, IBMCOPY initializes the diskette to the selected format. When the initialization is complete, the End-of-Job menu is displayed. The End-of-Job menu enables you either to initialize another diskette by pressing PF1, to return to the main menu by pressing PF2, or to exit IBMCOPY by pressing PF16.

11.4 DISPLAYING OR PRINTING AN IBM FORMAT DISKETTE DIRECTORY

IBMCOPY allows you to display or print the contents of an IBM diskette directory when you press PF5 from the main menu. While IBMCOPY can only transfer IBM Basic Data Exchange and Type H files, it can read the directory of any IBM diskette and can determine whether or not it can transfer the resident files.

When you press PF5 from the main menu, IBMCOPY requests you to identify the IBM volume and the desired type of directory report through the following parameters:

VOLUME — The name of the IBM volume for which IBMCOPY should generate a directory report. VOLUME defaults to blanks, or to the name of the volume last specified through this function of IBMCOPY. If the specified diskette is not mounted, IBMCOPY allows you to mount the diskette (refer to Section 11.2).

DEVICE — Determines whether IBMCOPY displays the directory report at the workstation or creates a print file. A printed report contains directory information for all fields not described by IBM as reserved or padded. A displayed report also contains all the volume information, but contains only those fields from the HDR1 records that are usually of interest. Note that you can print the full report at any time while displaying the report. A response of DISPLAY specifies a displayed directory report; a response of PRINTER creates a print file. DEVICE defaults to DISPLAY, or to the value last specified through this function of IBMCOPY.

IBMCOPY also contains a hidden GETPARM, identified by the prname SPECIAL, that allows you to include deleted HDR1 records in the directory report. The screen is not displayed because the deleted records are usually only required to analyze a problem. The deleted HDR1 records are included if DELETED = YES, and omitted if you accept the default response of NO. The following procedure allows you to display the hidden GETPARM.

```
PROCEDURE
RUN IBMCOPY
DISPLAY SPECIAL
```

11.4.1 Displaying an IBM Format Diskette Directory

When you specify DEVICE = DISPLAY, IBMCOPY generates the directory report and displays the Directory Display menu, which provides the following options:

PF Key	Function
1	Select another diskette
2	Display Volume Label and Error Map
3	Display Data Set Labels
15	Print all information
16	Return to the main menu

When you press PF2 from the Directory Display menu, IBMCOPY displays volume label and error map directory information. The report contains the Volume ID, System Code, Owner ID, Volume Surface, Sector Length, Volume Access, Label Extension, Special Requirements, Extent Arrangement, and Sector Sequence fields. The report also indicates whether the volume contains any defective cylinders or alternative record relocation. Refer to Subsection 11.4.3 for details on each field in the directory. From the Volume Label and Error Map screen, pressing PF1 returns control to the Directory Display menu; pressing PF15 creates a print file containing the entire directory report.

When you press PF3 from the Directory Display menu, IBMCOPY displays directory information for each data set on the volume. The report contains the Label Address (CCHHS), Dataset Name, Exchange Type, Multiple Volume Indicator, Volume Number, Block Length, Sector Length, Record Length, Beginning-of-Extent (BOE) Address, and End-of-Extent (EOE) Address fields, and indicates whether or not IBMCOPY can read the data set. Refer to Subsection 11.4.3 for details on each
field in the directory. If the diskette contains more data sets than can be described on one screen, you can view other screens by pressing the following PF keys.

PF Key	Result
2	Display the first screen
3	Display the last screen
4	Display the previous screen
5	Display the next screen
6	Move the display down one line
7	Move the display up one line

From the Data Set Label screen, pressing PF1 returns control to the Directory Display menu; pressing PF15 creates a print file containing the entire directory report.

The Directory Display menu also allows you to generate a directory report for a different IBM diskette by pressing PF1, create a complete printed report by pressing PF15, or return to the main menu by pressing PF16.

11.4.2 Printing an IBM Format Diskette Directory

When you specify DEVICE = PRINTER, IBMCOPY creates a print file that contains a complete directory report for the IBM diskette. Each field in the report is described in Subsection 11.4.3. The print file name is constructed by concatenating a unique, 4-digit number to IBMC, and the file resides in your default print library (SPOOLIB) on your default print volume (SPOOLVOL). The print file is printed according to your print mode defaults. Refer to the VS Programmer's Introduction for details about the effect of the default PRNTMODE value on print files.

11.4.3 Analyzing the Resulting Directory Report

IBMCOPY directory reports contain volume information and data set information. Volume information consists of fields from the IBM diskette's VOL1 and ERMAP records, and data set information consists of fields from the diskette's HDR1 records. Printed reports contain all of the available directory information; displayed reports omit less useful fields in the HDR1 records. Each field in each section is described as follows:

Volume Information

The following fields are contained in both displayed and printed reports, and reside in the diskette's VOL1 record:

Field	Description
Volume ID	The name of the volume when the diskette was initialized.
Volume Access	A character indicating whether or not more qualifications are necessary to access the diskette.
System Code	The name of the system that initialized the diskette. Diskettes initialized through IBMCOPY contain WANGVS in this field.
Owner ID	The name of the owner of the diskette.
Label Extension	The number of additional cylinders reserved for the diskette directory. IBMCOPY can only copy data from diskettes with a space in this field.
Volume Surface	The type of diskette as recorded in the diskette directory. This field can contain a value of DSDD (double-sided, double-density), DSSD (double-sided, single-density), or SSSD (single-sided, single-density).
Extent Arrangement	Indicates whether or not data sets (and labels) must be contiguous on the diskette.
Special Requirements	Indicates whether or not special requirements must be met to access data on this volume. IBMCOPY can only copy data from the diskette if this field contains a space.

Field	Description
Sector Length	The length of all sectors on the diskette on cylinders 1 through 76. A value in parentheses indicates that the diskette contains an invalid value. Because IBMCOPY can only read Basic Data Exchange and Type H diskettes, it can only read diskettes with 128 or 256 recorded in this field.
Sector Sequence	A number that determines the logical sequence of the physical sectors on the diskette. IBMCOPY requires a value of spaces, 0, or 1 to read files on the diskette.

ERMAP Fields

The ERMAP record indicates whether the diskette has any defective cylinders or any physical record relocation. If the diskette has defective cylinders, the directory report lists them in the format "Defective cylinders: xx yy." If the diskette contains physical record relocation, the directory report indicates the defective record method and lists the cylinder, head, and sector addresses of the defective records.

HDR1 Fields

The following fields are contained in the displayed and/or printed reports, and reside in the diskette's HDR1 records:

Field	Description
Label CCHHS Address	The cylinder, head, and sector address of the data set's entry in the label.
Dataset Name	The IBM name of the data set.
Block Length	The length of a block in the data set.
Record Attribute	Specifies whether or not blocking is used within the data set. This field must be blank for diskettes with the Exchange Type Indicator equal to H or a blank; thus, IBMCOPY can only copy data from diskettes with a blank in this field. The directory report contains an S if the records are spanned and a B if the records are blocked. This field is not included in a displayed directory report.
BOE CCHHSS Address	The cylinder, head, and sector address of the Beginning-of-Extent (BOE) pointer.
Sector Length	The physical record length (128, 256, 512, or 1024 bytes).
EOE CCHHSS Address	The cylinder, head, and sector address of the End-of-Extent (EOE) pointer.

Field	Description
R F	The record/block format that indicates fixed-length records residing in fixed-length blocks. The field can contain a blank or an F. This field must be blank if the Exchange Type Indicator is an H or a blank; thus, IBMCOPY can only copy data from diskettes with a blank in this field. This field is not included in a displayed directory report.
B I	The Bypass Indicator that determines if the data set is to be skipped in copy operations on IBM systems; IBMCOPY does not use this field. If the field is blank, the data set can be transferred; if the field contains a B, the data set is skipped when copying the diskette on an IBM system. This field is not included in a displayed directory report.
D S	The degree of data set security on IBM systems; IBMCOPY does not use this field. This field is not included in a displayed directory report.
W P	Indicates whether or not the data set can be written. If the field is blank, the data set can be both read or written. If the field contains a value, the data set can only be read. This field is not included in a displayed directory report.
Exchange Type	The type of diskette. IBMCOPY can read only diskettes in the Basic Data Exchange (BDE) or Type H (H) formats.
Multiple Volume	Indicates whether the data set spans more than one volume. , A blank indicates the data set resides only on this volume. A C indicates that the data set is continued to another diskette; an L reflects the last diskette in a series.
Volume Number	If the data set spans more than one volume, the sequence number of the current volume.
Creation Date	The date that the data set was created. This field is not included in a displayed directory report.
Record Length	The length of the records in the data set.
Offset to Next Record Space	For blocked records, indicates the starting position of the next record space in the data set's last sector (EOD). This field is not included in a displayed directory report.
Expiration Date	Indicates when the data set can be deleted or written over. This field is not included in a displayed directory report.

Field	Description
Verify/Copy Indicator	Indicates whether the data set has been verified (V) or copied (C). This field is not included in a displayed directory report.
Data Set Organization	Indicates whether or not defective records can be sequentially relocated. This field is not included in a displayed directory report.
EOD (End-of-Data) Address	Identifies the address of the next unused sector within the data set extent. If the address is the same as the BOE address, the data set contains no records. This field is not included in a displayed directory report.
Readable	Indicates whether or not IBMCOPY can read the data set.

11.5 COPYING DATA FROM AN IBM DISKETTE TO A VS DISK

IBMCOPY transfers IBM files to VS disk storage when you select PF1 from the main menu. IBMCOPY automatically converts the IBM files to the VS file format and, at your request, converts the IBM data from EBCDIC to ASCII. You can transfer a file, selected files, or all files on the IBM diskette. You can also automatically transfer the contents of several IBM diskettes into a single VS file through separate IBMCOPY transfers.

IBMCOPY displays the IBM to VS Input Definition screen when you select IBM to VS copying. The IBM to VS Input Definition screen allows you to identify the input IBM diskette and to select the input range through the following parameters. Alternatively, you can return to the main menu by pressing PF1.

COPY — The range of the copying operation. You can copy a single file, selected files, or an entire volume. The range is determined by one of the following responses. The default response is VOLUME.

- FILE — Copies a single file from the IBM diskette to VS disk storage. Specify the name of the IBM file in the FILE parameter on this screen. Subsection 11.5.1 discusses subsequent processing for a single file copy.
- SELECT — Copies only those files on the IBM diskette that you select. Subsection 11.5.2 describes subsequent processing for selected file copying.
- VOLUME — Copies the entire IBM diskette contents. Each file on the diskette is copied to an equivalent file on the VS. Subsection 11.5.3 presents subsequent processing for volume copying.

VOLUME — The name of the input IBM diskette. VOLUME defaults to blanks or to the name of the volume last specified through this function of IBMCOPY. If the specified IBM diskette is not mounted, IBMCOPY allows you to mount the diskette, (refer to Section 11.2).

FILE — The name of the file to be copied for single file transactions. Leave this parameter blank when the input range spans more than one file.

11.5.1 Copying a Single File from an IBM Diskette to a VS Disk

After you select the single file input range, and after you supply the file and volume names of the IBM file, the Single File Output Definition screen is displayed. The Single File Output Definition screen allows you to specify the VS file parameters, file organization, and storage characteristics for the output file. In addition, you can select whether or not the input file is translated from EBCDIC to ASCII. The Single File Output Definition screen requests this information through the following parameters:

FILE — The VS file name of the output file. Valid VS file names include any alphanumeric value of up to eight characters long. The default response is the IBM name of the file.

LIBRARY — The VS library name of the output file. Valid VS library names include any alphanumeric value of up to eight characters long.

VOLUME — The VS volume name of the output file. Valid VS volume names include any alphanumeric value of up to six characters long.

TRANSL — Determines whether or not the file is to be translated from EBCDIC to ASCII. If TRANSL = YES, the file is translated. If TRANSL=NO, the file remains in the EBCDIC character set. The default response is YES.

NOTE

Set TRANSL to NO for files containing numeric data because IBMCOPY performs only character translation. Files containing numeric data can be translated through the TRANSL utility after IBMCOPY has converted the files to the VS format (refer to Chapter 17).

RECORDS — The number of records to be allocated for the output file. The default response is the number of records in the IBM input file. If you intend to copy the data from several IBM diskettes into a single VS file, you should specify the number of records required for the entire file and set RELEASE = NO.

RELEASE — Determines whether or not allocated but unused space is to be released after the file is created. The default response, YES, releases the unused space; the unused space is not released if RELEASE = NO. RELEASE = NO is appropriate for cases where data from several IBM diskettes is to be transferred into a single VS file.

FILEORG — The file organization of the output VS file. If FILEORG is C, the output file remains in consecutive organization. (All files on an IBM Data Exchange format diskette are stored as consecutive files.) If FILEORG is I, IBMCOPY converts the IBM file to a VS indexed file. The default response is C. If you choose to create an indexed file, you can specify only a primary key. Also, you must specify the KEYLEN, KEYPOS, IPACK, and DPACK parameters (described in this list) to define the index of the file.

COMPRESS — Determines whether or not the output VS file is stored using data compression to conserve storage space. Refer to the VS Operating System Services Reference for details on data compression. If COMPRESS is YES, the output file is compressed. If COMPRESS is NO, data compression is not used. The default response is YES.

FILECLAS — The VS file class of the output file. Valid file classes are A to Z, @, #, $, and blank. Refer to the VS Programmer's Introduction for details on file classes. FILECLAS defaults to your default file class, set through PF2 of the Command Processor or the VS Procedure language SET statement.

KEYLEN — The length in bytes of the field within the file to be used as the primary key. The key length can be any value from 1 to the length of the record, but cannot exceed 255 bytes. KEYLEN defaults to blanks. You must specify KEYLEN if FILEORG is I.

KEYPOS — The starting byte position of the field to be used as the key field within the input file. Valid key position can be any byte from byte 1 to the last byte of the record. KEYPOS defaults to blanks. You must specify KEYPOS if FILEORG is I.

IPACK — The packing density in which the index blocks are stored on the output disk. The default response is 100. Refer to the VS Operating System Services Reference for further information on packing densities. You must specify IPACK if FILEORG is I.

DPACK — The packing density in which the data blocks are stored on the output disk. The default response is 100. Refer to the VS Operating System Services Reference for further information on packing densities. You must specify DPACK if FILEORG is I.

After you define the output file parameters, the IBM file is transferred to the VS. If the specified file parameters for the output file represent an existing file, you can rename the output file, scratch the existing file by pressing PF3, or append the current data to the existing file by pressing PF5. Note that if you append data to an existing VS file, the file attributes specified in the current operation must match those of the existing file.

When the copy completes, IBMCOPY displays the End-of-Job menu. From the End-of-Job menu, you can copy another IBM file by pressing PF1, return to the main menu by pressing PF2, or exit IBMCOPY by pressing PF16. You can then dismount the IBM diskette through the Command Processor.

11.5.2 Copying Selected Files from an IBM Diskette to a VS Disk

After you specify the selected files input range and identify the input IBM diskette, the File Selection screen is displayed. The File Selection screen displays a listing of all files on the IBM diskette and indicates how many files the diskette contains. If more than 20 files exist on the diskette, the file listing continues to an additional screen(s). You indicate that a file is copied by entering any nonblank character in the field associated with the file name. Pressing ENTER displays any subsequent File Selection screen. If all screens are displayed, pressing ENTER ends input definition and continues processing. You can return to the IBM to VS Input Definition screen by pressing PF1. For example, if you complete file selection on the second screen out of six File Selection screens, you can skip the subsequent screens and continue processing by pressing PF16.

When you select the files to be copied, the Selected Files Output Definition screen is displayed. The Selected Files Output Definition screen requires you to define the output library and volume locations of the selected files and to indicate whether or not the files are to be translated from EBCDIC to ASCII. The Selected Files Output Definition screen requests this information through the following parameters:

LIBRARY — The name of the VS library into which the files are to be copied. The default response is your default output library (OUTLIB). If this is a rerun of the IBM to VS copy sequence, the default response is the previously entered value.

VOLUME — The name of the VS volume onto which the files are to be copied. The default response is your default output volume (OUTVOL). If this is a rerun of the IBM to VS copy sequence, the default response is the previously entered value.

TRANSL — Determines whether or not the files are transferred from EBCDIC to ASCII. If TRANSL is YES, the files are translated. If TRANSL is NO, the files remain in the EBCDIC character set. The default response is YES.

NOTE

Set TRANSL to NO for files containing numeric data because IBMCOPY performs only character translation. Files containing numeric data can be translated through the TRANSL utility after IBMCOPY has converted the files to the VS format (refer to Chapter 17).

While the Selected Files Output Definition screen determines the output options for all the selected files, it does not allow you to change file organization, file location, or storage characteristics for individual files. However, if you press PF2 from the Selected Files Output Definition screen instead of ENTER, you can specify the output options individually for each file. If you press ENTER, all selected files are copied with the output options specified on the Selected Files Output Definition screen; each selected IBM file is transferred to the specified library and volume location with the IBM file name used as the VS file name.

If you press PF2 from the Selected Files Output Definition screen, IBMCOPY displays a Single File Output Definition screen for each selected file. The Single File Output Definition screens are identical to the screen used for single file copying, described in Subsection 11.5.1. Thus, you can change the file, library, and volume names of the output file, as well as the file organization and storage characteristics. After you specify the individual output file requirements, the file is copied. IBMCOPY then displays a Single File Output Definition screen for the next file on the diskette to be copied.

If the output file parameters for an output file represent an existing file, you can rename the output file, scratch the existing file by pressing PF3, or append the current data to the existing file by pressing PF5. Note that if you append data to an existing VS file, the file attributes specified in the current operation must match those of the existing file.

When all files have been copied, the End-of-Job menu is displayed. The End-of-Job menu enables you to copy more IBM data to the VS by pressing PF1, return to the main menu by pressing PF2, or exit IBMCOPY by pressing PF16. You dismount the IBM diskette through the Command Processor.

11.5.3 Copying an IBM Diskette to a VS Disk

After you specify the volume input range and identify the input IBM diskette, the Volume Output Definition screen is displayed. The Volume Output Definition screen is identical to the Selected Files Output Definition screen. The Volume Output Definition screen allows you to specify library and volume locations and the translation option for all the files on the volume, or to process each file individually to select more specific output options. Refer to Subsection 11.4.2 for a discussion of the Selected Files Output Definition screen and individual file output specification.

If the output file parameters for an output file represent an existing file, you can rename the output file, scratch the existing file by pressing PF3, or append the current data to the existing file by pressing PF5. Note that if you append data to an existing VS file, the file attributes specified in the current operation must match those of the existing file.

When all the files on the volume have been copied, the End-of-Job menu is displayed. The End-of-Job menu enables you to copy more IBM data to the VS by pressing PF1, return to the main menu by pressing PF2, or exit IBMCOPY by pressing PF16. You dismount the IBM diskette through the Command Processor.

11.6 COPYING DATA FROM A VS DISK TO AN IBM DISKETTE

IBMCOPY transfers VS disk files to an IBM diskette when you press PF2 from the main menu. IBMCOPY automatically converts the VS files to the IBM file format and, at your request, can convert the VS files from ASCII to EBCDIC. Because IBM Data Exchange format diskettes can only contain consecutive files, IBMCOPY converts VS indexed files to IBM consecutive files. You can transfer a single file, selected files in a library, or an entire VS library to the IBM diskette.

IBMCOPY displays the VS to IBM Input Definition screen when you select VS to IBM copying. The VS to IBM Input Definition screen allows you to identify the range and locations of VS files to be copied through the following parameters. Alternatively, you can return to the main menu by pressing PF1.

COPY — The range of the copying operation. A single file, selected files in a library, or an entire VS library can be copied. The range is determined by one of the following responses. The default response is LIBRARY.

- FILE — Copies a single file from the VS to the output IBM diskette. You must also supply the file, library, and volume names of the file through the FILE, LIBRARY, and VOLUME parameters on this screen. Subsection 11.6.1 describes subsequent processing for single file copying.

- SELECT — Copies only those files in the VS library that you select. You identify the VS library from which files are to be selected, through the LIBRARY and VOLUME parameters on this screen. Subsection 11.6.2 describes subsequent processing for selected file copying.

- LIBRARY — Copies the entire VS library to the output IBM diskette. Each file in the library is copied to an equivalent file on the output IBM diskette. You must identify the VS library through the LIBRARY and VOLUME parameters on the screen. Subsection 11.6.3 describes subsequent processing for volume copying.

VOLUME — The name of the VS disk volume on which the input VS file(s) resides. The default response is your default input volume (INVOL).

LIBRARY — The name of the VS library to be copied or that contains the file(s) to be copied. The default response is your default input library (INLIB).

FILE — The name of the file to be copied. Omit this parameter when the input range spans more than one file.

11.6.1 Copying a Single File from a VS Disk to an IBM Diskette

After you select a single file copy and identify the input file on the VS to IBM Input Definition screen, you specify the output file, along with copy options, on the Single File Output Definition screen. Because all IBM files on IBM Data Exchange format diskettes have fixed-length records, the Single File Output Definition screen allows you to select the pad character used to convert VS variable-length records. You can also select ASCII to EBCDIC character set conversion. You select the output file location and copy options through the following parameters:

FILE — The name of the output IBM file. You can supply an alphanumeric value of up to eight characters.

VOLUME — The name of the output IBM diskette. If the specified IBM diskette is not mounted, IBMCOPY allows you to mount the diskette, as described in Section 11.2. If the specified IBM diskette is not initialized, IBMCOPY allows you to initialize the volume without returning to the main menu. No default response is required unless you rerun the VS to IBM copy sequence, when VOLUME defaults to the previously entered value.

PADCHAR — The character used to extend variable-length VS records to fixed-length IBM records. Pad characters are hexadecimal zeroes or blanks; PADCHAR defaults to blanks.

TRANSL — Determines whether or not the file is translated from ASCII to EBCDIC. If TRANSL is YES, the file is translated. If TRANSL is NO, the file remains in the ASCII character set. The default response is YES.

NOTE

Set TRANSL to NO for files containing numeric data because IBMCOPY performs only character translation. Files containing numeric data can be translated through the TRANSL utility (refer to Chapter 17).

After you specify the output options, the VS file is copied to the IBM diskette. However, if the output IBM diskette already contains files, you must indicate whether the utility adds the file to the diskette or scratches all other files on the diskette before copying the new file. You select one of these options through the NEWINDEX parameter on the screen displayed after the Single File Output Definition screen. If NEWINDEX is NO (the default response), the file is added to the current diskette contents. If NEWINDEX is YES, any files already residing on the diskette are erased before the file is copied. Alternatively, you can return to the main menu by pressing PF16.

If the specified output file name is the same as a file already on the diskette, IBMCOPY signals the error and allows you to respecify the file, delete the file on the diskette by pressing PF3, or terminate the copy by pressing PF16.

After you copy the file, IBMCOPY displays the End-of-Job menu. From the End-of-Job menu, you can copy more VS data by pressing PF1, return to the main menu screen by pressing PF2, or exit IBMCOPY by pressing PF16. You can then dismount the IBM diskette through the Command Processor.

11.6.2 Copying Selected Files from a VS Disk to an IBM Diskette

If you choose to copy only selected files from a specifed VS library, IBMCOPY allows you to select the files to be copied on the File Selection screen. The File Selection screen displays a listing of all files in the VS library and indicates the number of files in the library. If more than 20 files reside in the library, the file listing continues on an additional screen(s). You indicate that a file is copied by entering a nonblank character in the field associated with the file name. To display any subsequent File Selection screen, press ENTER. If all File Selection screens are displayed, press ENTER to end input definition and to continue processing. Alternatively, you can return to the IBM Input Definition screen by pressing PF1. If, for example, you complete file selection on the second screen out of six File Selection screens, you can skip screens three through six and continue processing by pressing PF16.

When you select the files to be copied, the Selected Files Output Definition screen is displayed. The Selected Files Output Definition screen allows you to identify the output IBM diskette, select character set conversion, and determine the character used to extend VS variable-length records to IBM fixed-length records. Thus, the Selected Files Output Definition screen is similar to the Single File Output Definition screen (discussed in Subsection 11.6.1), except that the file name is not defined; the VS file name for each selected input file is used as the output IBM file name.

If you wish to specify the file name and/or copy options separately for each file, a Single File Output Definition screen is displayed for each file. To display the files separately, press PF2 from the Selected Files Output Definition screen. The output options on the Selected Files Output Definition screen display as default values on each Single File Output Definition screen, but can be overridden.

After you specify the output options, IBMCOPY copies all the selected files to the IBM diskette. However, if the output IBM diskette already contains files, you must indicate whether the utility adds the files to the diskette or scratches all other files on the diskette before copying the files. You select one of these options through the NEWINDEX parameter on the screen that is displayed following the Selected Files Output Definition screen. If you define the output options separately for each selected file, the screen is displayed following the first Single File Output Definition screen. If NEWINDEX is NO (the default response), the files are added to the current diskette contents. If NEWINDEX is YES, any files already residing on the diskette are deleted before the files are copied. Alternatively, you can return to the main menu by pressing PF16.

If sufficient space is not available on the diskette to hold all the selected files, IBMCOPY allows you either to skip copying the current file by pressing PF2, to continue the copy (making a multivolume file) by pressing PF3, or to mount a new diskette that can hold the file by specifying the name of the new diskette and pressing ENTER. For a multivolume file copy when one diskette is full, IBMCOPY requests you to mount a new IBM diskette.

If any output file name duplicates a file name already on the diskette, IBMCOPY signals the error and allows you either to respecify the file, to scratch the file on the diskette by pressing PF3, or to terminate the copy by pressing PF16.

After you copy the files, IBMCOPY displays the End-of-Job menu. From the End-of-Job menu, you can copy more VS data by pressing PF1, return to the main menu by pressing PF2, or exit IBMCOPY by pressing PF16. Then, you dismount the IBM diskette through the Command Processor.

11.6.3 Copying a Library from a VS Disk to an IBM Diskette

After you select a library copy and identify the input library on the VS to IBM Input Definition screen, specify the output diskette, along with the copy options, on the Library Output Definition screen. The Library Output Definition screen allows you to identify the output IBM diskette, to select character set conversion, and to determine the character used to extend VS variable-length records to IBM fixed-length records. Thus, the Library Output Definition screen is similar to the Single File Output Definition screen (discussed in Subsection 11.6.1), except that the file name is not defined. The VS file name for each file in the library becomes the output IBM file name.

If you wish to specify the file name and/or copy options separately for each file, a Single File Output Definition screen is displayed for each file in the library. To display each file in the library, press PF2 from the Library Output Definition screen. The output options you select on the Library Output Definition screen are displayed as default values on each Single File Output Definition screen; however, the default values can be overridden.

If the output IBM diskette already contains files, you must indicate whether to add the library's files to the diskette or to delete all files already on the diskette before copying the library. You select this option through the NEWINDEX parameter on a separate screen. This screen appears after the Library Output Definition screen. If you define the output options separately for each file, the screen appears after the first Single File Output Definition screen. If NEWINDEX is NO (the default response), the files are added to the current diskette contents. If NEWINDEX is YES, any files already on the diskette are deleted before the files are copied. Alternatively, you can return to the main menu by pressing PF16.

If sufficient space is not available on the diskette to hold all the files in the library, IBMCOPY allows you either to skip copying the current file by pressing PF2, to continue the copy (making a multivolume file) by pressing PF3, or to mount a new diskette that can hold the entire file by specifying the name of the new diskette and pressing ENTER. When one diskette is full for a multivolume copy, IBMCOPY requests you to mount a new IBM diskette.

If any input file name duplicates a file name already on the diskette, IBMCOPY signals the error and allows you to respecify the file, to delete the file on the diskette by pressing PF3, or to terminate the copy by pressing PF16.

After you copy the files, IBMCOPY displays the End-of-Job menu. From the End-of-Job menu, you can copy more data by pressing PF1, return to the main menu screen by pressing PF2, or exit IBMCOPY by pressing PF16. You can then dismount the IBM diskette through the Command Processor.

11.7 A SAMPLE IBMCOPY PROCEDURE

You can control IBMCOPY processing through the VS Procedure language. You can specify all IBMCOPY options and mount operations through a procedure. A complete list of IBMCOPY GETPARMs is provided in Appendix A, consult the VS Procedure Language Reference for details concerning VS Procedure syntax.

The following procedure mounts an IBM diskette and copies all files on the IBM diskette into a new library on the VS SYSTEM volume. Once the files are copied, the procedure exits IBMCOPY and dismounts the IBM volume. Note that the procedure dismounts the volume after IBMCOPY processing terminates because VS Procedure language DISMOUNT statements cannot be embedded in a RUN, ENTER, or DISPLAY sequence. The IBMCOPY mount routine can be embedded in a procedure since the mount routine does not use the VS Procedure language MOUNT statement. The procedure totally automates IBMCOPY processing; you only physically mount and dismount the IBM diskette.

```
PROCEDURE
RUN IBMCOPY
ENTER FUNCTION 4
ENTER MOUNTCOM DEVICE=039, VOLUME=IBMVOL
ENTER FUNCTION 1
ENTER OPTIONS COPY=VOLUME, VOLUME=IBMVOL
ENTER OUTPUT LIBRARY=Z00LIB, VOLUME=SYSTEM, TRANSL=NO
ENTER EOJ 16
DISMOUNT DISK IBMVOL
RETURN
```

CHAPTER 12
THE IOELOG UTILITY

12.1 INTRODUCTION

The IOELOG utility allows you to examine the contents of an I/O error log file. You can use this utility to evaluate the history and condition of equipment configured to a system. IOELOG enables you to review the types of errors that occur and how often they occur.

The I/O error log resides in the @SYSLOG file in the System Work library (@SYSWORK) on the System volume. The error log file is always present, even if it contains no errors. The VS Operating System records each occurrence of an error and produces an error log entry that contains the date and the time that each error occurred. You cannot use the active error log file as input by IOELOG. To analyze or print the error log, you must first copy it into a separate file from the Operator Console menu. You can purge the error log file from the same menu. If you use the purge option, the system creates a new error log file.

IOELOG provides the following functions:

- You can analyze on your workstation screen the entire I/O error log or portions of it. If you choose to analyze the error log, IOELOG summarizes the error log contents into the following categories:
  - Standard Errors (errors logged against any of the five device classes)
  - Nonstandard Errors (logged errors not specifically related to one of the five device classes)
  - System Initial Program Loads (IPLs)
- You can print the entire contents of the I/O error log or portions of it. The printout lists errors sequentially by the date and the time they occurred. They are not sorted by the type of error.

Standard errors are logged against five system devices: communications, disk drives, printers, tape drives, and workstations. Standard errors occur as a result of a Start I/O (SIO) command or a Control I/O (CIO) command. SIO commands initiate data transfer or a control operation. WRITE data to a disk is an example of a SIO command that transfers data. The SEEK command to a disk is an example of a SIO command that controls the operation of an I/O device.

The second type of Standard I/O error results from a CIO command. CIO commands have two primary purposes:

- To control memory diagnostics for I/O processors
- To control microcode reading, microcode loading, and programmable processor control

An example of a memory diagnostic command is REFERENCE, which reads specified memory in the I/O processor and compares what was read to a given data pattern. Examples of microcode reading and loading, and programmable processor control are RESTART Bus Processor and START Data Link Processor.

Both types of Standard I/O errors are classified as hard or soft errors. A soft error occurs when a command is successfully completed after one or more failures: that is, the error is recoverable. A hard error occurs when a command is not successfully completed after a number of attempts: that is, the error is not recoverable.

Nonstandard I/O errors include missing interrupts (delayed device response), lost extents to the free extent list in the VTOC on a disk (reported by the Supervisory Calls (SVCs) SCRATCH and UPDATFDR), page-in/page-out errors, redundant VTOC compromises (an I/O command to the VTOC failed due to a hard disk error), and machine check errors.

The error log also includes an IPL summary, listing the dates and times of the system IPLs.

IOELOG provides you with access to an on-line help facility for information about the utility. By pressing PF13, you can enter the help facility from most of the IOELOG screens. Pressing PF13 from a given screen brings you to the on-line help text section for that screen. Pressing PF1 from within the help facility returns you to IOELOG.

An overview of IOELOG processing is provided in Figure 12-1.

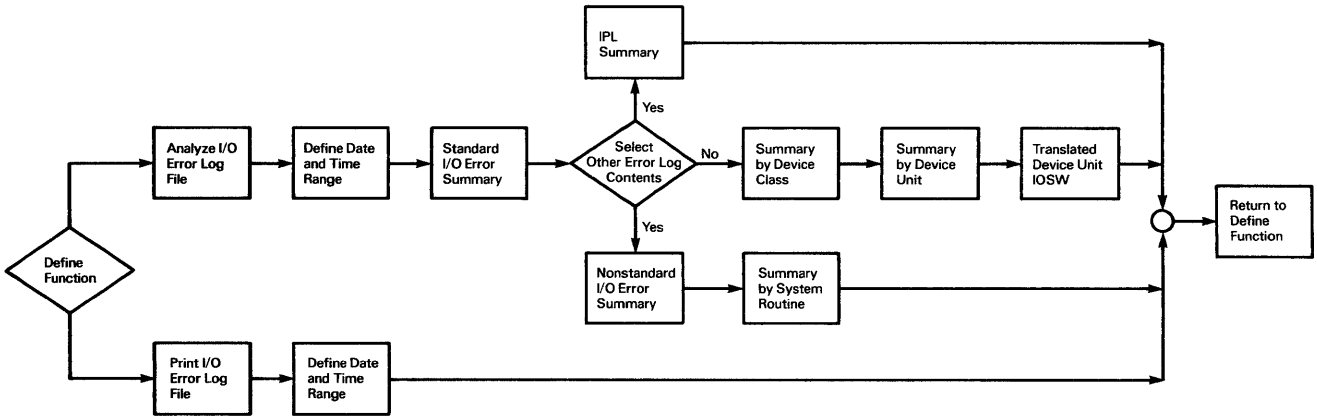


Figure 12-1. IOELOG Processing

12.2 COPYING THE ERROR LOG

To copy an I/O error log file from the Operator Console menu, press PF14 to display the Systems Options screen. From this screen, press PF9 to copy the I/O error log. The Copy I/O Error Log screen is displayed, which asks you to specify the names of the file, library, and volume to which the I/O error log is to be copied. The Copy I/O Error Log screen is shown in Figure 12-2.

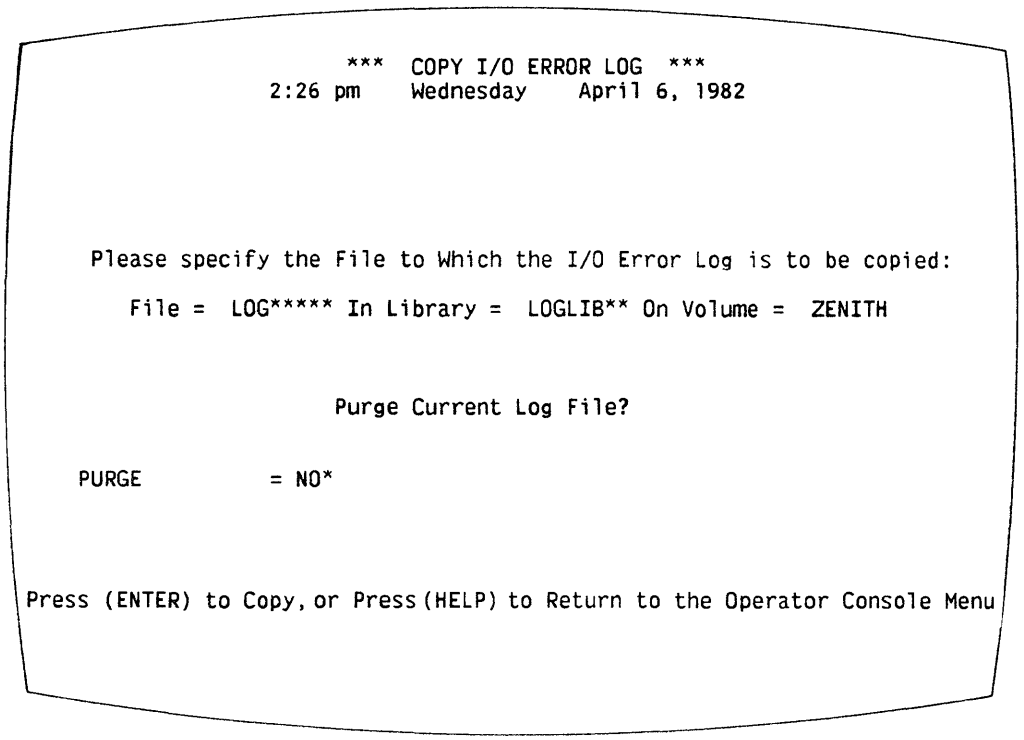


Figure 12-2. The Copy I/O Error Log Screen

If you press ENTER to copy the error log file, a message appears at the top of the screen stating whether or not the copy is successful. When the copy is complete, press HELP to return to the Operator Console menu.

12.3 RUNNING IOELOG

The Function Definition screen is the initial screen of the IOELOG utility, shown in Figure 12-3.

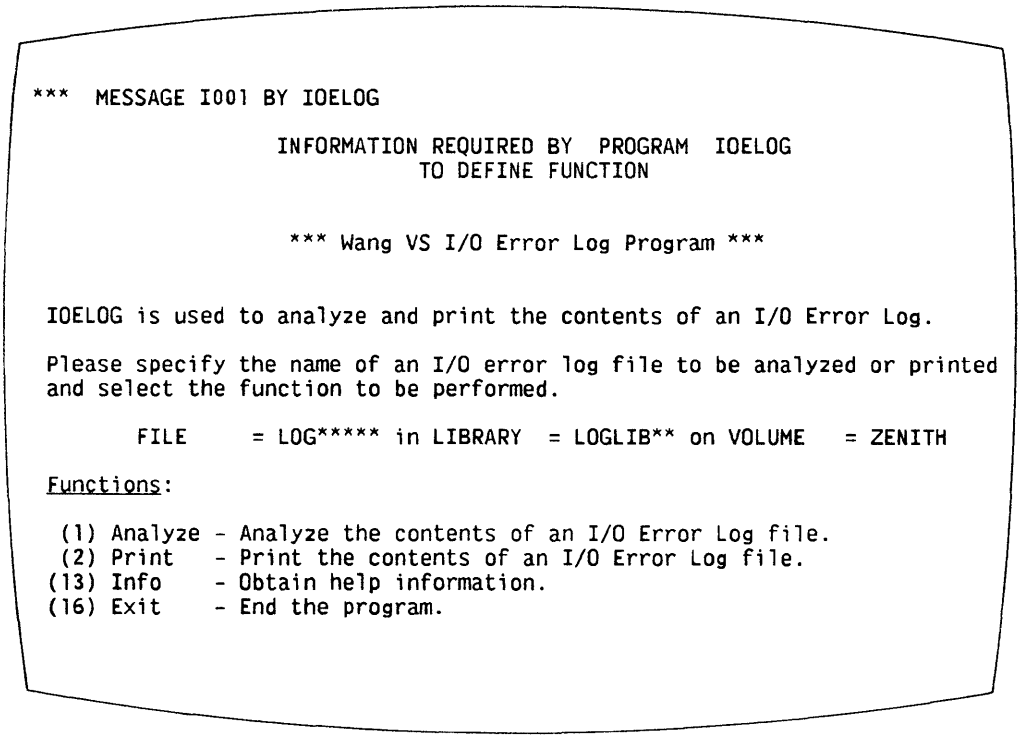


Figure 12-3. The Function Definition Screen

Use the Function Definition screen to identify the file location through the following parameters:

FILE — The name of the I/O error log file you wish to analyze or print.

LIBRARY — The name of the input library in which the copy of the error log file resides. LIBRARY defaults to your input library (INLIB) that you set previously through SET Usage Constants (PF2) of the Command Processor or a procedure.

VOLUME — The name of the input volume that contains the input library. VOLUME defaults to your input volume (INVOL) that you set previously through SET Usage Constants of the Command Processor or a procedure.

The Function Definition screen provides the following PF key options:

PF Key	Function
1	Analyzes the contents of an I/O error log file.
2	Prints the contents of an I/O error log file.
13	Provides you with on-line help information.
16	Exits the IOELOG utility.

After you select a function, processing continues.

12.4 ANALYZING AN I/O ERROR LOG FILE

To analyze an I/O error log on-line, enter the log file, library, and volume names and press PF1 from the Function Definition screen. The Range Definition screen displays the name and the location of the error log file to be analyzed, as shown in Figure 12-4. The date and time range of current error log entries is also displayed, including the number of records in the range.

```
***  MESSAGE I002 BY IOELOG

                INFORMATION REQUIRED BY  PROGRAM  IOELOG
                           TO DEFINE RANGE


                 *** Wang VS I/O Error Log Program ***


                              Analysis Option

File to be analyzed is:      LOG      in LOGLIB   on ZENITH
Date and time range is:      9/07/82  @ 20:14     to   9/08/82  @ 13:11
# records in range is:        68

Please specify the desired time range of the file to be analyzed:

              STARTDAY = *9/07/82       MM/DD/YY
              STARTIME = 20:14          HH:MM  (24 hour clock)
              ENDDAY   = *9/08/82       MM/DD/YY
              ENDTIME  = 13:11          HH:MM  (24 hour clock)

            Press PF1 to return to the function selection menu.
```

Figure 12-4. A Sample Range Definition Screen

IOELOG provides parameters that default to the date and time range of the error log. You can modify these parameters if you want to look at only a portion of the error log. The modifiable parameters include the following:

STARTDAY — The starting date used in error log analysis, in the form MM/DD/YY. The default value is the date of the first entry in the error log.

STARTIME — The starting time used in error log analysis, in the form HH:MM as based on a 24-hour clock. The default value is the time of the first entry in the error log.

ENDDAY — The end date used in error log analysis, in the form MM/DD/YY. The default value is the date of the last entry in the error log.

ENDTIME — The end time used in error log analysis, in the form HH:MM as based on a 24-hour clock. The default value is the time of the last entry in the error log.

You must use the format provided for STARTDAY and ENDDAY as it appears on the screen. If you use another format, a blinking field and an error message appear. In addition, if you enter values greater than 23:59 in the STARTIME or ENDTIME fields, the field blinks and an error message appears.

Press PF1 to return to the Function Definition screen, or ENTER to continue processing. If you press ENTER, the Standard I/O Error Summary screen is displayed. This screen contains a summary of Standard I/O errors logged against specific device classes.

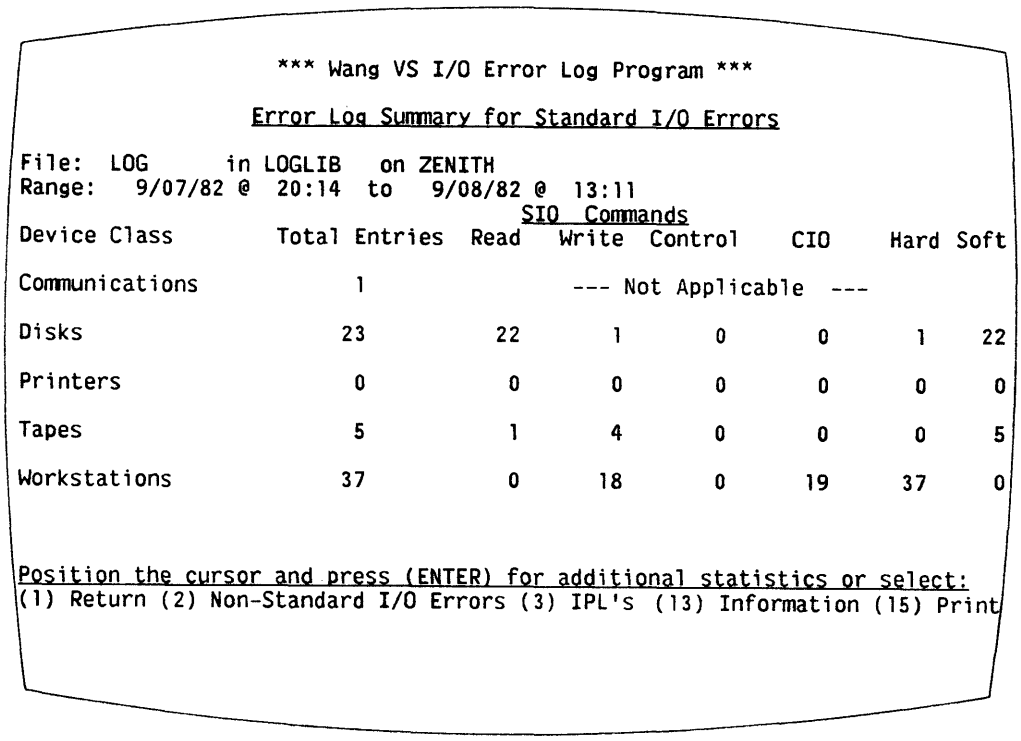
12.4.1 The Standard I/O Error Summary

The Standard I/O Error Summary screen displays the number of errors that are logged against each of the five device classes during the specified time range: communications, disk drives, printers, tape drives, or workstations. The total number of Standard I/O errors logged against each device class is given. This total is divided into two categories: errors that occurred as a result of an SIO or CIO command, and those errors that are classified as hard or soft.

The total number of CIO commands for each device appears in one column. SIO command errors are further divided into three categories: read, write, and control. Two separate columns list the total number of hard and soft errors that occurred for each device class.

The name and location of the error log file appear at the beginning of the summary, as well as the specified time range of the error log.

The Standard I/O Error Summary is illustrated in Figure 12-5.



*** Wang VS I/O Error Log Program ***

Error Log Summary for Standard I/O Errors

File: LOG in LOGLIB on ZENITH
Range: 9/07/82 @ 20:14 to 9/08/82 @ 13:11

Device Class	Total Entries	SIO Commands Read	Write	Control	CIO	Hard	Soft
Communications	1	--- Not Applicable ---					
Disks	23	22	1	0	0	1	22
Printers	0	0	0	0	0	0	0
Tapes	5	1	4	0	0	0	5
Workstations	37	0	18	0	19	37	0

Position the cursor and press (ENTER) for additional statistics or select:
(1) Return (2) Non-Standard I/O Errors (3) IPL's (13) Information (15) Print

Figure 12-5. A Sample Standard I/O Error Summary Screen

The Standard I/O errors contained in the log are categorized as follows:

Field Name	Refers to
SIO	Read: The total number of errors that occurred while reading data, as a result of an SIO command. Write: The total number of errors that occurred while writing data, as a result of an SIO command. Control: The total number of errors that occurred during a control operation such as a SEEK or a FORMAT on a disk, as a result of an SIO command. No transfer of data was involved.
CIO	The total number of errors that occurred while initiating diagnostic functions, loading or reading microcode, or performing processor control functions.
Hard	The total number of unrecoverable errors that occurred for a given device class.
Soft	The total number of recoverable errors that occurred for a given device class.

Not Applicable appears in the Communications device class because the errors cannot always be broken down into the categories listed. Only the total number of errors is given.

The Standard I/O Error Summary screen provides the following PF key options:

PF Key	Function
ENTER	Position the cursor next to a selected device class and press ENTER. An error summary is displayed that lists the total number of Standard I/O errors that occurred for a particular device class.
1	Returns to the Range Definition screen.
2	Displays the Nonstandard I/O Errors Summary, which lists I/O errors that are found in the error log and not reported against one of the five device classes. (Refer to Subsection 12.4.2.)
3	Displays the IPL Summary, which lists the date and time of each IPL of the system. (Refer to Subsection 12.4.3.)
13	Provides on-line help information.
15	Prints the screen contents.

Position the cursor next to any device class in the Standard I/O Error Summary screen and press ENTER. An error summary for the selected device class is displayed. The device class error summary lists all of the errors reported for that device, and allows you to analyze the errors for a particular device class in greater detail.

This chapter is limited to examples of error summary screens for disk devices. When you are analyzing disks or any of the four other device classes, you can press PF13 to access on-line help information. The help facility provides you with information relevant to the type of summary screen that is displayed.

To analyze errors logged against the disk device, position the cursor next to Disks and press ENTER. The Disk Summary screen is displayed. The Disk Summary screen includes the total number of errors that occurred for a given disk drive. As described in the Standard I/O Error Summary, the screen also displays the three categories of SIO command errors, CIO command errors, and the total number of hard and soft errors that occurred for each disk unit. The errors are listed by disk unit number. The device name, capacity, and type are also listed for each disk unit.

The date and time range appears at the beginning of the error log. Figure 12-6 illustrates a sample Disk Summary screen.

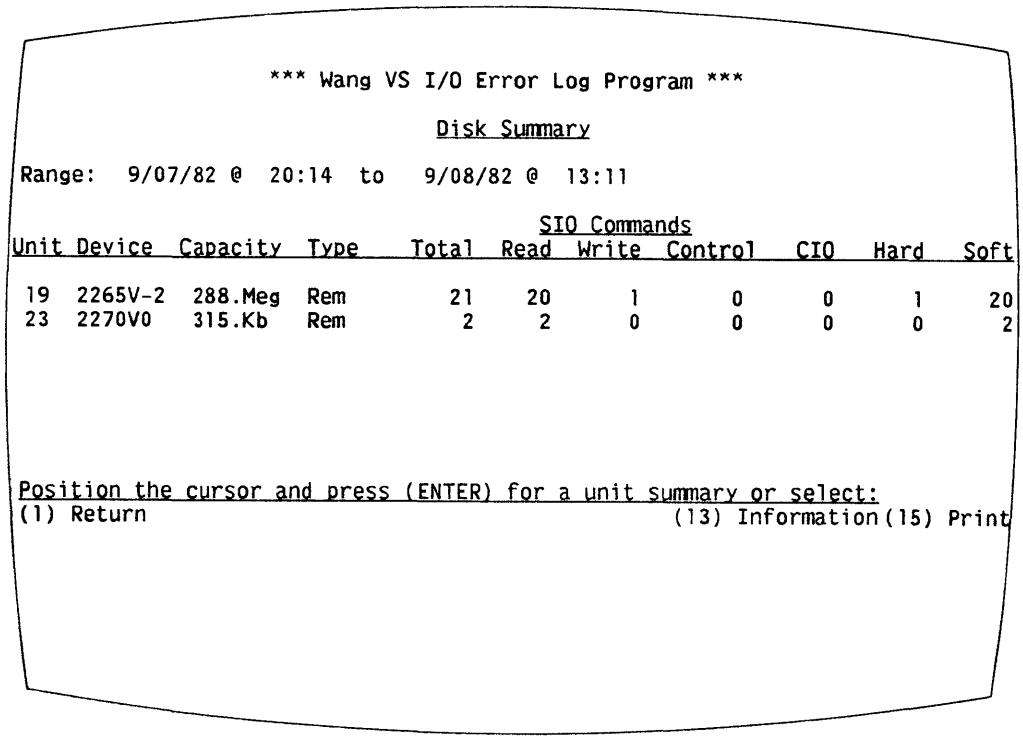


Figure 12-6. A Sample Disk Summary Screen

The Disk Summary screen displays the following fields:

Field Name	Refers to
Unit	The unit number of the disk drive.
Device	The device name that the system recognizes.
Capacity	The amount of disk storage available on that disk.
Type	The type of disk packs that can be mounted on the disk drive. The types displayed include Fixed (Fix), Removable (Rem), and Fixed/Removable (Fix/Rem).
Total	The total number of errors recorded by the log for a given disk drive.
SIO	Read: The total number of errors that occurred while reading data, as a result of an SIO command. Write: The total number of errors that occurred while writing data, as a result of an SIO command.

Field Name	Refers to
	Control: The total number of errors that occurred during a control operation, such as a SEEK or a FORMAT on a disk, as a result of an SIO command. No transfer of data was involved.
CIO	The total number of errors that occurred while initiating diagnostic functions, loading or reading microcode, or performing processor control functions.
Hard	The total number of unrecoverable errors that occurred for a given device class.
Soft	The total number of recoverable errors that occurred for a given device class.

The Disk Summary screen provides the following PF key options:

PF Key	Function
ENTER	Position the cursor next to a particular device unit number. An error summary for that device unit is displayed, providing an analysis of the types of Standard I/O errors that occurred for a particular disk unit.
1	Returns to the Standard I/O Error Summary screen.
2 (First)	Displays the first screen of errors, if more than one screen is required.
3 (Last)	Displays the last screen of errors, if the list requires more than one screen.
4 (Prev)	Displays the previous list of errors, if more than one screen is required.
5 (Next)	Displays the next list of errors, if the list requires more than one screen.
13	Provides on-line help information.
15	Prints the screen contents.

If you want to analyze errors logged against a particular disk unit, position the cursor next to a unit number in the Disk Summary screen and press ENTER. The Disk Unit Summary screen is displayed, listing all the errors for that unit, allowing you to analyze further the types of errors recorded.

Figure 12-7 illustrates a sample Disk Unit Summary screen.

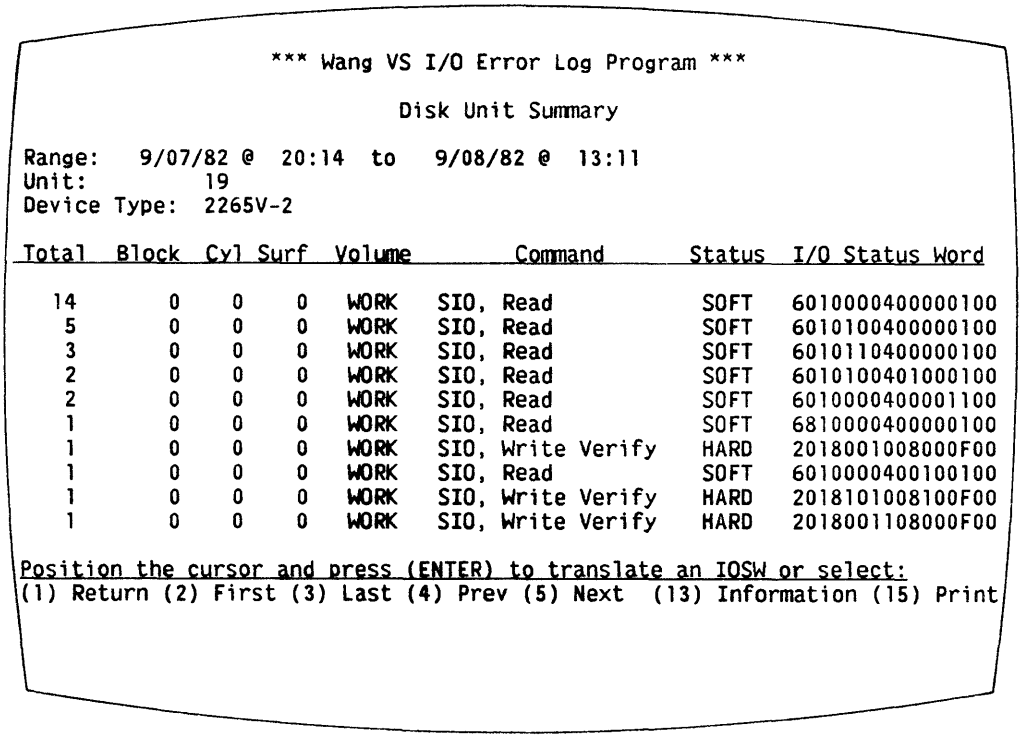


Figure 12-7. A Sample Disk Unit Summary Screen

The Disk Unit Summary screen categorizes the total number of errors that occurred for a given disk unit by listing the number of errors unique to a particular block, cylinder, surface, command, status, and I/O Status Word (IOSW). The IOSW is a hexadecimal representation of the status a device returns in response to an I/O command that the Operating System issues.

The following fields in the error log are displayed:

Field Name	Refers to
Total	The total number of errors detected for the device.
Block	The block number within the surface number (in decimal form) where an error occurred.
Cyl	The cylinder number (in decimal form) within the disk where an error occurred.
Surf	The surface number (in decimal form) within the cylinder where an error occurred.
Volume	The name (up to 6 characters) of the disk where an error occurred.

Field Name	Refers to
Command	The disk command, translated from the I/O Control Word (IOCW), that the device was executing at the time that an error occurred.
Status	The type of error detected: hard if the error is nonrecoverable, soft if it is recoverable.
IOSW	The hexadecimal representation of the status of a I/O command (IOCW) that the Operating System issues to a device.

The Disk Unit Summary screen provides the following PF key options:

PF Key	Function
ENTER	Continues processing and displays the Translated Disk IOSW screen, which provides you with the types of errors that occurred during an I/O command from the Operating System to a disk.
1	Returns to the Disk Summary screen.
2 (First)	Displays the first screen of errors, if more than one screen is required.
3 (Last)	Displays the last screen of error types, if the list requires more than one screen.
4 (Prev)	Displays the previous list of error types, if more than one screen is required.
5 (Next)	Displays the next list of error types, if the list requires more than one screen.
13	Provides on-line help information.
15	Prints the screen contents.

To translate an IOSW for a particular error type, position the cursor on the line describing the error and press ENTER. The Translated Disk IOSW screen is then displayed. This screen provides you with an analysis of the error conditions and retry indicators given in the IOSW. A sample Translated Disk IOSW is shown in Figure 12-8. The range of the log you are examining, the disk unit number, and the device type are displayed at the top of the screen.

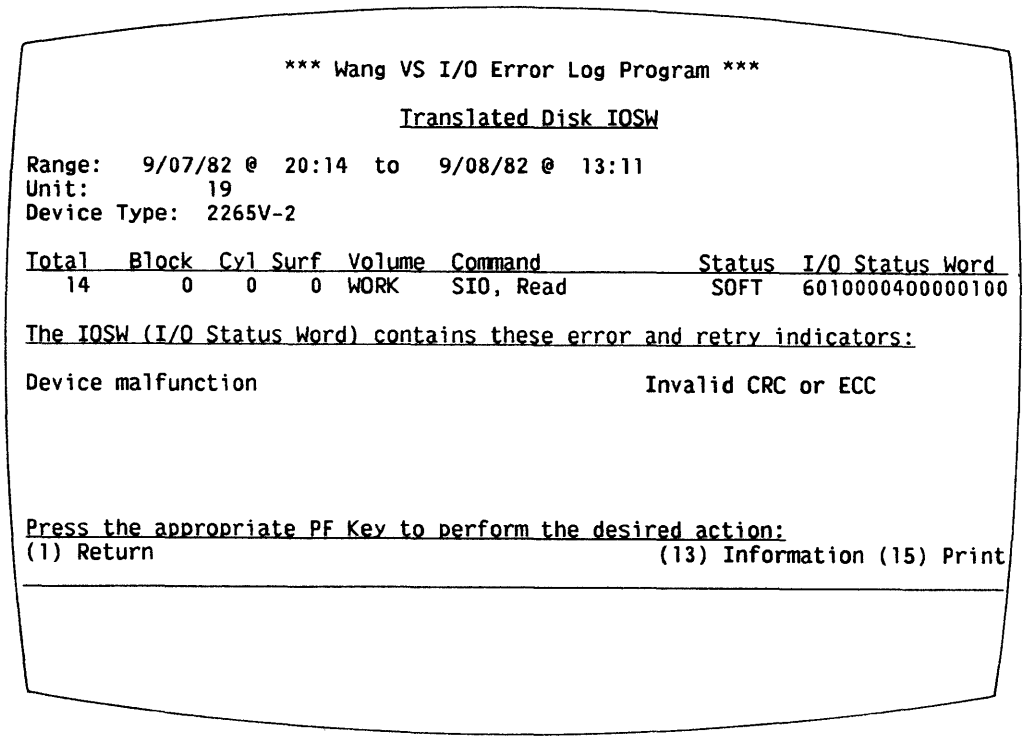


Figure 12-8. A Sample Translated IOSW Screen

The total number of errors that occurred for the error type you choose to examine is given. As in the Disk Unit Summary, the block, cylinder and surface and volume name on which an error occurred (if any) are listed, as well as the command type, error status, and IOSW in hexadecimal form. A breakdown of the error conditions and retry indicators is also given.

The Translated Disk IOSW screen provides you with the following PF key options:

PF Key	Function
1	Returns to the Disk Unit Summary screen.
13	Provides on-line help information.
15	Prints the screen contents.

12.4.2 The Nonstandard I/O Error Summary

You can analyze Nonstandard I/O errors by pressing PF2 from the Standard I/O Error Summary screen. The Nonstandard I/O Error Summary screen is displayed containing any errors recorded in the log that are not logged against one of the five device classes. The log file name and location are displayed at the top of the screen, as well as the date and time range of the error log you are analyzing.

A sample Nonstandard I/O Error Summary screen is shown in Figure 12-9.

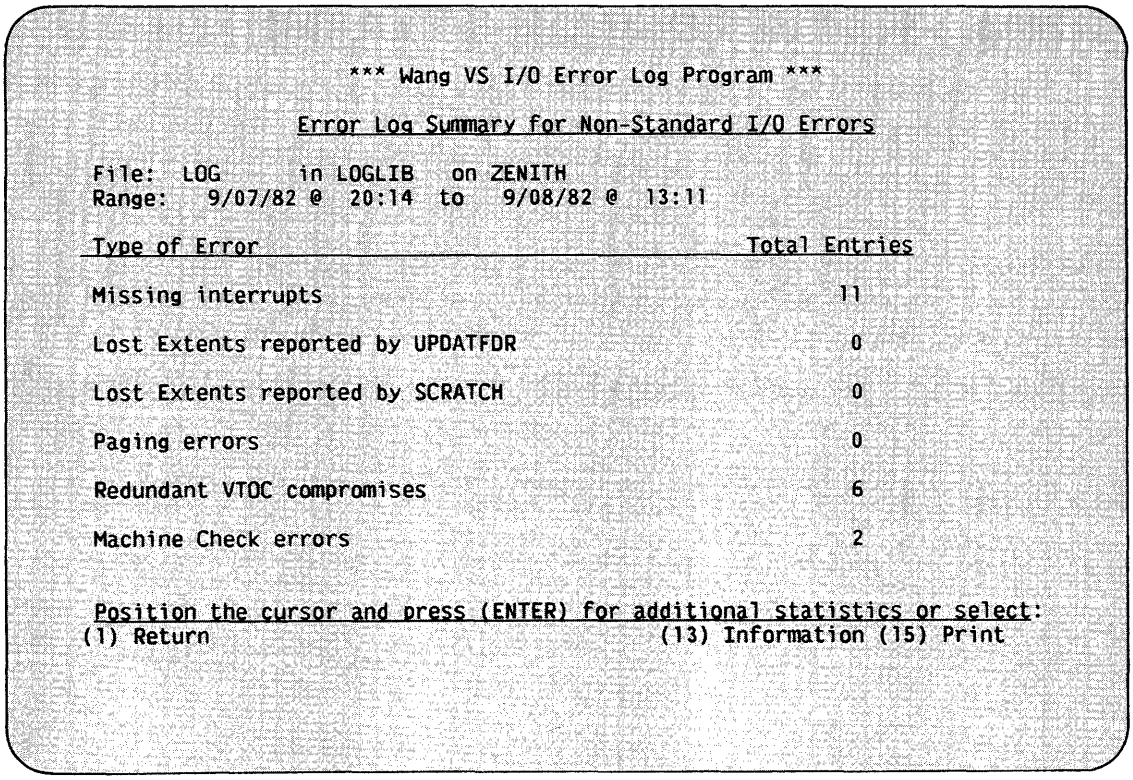


Figure 12-9. Sample Nonstandard I/O Error Summary Screen

The following describes the types of errors included in the Nonstandard I/O Error Summary screen:

Error	Meaning
Missing Interrupts	A device took longer to respond to a command than it should have.
Lost Extents Reported by UPDATFDR	The UPDATFDR Supervisory Call (SVC) cannot return an extent to the free extent list in the VTOC. This message usually does not indicate a disk error, but rather a problem with space in the VTOC.
Lost Extents Reported by SCRATCH	The SCRATCH SVC cannot return an extent to the free extent list in the VTOC. This message usually does not indicate a disk error, but rather a problem with space in the VTOC.
Paging Errors	An I/O command to the paging file failed due to a hard disk error. A paging file is a disk image of a task's memory.
Redundant VTOC Compromises	An I/O command to the VTOC failed due to a hard disk error.

Error	Meaning
Machine Check Errors	A memory parity error or an IOP error occurred. A memory parity error occurs when data is transmitted incorrectly. If an error that involves one bit occurs, the microcode detects the error and makes corrections. If the error involves two or more bits, the microcode detects but cannot correct the error. IOELOG reports memory parity errors that involve two or more bits. An IOP error occurs when an IOP cannot move data into or out of the devices needed to process the data.

The Nonstandard I/O Error Summary screen provides the following PF key options:

PF Key	Function
ENTER	Continues processing and displays a summary of errors that occurred for a particular system routine. Position the cursor next to any Nonstandard error before pressing ENTER.
1	Returns to the Standard I/O Error Summary screen.
13	Provides on-line help information.
15	Prints the screen contents.

You can analyze each Nonstandard error further by positioning the cursor next to an error and pressing ENTER. An error summary of the selected Nonstandard error is displayed. Figure 12-10 illustrates a sample Missing Interrupts Summary screen. The Missing Interrupts Summary screen provides you with the error type and the device on which the error occurred. The date and time range of the error log appear at the top of the screen.

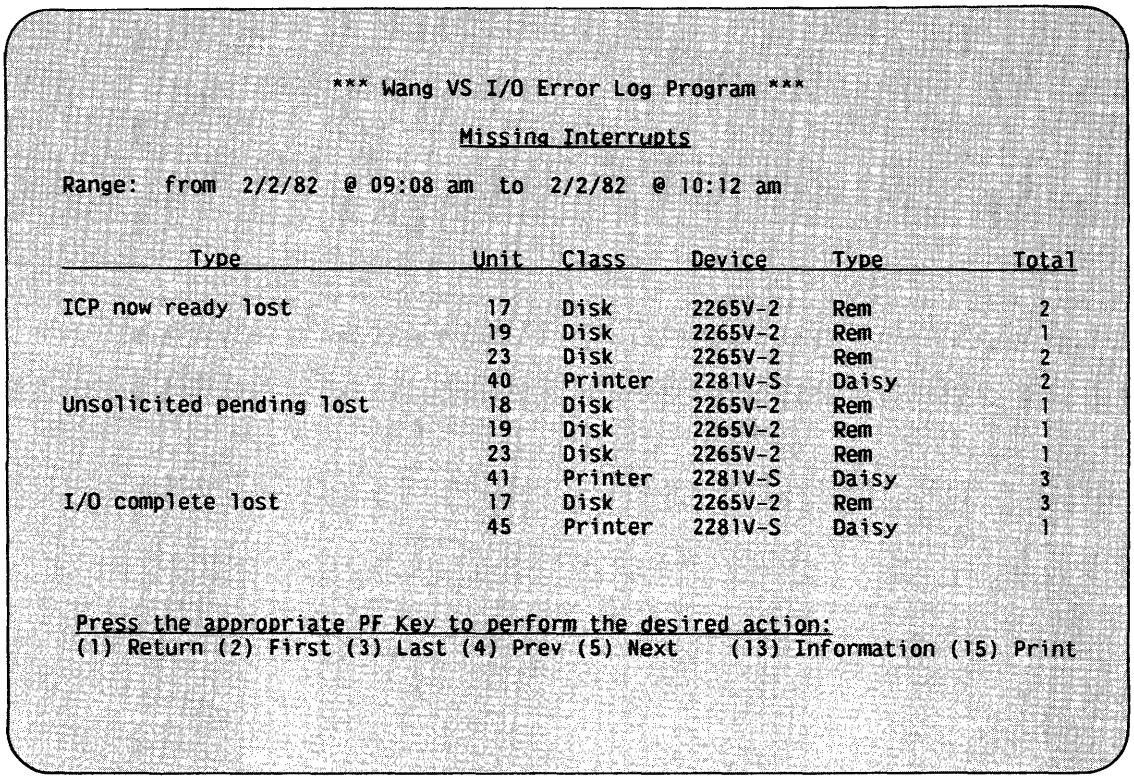
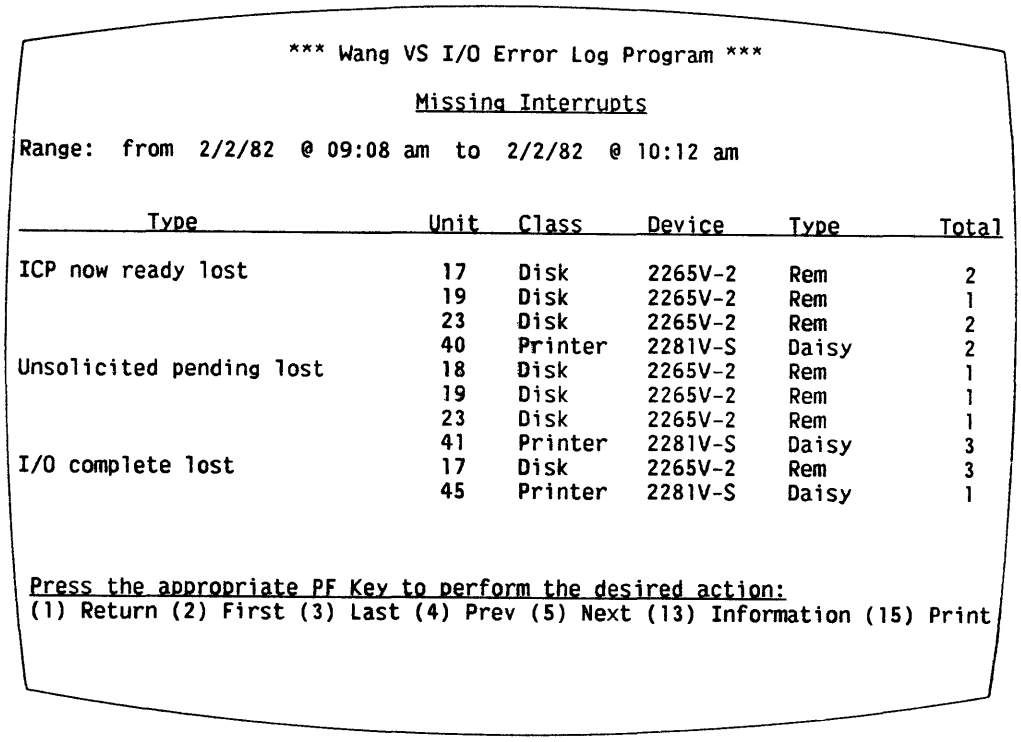


Figure 12-10. A Sample Missing Interrupt Summary Screen

The Nonstandard I/O Error Summary screen provides the following PF key options:

PF Key	Function
ENTER	Continues processing and displays a summary of errors that occurred for a particular system routine. Position the cursor next to any Nonstandard error before pressing ENTER.
1	Returns to the Standard I/O Error Summary screen.
13	Provides on-line help information.
15	Prints the screen contents.

You can analyze each Nonstandard error further by positioning the cursor next to an error and pressing ENTER. An error summary of the selected Nonstandard error is displayed. Figure 12-10 illustrates a sample Missing Interrupts Summary screen. The Missing Interrupts Summary screen provides you with the error type and the device on which the error occurred. The date and time range of the error log appear at the top of the screen.



*** Wang VS I/O Error Log Program ***

Missing Interrupts

Range: from 2/2/82 @ 09:08 am to 2/2/82 @ 10:12 am

Type	Unit	Class	Device	Type	Total
ICP now ready lost	17	Disk	2265V-2	Rem	2
	19	Disk	2265V-2	Rem	1
	23	Disk	2265V-2	Rem	2
	40	Printer	2281V-S	Daisy	2
Unsolicited pending lost	18	Disk	2265V-2	Rem	1
	19	Disk	2265V-2	Rem	1
	23	Disk	2265V-2	Rem	1
	41	Printer	2281V-S	Daisy	3
I/O complete lost	17	Disk	2265V-2	Rem	3
	45	Printer	2281V-S	Daisy	1

Press the appropriate PF Key to perform the desired action:
(1) Return (2) First (3) Last (4) Prev (5) Next (13) Information (15) Print

Figure 12-10. A Sample Missing Interrupt Summary Screen

The Missing Interrupts Error Summary screen includes the following:

Type	The type of missing interrupt that occurred.
Unit	The unit number of the device.
Device	The device name recognized by the system.
Type	The type of device on which the error occurred (i.e, Rem, Fixed, or Fix/Rem disk; Daisy or Chain printer, etc.)
Total	The total number of missing interrupts recorded by the log for a given device unit.

The Missing Interrupts Error Summary screen also provides the following PF key options:

PF Key	Function
1	Returns to the Nonstandard I/O Error Summary screen.
2 (First)	Displays the first screen of missing interrupt errors, if more than one screen is required.
3 (Last)	Displays the screen of missing interrupt errors, if the list requires more than one screen.
4 (Prev)	Displays the previous list of missing interrupt errors, if more than one screen is required.
5 (Next)	Displays the next list of missing interrupt errors, if the list requires more than one screen.
13	Provides on-line help information.
15	Prints the screen contents.

12.4.3 The IPL Summary

If the system records any IPLs in the I/O error log, press PF3 from the Standard I/O Error Summary screen to display the IPL Summary screen. The IPL Summary lists, in a left to right sequence, the date and time that each IPL occurred. The time and date range of the error log you analyze appears at the beginning of the summary, and a field of PF key options appears at the bottom of the screen.

A sample IPL Summary screen is shown in Figure 12-11.

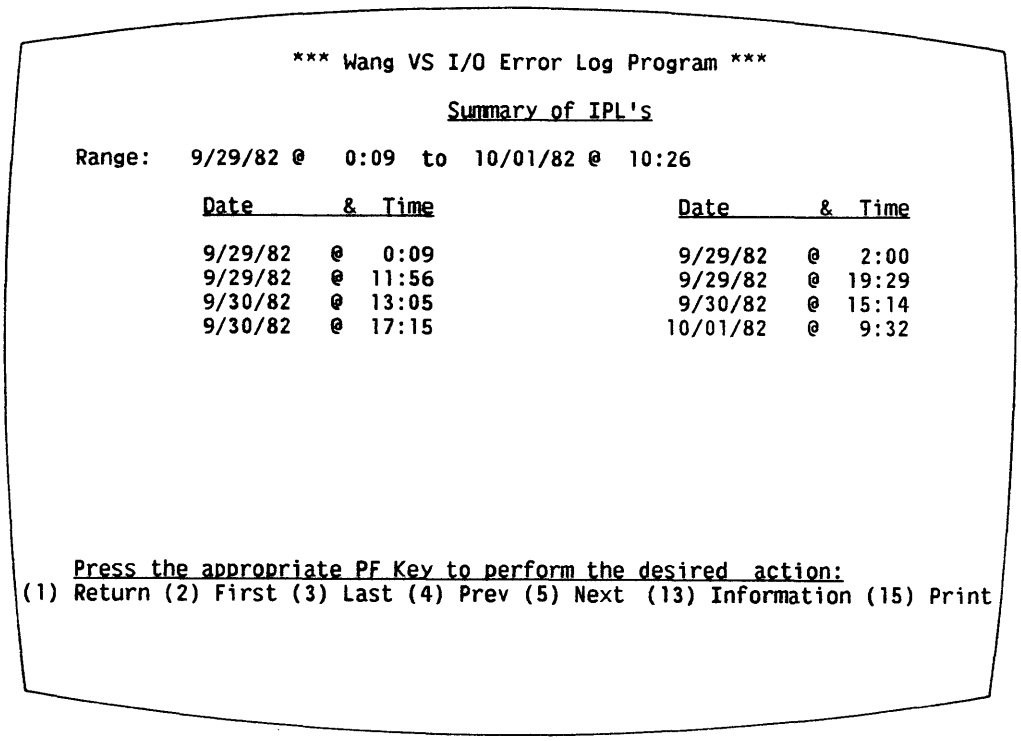


Figure 12-11. A Sample IPL Summary

The IPL Summary screen provides the following PF key options:

PF Key	Function
1	Returns to the Standard I/O Error Summary screen.
2 (First)	Displays the first screen of IPLs, if more than one screen is required.
3 (Prev)	Displays the previous list of IPLs, if the list requires more than one screen.
4 (Last)	Displays the last screen of IPLs, if more than one screen is required.
5 (Next)	Displays the next list of IPLs, if the list requires more than one screen.
13	Provides on-line help information.
15	Prints the screen contents.

12.5 PRINTING AN I/O ERROR LOG FILE

To print the contents of an I/O error log file, press PF2 from the Function Definition screen (see Figure 12-3). The Range Definition screen is displayed. This screen is similar to the Range Definition screen that is displayed when you choose to analyze an error log on your workstation (see Figure 12-4). The Range Definition screen displays the name and the location of the error log file you want to print. The date and time range of data contained in the error log is also displayed, including the number of records in the range. The screen also provides you with the modifiable beginning (STARTDAY and STARTIME) and ending (ENDDAY and ENDTIME) date and time range of the log, if you wish to print only a portion of the error log.

When you press ENTER, the error log or a portion is printed. An I/O error log printout contains the date and time an error is recorded by the log, the device(s) affected by the error, the command issued to the device by the Operating System, and the type of error that occurred as a result of that command. If a disk is affected by an I/O error, the location of the error on the disk is listed. The error occurred in the VTOC if a volume name is listed and file and library names are not. The date and time range of the error log appears at the top of each page on the printout.

When you print an error log, the log is sent to a print file. The system assigns the name ELOG followed by four unique digits (in the form ELOGXXXX). The file is stored in your default SPOOLIB and SPOOLVOL that you set through SET Usage Constants (PF2 from the Command Processor) or through a procedure. If you do not set any defaults, the print file is stored in the System Print Library (#PRT) on the System Volume.

When printing is completed, the Range Definition screen is displayed. You can press PF1 to return to the Function Definition screen, or print all or a portion of the error log.

A sample I/O error log printout is shown in Figure 12-12.

```
*** Wang VS I/O Error & IPL Log ***     Covering the time period from    9/29/82 at 12:09 to  9/29/82 at 14:51

                Active Task Unit #  IOCW(I/O Command Word) IOSW(I/O Status Word)          Retry          (For Disks Only)
Date     Time   User WS    #  Type   Command Description    Error Description            #  Volume Library File      Block Cyl Sur

9/29/82 12:09      ---   1     2   A0 048000 0800 000000  20190001 08000000
                         2246C     CIO, Load Microcode    Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Device powered off
                                                          Device processor not running

9/29/82 12:09      ---   1     2   82 0568DC 0054 011002  20193F01 00000000
                         2246C     SIO, Write             Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Screen damage alert
                                                          Device powered off
                                                          Device processor not running

9/29/82 12:20 BLB  39  14    19   43 06A990 0800 0D6F88  60100004 00000100            1 WORK   #BLBWORK ASM10003      0  41 149
                         2265V-2   SIO, Read              Soft Error
                                                          Device malfunction
                                                          Invalid CRC or ECC

9/29/82 12:38 JVA   2  13     2   A0 04F000 0800 000000  20190001 08000000
                         2246C     CIO, Load Microcode    Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Device powered off
                                                          Device processor not running

9/29/82 12:38 JVA   2  13     2   82 033EFC 0054 0002C6  20193F01 00000000
                         2246C     SIO, Write             Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Screen damage alert
                                                          Device powered off
                                                          Device processor not running

9/29/82 14:07 GAS  33  19    33   A0 03E000 0800 000000  20190001 08000000
                         2246C     CIO, Load Microcode    Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Device powered off
                                                          Device processor not running

9/29/82 14:07 GAS  33  19    33   82 074614 0054 000325  20193F01 00000000
                         2246C     SIO, Write             Hard Error
                                                          Device malfunction
                                                          Device or memory damage
                                                          No device processor code
                                                          Screen damage alert
```

Figure 12-12. A Sample I/O Error Log Printout

The following information is included in an I/O error log printout:

Item	Description
Date	The date of an entry in the error log.
Time	The time of an entry in the error log.
User	The User ID.
WS	A user's workstation number.
Task	The number of the task that is associated with the running program.
Unit #	The unit number of the device on which an error occurred.
Device Class & Type	The device classification (i.e., disk) and device type.

Item	Description
IOCW Command Description	The type of command that is issued by the Operating System to the device. A hexadecimal representation of the IOCW is also given.
IOSW Error Description	The status of the command previously issued by the Operating System to the device. A hexadecimal representation of the IOSW is also given.
Retry #	The number of times a command is issued to a device by the Operating System before it is successful (a soft error).

The following information appears only if a disk error occurred:

Item	Description
Volume	The name of the volume that is affected by an error.
Library	The name of the library that is affected by an error.
File	The name of the file that is affected by an error.
Cyl	The cylinder on which an error occurred, if any.
Sur	The surface on which an error occurred, if any.
Block	The block on which an error occurred, if any.

CHAPTER 13
THE LISTVTOC UTILITY

13.1 INTRODUCTION

The LISTVTOC utility performs a verification check on the Volume Table of Contents (VTOC) of a specified disk volume to determine whether or not the VTOC contains any bad blocks. This utility also provides an analysis of the contents of the VTOC. The results of the analysis for a specific volume consist of three listings:

- Space allocation for the volume.
- File names and attributes, by library.
- A VTOC map, which is a dump of the VTOC control blocks. (For an explanation of the VTOC, refer to the VS Operating System Services Reference.)

An overview of LISTVTOC is provided in Figure 13-1.

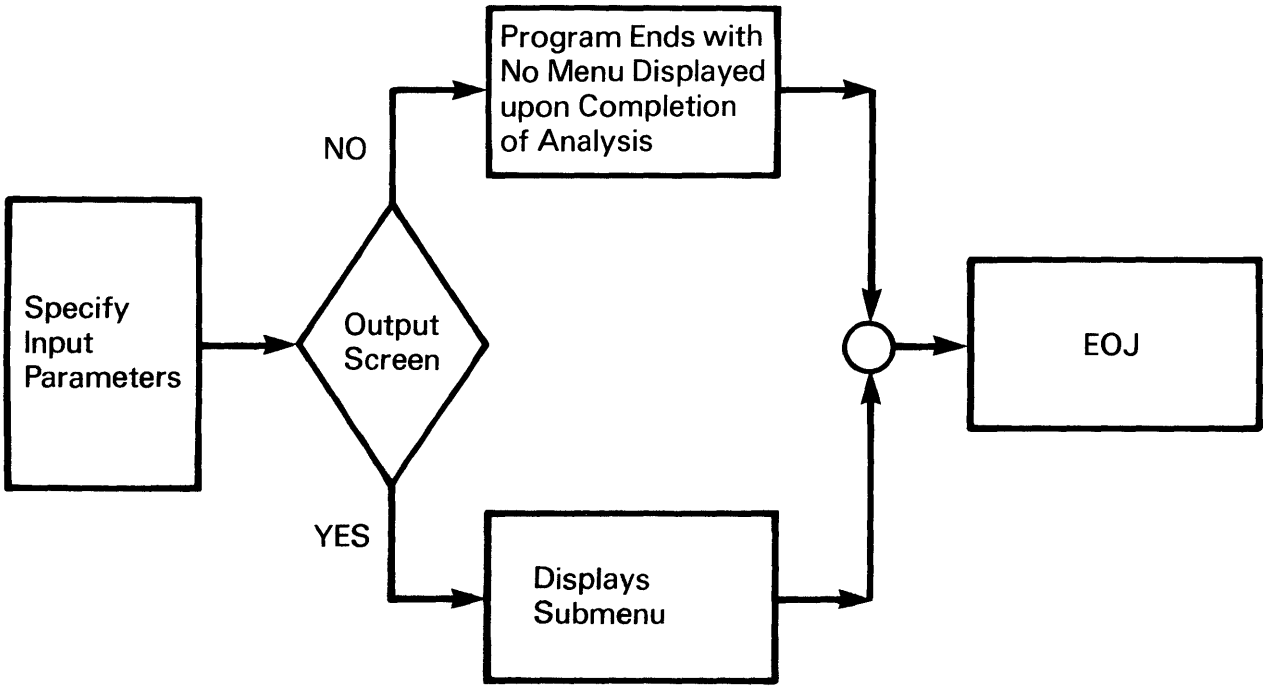


Figure 13-1. LISTVTOC Processing

13.2 DEFINING THE INPUT

When LISTVTOC processing begins, you must specify the following input parameters:

VOLUME	Volume name to be examined.
PRINT FILES	An alphabetical file listing, on the specified volume or in a specified library on the volume. The file attributes for each file are included in the listing. YES in PRINT FILES option determines whether or not you place the listing in a print file. NO causes the listing to be displayed on the workstation only. The default is NO.
LIBRARY	The library for which a File Name and Attribute listing is to be performed. If you leave this entry blank, the utility produces a listing (by library) of all the files on the specified volume.
PRINT VTOCMAP	A dump of the VTOC control blocks. If you enter YES, the VTOC map is placed in a print file. If you enter NO, the VTOC map can be displayed on the workstation screen, but no print file is created. You can review the dump by requesting DISPLAY VTOC CONTROL BLOCKS from the File Name and Attribute Listing menu. The default value is NO.
OUTPUT SCREEN	Determines whether or not the results of the LISTVTOC analysis are displayed on the workstation screen. When you enter YES, LISTVTOC displays a menu allowing you to view the results of the analysis, the file listings, and the VTOC map. When you enter NO, the program ends upon completion of the analysis with no menu displayed.
SELECT PRINT	The number of lines to appear on each page of the print file produced. Specify SELECT PRINT only when print files is requested. The default value is 45.

If you enter NO next to all input parameters, LISTVTOC creates a print file containing only the VTOC space and integrity analysis.

13.3 VTOC ANALYSIS SUBMENU

If you enter YES for the OUTPUT SCREEN parameter, upon completion of the VTOC analysis, LISTVTOC displays a submenu containing the following options:

PF Key	Function
1	Respecify the volume to be analyzed. The program is then restarted so that another volume can be analyzed.

PF Key	Function
2	Display current analysis result. This utility displays the Space Allocation listing containing the following information: • Total space on the volume, in blocks • Total allocated space on the volume, in blocks • Total available space on the volume, in blocks • Total space in the VTOC, in blocks, and the number of blocks allocated to each type of control block • Number of free blocks in the VTOC • Number of libraries and files on the volume • Number of free extents on the volume • Total available space on the volume, in blocks
3	Display file listing on the volume. File names and attributes are displayed in alphabetical order, by library.
4	Display VTOC control blocks. The VTOC map is displayed.
16	Terminate processing.

The VTOC map created by LISTVTOC displays the contents of the following control blocks in ASCII or hexadecimal:

FDAV	The File Diretory Available Space block that contains information for all unallocated areas on the volume.
FDX1	The first File Directory Index block that contains entries for each library on the volume and pointers to FDX2 records for each library.
FDX2	The second level of the File Directory Index that contains four FDX2 records in a 2048 block. The FDX2 records associate each file name with a library name, and contain pointers to the FDR block containing the file's FDR1 record.
FDR	The File Descriptor Record block that contains FDR1 records that describe the attributes of each file.

For more information about the VTOC control blocks, refer to the VS Operating System Services Reference.

13.4 A SAMPLE LISTVTOC PROCEDURE

You can control LISTVTOC workstation interaction through the VS Procedure language. A list of LISTVTOC GETPARMs is provided in Appendix A. Refer to the VS Procedure Language Reference for details concerning procedure syntax.

The following procedure creates a print file containing a VTOC space and integrity analysis, as well as a dump of the VTOC control blocks of the SYSTEM volume. The procedure eliminates workstation interaction by setting the SCREEN parameter to NO.

```
PROCEDURE
RUN LISTVTOC
ENTER INPUT VOLUME=SYSTEM, VTOCMAP=YES, SCREEN=NO
RETURN
```

CHAPTER 14
THE SORT UTILITY

14.1 INTRODUCTION

The SORT utility allows you to perform two different operations on VS files: sorting a file according to one or more key fields within the records and merging two or more sorted files. Within these operations, SORT allows you to perform the following functions:

- Sort a single data file into an order that you specify.
- Sort and merge up to 20 data files into a single, ordered output file.
- Merge files that are already sorted into a single output file.
- Select specified records from one or more input files and sort them into a single output file.
- Produce an output file that contains only the primary index key field from each record in the input file. You specify the sort order for the input primary index key field.

You can retrieve input files from tape (consecutive) or disk (consecutive or indexed). The SORT utility leaves the input file(s) intact unless you assign the same name to the input and output files. SORT always creates the output file, containing the sorted or merged records, as a consecutive file.

The SORT option of the utility takes an unordered file or files and puts it into a specified order, producing one ordered, consecutive, output file. You specify the order of the output file. When you sort two or more files, the utility automatically merges the files. The MERGE option takes two or more files that already have the same format and order, and combines them into one ordered, consecutive, output file. Both SORT and MERGE allow you to extract specific input records or parts of records for processing.

If you attempt to MERGE two or more files, any one of which is out of the specified order, the program terminates. The output file is already created at this point. It continues to exist despite the fact that it may contain no data.

To operate the SORT utility, perform the following steps:

1. Define the program options, i.e., SORT or MERGE.
2. Specify the input file(s) to be sorted or merged.
3. Specify the SORT/MERGE keys.
4. Define the output file.

An overview of SORT processing is provided in Figure 14-1.

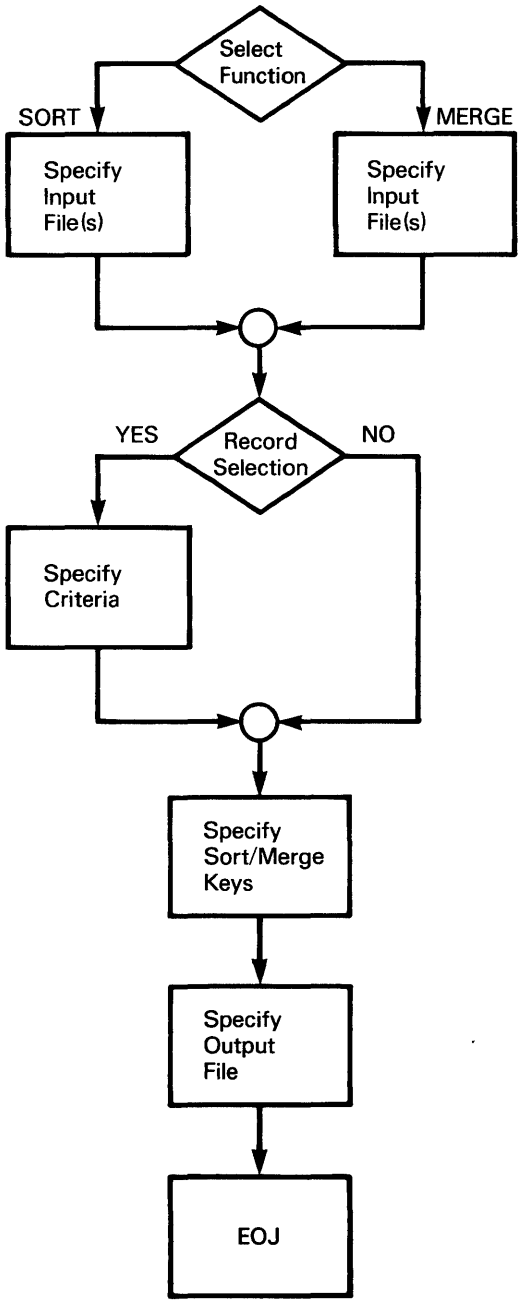


Figure 14-1. SORT Processing

14.2 DEFINING THE PROGRAM OPTIONS

When SORT processing begins, the Program Options screen is displayed. On this screen you select the type of program function (SORT or MERGE), determine the amount of memory to be used for processing the SORT or MERGE, and define the output selection criteria you need to process the files. To exit the SORT utility without defining the program options, press PF16. The program options are as follows:

FUNCTION — The processing operation that is performed. You can specify the function as SORT or MERGE. The SORT function sorts one or more files into a single output file. The MERGE function merges two or more ordered files into a single, sorted, output file. FUNCTION defaults to SORT.

MEMORY — The amount of main memory to be used as a work area. The main memory used affects the efficiency of the operation being performed. For most operations, the default value, 128K memory, is sufficient. If sufficient main memory is not available for the default value, the system determines the amount actually available and runs the utility in that amount of memory.

Enlarging the work area beyond the default value to improve SORT or MERGE processing time increases the competition with other users for main memory. The maximum amount of main memory available is displayed on the screen; it is dependent upon the size of the system CPU. The optimum amount of main memory for each SORT or MERGE can only be determined by experimenting with the actual operating environment. If the SORT or MERGE takes a lot of processing time or a large amount of main memory, you should run the operation in batch mode during the off-peak hours of the installation so that other users are not slowed down or temporarily halted. MEMORY defaults to 128K.

ADDROUT — A YES or NO option that produces a file of sorted, three-byte records used by RPG II programs. Each record is a positive binary number representing the relative number of a record in the input file. When you perform a sort with the ADDROUT option, the only output is an ADDROUT file. You cannot use the ADDROUT option with the MERGE option, nor can you combine it with a KEYOUT file. ADDROUT can be combined with a STABLE sort. ADDROUT defaults to NO. An example of an ADDROUT file is illustrated in Subsection 14.2.1. For more details on ADDROUT files, refer to the VS RPG II Language Reference.

KEYOUT — A YES or NO option that produces a single field, consecutive file by using the primary index key of the input file as the selection criteria. The input file must be indexed. You cannot use a KEYOUT file with the MERGE option, nor can you combine it with an ADDROUT file. KEYOUT can be combined with a STABLE sort. KEYOUT defaults to NO. An example of a KEYOUT file is illustrated in Subsection 14.2.1.

STABLE A YES or NO option that produces an output file. In this sorted output file, records with identical sort keys remain in the same order as the input order sequence. STABLE defaults to NO. An example of a STABLE sort is illustrated in Subsection 14.2.1.

The combinations of program options and output files are summarized in the following table:

		OPTION				
		ADDROUT	KEYOUT	STABLE	SORT	MERGE
OUTPUT FILE	ADDROUT	YES	NO	YES	YES	NO
	KEYOUT	NO	YES	YES	YES	NO
	STABLE	YES	YES	YES	YES	YES

After you enter the the program options, the Input Definition screen is displayed.

14.2.1 SORT Processing Examples

The following examples demonstrate the ADDROUT, KEYOUT, and STABLE options.

ADDROUT Output File Example

In the following example, the input is a consecutive file to be sorted on the NUMBER field in ascending order. The ADDROUT output file is a consecutive file consisting of 3-byte records containing the sequence numbers of the records in the input file. For example, Teller is record number 008, and is listed first in the output file because it has the lowest NUMBER in the sorted field. DiMello is record number 003, and is listed second because it has the second lowest NUMBER in the sorted field. The SORT utility cannot produce ADDROUT files from a MERGE or multiple input files. Each record is a positive binary number that is intended to be read by an RPG II program; ASCII Display Characters appear if you have the output file displayed at your workstation.

	INPUT FILE			ADDROUT OUTPUT FILE
RECORD SEQUENCE	NAME	NUMBER	ADDRESS	
001	Anders	2221	Greenwich St	008
002	Dalgliesh	8677	Edinburgh Rd	003
003	DiMello	1120	Roman Rd	005
004	Edwards	3001	E. 67th St	001
005	Funazi	2220	Hamden	007
006	Juliano	6676	W Haven	004
007	Martins	2231	Minton Ave	006
008	Teller	1110	Beacon St	002

KEYOUT Output File Example

In the following example, the input is an indexed file with three index keys. The primary index key is NAME, the first alternate key is NUMBER, and the second alternate key is ADDRESS. The input file is sorted on the NUMBER field in ascending order. The result is a KEYOUT file that contains only the contents of the primary index key field.

	INPUT FILE		KEYOUT OUTPUT FILE
NAME	NUMBER	ADDRESS	
Anders	2221	Greenwich St	Teller
Dalgliesh	8677	Edinburgh Rd	DiMello
DiMello	1120	Roman Rd	Funazi
Edwards	3001	E. 67th St	Anders
Funazi	2220	Hamden	Martins
Juliano	6676	W Haven	Edwards
Martins	2231	Minton Ave	Juliano
Teller	1110	Beacon St	Dalgliesh

STABLE Output File Example

In the following example, the input files are indexed and have three index keys. The primary index key is NAME, the first alternate key is NUMBER, and the second alternate key is ADDRESS. Input files 1 and 2 are to be sorted in descending order using the NUMBER field as the primary sort key. Note the order of the entries, with identical sort keys, in the output file showing the STABLE option. In this output file, the output records with the same key remain in the same order as the input order sequence. For comparison, an example of the output file showing the STABLE=NO option is included.

INPUT FILE 1

NAME	NUMBER	ADDRESS
Carter	1110	Hollywood Ave
Dahlman	1110	Boothbay St
Livieri	2220	Elm St
Lovell	2221	Daniel Blvd
Riccio	3001	Scott Dr

INPUT FILE 2

NAME	NUMBER	ADDRESS
Anders	2221	Greenwich St
Dalgliesh	8677	Edinburgh Rd
DiMello	1120	Roman Rd
Edwards	3001	E. 67th St
Funazi	2220	Hamden
Juliano	6676	W Haven
Martins	2231	Minton Ave
Teller	1110	Beacon St

OUTPUT FILE WHEN STABLE = YES:

Dalgliesh	8677	Edinburgh Rd
Juliano	6676	W Haven
Riccio	3001	Scott Dr
Edwards	3001	E. 67th St
Martins	2231	Minton Ave
Lovell	2221	Daniel Blvd
Anders	2221	Greenwich St
Livieri	2220	Elm St
Funazi	2220	Hamden
DiMello	1120	Roman Rd
Carter	1110	Hollywood Ave
Dahlman	1110	Boothbay St
Teller	1110	Beacon St

OUTPUT FILE WHEN STABLE = NO:

Dalgliesh	8677	Edinburgh Rd
Juliano	6676	W Haven
Edwards	3001	E. 67th St
Riccio	3001	Scott Dr
Martins	2231	Minton Ave
Anders	2221	Greenwich St
Lovell	2221	Daniel Blvd
Livieri	2220	Elm St
Funazi	2220	Hamden
DiMello	1120	Roman Rd
Dahlman	1110	Boothbay St
Carter	1110	Hollywood Ave
Teller	1110	Beacon St

14.3 SPECIFYING THE INPUT FILE TO BE SORTED OR MERGED

After you select the program options, specify the input files you want to sort or merge. Specify the file, library, and volume names for each input file you want to process. You also specify whether SORT will open indexed and consecutive input files in Shared mode. If you select the MERGE option (through the Program Options screen), you can now specify up to 20 input files. The Input Definition screen automatically reappears until you enter an unaltered screen. If you previously selected the SORT option, you can specify only one input file, unless you specify MOREFILE=YES. The MOREFILE field enables you to specify more than one input file. You can also select specific records from a file for use in a SORT or MERGE operation through use of the Input Definition screen. (Refer to Subsection 14.3.3 for more information on selecting records.)

14.3.1 Defining the Input

When defining the input file, you must also specify the file input device. The file input device can be either tape or disk. If the input device is tape, you must specify the file sequence number and the maximum number of input records. You do not need to specify file sequence number and number of records for disk files.

The input parameters to be defined are as follows:

INPUT FILE — The file name of the first input file you want to sort or merge. If the MERGE option has been selected, you can specify up to 20 input files. The Input Definition screen automatically appears again until you enter an unaltered screen. There is no default value for INPUT FILE.

LIBRARY — The name of the library in which the input file(s) reside(s). The default value for LIBRARY is your INLIB, previously set through SET Usage Constants of the Command Processor or a Procedure language SET statement.

VOLUME — The name of the volume on which the input file(s) reside(s). The default value for VOLUME is your INVOL, previously set through SET Usage Constants of the Command Processor or a Procedure language SET statement.

SELECT — A YES or NO option. You can select specific input records for use in a SORT or MERGE operation. If you specify SELECT=YES, only records matching specified selection criteria are used to produce the output file. After you enter SELECT=YES, select the criteria you want through the Select screen that is displayed next. (For additional information on defining SELECT criteria, refer to Section 14.3.3.) You exit the Select screen by pressing PF16. If you specify SELECT=NO, all the records in the input file are processed, and the Select screen is not shown. SELECT defaults to NO.

MOREFILE — A YES or NO option. You can sort more than one input file (and automatically merge the files) into a single output file. If you specify MOREFILE=YES, you must specify all input file names before continuing SORT processing. Additional input files are entered through additional Input Definition screens that display automatically if you select the MOREFILE=YES option. Notice the additional Input Definition screens do not include the SELECT and MOREFILE options. When you enter an unaltered screen, the Input Definition screens are terminated. MOREFILE defaults to NO.

SHARED — A YES or NO option. You can direct SORT to open consecutive and indexed input files in Shared mode. If you specify YES, SORT then displays the Lock screen; refer to Subsection 14.3.2. If you specify NO, SORT opens the input file in Input mode. SHARED defaults to NO.

FILE INPUT DEVICE — The type of device on which the input file(s) reside(s). If the input device is disk, no further information is required to complete the input definition. If the input device is tape, you must define two further parameters: FILESEQ and RECORDS. FILE INPUT DEVICE defaults to DISK.

FILESEQ (tape only) — The sequence number of the input file on the tape. FILESEQ defaults to 1.

RECORDS (tape only) — The approximate number of records that are to be processed in the file. (You can find the number of records by displaying the file attributes. Press PF5, Manage FILES/LIBRARIES, on the Command Processor.) The program compares the tape file label to the number of RECORDS specified. If you underestimate the number of records in a file, you may have to select RESTART. (For further details on RESTART, refer to Section 14.5.1.) The smallest possible overestimation is the most efficient. RECORDS defaults to 1000.

14.3.2 Sorting Files in Shared Mode

If you specify Shared mode for an input file, the Lock screen prompts you to indicate whether you want the file to be locked. Enter YES or NO in the LOCK field; the default is YES. If you lock a file, no updates to the file can occur while you are sorting it. If you specify NO, no lock is placed on the file, and there is no need to specify the TIMEOUT and BYPASS options.

If a file is held for update by another user, the TIMEOUT field specifies the length of time that SORT waits to open the file in Shared mode with a lock. You can specify a timeout for a file if LOCK is equal to YES. Enter a value from 0 to 255 seconds or NO; the default is 10 seconds. If you specify NO in the TIMEOUT field, there is no timeout, and SORT waits indefinitely until it can lock the file.

The BYPASS field allows you to specify whether the file should be skipped if the timeout expires. Values for BYPASS are YES and NO; the default is NO.

If BYPASS is YES and the timeout expires, SORT skips the file. If BYPASS is NO and the timeout expires, the Lock screen reappears with the message

File XXXXXXXX in XXXXXXXX on XXXXXX is held by user XXX

You can then redefine the LOCK, TIMEOUT, and BYPASS options and press ENTER to continue with the sort operation. You can also press PF1 to skip the file on which the timeout occurred.

14.3.3 Defining SELECT Criteria

If you specify SELECT=YES, the SORT utility selects the records to be sorted or merged on the basis of the relationships defined on the Select screen. The utility selects individual input records by testing them against comparison values you enter through this screen. For example, if you are processing a payroll file, you may only want to process the records of part-time employees. The records of part-time employees all have a "P" in the first byte. In this example, the test criteria is that all input files must equal 'P' in position 1 in order to be processed. If the specified relation is equal, the record is selected for sorting. Otherwise, it is not sorted or merged, and is not transferred to the output file. Figure 14-2 illustrates this example.

Besides being tested for a literal data value, a field can also be tested against another field in the same record. For example, you can specify a condition where if position 1 equals position 43, the record can be processed.

When processing a file, you can test additional criteria by using the optional CONECTn field. You can specify the complex relations by combining simple relations with the AND connector. Alternative relations are specified with the OR connector. For example, if a payroll record must be part-time and active, then the test criteria must specify that all input files must equal 'P' in position 1 and must equal 'A' in position 3 in order to be processed. Figure 14-2 illustrates this example of the CONECTn field entered on the Select screen. Note that the test statements in Figure 14-2 are processed by the SORT utility as follows: (Statement 1) OR (Statement 2) OR (Statement 3 AND Statement 4).

14.3.2 Defining SELECT Criteria

If you specify SELECT=YES, the SORT utility selects the records to be sorted or merged on the basis of the relationships defined on the Select screen. The utility selects individual input records by testing them against comparison values you enter through this screen. For example, if you are processing a payroll file you may want part-time employees only processed. The records of part-time employees all have a "P" in the first byte. In this example, the test criteria is that all input files must equal 'P' in position 1 in order to be processed. If the specified relation is equal, the record is selected for sorting. Otherwise, it is not sorted or merged, and is not transferred to the output file. Figure 14-2 illustrates this example.

Besides testing for a literal data value, a field can also be tested against another field in the same record. For example, you can specify a condition so that if position 1 equals position 43, the record can be processed.

When processing a file, you can test additional criteria by using the optional CONECTn field. You can specify the complex relations by combining simple relations with the AND connector. Alternative relations are specified with the OR connector. For example, if a payroll record must be part-time and active, then the test criteria is that all input files must equal 'P' in position 1 and equal 'A' in position 3 in order to be processed. Figure 14-2 illustrates this example of the CONECTn field entered on the Select screen. Note that the test statements in Figure 14-2 are processed by the SORT utility as follows: (Statement 1) OR (Statement 2) OR (Statement 3 AND Statement 4).

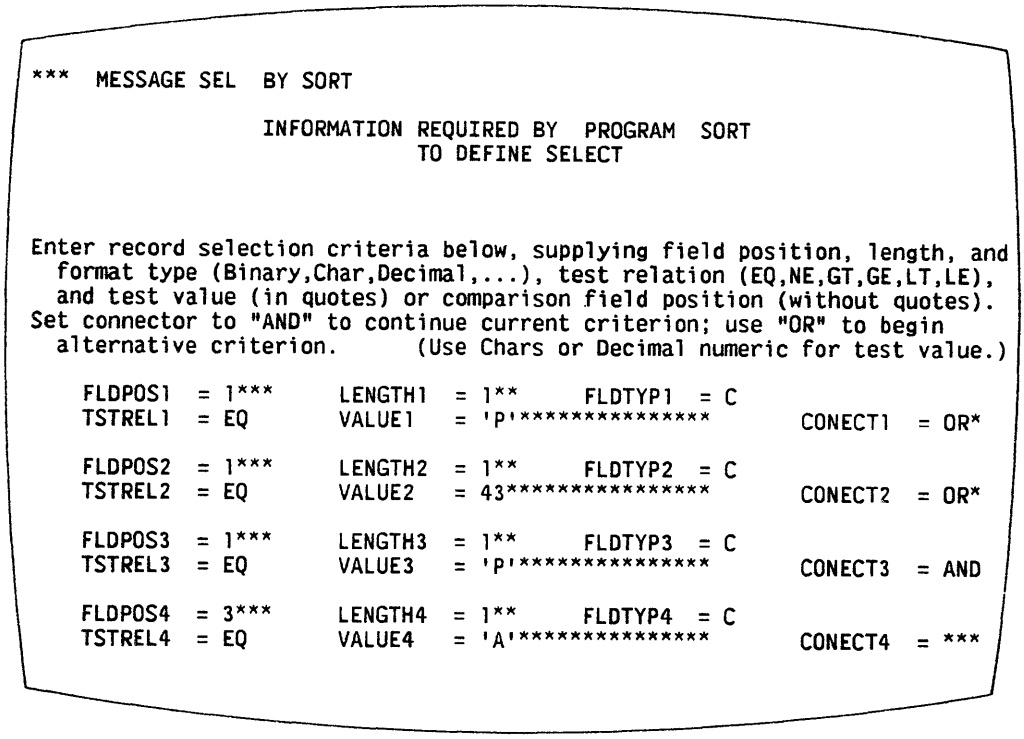


Figure 14-2. A Sample Select Screen

You select from the following options to sort or merge records. The 'n' appended to each parameter is a numeric value that SORT automatically assigns. All parameters with the same numeric values are combined to specify a test relationship for data comparisons. For example, FLDPOS1, LENGTH1, FLDTYP1, TSTREL1, VALUE1, and CONECT1 are combined to identify selection criteria number one.

FLDPOSn — The starting field position in the record that is to be used in the comparison. The first byte of a record occupies position 1. When building a complex relation, the field positions do not have to be in ascending numeric order. For example FLDPOS1=5, FLDPOS2=3 are as acceptable as FLDPOS1=3, FLDPOS2=5. No default value exists for FLDPOSn.

LENGTHn — The number of bytes in the field to be tested, starting at FLDPOSn. No default value exists for LENGTHn.

FLDTYPn The data format of VALUEn. The field type options are:

C	Character data (alphanumeric)
B	Binary
D	External decimal, sign trailing in separate byte
L	Zoned decimal, sign leading in separate byte
P	Packed decimal, sign trailing included in last byte
Z	Zoned decimal, sign trailing included in last byte

FLDTYPn defaults to C.

TSTRELn The type of comparison to be made between the input data (starting at FLDPOSn) and VALUEn. The test relationship options are:

EQ	Equal
NE	Not equal
GT	Greater than
GE	Greater than or equal to
LT	Less than
LE	Less than or equal to

No default value exists for TSTRELn.

VALUEn The value to be compared to the field in each input record beginning at FLDPOSn and extending for LENGTHn bytes. VALUEn can be a literal data value enclosed in single or double quotes. VALUEn can also be the starting position of another field in the record. If you specify a literal data value, even if it is a numeric data value, you must enclose it in quotes. If you do not type the quotes, VALUEn is treated as a field position. Eliminating the quotation marks can result in an erroneous output file or an aborted sort or merge. If you specify the starting position of another field in the record, it must be of the same type and length as the original field. Specifying the incorrect type or length can result in an erroneous output file or an aborted sort or merge. VALUEn can be a maximum of 16 bytes long for a literal data value or for external, zoned or packed decimal data. VALUEn can be a maximum of 256 bytes long for character data in a field-to-field comparison. VALUEn must be two or four bytes long for binary data. No default value exists for VALUEn.

CONECTn The logical connector you use between selection criteria. The choices are AND and OR. AND combines two or more simple comparisons into a single complex criterion. OR indicates the start of a new alternative criterion. You can specify up to 32 comparisons for each SORT or MERGE; for example, the number of ANDs and ORs used for each SORT or MERGE can be a maximum of 31. CONECTn is an optional field. A blank CONECTn field signifies the end of the comparisons in a single criterion. CONECTn defaults to blank.

14.4 DEFINING THE SORT/MERGE KEYS

After you specify the input file(s) and the selection criteria (if any), the Sort/Merge Keys screen is displayed. At the Sort/Merge Keys screen you select the data field(s) by which the file is to be sorted or merged. These selected data field(s) are the sort/merge keys (to be distinguished from index keys). To define the sort/merge keys, you must specify the following: the position of the data field in the record, the length of the data field in bytes, the data type of the key field (character, binary, decimal, etc.), the order of the output file (ascending or descending), and the number of sort/merge keys to be used. You can specify up to eight keys for each sort or merge.

The options for the sort/merge keys are as follows. The 'n' appended to each parameter is a number SORT automatically assigns. All parameters with the same number are combined to specify an individual sort/merge key. For example, POST1, LENGTH1, TYPE1, and ORDER1 are combined to identify sort/merge key number one.

NUMBER OF KEYS — The number of keys you use for each sort or merge. A maximum of eight keys are allowed.

POSTn — Starting position of the key field within the record. The first byte of a record is designated as position 1. POSTn has no default value.

LENGTHn — The length of the key field in bytes. LENGTHn has no default value.

TYPEn — The data type of the key field. The options are:

- C Character data (alphanumeric)
- B Binary
- D External decimal, sign trailing in separate byte
- F Floating point
- L Zoned decimal, sign leading in separate byte
- P Packed decimal, sign trailing included in last byte
- Z Zoned decimal, sign trailing included in last byte

TYPEn defaults to C.

ORDERn — The output file is sorted or merged on the selected key. The options are as follows.

- A Ascending order (low to high)
- D Descending order (high to low)

ORDERn defaults to A.

If you specify more than one key in the Sort/Merge Keys screen, the SORT utility uses the keys in the order in which they are listed. For example, the sort is performed first on Key 1, then on Key 2, then on Key 3, etc. The sort/merge keys can be used to sort more than one field or to sort on the first key specified, with the additional keys being used to sort duplicate values within the first key. When you have entered all the sort/merge keys, the SORT utility processes the file(s).

14.5 DEFINING THE OUTPUT FILE

After SORT processes the file(s), the Output Definition screen is displayed. You must now define the output file to be created. You can have SORT produce the output file in two different ways. First, you can create a new file as the output file. Second, if the output file size is less than or equal to the input file size, the newly created output can replace the input file. If you create a new file as output, assign different names to the input file and output file. If the input file is to be replaced by the newly created output, assign the same name to the output file as the input file. The utility then displays a message informing you that a file already exists with this name. Press PF3 to scratch the previous input file; the newly created output file then takes its place.

The output volume and library names are also required. LIBRARY and VOLUME default to your OUTLIB and OUTVOL. The output file can be compressed or uncompressed. The file output device can be tape or disk. If the output device is tape, you must specify the file sequence number. File sequence number is not specified for disks.

The REPLACE option facilitates the running of the SORT utility as a procedure. REPLACE allows the replacement of the input file by the newly created output file of the same name without reference to a screen prompt. Interactive users have a more general replacement option available to them; as mentioned above, naming the output file the same name as the input file. Consult the VS Procedure Language Reference for more information on writing and running procedures.

After you define the output, the SORT utility creates the output file and returns you to the Command Processor. A SORT completion message appears on the screen. If the SORT utility is unable to proceed correctly, an error message and a return code appears. (Refer to Appendix D for more information on return codes.) In some cases, during an unsuccessful SORT or MERGE, the option to restart or terminate the processing is also displayed. (Refer to Subsection 14.5.1 for more information on RESTART.)

The options for defining the output file are:

OUTPUT FILE	The name of the output file to be created. No default value exists for output file name.
LIBRARY	The name of the library in which the output resides. LIBRARY defaults to your OUTLIB, previously set through SET Usage Constants of the Command Processor.
VOLUME	The name of the volume on which the output resides. VOLUME defaults to your OUTVOL, previously set through SET Usage Constants of the Command Processor.
REPLACE	A YES or NO option. You can replace the input file without a prompt to scratch the existing file. REPLACE defaults to NO. The REPLACE option does not appear if you selected the MERGE function, if there is more than one input file, if the input file is indexed, if the input file is opened in Shared mode, or if you are creating an ADDROUT or KEYOUT file.
COMPRESS	A YES or NO option. You can compress data on the output file. Compression can save disk space. COMPRESS defaults to YES for variable-length input, and to NO for fixed-length input.
FILE OUTPUT DEVICE	The type of device on which the output file is to reside. If the output device is disk, no further information is required to complete the output definition. If the output device is tape, you must specify the file sequence number. The default value is DISK.
FILESEQ (tape only)	The sequence number of the output file on the tape. FILESEQ defaults to 1.

14.5.1 Restarting the SORT Utility

An unsuccessful SORT or MERGE will halt, allowing you to restart or terminate the program run. For example, the SORT utility will not proceed if the actual number of records in the input file is more than you specified (at the Input Definition screen). An error message is displayed, the SORT utility provides you with an actual record count, and the option to either restart (using the actual record count) or terminate the processing.

If you select to restart by pressing PF1, you can do this in two different ways depending upon whether the input file resides on tape or disk. If the input file is on tape, you only have to rerun the SORT utility by using the actual record count. No further data entry operations are necessary because SORT saves the previously entered GETPARMs when it issues the RESTART message. If the input file is on disk, the problem may result from a prior system failure, and you should reorganize the file through the COPY utility, before retrying the SORT utility. (Refer to Chapter 2 for more information on file reorganization.)

If the option to terminate is selected, press PF16. The SORT utility is then terminated, and a return code is issued; an output file is not produced.

14.6 A SAMPLE SORT PROCEDURE

You can control SORT processing through the VS Procedure language. You can also specify SORT input and output options, as well as selection criteria and key options. Appendix A contains SORT GETPARMs. Consult the VS Procedure Language Reference for details about procedure syntax.

In the sample SORT procedure that follows, all defaults of the Program Options screen have been accepted. The output file is sorted in descending order. Selection criteria specify that the value of the first byte of the output records must be greater than or equal to the value at byte 10 of the input records, or greater than a literal value of A.

Note that the value at VALUE2 is enclosed in single quotes within double quotes. 'A' is a literal value specified in the Select screen. Because VS Procedure language uses quotation marks as delimiters, both sets of quotes are necessary. If you eliminate one set of quotation marks when writing the SORT procedure, or if both sets of quotes are the same, the procedure returns an error message indicating incorrect punctuation and a return code of 4. (Appendix D contains return code explanations.)

```
PROCEDURE
RUN SORT
ENTER OPTIONS
ENTER INPUT FILE=SAVE3, LIBRARY=DJDCOPY, VOLUME=NEWSYS, SELECT=YES
ENTER SELECT FLDPOS1=1, LENGTH1=1, TSTREL1=GE, VALUE1=10,
    CONECT1=OR, FLDPOS2=1, LENGTH2=1, TSTREL2=GT, VALUE2="'A'"
ENTER KEYS POST1=1, LENGTH1=1, ORDER1=D
ENTER OUTPUT FILE=FILE1, LIBRARY=PROC1
RETURN
```

CHAPTER 15
THE TAPECOPY UTILITY

15.1 INTRODUCTION

The TAPECOPY utility performs copy functions involving tape files as the input and/or output files. It also can be used to transfer disk files. TAPECOPY performs the following functions:

- Copies files, including word processing documents and IBM files, from tape to disk, disk to tape, tape to tape, and disk to disk.
- Invokes the TRANSL utility to translate file contents to or from the EBCDIC and ASCII character sets.
- Invokes the TRANSL utility to manipulate the fields within a record you are copying.

> **NOTE**
>
> Even though it is possible, TAPECOPY is not recommended for disk-to-disk copying. The COPY utility is more efficient for disk-to-disk copying.

An overview of TAPECOPY processing is provided in Figure 15-1.

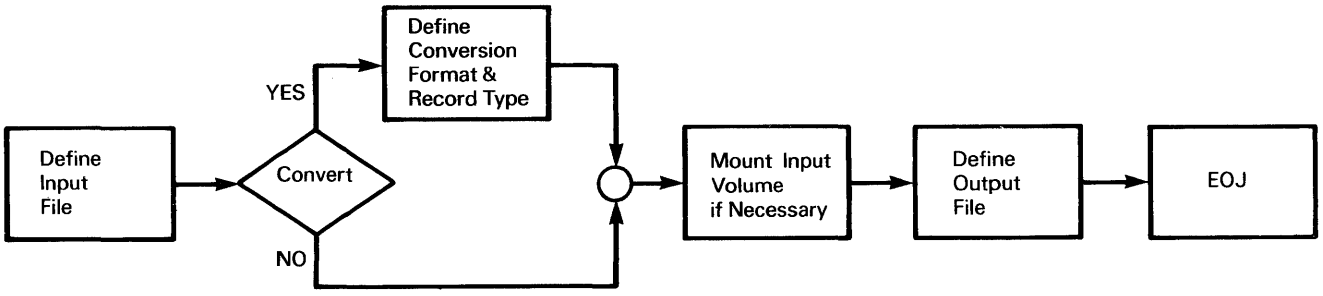


Figure 15-1. TAPECOPY Processing

15.2 DEFINING THE INPUT FILE, TAPE, OR DISK

When processing begins, specify the input file parameters. Specify a disk file by entering the file, library, and volume names. You specify a tape file by entering the file, library, and volume names (labeled tapes only) or by entering the volume name and the file sequence (FSEQ) number. If you enter the file and library names in addition to the file sequence number and volume name, TAPECOPY positions the tape according to the FSEQ number and then verifies the file label.

You must copy files that reside on more than one tape volume in the correct sequence. Refer to Section 15.6 for details on multivolume copying.

You also specify the following parameters:

DEVICE — The device containing the file you want to copy. Valid responses are TAPE and DISK; the default response is TAPE.

CONVERT — The character set conversion option you want. An entry in this field invokes the TRANSL utility. The input is copied without translation if the default response of N is accepted.

N	No Conversion
E	EBCDIC to ASCII Conversion
A	ASCII to EBCDIC Conversion
BA	BCD Format A to ASCII Conversion
AB	ASCII to BCD Format A Conversion
HA	BCD Format H to ASCII Conversion
AH	ASCII to BCD Format H Conversion

For transactions to BCD Format A, the ASCII characters plus sign (+), left parenthesis ((), right parenthesis ()), equals sign (=), and apostrophe (') are converted to blanks due to the definition of the A Format. Thus, data containing these characters should be converted through the BCD Format H option.

MULTYPE — The record type. N indicates the file contains a single record type. Y indicates a multiple record type file. The default response is N. This field must be specified when you invoke the TRANSL utility.

NOTE

If you invoke TRANSL, its Field Attributes Table is displayed. If MULTYPE=Y, the screen requesting identifying characters for each record type is displayed. Refer to Chapter 17 for more information.

If the input file is on tape, enter the file sequence number and label type as follows:

FSEQ — The sequence number of the file on the tape. A tape file is identified by the volume name and FSEQ.

If you do not supply the FSEQ number for a labeled tape (FSEQ=0), TAPECOPY uses the file and library names to position the tape. If you do supply the FSEQ number, TAPECOPY uses it to position the tape and check the file name against the label. If a discrepancy occurs, an error message informs you of the conflict. You then have the option of respecifying the input file name or leaving the field blank. If you leave the field blank, TAPECOPY uses only the FSEQ number to position the tape. If you do not enter the FSEQ number, TAPECOPY uses the file name to position the tape.

When you rerun TAPECOPY in the same processing session, the FSEQ number is automatically incremented by one. The allowable range is from 1 to 9999; the default response is 1.

LABEL — The type of tape label. Enter AL for ANSI, IL for IBM, or NL for No label. The default response is AL.

If the specified tape or disk volume is not already mounted, TAPECOPY allows you to initiate a mount operation by supplying a device number and pressing PF4. You can respecify the volume name again.

15.3 DEFINING TAPE FILE CHARACTERISTICS FOR UNLABELED TAPES

TAPECOPY obtains input file characteristics and default output file characteristics for disks and labeled tapes from input file labels. If the file, however, resides on an unlabeled tape, TAPECOPY displays the Define Tapefile Screen for you to enter the file characteristics as follows. (Default values for output parameters are selected from the input file.)

RECFORM — The tape record format. The following record formats are valid:

- F Fixed length
- V Variable length (Wang format)
- I Variable length (IBM format)
- U Undefined length

LARGEST RECSIZE — The length of the record in bytes. For variable-length records, enter the length of the largest record. The maximum record size for all record formats is 2048. The following table summarizes the legal minimum record sizes for all record formats:

Record Format	Minimum Record Size Input	Minimum Record Size Output
F	12	18
V,I	4	4
U	12	18

BLOCKED — Indicates whether the file is blocked or unblocked. Files with fixed-length records can be blocked or unblocked. Files with variable-length records in Wang format are always blocked. For input only, files with variable-length records in IBM format can be blocked or unblocked. Variable length IBM format output records must be blocked. Files with records of undefined length are unblocked. A response of Y indicates blocked records; a response of N indicates unblocked records. The default response is Y.

LARGEST BLOCKSIZE — The block length in bytes. The maximum block size is 32K (32,768 bytes). The mimimum block size is 12 bytes for input files and 18 bytes for output. The block size entered for a fixed-length record must be an even multiple of the record size. The block size for variable-length records in Wang format must be at least 4 bytes longer than the maximum record size. The block size for variable-length records in IBM format must be at least 8 bytes longer than the maximum record size.

COMPRESS — Selects file compression. File compression saves tape or disk space. A response of Y compresses the file; a response of N does not compress the file.

If you specify a tape file as the output file, the next display requests you to enter the tape file parameters. (The parameters are described in Section 15.4.)

15.4 DEFINING AN OUTPUT TAPE FILE

After you enter the input file parameters, specify the output file parameters. Labeled tape files are identified by file, library, and volume names. When you copy a word processing document, you must provide file and library names that can be recognized by VS Word Processing, if the document is to be accessed by VS Word Processing. Word processing file names consist of four digits and word processing library names are in the form of DOCMNTX or DOCMNTXX. X represents an uppercase document library letter, and XX represents a lowercase document library letter. Refer to the VS Programmer's Guide to VS/IIS for details.

Unlike input tape files, you must supply the FSEQ number for both labeled and unlabeled output tape files. An unlabeled tape file is identified by volume name and FSEQ number. Specify the required parameters as follows:

DEVICE — The device on which the output file is placed. DEVICE can be TAPE or DISK. The default value for the output device is DISK when the input device is TAPE. The default value for the output device is TAPE when the input device is DISK.

When the output file is on tape, specify the file sequence number and the label type as follows:

FSEQ — The sequence number of the file on the tape. You must provide the file sequence number for both labeled and unlabeled tapes.

LABEL — The label type of the output tape volume. Specify AL for ANSI, IL for IBM, or N for NO label. TAPECOPY enters the library and file names into the output tape label using the format library name.file name. (For example, Userlib.Userfil.) For a tape with an IBM label, TAPECOPY performs an automatic translation of the label into EBCDIC.

15.5 DEFINING AN OUTPUT DISK FILE

If the output file resides on a disk, specify the following output disk file parameters. (Default values are taken from the input file.)

NRECS — The approximate number of records in the output file. The specified number must be large enough to provide sufficient disk space for the file. This field is not displayed when the input file is a disk file.

FILEORG — The organization of the file. A response of C indicates a consecutive file; I represents an indexed file; X is a program file; P indicates a print file. The default response is C.

RECFORM — The format of the records in the file. A response of F indicates fixed-length records; a response of V indicates variable-length records. The default response is V.

COMPRESS — Selects file compression. File compression can save space. A response of Y compresses the file; a response of N does not compress the file. The default response is Y. For more information about file compression, refer to the VS Operating System Services Reference.

FILECLAS — The file protection class. This code determines which users are allowed to access this file. Specify A to Z, #, @, $, or blank. For more information about file protection classes, refer to the VS Programmer's Introduction.

RETAIN — The file retention period. Specify the number of days the file is protected from being scratched or renamed.

If the output file is indexed, specify the following parameters:

KEYLEN	The length of the key field
KEYPOS	The starting byte position of the key field
IPACK, DPACK	The packing density for index and data blocks. The allowed range is 1 percent to 100 percent. The default response is 100 percent. For more information about packing densities, refer to the VS Operating System Services Reference.

If the specified output volume is not mounted, a message is displayed directing you to mount or respecify the volume name.

After you enter the output parameters, the output file is created. When the copy completes, TAPECOPY indicates how many records are in the output file, and allows you to run TAPECOPY again by pressing PF1 or terminate processing by pressing PF16.

15.6 MULTIVOLUME TAPE COPYING

TAPECOPY can copy a labeled tape file that resides on two or more volumes. The utility does not allow you to copy less than the full set. TAPECOPY displays a message when the end of a volume is reached. You are then requested to mount the next input volume by specifying the volume name and device number. TAPECOPY checks the newly mounted volume for the correct volume sequence number, and then continues reading the input file.

Multivolume output is handled similarly to multivolume input. TAPECOPY displays a message when it reaches the end of an output tape. You are requested to mount the next volume by specifying the volume name and device number. The write operation continues on the next volume.

15.7 A SAMPLE TAPECOPY PROCEDURE

You can control TAPECOPY processing through the VS Procedure language. Appendix A contains TAPECOPY GETPARMs. Refer to the VS Procedure Language Reference for details about procedure syntax.

The following procedure copies a disk file to the twelfth file on an AL-tape. Character set conversion is not selected. Because the input disk file characteristics (record and block lengths) are appropriate, the default values obtained for the TAPEFILE prname are used by the procedure. The procedure exits TAPECOPY when the file is copied.

```
PROCEDURE
RUN TAPECOPY
ENTER INPUT FILE=KANGAROO, LIBRARY=ZOOLIB, VOLUME=SYSTEM, DEVICE=DISK
ENTER OUTPUT FILE=KANGAROO, LIBRARY=ZOOLIB, VOLUME=MYTAPE,
        DEVICE=TAPE, FSEQ=12, LABEL=AL
ENTER TAPEFILE
ENTER EOJ 16
RETURN
```

CHAPTER 16
THE TAPEINIT UTILITY

16.1 INTRODUCTION

Before you can store data on a tape volume, you must initialize the tape volume. The functions provided by TAPEINIT are as follows:

- Initialize new (and reinitialize old) tape volumes for both 7- and 9-track tapes. All information previously stored on the volume is destroyed.
- Initialize an IBM labeled tape volume (9-track only).
- Mount a tape that is not initialized rather than using the standard mount procedure.

An overview of TAPEINIT processing is provided in Figure 16-1.

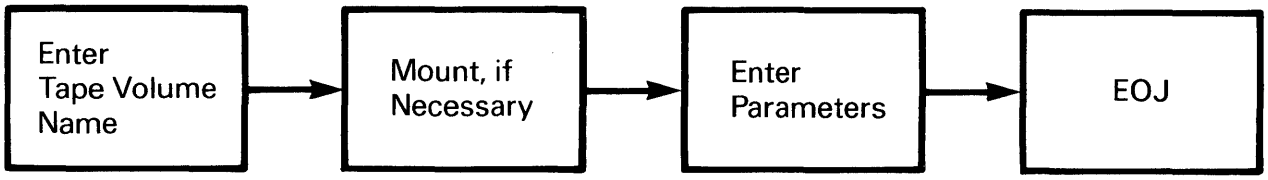


Figure 16-1. TAPEINIT Processing.

16.2 SPECIFYING THE VOLUME

The Volume Definition screen requests you to enter the name of the tape volume you want to initialize. If you have not mounted the tape volume, press PF4 to initiate the mount procedure. The Mount screen then displays a list of the available devices. Type in a device number and press ENTER. TAPENINT processing continues after the completion of the mount procedure. If you have mounted the tape, the Mount option is not displayed. If you mount a 7-track tape, TAPEINIT processing continues, as described in Section 16.3. If you mount a 9-track tape, subsequent TAPEINIT processing is described in Section 16.4.

16.3 7-TRACK TAPE INITIALIZATION

After you mount a 7-track tape volume, TAPEINIT requests you to enter the following initialization parameters. Note that 7-track tapes are nonlabeled.

PARITY — Selects the error-checking system used for the tape's data. If you accept the default response of EVEN, the parity bit is set to 0 or 1, as required. This yields an even number value when all bits in the character are totaled. If you set PARITY = ODD, the total of all bits in the character is an odd number.

DENSITY — Determines the number of bits per inch (bpi) to be stored on each track of the tape you want to initialize. Data is stored at 800 bpi (the default).

TAPEINIT initializes the tape at this point. After the tape is initialized, you can start TAPEINIT processing again by pressing PF1 or terminate processing by pressing PF16.

16.4 9-TRACK TAPE INITIALIZATION

After you mount a 9-track tape volume, TAPEINIT permits you to specify the following three label attributes:

LABEL TYPE — Specify one of the following label types. The default response is AL.

- AL — The Wang standard (ANSI) label type
- IL — An IBM label type
- NL — No label is placed on the tape

NOTE

For the IBM label type, only IBM OS tape labels are supported.

OWNER — The person responsible for tape contents. You can leave this option blank, but you must fill in this option when you wish to restrict access rights. The owner name becomes part of the label. The owner name is up to 14 characters long for AL tapes, and up to 10 characters long for IL tapes.

DENSITY — The recording density for this particular tape. Density is restricted by the tape drive model. Dual density drives support densities of 800 or 1600 bpi. Tridensity drives support densities of 800, 1600 or 6250 bpi.

After you specify the label attributes, TAPEINIT initializes the tape. When the tape is initialized, you can start TAPEINIT processing again by pressing PF1 or terminate processing by pressing PF16.

16.5 A SAMPLE TAPEINIT PROCEDURE

You can control TAPEINIT processing through the VS Procedure language. You can specify all TAPEINIT options through a procedure. Procedure language MOUNT and DISMOUNT statements, however, cannot be embedded in a RUN, ENTER (or DISPLAY) sequence; they must precede the RUN statement or follow the last ENTER statement. A TAPEINIT mount operation cannot be embedded in the procedure because TAPEINIT uses a respecification GETPARM. Appendix A contains a list of TAPEINIT GETPARMs. Consult the VS Procedure Language Reference for details about procedure syntax.

The following procedure logically mounts and initializes a 9-track standard label tape with a density of 1600 bpi and an owner specified as ME. The procedure logically dismounts the tape when TAPEINIT processing ends. Because all TAPEINIT processing is automated, you (or the system operator if the procedure is run in background mode) need only physically mount and dismount the tape on the tape drive.

```
PROCEDURE
MOUNT TAPE MYTAPE ON 040 WITH NO LABEL FOR EXCLUSIVE USAGE
RUN TAPEINIT
ENTER INPUT VOLUME=MYTAPE
ENTER TAPE LABEL=AL, OWNER=ME, DENSITY=1600
ENTER EOJ 16
DISMOUNT TAPE MYTAPE
RETURN
```

CHAPTER 17
THE TRANSL UTILITY

17.1 INTRODUCTION

The TRANSL utility enables you to translate files from one character set to another, using a standard ASCII-to-EBCDIC, EBCDIC-to-ASCII, or user-defined translation table. You must identify an input file, select a translation function, define field manipulation options (indicating the fields from the input file contained in the output file), and determine the output file specifications. The TRANSL utility creates a translated output file, enabling you to transfer files among systems with different character sets, or to copy a file and to change certain characters. TRANSL performs the following functions:

- Translates files from ASCII-to-EBCDIC or EBCDIC-to-ASCII. When you select either of these functions, translation is performed using a standard conversion table.

- Translates files to or from a nonstandard character set. A user-defined translation table performs the translation. This function enables you to create, retrieve, or modify a translation table, as well as save the table for subsequent file translation.

- Performs field manipulation functions, bypassing all translation activity. These functions enable you to rearrange, insert, or delete input record fields to create a modified output record.

- Performs both field manipulation functions and translation functions in a combined operation. This function enables you to rearrange, insert, or delete input record fields as a translated output file is created.

- Performs selective translation of user-specified multiple record types. You specify indicator characters that identify each record type.

Operating the TRANSL utility involves the following steps. You can also control processing through the VS Procedure language, as described in Section 17.6.

- Specify the input file and a TRANSL function. You can optionally create, select, or modify a user-defined translation table that performs translation functions.

- Specify multiple record types (optional) to selectively translate the records of a file.

- Define field manipulation options to enable the utility to perform translation functions.

- Indicate the output file specifications.

An overview of TRANSL processing is provided in Figure 17-1.

17.2 DEFINING THE INPUT

The initial screen of the TRANSL utility is the Input Definition screen. You specify the file, library, and volume name of the file to be translated or reorganized. You also specify the following parameters:

TYPES Permits TRANSL to selectively translate a file's records. You specify multiple record types by indicating YES in the TYPES field. The Types screen is displayed after the Input Definition screen is processed, unless you set CODE=OPTION. If you request the TABLE option the Types screen is displayed after the Outtable screen. The TYPES option defaults to NO.

CODE Determines the type of translation to be performed. You can select one of the following codes:

ASCII Translates ASCII code to EBCDIC code.

EBCDIC Translates EBCDIC code to ASCII code.

TABLE Allows you to create a translation table, or to access and modify a previously defined table. This option converts files to or from nonstandard codes.

NONE Allows you to use the field manipulation features of the TRANSL utility without any character translation. When you specify NONE (no translation) in the CODE field of the Input Definition screen, all translation table activities are skipped, and the Options screen is displayed.

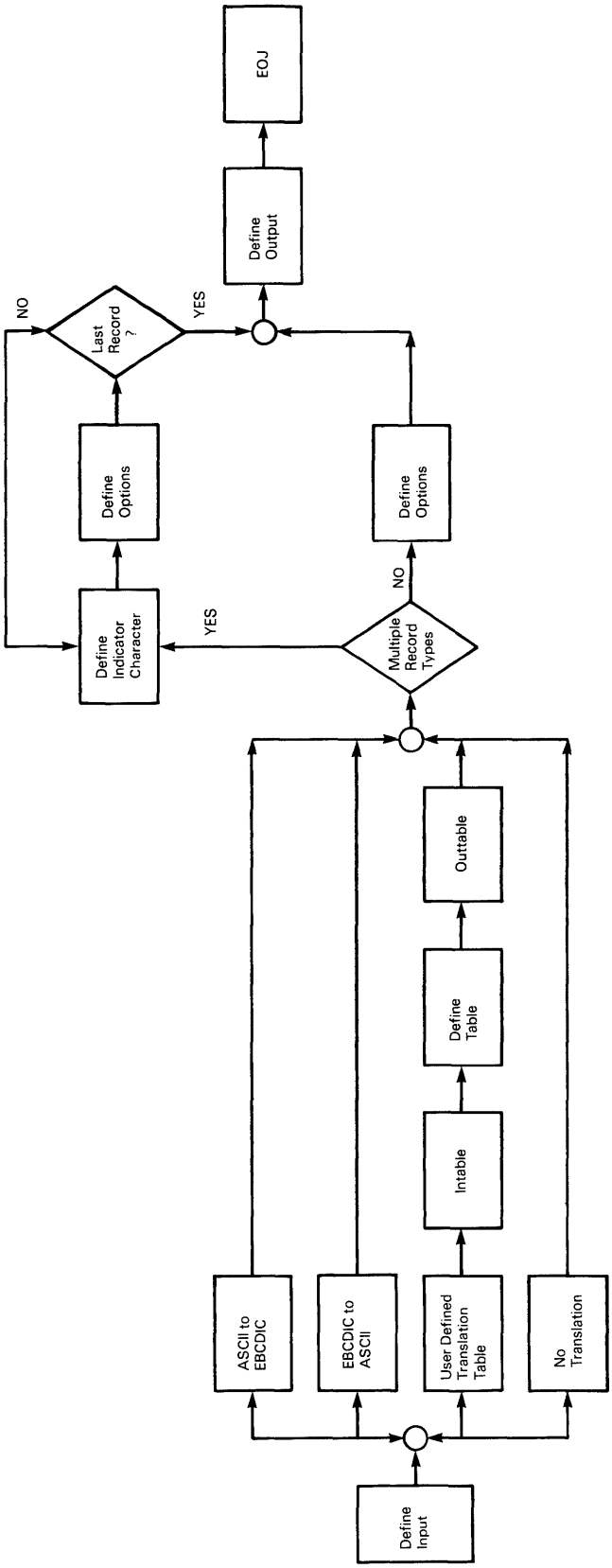


Figure 17-1. TRANSL Processing

You need not specify the whole word for each option; the initial letter is sufficient. The CODE option defaults to EBCDIC.

17.2.1 Performing a Standard Translation

If you select the ASCII-to-EBCDIC conversion process, TRANSL automatically translates each ASCII character contained in the input file to the corresponding EBCDIC character. Both ASCII-to-EBCDIC and EBCDIC-to-ASCII conversions are processed by use of a predefined conversion table. You also can have a file translated to or from nonstandard code by defining a unique translation table. (Refer to Subsection 17.2.2.) If you indicate a standard code translation (assuming that you do not select Multiple Record Types), the Options screen is displayed next. (The Options screen is discussed in Section 17.4.) You must identify the output file's field attributes on the Options screen before the utility can create an output file and perform the selected translation.

17.2.2 Defining a Translation Table

You can create or select a user-defined translation table by specifying TABLE in the CODE parameter of the Input Definition screen. If you specify TABLE, the Intable screen is displayed. To create a new translation table, you need not specify the parameters of the Intable screen. You press ENTER to process the Intable screen and to display the default Table screen. The Outtable screen appears after the translation table is processed. The Outtable screen allows you to assign the table to a file. To access a previously defined translation table, you specify the file, library, and volume in which the table is contained as the parameters of the Intable screen. When you process the Intable screen, the specified Table screen is displayed.

A Translation Table is a 256-byte string used to convert one character set to another. When you create a new table or modify an existing table, the Table screen displays a table containing a complete 256 two-digit set of hexadecimal values. If you create a new translation table, a default table is displayed. The default table contains each hexadecimal value assigned to itself; you need to indicate only altered values. The default table is shown in Figure 17-2. The hexadecimal values 00 through 0F are contained in the first row of the table. Two hexadecimal digit locations are reserved for each value.

For example, to change all ASCII uppercase and lowercase characters to uppercase, you change the hexadecimal values of HEX code 61 through 7A to 41 through 5A, as shown in Figure 17-3. Each character value in the lowercase ASCII alphabet is translated into the corresponding uppercase character.

```
***  MESSAGE T002 BY TRANSL

                INFORMATION REQUIRED BY  PROGRAM  TRANSL
                          TO DEFINE TABLE


Please modify as desired and/or select from choices listed

    HEX00#0F = 00010203 04050607 08090A0B 0C0D0E0F
    HEX10#1F = 10111213 14151617 18191A1B 1C1D1E1F     -----------------
    HEX20#2F = 20212223 24252627 28292A2B 2C2D2E2F
    HEX30#3F = 30313233 34353637 38393A3B 3C3D3E3F      Please select:
    HEX40#4F = 40414243 44454647 48494A4B 4C4D4E4F
    HEX50#5F = 50515253 54555657 58595A5B 5C5D5E5F      ENTER - Continue
    HEX60#6F = 60616263 64656667 68696A6B 6C6D6E6F
    HEX70#7F = 70717273 74757677 78797A7B 7C7D7E7F      PF1 - Return to
    HEX80#8F = 80818283 84858687 88898A8B 8C8D8E8F       INTABLE screen
    HEX90#9F = 90919293 94959697 98999A9B 9C9D9E9F
    HEXA0#AF = A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF
    HEXB0#BF = B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF      PF14 - Further
    HEXC0#CF = C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF          information
    HEXD0#DF = D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF
    HEXE0#EF = E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF     -----------------
    HEXF0#FF = F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF
```

Figure 17-2. The Default Translation Table

```
***  MESSAGE T002 BY TRANSL

                INFORMATION REQUIRED BY  PROGRAM  TRANSL
                          TO DEFINE TABLE


Please modify as desired and/or select from choices listed

    HEX00#0F = 00010203 04050607 08090A0B 0C0D0E0F
    HEX10#1F = 10111213 14151617 18191A1B 1C1D1E1F     -----------------
    HEX20#2F = 20212223 24252627 28292A2B 2C2D2E2F
    HEX30#3F = 30313233 34353637 38393A3B 3C3D3E3F      Please select:
    HEX40#4F = 40414243 44454647 48494A4B 4C4D4E4F
    HEX50#5F = 50515253 54555657 58595A5B 5C5D5E5F      ENTER - Continue
    HEX60#6F = 60414243 44454647 48494A4B 4C4D4E4F
    HEX70#7F = 50515253 54555657 58595A7B 7C7D7E7F      PF1 - Return to
    HEX80#8F = 80818283 84858687 88898A8B 8C8D8E8F       INTABLE screen
    HEX90#9F = 90919293 94959697 98999A9B 9C9D9E9F
    HEXA0#AF = A0A1A2A3 A4A5A6A7 A8A9AAAB ACADAEAF
    HEXB0#BF = B0B1B2B3 B4B5B6B7 B8B9BABB BCBDBEBF      PF14 - Further
    HEXC0#CF = C0C1C2C3 C4C5C6C7 C8C9CACB CCCDCECF          information
    HEXD0#DF = D0D1D2D3 D4D5D6D7 D8D9DADB DCDDDEDF
    HEXE0#EF = E0E1E2E3 E4E5E6E7 E8E9EAEB ECEDEEEF     -----------------
    HEXF0#FF = F0F1F2F3 F4F5F6F7 F8F9FAFB FCFDFEFF
```

Figure 17-3. A Modified Translation Table

When you process the Table screen, the Outtable screen is displayed. The Outtable screen requests you to specify a file, a library, and a volume in which the Translation Table is saved for future use by TRANSL. You can press PF16 to continue processing without saving the table, or press ENTER to process the file assignment. If a table already exists in the file, library, and volume specified on the Outtable screen, you can press PF2 to delete the existing table and then replace it with the newer one.

After you process the Outtable screen, the Options screen is displayed. You must identify the output file's field attributes on the Options screen, before the utility can create an output file and perform the selected translation. (Refer to Section 17.4.)

17.2.3 Enabling Field Manipulation Without Translation

You can bypass all translation activities and only perform field manipulation functions by specifying NONE in the CODE field of the Input Definition screen. When you enter NONE on the Input Definition screen (assuming that Multiple Record Types is not indicated), the Options screen is displayed. The Options screen requests you to make field manipulation selections by indicating the output file's field attributes. Field manipulation options are discussed in Section 17.4.

17.3 DEFINING MULTIPLE RECORD TYPES

If you specify YES for the TYPES option on the Input Definition screen, the Types screen is displayed after you process the Input Definition screen. (Refer to Figure 17-4.) The Types screen is displayed when you specify the ASCII-to-EBCDIC, EBCDIC-to-ASCII, or No Translation function and indicate Multiple Record Types. However, if you specify the Table option, the Types screen is displayed after the Outtable screen.

When you indicate multiple record types, you select indicator characters to identify various types of records. TRANSL reorganizes each record type in a different manner. An Options screen is displayed after each Types screen. The Options Screen enables you to specify field manipulation options and to define field attributes for each multiple record type.

```
***  MESSAGE 1007 BY TRANSL

                INFORMATION REQUIRED BY  PROGRAM  TRANSL
                          TO DEFINE TYPES



Please define the character(s) and location(s) to be used in identifying the
record types in the file. Each character may be specified as one ASCII
character or two HEX digits.
Leave this screen blank to indicate the end of the different record types.

Record type 01 will be identified by:
                CHAR1    = ** in COLUMN1  = **** ,  SWITCH1  = YES
            and CHAR2    = ** in COLUMN2  = **** ,  SWITCH2  = YES
            and CHAR3    = ** in COLUMN3  = **** ,  SWITCH3  = YES

          The above indicators are as found in which character set ?
                    CHARSET  = INPUT*    (INPUT or OUTPUT)

Note:  SWITCH(1,2,3) = "YES", Means that this character's presence will
       be used to identify this record type.  SWITCH(1,2,3) = "NO", Means
       That its absence will be used to identify this record type.
```

Figure 17-4. The Types Screen

The Types screen requests that you specify the character(s) you use to identify each record type. You can identify a record type by either the presence or absence of the specified character(s), known as the indicator character(s). You must supply the following parameters before TRANSL processing continues:

CHAR — A character or value used to distinguish record types. You can specify one ASCII character or two hexadecimal digits as the indicator character to identify each record type.

COLUMN — The column number of the input record in which CHAR appears.

SWITCH — Specifies whether a record type can be identified by the presence or absence of the character indicated in the CHAR field. YES indicates that the presence of the character identifies the record type; NO indicates that the absence of the character identifies the record type.

CHARSET — Determines which character set, INPUT or OUTPUT, contains the specified indicator CHAR.

The Types screen is displayed recursively with the Options screen, so that you can define the field attributes of each record type defined on each Types screen. You display the Output Definition screen by pressing ENTER and processing a Types screen in which all fields are left blank.

17.4 SELECTING FIELD MANIPULATION OPTIONS

You must define field manipulation selections on the Options screen before the utility performs the selected translation. Field manipulation options specify the order, length, and data type of each field to be contained in the output file. You can also indicate deletion, insertion, or rearrangement of input file fields to be contained in the output file. The Options screen is displayed in Figure 17-5.

```
***  MESSAGE 0001 BY TRANSL

                INFORMATION REQUIRED BY  PROGRAM  TRANSL
                          TO DEFINE OPTIONS


Press ENTER after defining the necessary fields

                                                                  (Code I,D
Field attributes:                                               or leave blank)
       Position(Input)     Length(Bytes)      TYPE  (B or C)   Insert   Delete
FLD1 :  POST01   = ****    LENGTH01 = ****     TYPE01   = *      SWITCH01 = *
FLD2 :  POST02   = ****    LENGTH02 = ****     TYPE02   = *      SWITCH02 = *
FLD3 :  POST03   = ****    LENGTH03 = ****     TYPE03   = *      SWITCH03 = *
FLD4 :  POST04   = ****    LENGTH04 = ****     TYPE04   = *      SWITCH04 = *
FLD5 :  POST05   = ****    LENGTH05 = ****     TYPE05   = *      SWITCH05 = *
FLD6 :  POST06   = ****    LENGTH06 = ****     TYPE06   = *      SWITCH06 = *
FLD7 :  POST07   = ****    LENGTH07 = ****     TYPE07   = *      SWITCH07 = *
FLD8 :  POST08   = ****    LENGTH08 = ****     TYPE08   = *      SWITCH08 = *

*Options for field type: B=Binary,C=Character
Notes: If all 8 fields are used the option to define more fields will appear
  Selecting delete(D) causes this input field not to be transferred to output
  Selecting insert(I) creates a new field in output (Leave position zero )
```

Figure 17-5. The Options Screen

If the input file is a fixed-length record file, you must account for all bytes of the input record on the Options screen. If necessary, the Delete option is available to indicate that a specified field is not to be included in the output record. If the file is a variable-length record file, and you do not want to include a specified field in the output record, you can omit the field on the Options screen. The field is not included in the output record.

The field definitions on the Options screen are arranged in the order in which they appear in the output record. FLD1 is the first field in the output record, FLD2 is the second output field, and so on. The table displays eight fields at a time. When you fill in the eight fields, the program displays another eight fields. You can specify a maximum of 24 fields. You specify the following parameters on the Options screen:

POST — The position of the field as it is contained in the input record. For example, if a field that begins at byte 11 in the input record is to be the first field in the output record, FLD1 has a position value of 11. If the output record is to have the same order as the input record, the fields' positions must be listed in the order in which they appear in the input record. Fields of the same data type, except zoned fields, can be grouped. (The first position of a record is considered to be 1, not 0.)

LENGTH — The input field length in bytes.

TYPE — The data type of the input field. The data type can be Binary (B), Character (C), Packed (P), or Zoned (Z). Binary fields are not translated. When Zoned decimal characters are translated, the sign is automatically included in the low-order byte of the field.

SWITCH — Enables you to indicate insertion or deletion of fields to be contained in the output file. The SWITCH field defaults to blanks. The default has no effect on either the input or output field. The SWITCH option can also have the following values:

D — DELETE indicates that the input field is not to be copied to the output file. This option is significant only for fixed-length files. If you do not account for all bytes of a fixed-length input record, the input and output record lengths differ, and an Error screen appears. You press PF1 to redesign the field attributes, or press PF16 to terminate the TRANSL utility. This does not apply to variable-length records.

I — INSERT indicates that a new field of a specified length is to be inserted in the output file. You must specify the position of the field to be inserted as a blank or zero, (it has no position in the input file). To insert blanks in bytes 21 to 30 in the output file, you indicate the fields that occupy bytes 1 to 20. The next field is given a POSITION of 0, a LENGTH of 10, and a SWITCH option of I (INSERT). If TYPE is Character, a field of blanks is inserted; If TYPE is Binary, a field with a value of Hex '00' is inserted. If TYPE is Packed or Zoned, a field with a value of decimal zero is inserted.

When an indexed file is translated, the Indexed screen is displayed. You can change the file's primary key position and length in the KEYPOS and KEYLEN fields. If you want these values to be identical to those of the input file, leave them blank. No duplicate keys are allowed because TRANSL cannot create an alternate-indexed output file.

To process the Options screen and display the next screen, you press ENTER. However, if you indicate multiple record types on the Input Definition screen, you also must specify the record types on the Types screen. The Types screen appears after the Input Definition screen:

17.5 DEFINING THE OUTPUT FILE

When the TRANSL options are processed, the Output screen is displayed. The Output screen requests you to specify the output file parameters. You supply the file, library, and volume names of the output file, plus the following parameters:

RECORDS — Indicates the number of records obtained from the input file.

RETAIN — Indicates that the file is protected from deletion. A file can be protected for a maximum of 999 days and a minimum of zero days.

FILECLAS — The protection class of the output file. The protection class determines which users can select the file. The following file protection classes are available: #, $, @, blank, A to Z. (For more information on fileclass options, consult the VS Programmer's Introduction.)

RELEASE — Indicates whether or not unused storage extents, previously allocated for data, can be released for use by other files. RELEASE is a YES or NO indicator. RELEASE defaults to the input file specification.

DEVICE — Indicates the device to which the output file is transferred. You set DEVICE to DISK to copy the file to diskette or hard disk. If the output file is a print file, you indicate PRINTER as the device. (The TRANSL utility does not copy to tape.) DEVICE defaults to DISK.

When TRANSL activity is complete, an output file is created, containing the translations and manipulations you specify. The input file still exists in its original form. The EOJ screen is displayed, and you can either exit from the TRANSL utility by pressing PF16 or rerun the utility by pressing PF1.

17.6 A SAMPLE TRANSL PROCEDURE

You can control TRANSL processing through the VS Procedure language. You can specify TRANSL options and output file organization. A complete list of TRANSL GETPARMs is provided in Appendix A. Consult the VS Procedure Language Reference for details about procedure syntax.

The following sample procedure converts a 120-byte indexed file, containing only character fields, from the ASCII to the EBCDIC character set. The procedure specifies the input file and does not select output file reorganization. The procedure displays a set of default file parameters for the output file and finally exits the utility.

```
PROCEDURE
RUN TRANSL
ENTER INPUT FILE=DATASCII, LIBRARY=TRANSME, VOLUME=SYSTEM, CODE=EBCDIC
ENTER OPTIONS POST01=0001, LENGTH01=0120, TYPE01=C
ENTER INDEXED
DISPLAY OUTPUT FILE=DATEBCDI, LIBRARY=TRANSED, VOLUME=SYSTEM
ENTER EOJ 16
RETURN
```

CHAPTER 18
THE VERIFY UTILITY

18.1 INTRODUCTION

The VERIFY utility performs a comprehensive check on the validity of the various components of an indexed file and identifies any damage. You should be familiar with the internal structure of VS indexed files before attempting to use VERIFY. For more information on the internal structure of VS indexed files, refer to the VS Operating System Services Reference. The VERIFY utility performs the following functions:

- Checks either a single indexed file, all the indexed files in a library, or all the indexed files in all the libraries on a volume
- Validates only the primary index structure, or both the primary index structure and all alternate indices
- Validates the consistency between VTOC entry (FDR1) fields and the file data structures and Index structures
- Validates the consistency between entries in the Alternate Index Descriptor (AXD1) block and file index structures
- Validates the consistency between the primary and alternate index structures of indexed files
- Validates the completeness of the alternate index structure against the data structure
- Validates the physical integrity of the file
- Optionally displays error messages as the system detects errors
- Displays the summary statistics of the file, at the end of processing each file, and recommends recovery procedures for the file if the system detects damage
- Produces a printed copy containing summary information on the file characteristics and, if the system detects any errors, an error report is produced.

In the process of validating a file, VERIFY detects and automatically records all permanent I/O errors. VERIFY is designed to continue processing even if the input file contains significant structural errors, in order to extract as much information as possible about the file, library, or volume. If, however, you request an error display, you can exit VERIFY processing at any point after the utility displays an error.

Use of the VERIFY utility, in conjunction with the BACKUP utility or after a system crash, can aid System Operations by ensuring quick discovery of any indexed file damage.

The operation of the VERIFY utility is as follows and is documented in the indicated sections. You can run the VERIFY utility through a procedure by using the VS Procedure language.

- You specify the input to be verified and indicate whether or not all encountered errors are to be displayed on the workstation screen.
- VERIFY performs tests on the user-specified input, and if you specify, VERIFY displays any error messages and a summary of the test results of the file.
- VERIFY displays end-of-job results.
- You specify a hard copy of the VERIFY results. The utility prints a Summary Report on the file characteristics and if the system detects any errors, an Error Report is processed.

An overview of VERIFY processing is provided in Figure 18-1.

18.2 SPECIFYING THE INPUT OPTIONS

To begin processing, the VERIFY utility displays the Options screen, and you specify the input you want to verify. VERIFY has three input options: a single indexed file (FILE), all the indexed files in a library (LIBRARY), or all the indexed files in all libraries on a volume (VOLUME). If you specify FILE as the input and the file is not an indexed file, VERIFY displays a message. You can either respecify the input file or cancel processing. If you do not have access authorization for a specified file, VERIFY displays an error message, and the file is not verified. You choose between respecifying the file (if the FILE input option is chosen), skipping the file and going to the next file (if the LIBRARY or VOLUME input option is chosen), or cancelling VERIFY processing.

During a library or volume verification, the files are verified in the order in which they appear in the VTOC, not in alphabetical or numerical order. You do not need to know the name(s) of any indexed file(s) during library or volume verify. Also, you do not need to know the name of any library during volume verification. VERIFY finds any indexed files present in a library or on a volume.

The FILE, LIBRARY, or VOLUME you choose for verification is referred to in this chapter as the "input unit." When you specify the input unit, you can choose to verify either all the indices (primary and alternate) verified or the primary index only. You also specify whether or not error messages are to be displayed at the workstation.

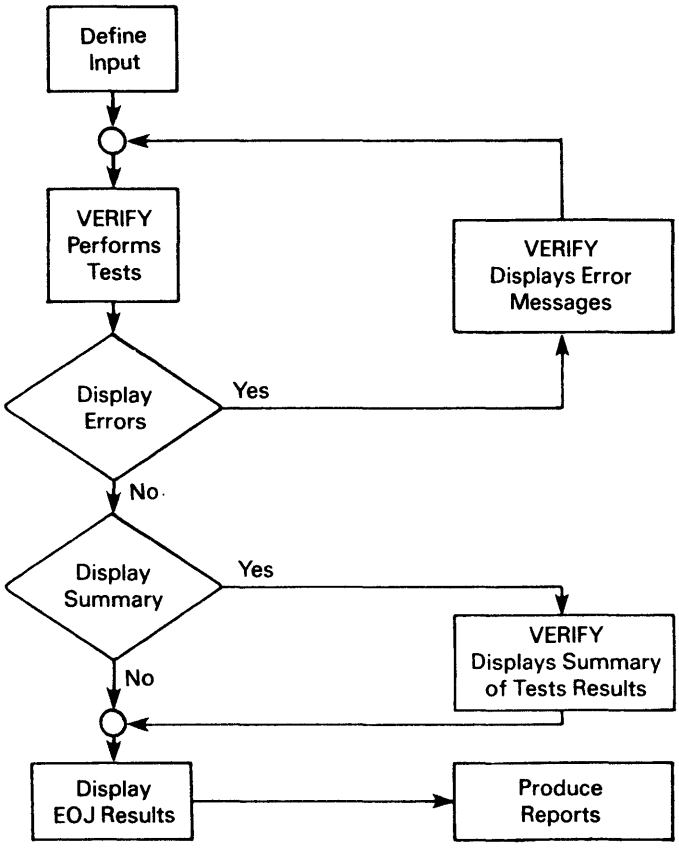


Figure 18-1. VERIFY Processing

The Options screen is shown in Figure 18-2.

```
***  MESSAGE 0001 BY VERIFY

                INFORMATION REQUIRED BY  PROGRAM  VERIFY
                           TO DEFINE OPTIONS


                        ***  WANG VS VERIFY Program  ***
          Please specify the following information.

Depending  on the options chosen, program will verify:

RANGE     = FILE***              FILE    - File in the library on the volume.
                                 LIBRARY - All indexed files in the library.
                                 VOLUME  - All indexed files on the volume.

FILE      = ********   in LIBRARY  = ******** on  VOLUME    = ******


VERIFY    = ALL****  INDICES     ALL     - Primary and alternate indices.
                                 PRIMARY - Primary index only.


ERROR  DISPLAY  = YES            YES - All errors will be displayed.
                                 NO  - Errors will not be displayed.
```

Figure 18-2. VERIFY Options Screen

The input options are as follows:

RANGE — Specifies the input unit to be verified. Three options are available: FILE, LIBRARY, or VOLUME. If you specify RANGE=FILE, a single file is verified. If you specify RANGE=LIBRARY, all the indexed files in the library are verified. If you specify RANGE=VOLUME, all the indexed files in all libraries on the volume are verified. You do not need to specify the whole word for each option; the initial letter is sufficient. RANGE defaults to FILE.

FILE — The name of the input file to be verified. You specify the file name if RANGE is FILE. If RANGE is LIBRARY or VOLUME, no entry is made for this parameter. There is no default for FILE.

LIBRARY — The name of the input library to be verified. You specify the library name if RANGE is FILE or LIBRARY. If you specify RANGE=VOLUME, the library name remains blank. LIBRARY defaults to your INLIB, previously set through Set Usage Constants of the Command Processor.

VOLUME — The name of the input volume to be verified. VOLUME defaults to your INVOL, previously set through SET Usage Constants of the Command Processor.

VERIFY — Specifies the index(es) to be verified. Two options are available: ALL or PRIMARY. If you specify VERIFY=ALL, the primary index and all alternate indices are verified. If you specify VERIFY=PRIMARY, only the primary index is verified. VERIFY defaults to ALL.

ERROR DISPLAY — Determines whether or not error messages are displayed at the workstation. If you specify ERROR DISPLAY=YES, error messages, as they are detected, and summary statistics of each file are displayed on the workstation. If you specify ERROR DISPLAY=NO, the error messages and summary statistics are not displayed, and you must rely on the Summary Report to identify errors. (Refer to Section 18.7 for more information about the Summary Report.) ERROR DISPLAY defaults to YES. (Refer to Section 18.4 for more information on error message displays.)

18.3 TESTING FILES

After you enter the Options screen, VERIFY performs a comprehensive check on the validity of each indexed file. The tests for each file, which are documented in the indicated sections, are as follows:

- Validation of the VTOC entry (FDR1) fields. (Refer to Subsection 18.3.1.)
- If alternate indices are present, validation of the entries in the Alternate Index Descriptor (AXD1) block. (Refer to Subsection 18.3.2.)
- Validation of the consistency between the primary and alternate index structures of indexed files. (Refer to Subsection 18.3.3.)
- Validation of the completeness of the alternate index structure. (Refer to Subsection 18.3.4.)
- Validation of the physical integrity of the file. (Refer to Subsection 18.3.5.)

18.3.1 Validating the VTOC Entry (FDR1) Fields

To validate the VTOC entry (FDR1) fields, VERIFY examines the FDR1 fields to determine if the pointers to the root index block and the first data block are within the file: that is, are less than the pointer to the last block. VERIFY also determines if the number of index levels specified is legal; the maximum number of index levels is 4.

If both pointers, or one pointer and the number of index levels are invalid, the VTOC entry for the file is invalid, and you cannot gain access to the file. Also, neither the index structure nor the data chain of the file can be verified.

If only one pointer is invalid and the number of index levels are legal, the program attempts to verify the integrity of the data chain. Depending on the result of this check, VERIFY displays a message informing you that the data chain is either broken or intact. In either case, VERIFY cannot check the validity of the index structure. If you press PF16, the verification of the file ends. Also, if the input unit is a library or volume, VERIFY proceeds to the next file.

VERIFY also checks that the number of data records specified in FDR1 is the same as the number actually counted by VERIFY. If a discrepancy appears, an error code is displayed. (Refer to Subsection 18.7.1 for the meaning of the error code.) This display informs you of the discrepancy and provides both the specified count and the actual count. You can continue to verify the file (press ENTER), or end verification (press PF16). If the input unit is a library or volume, VERIFY proceeds to the next file.

18.3.2 Validating the Entries in the Alternate Index Descriptor Block

To validate the entries in the Alternate Index Descriptor (AXD1) block, VERIFY checks each alternate index to determine if the pointer to the alternate index root block and the pointer to the first alternate index block are within the file, that is, are not greater than the pointer to the last block. At the same time, VERIFY also checks that the number of index levels, in each alternate index, is feasible.

If one or more of these pointers is invalid, VERIFY informs you that the AXD1 block contains invalid data and that the alternate index cannot be verified. The program then attempts to verify the remaining alternate indices.

18.3.3 Validating Primary and Alternate Index Structure Consistency

After the FDR1 and AXD1 blocks are checked, VERIFY performs a series of consistency checks on the primary and alternate index structures. The checks ensure that the following conditions exist:

- The chain pointer in the last block on each index level is equal to hexadecimal 'FFFFFF'.
- The largest key in each lower level index and data block is equal to the key in the corresponding entry in the index structure, excluding the largest key in the last block at each level, which must equal hexadecimal 'FFFFFF'.
- All keys are in ascending order at data level.
- All keys are in ascending order at each index level, except for the lowest level of alternate index structures where duplicate alternate keys are allowed.

- If two or more alternate keys are equal, the corresponding primary keys are in ascending order.

When the system first reports a pointer error or a key discrepancy, VERIFY displays an error message. You can either continue to the next check by pressing ENTER, obtain a more detailed error display by pressing PF1, or end verification of the file by pressing PF16.

If a pointer inequality is detected in the primary index structure, VERIFY assumes that the index pointer is correct and tries to verify the file on this basis. If this assumption leads to another discrepancy, VERIFY tries the chain pointer. If this attempt also fails, VERIFY informs you that the primary index structure cannot be verified. If you specify a library or a volume as input, VERIFY proceeds to the next file.

Pointer inequalities in the alternate index structures are processed in the same way as primary index structures, with one exception; if a second discrepancy occurs, you are informed that this alternate index cannot be verified. VERIFY then proceeds to the next alternate index structure.

18.3.4 Validating the Completeness of the Alternate Index Structures

If the alternate indices are structurally consistent, VERIFY tests them to ensure they are complete. Tests on the alternate indices are run concurrently with the check on the structural consistency of the primary index. If any errors are found in the primary index, testing the completeness of the alternate index structures is suspended. Errors found by this check are recorded in the Error Report, but are not displayed individually. Errors are counted and recorded as errors in the alternate indices.

VERIFY checks to ensure that the following conditions exist:

- An alternate path contains an alternate key/primary key pair, only one alternate key/primary key pair exists in that alternate path where the primary key is the same as in the record. The following example illustrates an alternate path with valid key pairs (alternate path 1) and an alternate path with invalid key pairs (alternate path 2):

Alternate path 1		Alternate path 2	
alternate key	primary key	alternate key	primary key
001	123	001	123
002	123	001	123

- An item in the alternate index contains a primary key, VERIFY checks that a record with this primary key exists and that the record access mask indicates that you can select it through the alternate index.

- The alternate key in the index is equal to the alternate key in the record.

18.3.5 Validating the Physical Integrity of the File

In the process of verifying the file, VERIFY reads all index and data blocks and detects all permanent I/O errors. If the system detects an I/O error when reading a primary index block, VERIFY attempts to check the integrity of the data chain. VERIFY then informs you that the index structure of the file cannot be verified and, depending on the outcome of the data chain check, that the data chain is either intact or broken.

If an I/O error is found in an alternate index block, VERIFY informs you that this alternate index cannot be verified, and proceeds to the remaining alternate index paths (or structures).

If an I/O error is found in a data block, VERIFY displays an error message, the number of the bad block, and if appropriate, the recommended corrective action. You can either continue to the next block in the same file (if any) by pressing ENTER, display the range of keys affected by pressing PF1, or end file verification and proceed to the next one by pressing PF16. If VERIFY encounters problems when expanding a compressed data record, the processing is the same as for an I/O error.

18.4 READING THE ERROR MESSAGE DISPLAYS

If you specify ERROR DISPLAY=YES through the Options screen, VERIFY displays an error message for each error encountered, as the files are validated. As each error message is displayed, VERIFY awaits your response. In all cases, you can resume file verification by pressing ENTER. For some types of errors, VERIFY allows you to display more detailed information on the error by pressing PF1 or terminate VERIFY without completing file verification by pressing PF16.

The three displays used for error messages are as follows:

ERROR1 — Reports errors that do not permit any further verification of the file (e.g., errors in the VTOC or in a root block) or that are not serious enough to require corrective action immediately (e.g., access consistency errors in an alternate index tree). In either case, you must press ENTER to invoke the next error message or summary display.

ERROR2 — Reports errors for which no greater level of detail can be provided by the VERIFY utility (e.g., "primary index tree cannot be fully verified"). From this display, you can either resume file verification by pressing ENTER or end file verification by pressing PF16.

ERROR3 — Reports errors for which VERIFY can provide a more detailed explanation. For example, "key inconsistency detected" can be followed up with a display showing the error details for this error by pressing PF1. You also can continue file verification by pressing ENTER or end file verification by pressing PF16.

18.5 READING THE SUMMARY DISPLAY

If you previously specify ERROR DISPLAY=YES through the Options screen, VERIFY provides a Summary screen in addition to the individual error message displays. The Summary screen displays summary information on the file just verified and, if appropriate, recovery procedures for the file if it is damaged. (Refer to Subsection 18.5.1 for more information on determining corrective action for a damaged file.) The Summary screen for each file is displayed only after VERIFY detects and displays all errors in that file. To gain access to the Summary screen in the most direct way, you press ENTER after each error message is displayed, until the Summary screen appears. From the Summary screen, you must press ENTER to continue to the next file you want to verify or to the End-of-job screen.

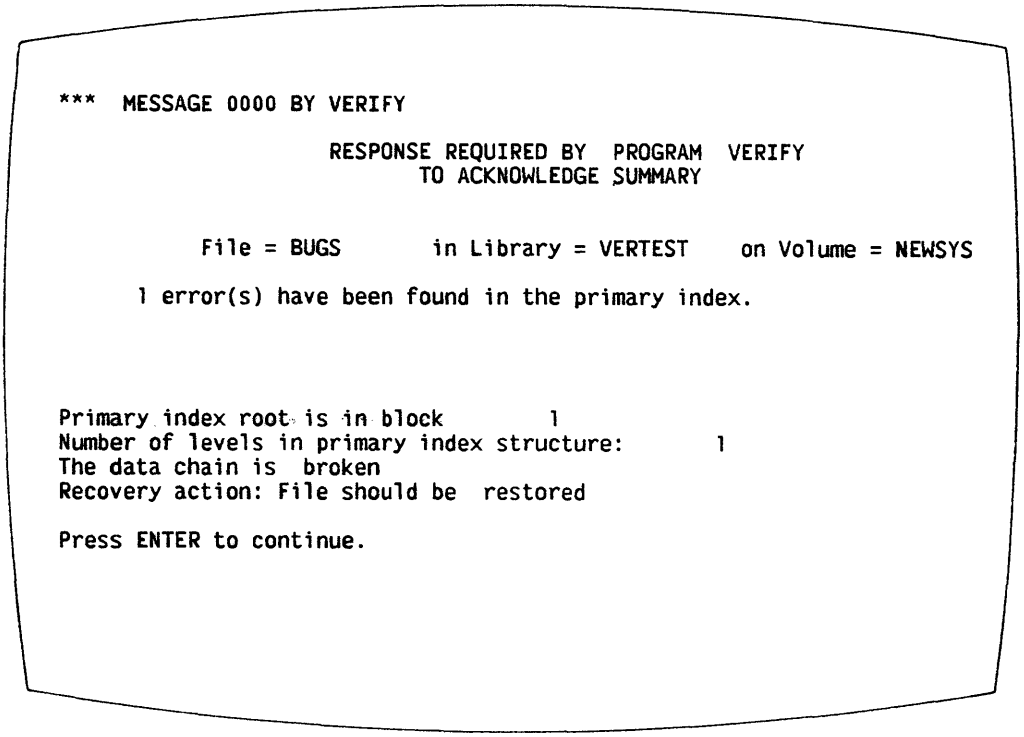


Figure 18-3. A Sample Summary Screen

18.5.1 Determining Corrective Action for a Damaged File

In addition to a summary of errors and other statistics, the Summary screen contains, when appropriate, a recommended course of action to recover a damaged file. There are three possible recommended actions:

- For files with no errors or access consistency errors only, no immediate corrective action is necessary.

- For certain types of index errors, running the COPY utility with the REORG option repairs the error. (Refer to Chapter 2, The COPY Utility.)

- For more serious errors, including a broken data chain, restoring the backup copy of the file or recreating the file manually are the only corrective actions you can take.

18.6 READING THE END-OF-JOB RESULTS

After the entire input unit is processed, VERIFY displays the name(s) of the file, library, and/or volume verified, and a message informing you that VERIFY processing is completed. If you prematurely terminated VERIFY processing by pressing PF16, then the End-of-job screen displays that verification of the input unit is incomplete when processing is terminated. In either case, you can rerun the program by pressing PF1, or end processing by pressing ENTER or PF16.

18.7 PRODUCING ERROR AND SUMMARY REPORTS

VERIFY automatically produces print files containing a Summary Report and, if any errors are detected, an Error Report is processed. The Summary Report cites the number of errors in the file (including no errors), the number of records in the file (including no records), the number of data blocks, primary index blocks, and alternate index blocks allocated, the total number of unused blocks, and the total number of blocks used (the sum of the data, index, and alternate index blocks). Although the Summary Report lists each error code, the report does not contain the meaning of the code. (Refer to Section 18.7.1 for the meanings of the error codes and recommended course of action.)

The Error Report identifies the file that is processed and lists the errors.

Examples of the Summary and Error Reports are shown in Figures 18-4 and 18-5.

```
WANG VS VERIFY PROGRAM                          SUMMARY REPORT                          13:30 09/11/81  PAGE     1

                                 ERROR  PRIMARY INDEX  ALTERNATE INDICES              DATA  PRIMARY INDEX  ALTERNATE INDEX  TOTAL UNUSED  TOTAL
VOLUME  LIBRARY   FILE           CODE   ERRORS         ERRORS          RECORDS        BLOCKS  BLOCKS         BLOCKS           BLOCKS        BLOCKS

NEWSYS  VERTEST   BUGS           08           1                0       UNKNOWN       UNKNOWN  UNKNOWN        UNKNOWN          UNKNOWN       UNKNOWN

END OF VERIFY SUMMARY REPORT.
```

Figure 18-4. A Sample Summary Report

```
WANG VS VERIFY PROGRAM                          ERROR REPORT                            13:30 09/11/81  PAGE     1

VOLUME  LIBRARY  FILE      ERROR MESSAGE

NEWSYS  VERTEST  BUGS      Invalid block length was detected in      data block number       0.

END OF VERIFY ERROR REPORT.
```

Figure 18-5. A Sample Error Report

18.7.1 Interpreting an Error Code

VERIFY has no displayed return codes. In the printed Summary Report, however, VERIFY assigns an error code to each file. The error codes and their meanings are listed as follows:

Error Code	Meaning	Recommended Corrective Action
0	No errors.	None
1	Access inconsistency errors in alternate indices only.	COPY/REORG can restore damaged index path.
2	Alternate indices cannot be verified. (AXD1 block is bad.) Primary index structure, however, is intact.	File must be restored from the backup volume. Not enough data is available for COPY/ REORG to reconstruct indices.
4	Primary index errors found. (FDR1 block or primary index block is bad.) Data chain, however, is intact.	COPY/REORG can reconstruct index unless FDR1 block is bad. In that case, the file must be restored
8	Data chain is broken.	File must be restored from the backup.
16	File cannot be fully verified. VERIFY could not get enough information to diagnose the problem. The volume may have a bad VTOC.	You can run LISTVTOC to diagnose VTOC problems. (Refer to Chapter 13.) In this case, the file must be restored from the backup or recreate it manually. Error Code 16 can also indicate that you terminated verification of the file.

18.8 A SAMPLE VERIFY PROCEDURE

You can control VERIFY processing through the VS Procedure language. A complete list of VERIFY GETPARMs is given in Appendix A. Consult the VS Procedure Language Reference for details concerning procedure syntax.

The following example shows a procedure to run VERIFY to validate a library (VETEST on volume SYSTEM) without any workstation displays. If a file is found that cannot be validated, the procedure causes VERIFY to skip to the next file in the library. Any Summary Reports and Error Reports, are written into a print file.

```
PROCEDURE
RUN VERIFY
ENTER OPTIONS  RANGE=LIBRARY, LIBRARY=VETEST, VOLUME=SYSTEM, VERIFY=ALL,
   DISPLAY=NO
ENTER ENDOFJOB 16
RETURN
```

CHAPTER 19
THE COMPRESS-IN-PLACE (CIP) UTILITY

19.1 INTRODUCTION

Compress-In-Place (CIP) enables you to consolidate free extents on nonsystem disks without performing a full volume backup and volume restore. It provides useful free space and improves performance by reducing disk seek time.

CIP enables you to compress only nonsystem volumes, those that you can obtain exclusively. CIP must be run by a task with a Segment 2 size of at least 512K. For system volumes, CIP is available as an option of the VS Stand-Alone Utility System on the VS25, VS45, VS85, VS90, and VS100; for more information, refer to the VS System Administrator's Reference.

Figure 19-1 gives an overview of CIP processing.

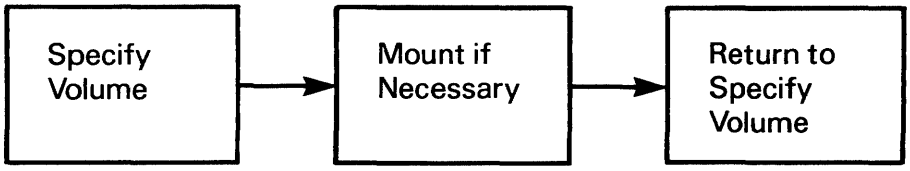


Figure 19-1. CIP Processing

19.2 RUNNING CIP

When CIP processing begins, the Input Definition screen asks you to specify the volume you wish to compress. The volume must be mounted for exclusive use. Figure 19-2 illustrates the Input Definition screen.

After specifying the volume name, press ENTER. If the volume that you specified is not mounted, the Mount screen prompts you to specify the device where you will mount the disk and waits for you to mount it. Press ENTER after the disk is mounted.

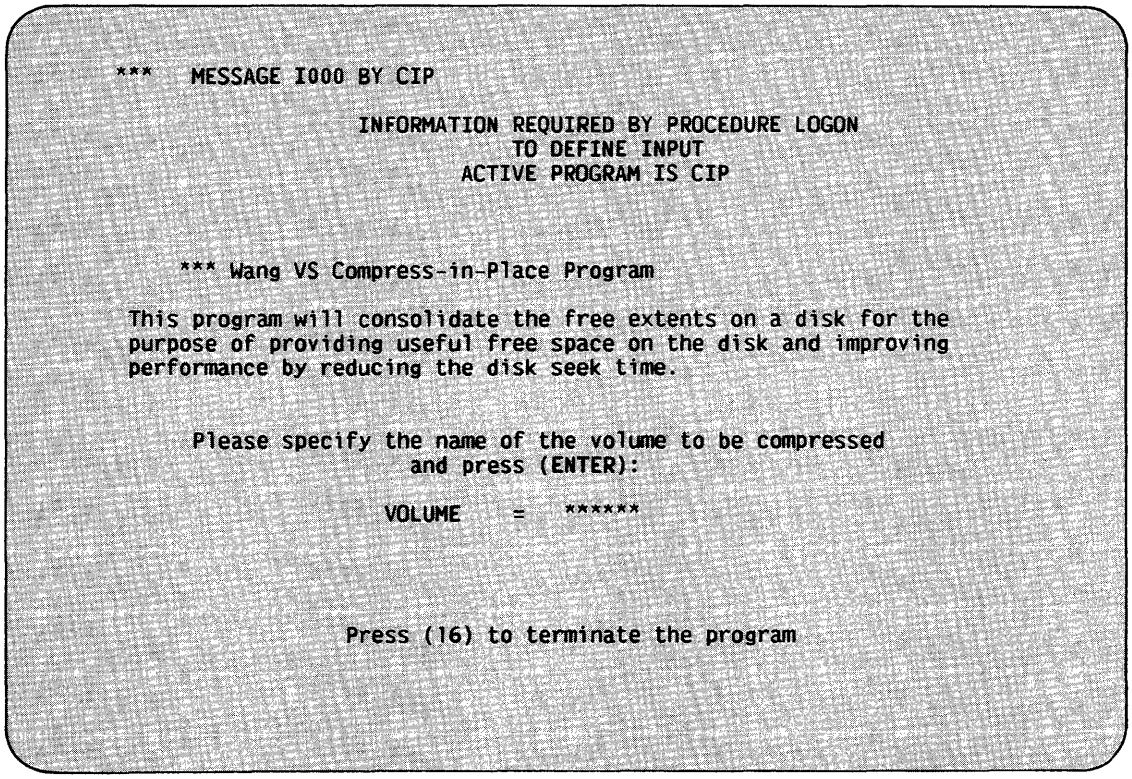


Figure 19-2. The CIP Input Definition Screen

NOTE

Before you run CIP, it is recommended that you run LISTVTOC to ensure that the VTOC is intact. If the VTOC has been damaged, do not use CIP. Instead, run the Backup and Restore options of the BACKUP utility. (For more information about LISTVTOC, refer to Chapter 13. BACKUP is described in the VS System Operator's Reference.)

When the compress operation is complete, CIP displays the Input Definition screen again. You can then run CIP again, or you can exit from CIP.

Several error messages are possible with this utility. If an I/O error occurs, CIP displays a message that includes the error I/O status word in hexadecimal digits. You must press ENTER to acknowledge the error, and CIP is then cancelled.

It is not appropriate to use CIP if a volume contains a large number of files. Currently, the limit is approximately 26,000 extents per disk. If the number of files exceeds the limit, CIP displays a message, and you must press ENTER to acknowledge the condition. In this case, you must do a full backup and restore to compress the volume.

You can minimize disk fragmentation if you run BACKUP to fully compress the disk and then use CIP on a regular basis.

If the volume is not crash- or media-tolerant and the CIP utility is cancelled abnormally, run LISTVTOC to be sure the VTOC is intact. The data will always be intact in this case. If the VTOC has been damaged, run the Backup and Restore options of the BACKUP utility to restore the disk.

19.3 A SAMPLE CIP PROCEDURE

You can control CIP processing through the VS Procedure language. You can specify all CIP options and mount operations through a procedure. Appendix A provides a complete list of CIP GETPARMs. For detailed information about the syntax of the VS Procedure language, refer to the VS Procedure Language Reference.

The following procedure mounts and compresses a volume. After the volume is compressed, the procedure exits CIP and dismounts the volume. The procedure dismounts the volume after the completion of CIP processing because VS Procedure language DISMOUNT statements cannot be embedded in a RUN, DISPLAY, or ENTER sequence. The CIP mount operation can be embedded in a procedure because it does not use the VS Procedure language MOUNT statement. The procedure totally automates CIP processing; you only physically mount and dismount the volume.

```
PROCEDURE
RUN CIP
ENTER INPUT VOLUME = ZENITH
ENTER MOUNT DEVICE = 018
ENTER INPUT 16
DISMOUNT DISK ZENITH
RETURN
```

CHAPTER 20
THE IOTRACE UTILITY

20.1 INTRODUCTION

The IOTRACE utility enables you to selectively capture a history of the I/Os issued to one or more devices for later analysis. This option is especially useful for attempts to solve intermittent problems related to device I/O.

IOTRACE monitors the I/O trace table and records I/O information for a range of devices, based on criteria you define. You can retain captured data in a small disk file that can be collected and analyzed at a convenient time.

You can minimize the effect on the normal operating environment by running IOTRACE from a procedure submitted as a background task. Refer to Section 20.6 for an example of an IOTRACE procedure.

Figure 20-1 provides an overview of IOTRACE processing.

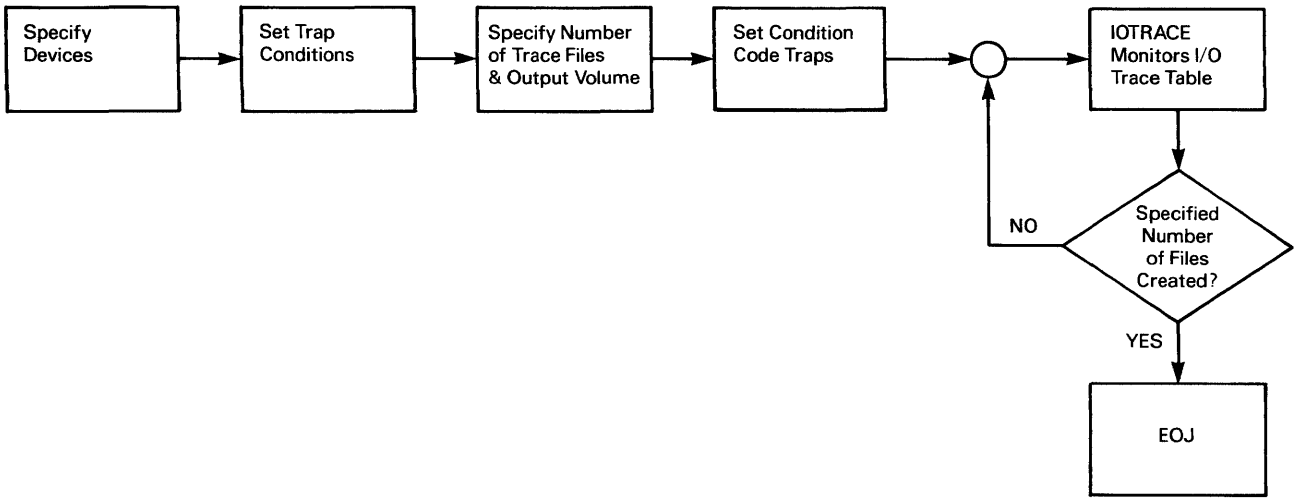


Figure 20-1. IOTRACE Processing

20.2 RUNNING IOTRACE

When you run the IOTRACE utility, the Input Definition Screen, illustrated in Figure 20-2, prompts you to enter the range of devices you want to trace. IOTRACE displays a default range tailored to your system; you can accept the default range or enter a different range in the pseudoblanks. Enter the lowest numbered device in the LOWDEV field and the highest numbered device in the HIGHDEV field. If you wish to trace only one device, enter that number in both fields.

```
***  MESSAGE 0001 BY TRACE

                INFORMATION REQUIRED BY PROGRAM IOTRACE
                            TO DEFINE INPUT

            *** WANG VS I/O TRACE UTILITY ***

  Please specify the following information for device I/O tracing:

  Range of devices to be traced  LOWDEV  =  000    HIGHDEV = 254

 Set at least one trap condition to close the trace file.

    IOSWTRAP = XXXXXXXXXXXXXXX   SIOTRAP = XXXXXXXXXXXXXXX  X= don't care
    CIOTRAP  = XXXXXXXXXXXXXXX   HIOTRAP = XXXXXXXXXXXXXXX

  Output volume name for trace file VOLUME   = ZENITH

  Desired number of trace files     TRAPREQ  = 001


                Press (ENTER) to continue, (16) to EXIT
```

Figure 20-2. A Sample IOTRACE Input Definition Screen

You must set one or more trap conditions. To set traps on the IOSW (I/O Status Word), SIO (Start I/O) instruction, CIO (Control I/O) instruction, or HIO (Halt I/O) instruction, you specify mask values for comparison against the designated I/O types. Enter the desired test values in hexadecimal characters. IOTRACE does not verify that the input values are valid hexadecimal characters.

The VOLUME field allows you to enter the name of the volume that will contain the trace files. The default is the System volume. The volume that you specify must have enough free space to store files.

Finally, you can specify a maximum number of trace files to be created. Enter the number in the TRAPREQ field; the default is 1. If IOTRACE creates the number of files that you specify, IOTRACE processing ends, and the Command Processor menu reappears.

After specifying the range of devices, the traps desired, the output volume, and the number of trace files, press ENTER. IOTRACE then displays the Condition Code Traps screen; Figure 20-3 illustrates this screen.

```
***  MESSAGE 0020 BY TRACE

                INFORMATION REQUIRED BY PROGRAM IOTRACE
                            TO DEFINE CCTRAPS


Each IOCW entry in the trace table contains the condition code
 returned by the instruction which issued the I/O.

Enter (Y) for any condition code to be trapped.

The trace file closes when the trap condition occurs.

CONDITION TYPE          START I/O     CONTROL I/O     HALT I/O
DEVICE BUSY             SIODB = N     CIODB = N       HIODB = N
IOP BUSY                SIOIB = N     CIOIB = N       HIOIB = N
IOP NOT OPERATIONAL     SIOIN = N     CIOIN = N       HIOIN = N

              Press (ENTER) to continue, (1) to RETURN
```

Figure 20-3. The Condition Code Traps Screen

The Condition Code Traps screen enables you to define additional trap conditions. To set a trap for one or more of the conditions, specify Y and press ENTER. The condition stored in the IOCW (I/O Command Word) element of the trace table is the condition code from the SIO, CIO, and HIO instructions. For more information about these instructions, refer to the VS Principles of Operation.

NOTE

If you did not specify a mask value on the Input Definition screen, you must set a trap on the Condition Code Traps screen.

20.3 IOTRACE PROCESSING

When you press ENTER from the Condition Code Traps screen, IOTRACE begins to monitor the I/O trace table and to test the entries in the table against the criteria that you specified. IOTRACE tests the trace table entries in the following order:

- The device for this I/O must be within the specified device range.

- If you specified a mask value for this type of I/O, IOTRACE compares the mask value to the IOSW or IOCW of the trace table entry. A match triggers a trap.

- If you specified a condition code, IOTRACE compares that code against the condition code of the trace table entry. A match triggers a trap.

When a trap occurs, IOTRACE stops testing and captures the next 50 to 100 I/Os from all devices in the system trace table. This ensures that any problem triggered by the trap is recorded. IOTRACE records these I/Os in a trace file, and this file is then closed and named. The file name consists of the day in the current month and the time stamp. For example, if a trace occurs at 3:23 pm and 15 seconds, on the eighth day of the month, the file name is 08152315. The trace file resides in the library @IOTRACE on the specified volume.

When the trace file closes, all masks are enabled, and IOTRACE continues to monitor the system trace table until it creates the number of trace files that you specified. IOTRACE processing then ends, and the Command Processor menu reappears.

If you cancel IOTRACE processing when there are valid entries in the IOTRACE work file, IOTRACE attempts to save the data in a disk file. The file is named as described above.

20.4 TRACE FILE FORMAT

Each trace file consists of 10 records that have a length of 2048 bytes. The first nine records contain the trace table entries, and the tenth record contains the system and program parameters when the trace file was created. You can use the following descriptions of the record formats to analyze the collected data.

20.4.1 Format of the First Nine Records

The first byte of each record is the Trace Table Overrun flag. A value of "FF" (in hexadecimal) indicates an overrun; there is so much system activity that the record may not contain all of the trace table entries. A value of "00" (in hexadecimal) indicates no overrun.

The next 15 bytes contain a date and time stamp in ASCII format. The date and time stamp has the form YYMMDD HHMMSSTH, where YY is the year, MM is the month, DD is the day, HH is the hour, MM is the minute, SS is the second, and TH is the second in hundredths.

The rest of each record consists of the trace table entries logged by IOTRACE. A trace table entry has a length of 16 bytes, and it has the following format:

Byte	Description
1	Device Number (CP3)
1-2	Physical Device Address (non-CP3)
3	Interrupt qualifier (non-CP3)
4	Type code and condition code that result from applying logical OR to the type code and the condition code returned by the SIO, CIO, or HIO command
5-13	IOSW or IOCW entry
14-16	Time stamp

Figure 20-4 illustrates the format of trace table entries for CP3 systems (VS50 and VS80). The format for other systems is illustrated in Figure 20-5.

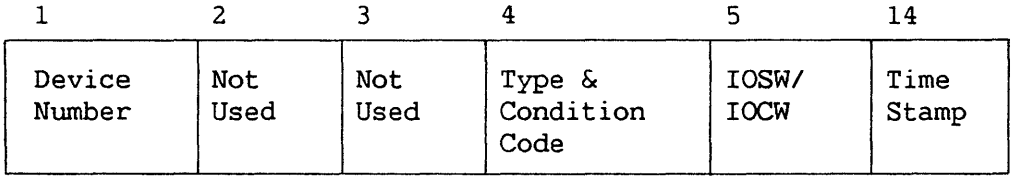


Figure 20-4. CP3 Trace Table Entry Format

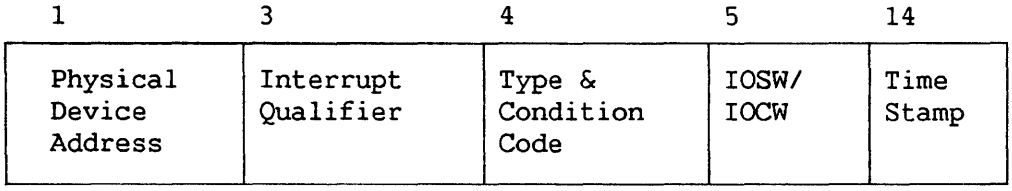


Figure 20-5. Non-CP3 Trace Table Entry Format

Figure 20-6 gives an example of a trace table entry for a CP5 (VS25 or VS45) system.

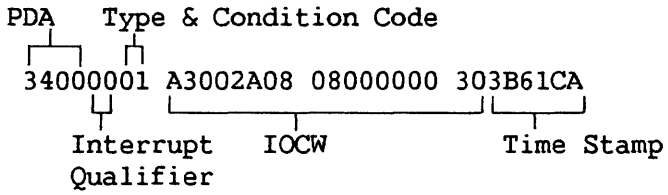


Figure 20-6. A Sample CP5 Trace Table Entry

Note that the first record may not contain the earliest data collected if tracing has been active for a period of time long enough to overwrite some of the trace file records. Use the time stamp at the beginning of each record to determine the sequence of the trace table entries. If IOTRACE encounters a trap condition after a short period of time, every record in the trace file may not contain valid information.

20.4.2 Format of the Tenth Record

The tenth record has the following format:

Byte	Description
0	CP type.
1	Type of trap taken. Byte 1 contains one of the following hexadecimal values:

Value	Description
00	IOSW trap
01	SIO trap
02	CIO trap
03	HIO trap
10	Device Busy Condition Code
20	IOP Busy Condition Code
40	IOP not operational
FF	Program cancelled by user

4-7	Version number of the Operating System.
8-9	Low device in the system trace range.
10-11	High device in the system trace range.
12-13	Low device in the program trace range.
14-15	High device in the program trace range.
16-23	Definition of the IOSW trap value.
24-31	Definition of the SIO trap value.
32-39	Definition of the CIO trap value.
40-47	Definition of the HIO trap value.
48-49	Trace file record containing the element that caused trap processing to begin.
50-51	Device count of devices traced.

The rest of the tenth record consists of a table that lists the devices traced. Each entry in the table has a length of 4 bytes, and it has the following format:

Byte	Description
1	Logical device number
2	Device type code
3-4	PDA (Physical Device Address) of the device (for non-CP3 systems)

20.5 IOTRACE ERROR MESSAGES

IOTRACE can issue the following error messages:

Number	Text	Description
0000	SORRY, GETHEAP failed	IOTRACE was not able to obtain buffer space in its Segment 2 for work areas.
0002	The LOW device specified is greater than the HIGH device	The low device number must be less than or equal to the high device number.
0005	The only device specified is invalid	If only one device has been specified and this device is not valid, no tracing can occur.
0006	No valid devices in specified range	No valid devices were found in the specified range for this system. Therefore, no tracing can occur.
0007	Invalid response to option	You entered an invalid response in one of the displayed fields. The field is blinking.
0090	Unable to establish message port	If the system is not tracing I/O activity, this program cannot trace.

20.6 A SAMPLE IOTRACE PROCEDURE

You can control IOTRACE processing through the VS Procedure language. Appendix A provides a complete list of IOTRACE GETPARMs. For detailed information about the syntax of the VS Procedure language, refer to the VS Procedure Language Reference.

The following procedure monitors the I/O trace table for a single device. Thus, LOWDEV and HIGHDEV have the same value. The procedure sets a trap for the completion with error of an I/O operation to the device. The procedure requests the creation of a maximum of three trace files and designates ZENITH as the output volume.

```
PROCEDURE
RUN IOTRACE
ENTER INPUT LOWDEV = 020, HIGHDEV = 020, IOSWTRAP = 2XXXXXXXXXXXXXX,
    VOLUME = ZENITH, TRAPREQ = 3
ENTER CCTRAPS
RETURN
```

CHAPTER 21
THE POOLSTAT UTILITY

21.1 INTRODUCTION

The POOLSTAT utility enables you to monitor the utilization of page pools on VS25, VS45, VS85, VS90, and VS100 systems. Through the DISKINIT utility, you allocate a page pool to a volume; for more information about DISKINIT, refer to Chapter 6. Refer to the VS Release 6.20 Software Bulletin for more information about page pools.

The system monitors the use of the page pools and issues warnings when a page pool nears capacity. Through the POOLSTAT utility, you can view these statistics at any time.

21.1 RUNNING POOLSTAT

When POOLSTAT processing begins, the Pool Utilization screen displays the utilization of up to three page pools. Figure 21-1 illustrates a sample Pool Utilization screen.

For each page pool, the Pool Utilization screen provides the following information:

Field	Description
Volume	The name of the volume on which the page pool resides.
Capacity	The actual size of the page pool.
Current Usage	The amount of space being used by tasks assigned to the page pool, and the percentage of the pool capacity that this represents.
Peak Usage	The maximum amount of space used by tasks assigned to the page pool since the pool was initialized, and the percentage of the pool capacity that this represents. A page pool is initialized at IPL if the volume is enabled for paging; otherwise, it is initialized when you enable the volume for paging through the Operator's Console.

Field	Description
Memory Commitment	The total of the Segment 2 sizes currently assigned to the page pool, and the percentage of the pool capacity that this represents.
User Count	The current number of tasks assigned to the page pool.
REF Rate (Immediate)	The number of I/Os to the pool during the last second.
REF Rate (20% Decay)	The average pool I/O rate weighted in favor of the last several seconds.

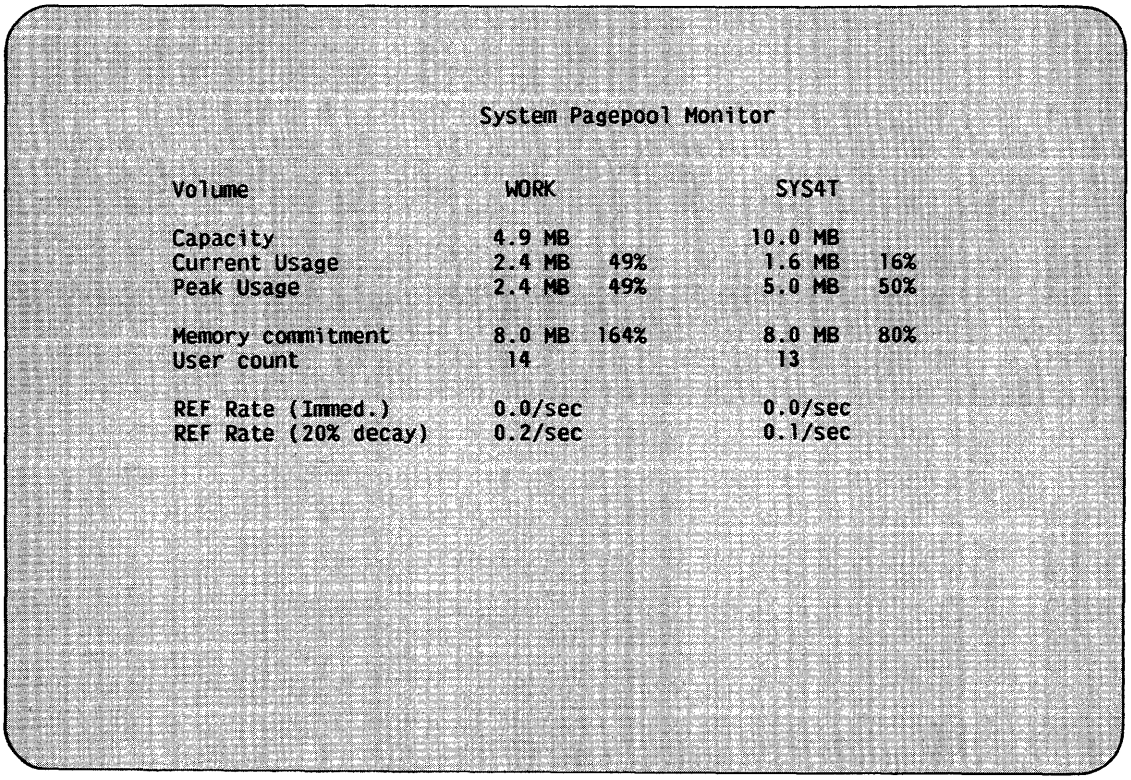


Figure 21-1. Sample Pool Utilization Screen

If a page pool has been allocated a capacity greater than 32 Mb, the Pool Utilization screen displays an asterisk next to the page pool's capacity and advises you that the excess capacity will not be used.

It is recommended that you run POOLSTAT when you first establish a page pool on your system. The statistics on current and peak usage can help you to determine if the page pool capacity is adequate for the paging requirements of the tasks assigned to the page pool.

CHAPTER 22
THE SORTINT UTILITY

22.1 INTRODUCTION

The SORTINT utility is an international version of the SORT utility. You can use SORTINT to sort up to 20 files into a single, ordered output file according to standard ASCII sequence or an external collating sequence. SORTINT also provides the option to reformat the output record. SORTINT supports all of the standard features of the SORT utility except for tape input and output, shared consecutive files, and the Merge function. (For more information about the SORT utility, refer to Chapter 14.)

SORTINT enables you to perform the following functions:

- Sort a single file according to standard ASCII or an external collating sequence.
- Sort up to 20 files into a single, ordered output file according to standard ASCII or an external collating sequence.
- Select specified records from one or more input files and sort them into a single output file according to standard ASCII or an external collating sequence.
- Produce an output file that contains only the primary index key field from each record in the input file. You specify the sort order for the primary index key field.
- Produce an ordered output file of 3-byte records to be used by RPG II programs.
- Reformat the record layout of the output file. By reformatting the output record, you can, for example, modify the length and sequence of fields in the record or include only selected fields in the record.

The ability to sort according to an external collating sequence enables you to accommodate international character sets or sort orders other than standard ASCII. SORTINT prompts you to specify the name of the file that defines the external collating sequence. Through the STABLEMT utility, you can create or modify files that define external collating sequences; for more information about STABLEMT, refer to Chapter 23.

You can access input files (consecutive or indexed) from disk. The SORTINT utility leaves the input file(s) intact unless you assign the same name to the input and output files. The output file is always a consecutive file.

Figure 22-1 provides an overview of SORTINT processing.

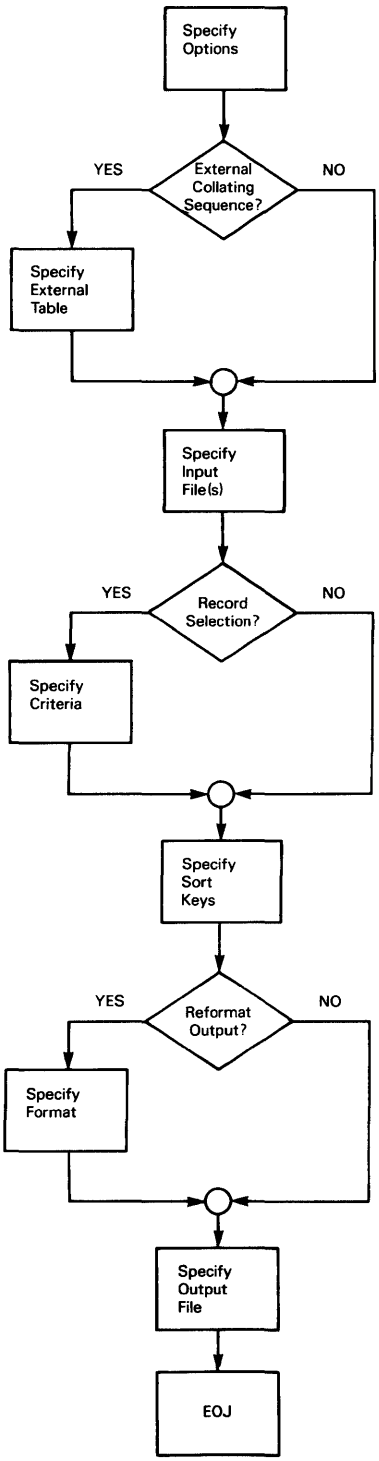


Figure 22-1. SORTINT Processing

SORTINT requires a 200% overhead in disk space. Because DMS does not support external collating sequences, you cannot use an output file sorted according to an external collating sequence as input to other processes (e.g., KEYFILE creation).

22.2 SELECTING THE PROGRAM OPTIONS

When SORTINT processing begins, the SORTINT Options screen prompts you to select the options needed to perform the sort operation. After selecting the options, press ENTER. Figure 22-2 illustrates the SORTINT Options screen.

```
*** MESSAGE 0001 BY SORTI

              INFORMATION REQUIRED BY PROGRAM SORTINT
                          TO DEFINE OPTIONS


You are now running Wang VS Sort/Merge Program, International Version

        FUNCTION = SORT*            (SORT or MERGE)
        MEMORY   = 128 K  (Memory used, Maximum size = 428 K)

Do you want an ADDROUT output file ?          ADDROUT  = NO* (YES or NO)
Do you want a KEYOUT output file ?            KEYOUT   = NO* (YES or NO)
Do you require a STABLE sort ?                STABLE   = NO* (YES or NO)
EXTERNAL collating sequence ?                 EXTERNAL = NO* (YES or NO)
REFORMAT output record ?                      REFORMAT = NO* (YES or NO)
Do you want to SHARE input files ?            SHARED   = NO* (YES or NO)
```

Figure 22-2. The SORTINT Options Screen

The options available from the SORTINT Options screen are summarized below.

Option	Description
FUNCTION	The type of processing to be performed. Specify SORT or MERGE; the default is SORT. Because SORTINT does not support the Merge function, it calls the SORT utility if you specify MERGE. Refer to Chapter 14 for a description of the SORT utility.

Option	Description
MEMORY	The amount of main memory to be used by SORTINT as a work area. The main memory used affects the efficiency of the operation being performed. For most operations, the default, 128K, is sufficient. If the default amount of main memory is not available, the system determines the amount actually available and runs the utility in that amount of memory. The maximum amount of available main memory depends on your system. If you enlarge the work area beyond the default value to improve SORTINT processing time, you increase the competition with other users for main memory. You can only determine the optimum amount of main memory for each operation by experimenting with the actual operating environment. If the sort operation requires a lot of processing time or a large amount of main memory, you should run the operation in batch mode during off-peak hours so that other users are not slowed down or temporarily halted.
ADDROUT	A YES or NO option that produces an ordered output file of 3-byte records to be used by RPG II programs; the default is NO. Each record is a positive binary number that represents the relative number of a record in the input file. When you perform a sort with the ADDROUT option, the only output is an ADDROUT file. You cannot use the ADDROUT option with the MERGE option, nor can you combine it with a KEYOUT file or a reformatted output file. You can use the ADDROUT option in conjunction with the STABLE or EXTERNAL option. Subsection 14.2.1 illustrates an example of an ADDROUT file. For more information about ADDROUT files, refer to the VS RPG II Language Reference.
KEYOUT	A YES or NO option that produces an ordered output file that consists of the primary index key field in the input file; the default is NO. The input file must be indexed. You cannot use the KEYOUT option with the MERGE option, nor can you combine it with a ADDROUT file or a reformatted output file. You can use the KEYOUT option in conjunction with the STABLE or EXTERNAL option. Subsection 14.2.1 illustrates an example of a KEYOUT file.
STABLE	A YES or NO option that produces an ordered output file in which records with identical sort keys remain in the same order as the input order sequence. STABLE defaults to NO. Subsection 14.2.1 illustrates an example of a STABLE sort.
EXTERNAL	A YES or NO option that allows you to specify a file that defines a collating sequence other that ASCII. EXTERNAL defaults to NO. You cannot use the EXTERNAL option with the MERGE option.

Option	Description
REFORMAT	A YES or NO option that allows you to reformat the record layout of the output file. REFORMAT defaults to NO. You cannot use the REFORMAT option with the MERGE option, nor can you combine it with a ADDROUT or KEYOUT file. You can use the REFORMAT option in conjunction with the STABLE or EXTERNAL option.
SHARED	A YES or NO option that allows input files to be shared. The input files must be indexed. The SHARED option defaults to NO.

NOTE

If you specify NO to both the EXTERNAL and REFORMAT options, SORTINT calls the SORT utility to perform the sort operation. Refer to Chapter 14 for a description of SORT processing.

Table 22-1 summarizes the relationship between the program options and the types of output files available.

Table 22-1. SORTINT Options and Output Files

Output File	Option						
	ADDROUT	KEYOUT	STABLE	EXTERNAL	REFORMAT	SORT	MERGE
ADDROUT	Yes	No	Yes	Yes	No	Yes	No
KEYOUT	No	Yes	Yes	Yes	No	Yes	No
STABLE	Yes	Yes	Yes	Yes	Yes	Yes	Yes
REFORMAT	No	No	Yes	Yes	Yes	Yes	No

22.3 SPECIFYING THE EXTERNAL COLLATING SEQUENCE

If you specify YES to the EXTERNAL option, SORTINT displays a screen that prompts you to identify the file that defines the external collating sequence. Enter the name, library, and volume of the file and press ENTER. The default library is TRNTABLE, and the default volume is the value defined for INVOL through the Set Usage Constants command of the Command Processor or through a Procedure language SET statement. Figure 22-3 illustrates the Specify External Collating Sequence screen.

You use the STABLEMT utility to create or modify files that define external collating sequences. Refer to Chapter 23 for more information.

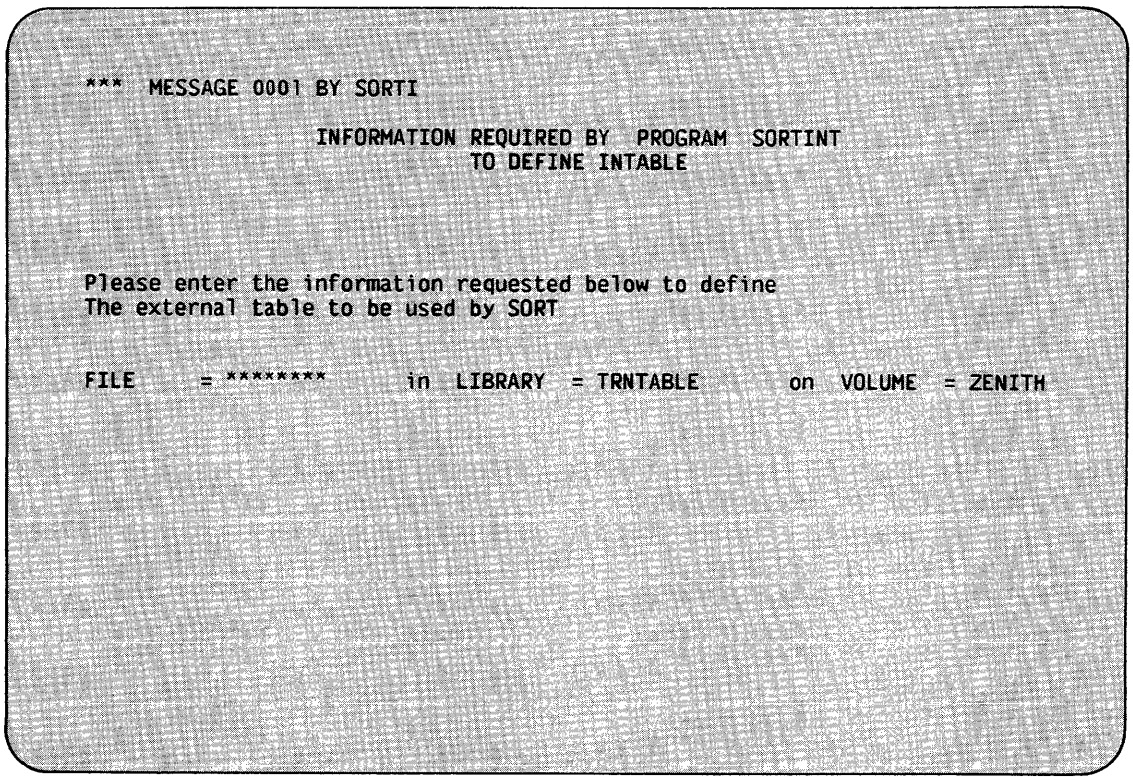


Figure 22-3. The Specify External Collating Sequence Screen

22.4 SPECIFYING THE INPUT FILE

After you select the program options and, if necessary, specify the file that defines the external collating sequence, SORTINT displays the Input Definition screen. The Input Definition screen prompts you to specify the name, library, and volume of each input file you want to process. To specify up to 20 input files, enter YES in the MOREFILE field. The Input Definition screen will then continue to appear until you press ENTER from an unaltered screen. To select specific records from the input files for use in the sort operation, enter YES in the SELECT field. After specifying the input file, press ENTER. Figure 22-4 illustrates the Input Definition screen.

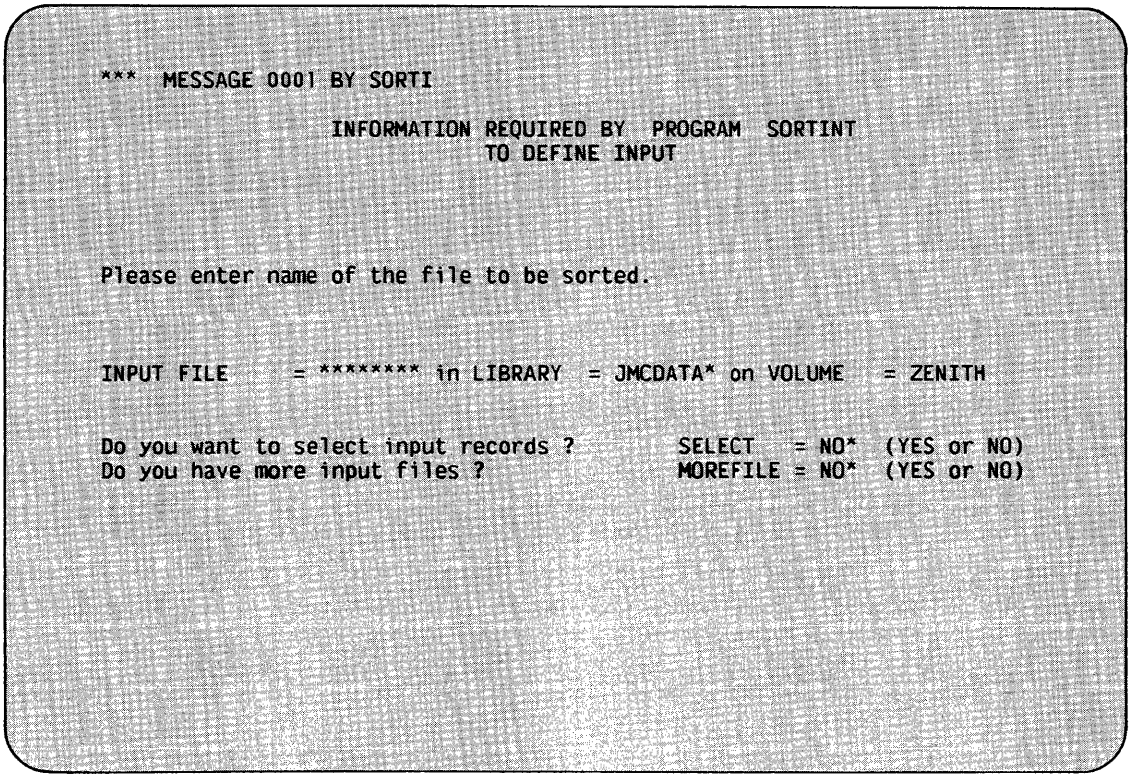


Figure 22-4. The SORTINT Input Definition Screen

The input file must be a disk file. You can use the SORT utility to sort tape files; refer to Chapter 14 for more information. To specify the input file, you enter values in the following fields:

Field	Description
FILE	The name of the input file to be sorted.
LIBRARY	The name of the library in which the input file resides. The default is the value defined for INLIB through the Set Usage Constants command of the Command Processor or through a Procedure language SET statement.
VOLUME	The name of the volume on which the input file resides. The default is the value defined for INVOL through the Set Usage Constants command of the Command Processor or through a Procedure language SET statement.
SELECT	A YES or NO option that allows you to select specific records from the input files for use in the sort operation. The default is NO. If you specify NO, SORTINT processes all of the records in the input files.

Field	Description
MOREFILE	A YES or NO option that allows you to specify additional input files. The default is NO. If you specify YES, you specify the additional input files on additional Input Definition screens. To indicate that you have specified all of the input files, press ENTER from an unaltered screen. The additional Input Definition screens do not contain the SELECT and MOREFILE fields.

22.5 DEFINING SELECTION CONDITIONS

If you specify YES to the SELECT option, SORTINT displays the Select screen. SORTINT then selects individual input records for the sort operation according to the selection conditions that you define on the Select screen. You can specify up to 32 selection conditions through the Select screen; you use the connectors AND and OR to define the relationships between the individual conditions.

Figure 22-5 illustrates a sample Select screen. In Figure 22-5, the first selection condition specifies that an input record must equal 'P' in Position 1. As the second selection condition illustrates, you can test a field against another field in the same record; the second selection condition specifies that Position 1 must equal Position 43 for a record to be processed by SORTINT. The OR connector defines the relationship between these two conditions; thus, SORTINT selects a record if it meets one of the conditions.

```
***  MESSAGE SEL  BY SORTI

                INFORMATION REQUIRED BY  PROGRAM  SORTINT
                          TO DEFINE SELECT


 Enter record selection criteria below, supplying field position, length, and
   format type (Binary,Char,Decimal,...), test relation (EQ,NE,GT,GE,LT,LE),
   and test value (in quotes) or comparison field position (without quotes).
 Set connector to "AND" to continue current criterion; use "OR" to begin
   alternative criterion.      (Use Chars or Decimal numeric for test value.)

    FLDPOS1  = 1***     LENGTH1  = 1**     FLDTYP1  = C
    TSTREL1  = EQ       VALUE1   = 'P'***************       CONECT1  = OR*

    FLDPOS2  = 1***     LENGTH2  = 1**     FLDTYP2  = C
    TSTREL2  = EQ       VALUE2   = 43****************       CONECT2  = OR*

    FLDPOS3  = 1***     LENGTH3  = 1**     FLDTYP3  = C
    TSTREL3  = EQ       VALUE3   = 'P'***************       CONECT3  = AND

    FLDPOS4  = 3***     LENGTH4  = 1**     FLDTYP4  = C
    TSTREL4  = EQ       VALUE4   = 'A'***************       CONECT4  = ***
```

Figure 22-5. A Sample Select Screen

Figure 22-5 also shows you how to specify a complex condition by combining simple conditions with the AND connector. The complex condition specifies that an input record must equal 'P' in Position 1 and 'A' in Position 3. The AND connector defines the relationship between the two simple conditions; thus, SORTINT selects a record only if it meets both conditions. The OR connector defines the relationship between this complex condition and the first two selection conditions. Thus, SORTINT processes the selection conditions as follows: (Condition 1) OR (Condition 2) OR (Condition 3 AND Condition 4).

For each selection condition, you specify the parameters summarized below. The 'n' appended to each parameter is a numeric value that SORTINT automatically assigns. All parameters with the same numeric value are combined to specify a selection condition. For example, FLDPOS1, LENGTH1, FLDTYP1, TSTREL1, VALUE1, and CONECT1 are used to specify the first selection condition.

Parameter	Description
FLDPOSn	The starting position of the field to be used in the condition. The first byte of a record occupies Position 1. When you specify complex conditions, the field positions do not have to be in ascending order. Thus, FLDPOS1 can be 5 while FLDPOS2 is 3.
LENGTHn	The number of bytes in the field to be tested, beginning with FLDPOSn.
FLDTYPn	The data format of VALUEn. The options are as follows:

Option	Description
C	Character data (alphanumeric)
B	Binary
D	External decimal, trailing sign in separate byte
L	Zoned decimal, leading sign in separate byte
P	Packed decimal, trailing sign in last byte
Z	Zoned decimal, trailing sign in last byte

FLDTYPn defaults to C.

TSTRELn — The type of comparison to be made between the input data (starting at FLDPOS1) and VALUEn. The options are as follows:

Option	Description
EQ	Equal
NE	Not equal
GT	Greater than
GE	Greater than or equal to
LT	Less than
LE	Less than or equal to

Parameter	Description
VALUEn	The value to be compared to the field that begins at FLDPOSn and extends for LENGTHn bytes in each input record. VALUEn can be a literal data value or the starting position of another field in the record. If you specify a literal data value (character or number), you must enclose the value in single or double quotes. SORTINT interprets a value not enclosed in quotes as a field position. If you specify the starting position of another field, the field must be of the same type and length as the original field. VALUEn has a maximum length of 16 bytes for literal character data or for external, zoned, or packed decimal data. VALUEn must be two or four bytes long for binary data. When you specify a field-to-field comparison, the two fields can have a maximum length of 256 bytes for character data.
CONECTn	The logical connector that you use to define the relationship between the selection conditions. The options are AND and OR. The AND connector combines two or more simple selection conditions into a single complex condition. The OR connector indicates the start of a new alternative condition. You can specify up to 32 selection conditions for a sort operation; thus, the maximum number of logical connectors is 31. A blank CONECTn field indicates the end of the selection conditions.

22.6 SPECIFYING THE SORT KEYS

After you specify the input file(s) and the selection conditions (if any), SORTINT displays the Sort Keys screen. The Sort Keys screen prompts you to specify the fields by which the input records are to be sorted. The fields that you specify are the sort keys (to be distinguished from the index keys). Figure 22-6 illustrates the Sort Keys screen.

You can specify a minimum of one and a maximum of eight sort keys. Enter the number of sort keys in the NUMBER OF KEYS field; the default is 1.

For each sort key, you specify the parameters summarized below. The 'n' appended to each parameter is a numeric value that SORTINT automatically assigns. All parameters with the same numeric value are combined to specify a sort key. For example, POST1, LENGTH1, TYPE1, and ORDER1 are used to specify the first sort key.

Parameter	Description
POSTn	The starting position of the field to be used as the sort key. The first byte of a record occupies Position 1.
LENGTHn	The number of bytes in the field to be used as the sort key.

Parameter	Description
TYPEn	The data type of the field to be used as the sort key. The options are as follows:

Option	Description
C	Character data (alphanumeric)
B	Binary
D	External decimal, trailing sign in separate byte
F	Floating point
L	Zoned decimal, leading sign in separate byte
P	Packed decimal, trailing sign in last byte
Z	Zoned decimal, trailing sign in last byte

TYPEn defaults to C.

ORDERn — The order in which the input records are to be sorted by the sort key. You can specify ascending (e.g., 1, 2, 3, or A, B, C) or descending (e.g., 3, 2, 1, or C, B, A) order. Enter A (Ascending) or D (Descending) as the value of ORDERn; the default is A.

```
***  MESSAGE 0001 BY SORTI

                INFORMATION REQUIRED BY  PROGRAM  SORTINT
                             TO DEFINE KEYS


Please specify Sort/Merge keys:

NUMBER OF KEYS     = 1  (Less than or equal to 8)
Key attributes:
         Position(>0)      Length in Bytes  Sort Key Type   Sort Order(A,D)
Key 1:   POST1    = ****  LENGTH1  = ***   TYPE1    = C    ORDER1   = A
Key 2:   POST2    = ****  LENGTH2  = ***   TYPE2    = C    ORDER2   = A
Key 3:   POST3    = ****  LENGTH3  = ***   TYPE3    = C    ORDER3   = A
Key 4:   POST4    = ****  LENGTH4  = ***   TYPE4    = C    ORDER4   = A
Key 5:   POST5    = ****  LENGTH5  = ***   TYPE5    = C    ORDER5   = A
Key 6:   POST6    = ****  LENGTH6  = ***   TYPE6    = C    ORDER6   = A
Key 7:   POST7    = ****  LENGTH7  = ***   TYPE7    = C    ORDER7   = A
Key 8:   POST8    = ****  LENGTH8  = ***   TYPE8    = C    ORDER8   = A

 * Key Type:    B=Binary, C=Character, D=Decimal, F=Floating, P=Packed
                Z=Zoned decimal, L=Zoned decimal sign leading
 * Sort Order:  A=Ascending, D=Descending
```

Figure 22-6. The Sort Keys Screen

If you specify more than one sort key, SORTINT uses the keys in the order in which they were specified. For example, if you specify three sort keys, the sort is performed first on Key 1, then on Key 2, and finally on Key 3. You can use multiple sort keys to sort more than one field, or to sort on the first key specified and to use the additional keys to sort duplicate values within the first key. After specifying all of the sort keys, press ENTER.

22.7 DEFINING THE OUTPUT FILE FORMAT

If you specify YES to the REFORMAT option, SORTINT now displays a screen that prompts you to define the format of records in the output file. By reformatting the output record, you can modify the length and sequence of fields in the record, or you can include only selected fields in the record. Figure 22-7 illustrates a sample Output Format screen.

```
***  MESSAGE 0001 BY SORTI

                INFORMATION REQUIRED BY  PROGRAM  SORTINT
                              TO DEFINE FORMAT


Please specify Output file format:


OUTPUT RECORD  LENGTH    = 100*       PAD       = 20
       Position(Input)   Length(Bytes)      Position(Output)
FLD1:  INPOS1   = 1***   LENGTH1  = 10*   OUTPOS1  = 1***
FLD2:  INPOS2   = 7***   LENGTH2  = 10*   OUTPOS2  = 11**
FLD3:  INPOS3   = 20**   LENGTH3  = 1**   OUTPOS3  = 30**
FLD4:  INPOS4   = ****   LENGTH4  = ***   OUTPOS4  = ****
FLD5:  INPOS5   = ****   LENGTH5  = ***   OUTPOS5  = ****
FLD6:  INPOS6   = ****   LENGTH6  = ***   OUTPOS6  = ****
FLD7:  INPOS7   = ****   LENGTH7  = ***   OUTPOS7  = ****
FLD8:  INPOS8   = ****   LENGTH8  = ***   OUTPOS8  = ****
FLD9:  INPOS9   = ****   LENGTH9  = ***   OUTPOS9  = ****
FLD10: INPOS10  = ****   LENGTH10 = ***   OUTPOS10 = ****
FLD11: INPOS11  = ****   LENGTH11 = ***   OUTPOS11 = ****
FLD12: INPOS12  = ****   LENGTH12 = ***   OUTPOS12 = ****
```

Figure 22-7. A Sample Output Format Screen

You must specify the length of the output record and a padding character if needed. The output record has a minimum length of 1 byte and a maximum length of 2048 bytes. The padding character must be a valid hexadecimal code. The default padding character is 20 (i.e., blank).

For each field in the output record, you specify the parameters summarized below. The 'n' appended to each parameter is a numeric value that SORTINT automatically assigns. All parameters with the same numeric value are combined to define a field. For example, INPOS1, LENGTH1, and OUTPUT1 are used to define one field in the output record.

Parameter	Description
INPOSn	The starting position of the field in the input record.
LENGTHn	The length of the field in the output record.
OUTPOSn	The starting position of the field in the output record. It cannot overlap or exceed the output record length that you defined.

You can define up to 12 fields in the output record. After defining the output record length, the padding character, and the fields, press ENTER.

22.8 SPECIFYING THE OUTPUT FILE

After SORTINT processes the file(s), the Output Definition screen prompts you to specify the name, library, and volume of the output file. You can also choose to compress the data in the output file. Figure 22-8 illustrates the Output Definition screen.

```
*** MESSAGE 0001 BY SORTI

                INFORMATION REQUIRED BY PROGRAM SORTINT
                          TO DEFINE OUTPUT


Please specify the output file below:

OUTPUT FILE     = ******** in LIBRARY  = JMCDATA* on VOLUME   = ZENITH
Replace input file with same name?    REPLACE  = NO*    (YES or NO)
Do you want the file compressed?      COMPRESS = NO*    (YES or NO)
```

Figure 22-8. The SORTINT Output Definition Screen

To create a new file as output, you must assign different names to the input and output files. If the output file is less than or equal to the input file in size, you can replace the input file with the newly created output. To replace the input file, you assign the same name to the output file as the input file. SORTINT then displays a screen that prompts you to press PF3 to scratch the existing file. When you press PF3, the newly created output file replaces the input file. If you specify YES to the REPLACE option, SORTINT replaces the input file without first prompting you to scratch the existing file.

The output file must be a disk file. You can use the SORT utility to produce tape files as output; refer to Chapter 14 for more information. To specify the output file, you enter values in the following fields:

Field	Description
FILE	The name of the output file.
LIBRARY	The name of the library in which the output file will reside. The default is the value defined for OUTLIB through the Set Usage Constants command of the Command Processor or through a Procedure language SET statement.
VOLUME	The name of the volume on which the output file will reside. The default is the value defined for OUTVOL through the Set Usage Constants command of the Command Processor or through a Procedure language SET statement.
REPLACE	A YES or NO option that allows you to replace the input file without a prompt to scratch the existing file. The default is NO. The REPLACE option facilitates the use of a procedure to run SORTINT. The REPLACE option does not appear if there is more than one input file, if the input file is indexed, or if you are creating an ADDROUT or KEYOUT file.
COMPRESS	A YES or NO option that allows you to compress data in the output file. The default is YES for variable-length input and NO for fixed-length input.

After you specify the output file and press ENTER, SORTINT creates the output file and returns you to the Command Processor. A SORTINT completion message appears on the screen. If SORTINT cannot proceed correctly, an error message and return code appear; refer to Appendix D for a list of the SORTINT return codes.

In some circumstances, during an unsuccessful sort operation, SORTINT gives you the option to restart or to terminate processing. Press PF1 to restart processing; press PF16 to terminate processing. If you press PF16, SORTINT processing ends, and a return code is issued; no output file is created. Because the problem may result from a previous system failure, you should reorganize the file through the COPY utility before you run SORTINT again. For more information on file reorganization, refer to Chapter 2.

22.9 A SAMPLE SORTINT PROCEDURE

You can control SORTINT processing through the VS Procedure language. Appendix A provides a complete list of SORTINT GETPARMS. For detailed information about the syntax of the VS Procedure language, refer to the VS Procedure Language Reference.

The following procedure illustrates the use of the EXTERNAL and REFORMAT options. The output file is sorted in descending order. The selection condition is that an input record is greater than or equal to 'A' in Position 1. Note that you must enclose 'A' in double quotes. Because the VS Procedure language uses quotation marks as delimiters, both sets of quotes are necessary. If you eliminate one set of quotes, or if both sets of quotes are the same, the procedure returns an error message and a return code of 4. (Refer to Appendix D for a list of return codes.)

```
PROCEDURE
RUN SORTINT
ENTER OPTIONS EXTERNAL = YES, REFORMAT = YES
ENTER INTABLE FILE = EXTERNAL, LIBRARY = TRNTABLE, VOLUME = ZENITH
ENTER INPUT FILE = BROKER, LIBRARY = JMCDATA, VOLUME = ZENITH,
    SELECT = YES
ENTER SELECT FLDPOS1 = 1, LENGTH1 = 1, TSTREL1 = GE, VALUE1 = "'A'"
ENTER KEYS POST1 = 1, LENGTH = 1, ORDER1 = D
ENTER FORMAT LENGTH = 100, PAD = 20, INPOS1 = 1, LENGTH1 = 10,
    OUTPOS1 = 1, INPOS2 = 7, LENGTH2 = 10, OUTPOS2 = 11, INPOS3 = 20,
    LENGTH3 = 1, OUTPOS1 = 30
ENTER OUTPUT FILE = BROKER1, LIBRARY = JMCSORT, VOLUME = ZENITH
RETURN
```

CHAPTER 23
THE STABLEMT UTILITY

23.1 INTRODUCTION

The STABLEMT utility enables you to create or modify files that define an external collating sequence. The use of an external collating sequence enables you to accommodate international character sets or sort orders other than standard ASCII. You can then direct the SORTINT utility to sort files according the external collating sequence that you defined rather than according to standard ASCII sequence. (For more information about SORTINT, refer to Chapter 22.)

The file that contains an external collating sequence consists of two translation tables. The primary collating sequence table is a simple one-to-one translation table that assigns a weight to each hex code representing a character. You also have the option to define a two-to-one or one-to-two translation table. A two-to-one translation table specifies pairs of characters that are to be sorted as a single character. A one-to-two translation table specifies a single character that is to be sorted as two characters.

Figure 23-1 gives an overview of STABLEMT processing.

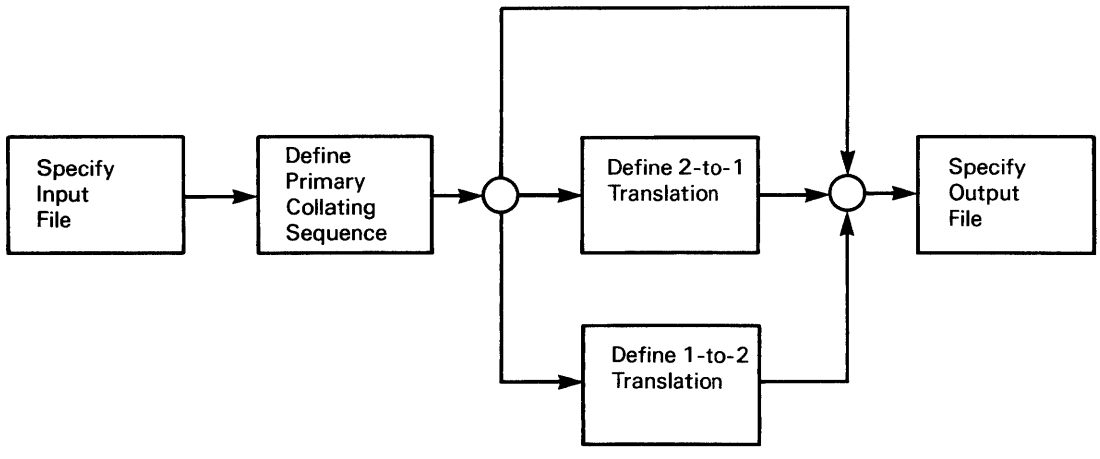


Figure 23-1. STABLEMT Processing

23.2 SPECIFYING THE INPUT FILE

When STABLEMT processing begins, the Input Definition screen prompts you to specify the name, library, and volume of the file you want to edit. If you are creating a new file, leave the fields blank. Figure 23-2 illustrates the Input Definition screen.

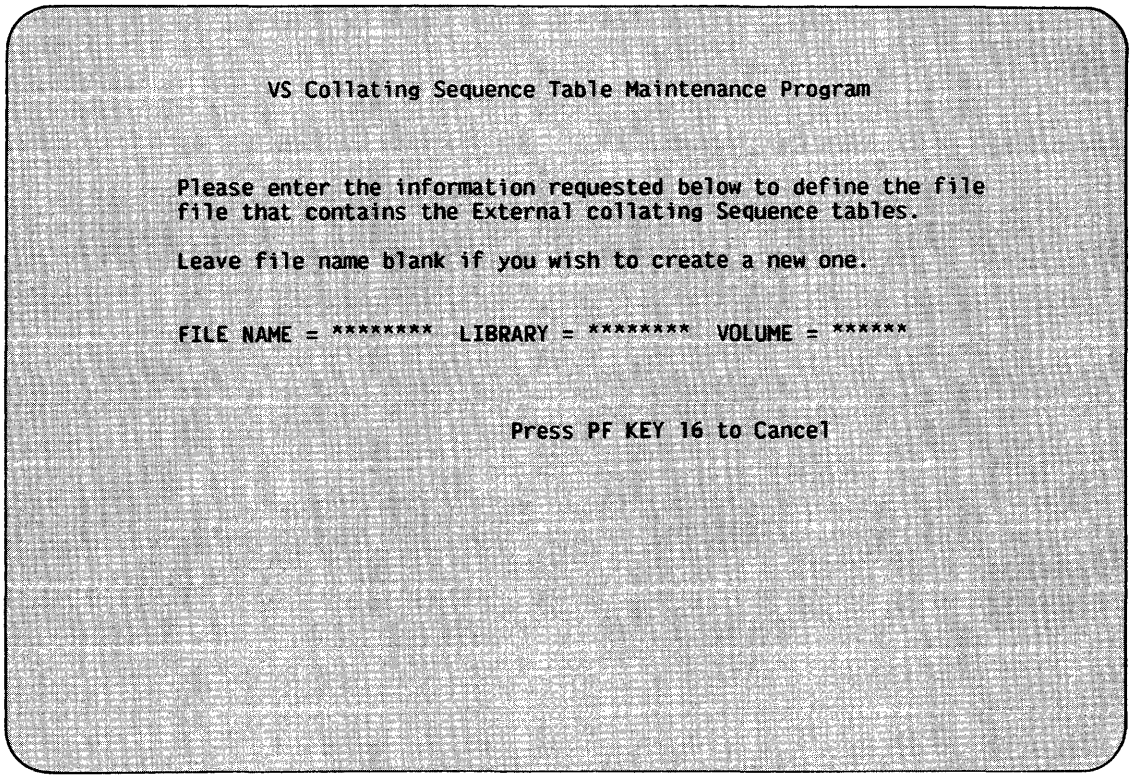


Figure 23-2. The STABLEMT Input Definition Screen

After specifying the input file, press ENTER. You can press PF16 to exit from STABLEMT without creating or editing a file.

23.3 DEFINING THE PRIMARY COLLATING SEQUENCE TABLE

When you press ENTER from the Input Definition screen, STABLEMT displays the primary collating sequence table in the file that you are editing. If you are creating a new file, the table displayed by STABLEMT represents the standard ASCII sequence. Figure 23-3 illustrates the Primary Collating Sequence Table screen; the collating sequence is standard ASCII.

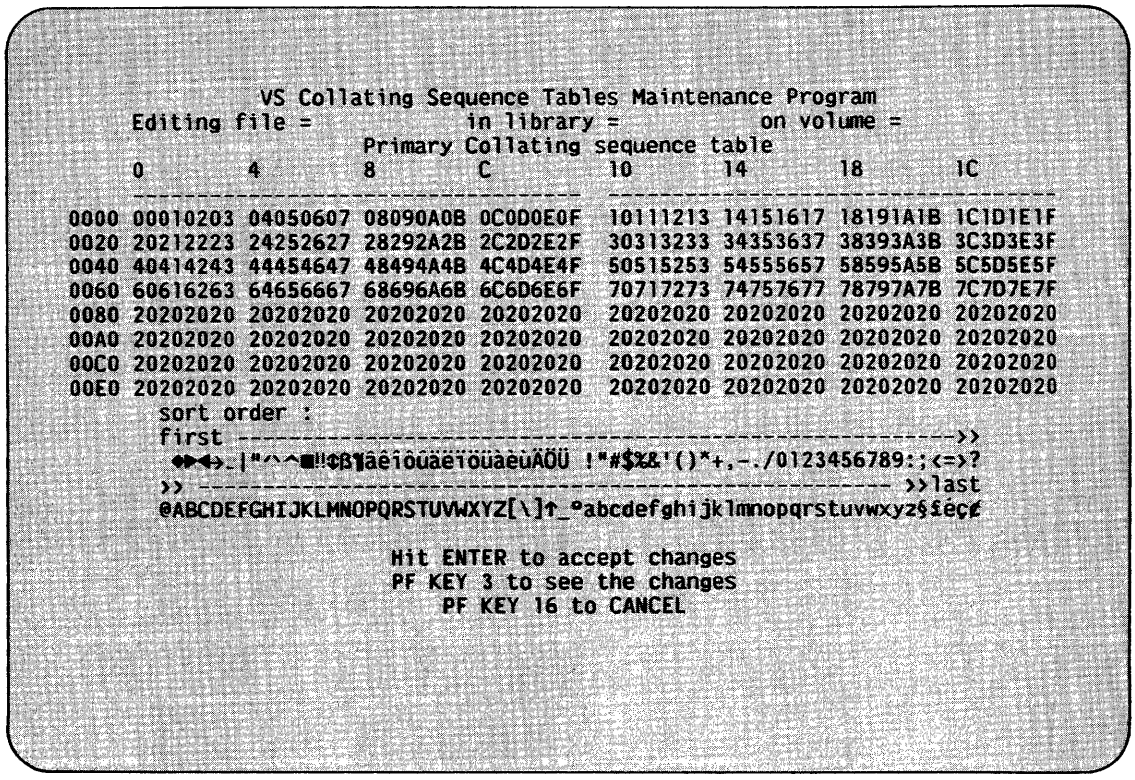


Figure 23-3. A Sample Primary Collating Sequence Table Screen

The primary collating sequence table is a simple one-to-one translation table that assigns a weight to each hex code representing a character. The hex position of a character contains its weight for a sort operation. In the standard ASCII sequence, as in Figure 23-3, the weight of each character is equal to the character's CRT hex code.

When you are defining the primary collating sequence table, consider the following:

- Characters with greater weights are sorted after characters with lower weights. For example, in Figure 23-3, the character b (hex code 62) has a weight of 62, and the character a (hex code 61) has a weight of 61. Therefore, the character b is sorted after the character a.

- Characters with the same weight are sorted as equal to each other.

- To sort a non-English character between two existing characters, you must leave an open weight between the two characters. For example, to sort CH between C and D, you must leave an open weight between these characters and then assign that open weight to CH when you define the two-to-one translation table.

- You place the weight of a character in the table location that corresponds to the CRT hex code of the character.

- STABLEMT does not use locations 80 to FE because characters with hex codes greater than 7F have no meaning in Data Processing mode. You can place values in those locations, but the values have no meaning.

- You cannot assign a weight of FF to a character. You use location FF in the table to indicate whether you want to define a two-to-one or one-to-two translation table.

STABLEMT displays beneath the primary collating sequence table the characters in the current sort order. If you modify the primary collating sequence table and want to see the new sort order, press PF3. For example, to sort ç as equal to the character c, you place a weight of 63 in position 7E (the CRT hex code of ç) because the character c has a weight of 63. If you make this change and then press PF3, the screen shown in Figure 23-4 appears.

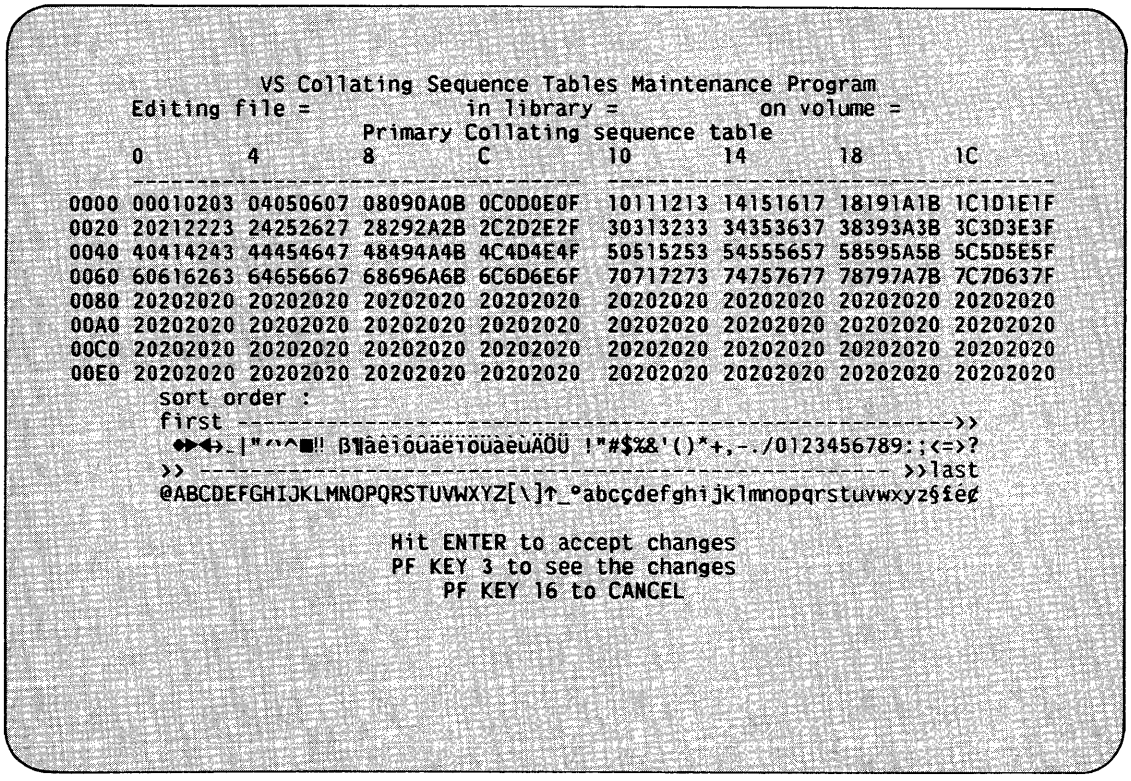


Figure 23-4. A Modified Primary Collating Sequence Table Screen

If you want to define a two-to-one translation table, enter the value 01 in position FF. Enter the value 02 in position FF if you want to define a one-to-two translation table. A value of FF in position FF indicates that you do not want to define an additional translation table. You cannot define both a two-to-one translation table and a one-to-two translation table as part of an external collating sequence.

After you define the primary collating sequence table, press ENTER. You can press PF16 to return to the Input Definition screen.

23.4 DEFINING A TWO-TO-ONE TRANSLATION TABLE

If you enter the value 01 in position FF on the Primary Collating Sequence Table screen, STABLEMT next displays the Two-to-One Translation Table screen. You use this screen to specify pairs of characters that are to be sorted as a single character. You can specify a maximum of 12 pairs. For example, Figure 23-5 shows how to assign a sort weight of 44 to the pairs CH and Ch and a weight of 64 to the pair ch.

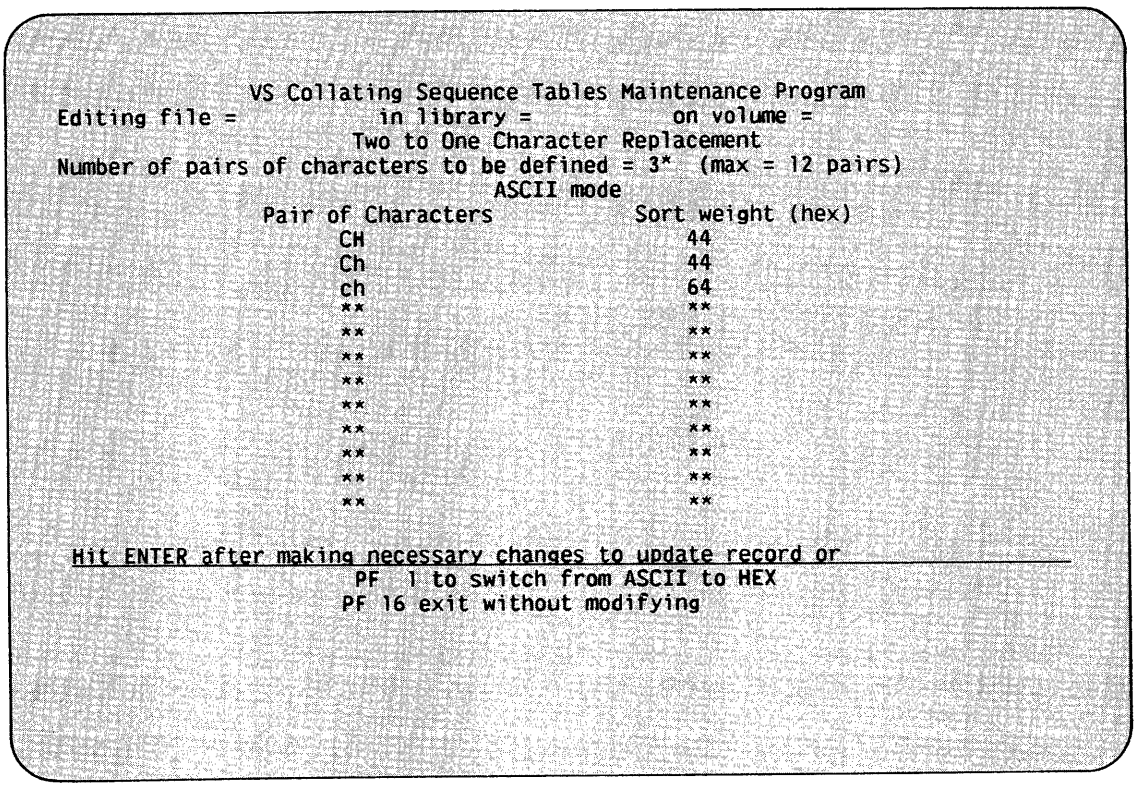


Figure 23-5. A Sample Two-to-One Translation Table Screen

Press PF1 if you want to use hexadecimal rather than ASCII representation. Note that STABLEMT does not retain the displayed values when you change the mode of representation. Figure 23-6 displays in hexadecimal the same information as Figure 23-5. You can press PF1 to return to ASCII from hexadecimal representation.

After you define the two-to-one translation table, press ENTER. You can press PF16 to return to the Primary Collating Sequence Table screen without defining a two-to-one translation table.

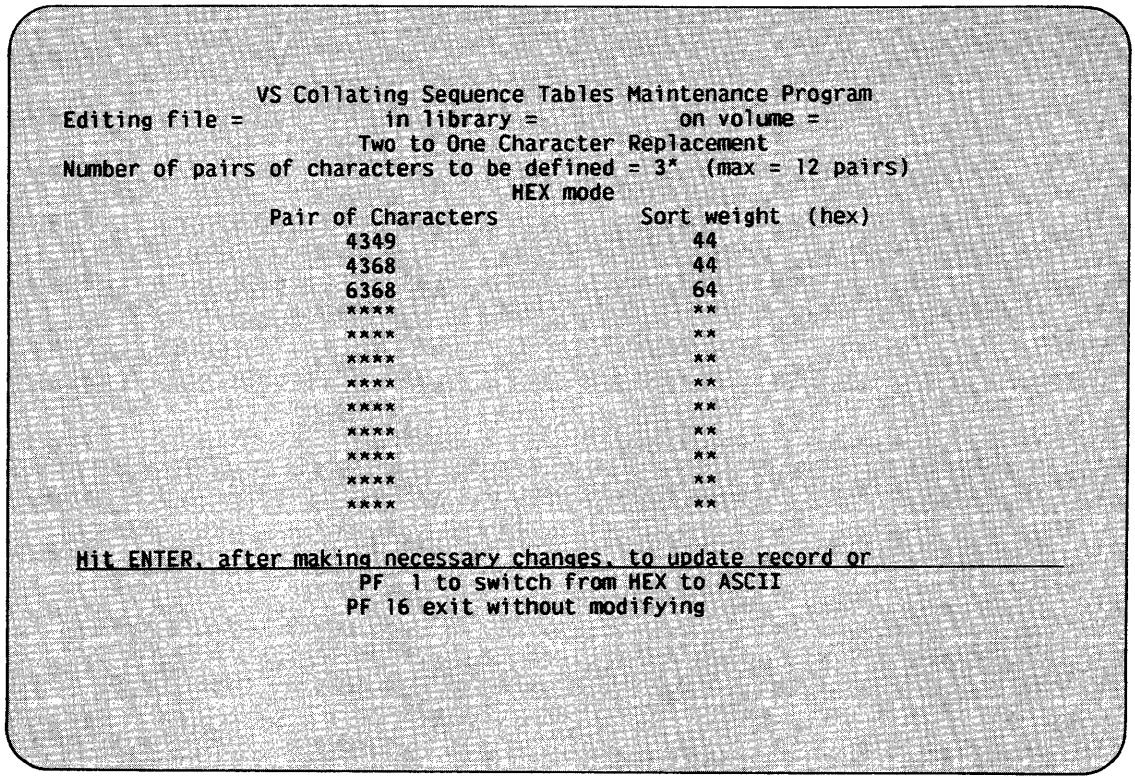


Figure 23-6. A Sample Two-to-One Translation Table in Hex Mode

23.5 DEFINING A ONE-TO-TWO TRANSLATION TABLE

If you enter the value 02 in position FF on the Primary Collating Sequence Table screen, STABLEMT next displays the One-to-Two Translation Table screen. You use this screen to specify single characters that are to be sorted as a pair of characters. You can specify a maximum of 12 characters. For example, Figure 23-7 shows how to sort umlauted vowels as pair of vowels.

Press PF1 if you want to use hexadecimal rather than ASCII representation. Note that STABLEMT does not retain the displayed values when you change the mode of representation. Figure 23-8 displays in hexadecimal the same information as Figure 23-7. You can press PF1 to return to ASCII from hexadecimal representation.

After you define the one-to-two translation table, press ENTER. You can press PF16 to return to the Primary Collating Sequence Table screen without defining a one-to-two translation table.

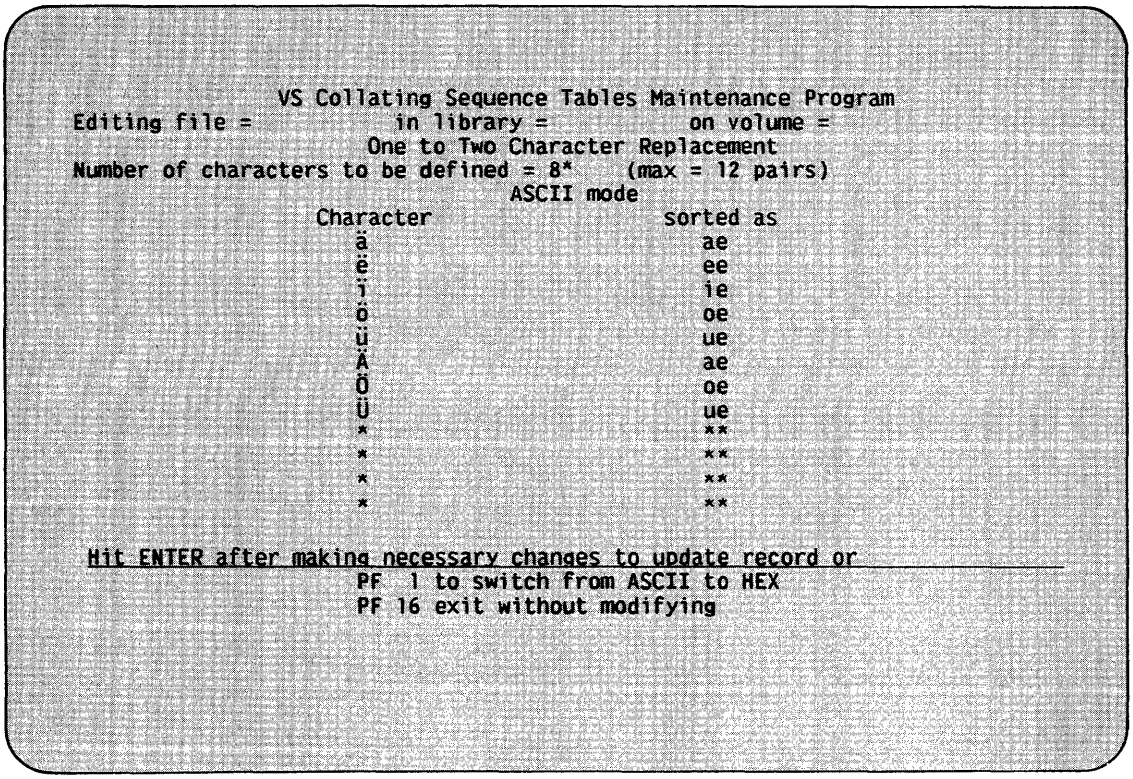


Figure 23-7. A Sample One-to-Two Translation Table Screen

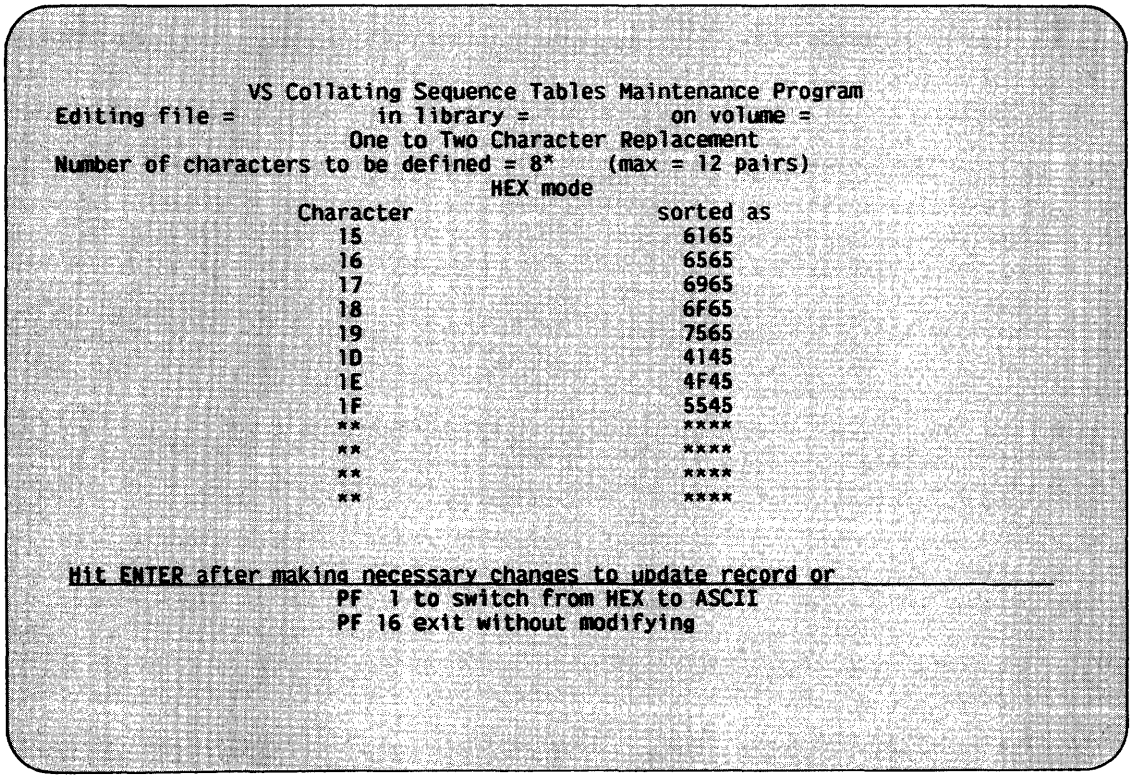


Figure 23-8. A Sample One-to-Two Translation Table in Hex Mode

23.6 SPECIFYING THE OUTPUT FILE

After you define the translation tables, the Output Definition screen prompts you to specify the name, library, and volume of the output file. If you specify the name of an existing file, STABLEMT displays a screen that prompts you to specify another file name or to press PF3 to scratch the existing file. Figure 23-9 illustrates the STABLEMT Output Definition screen.

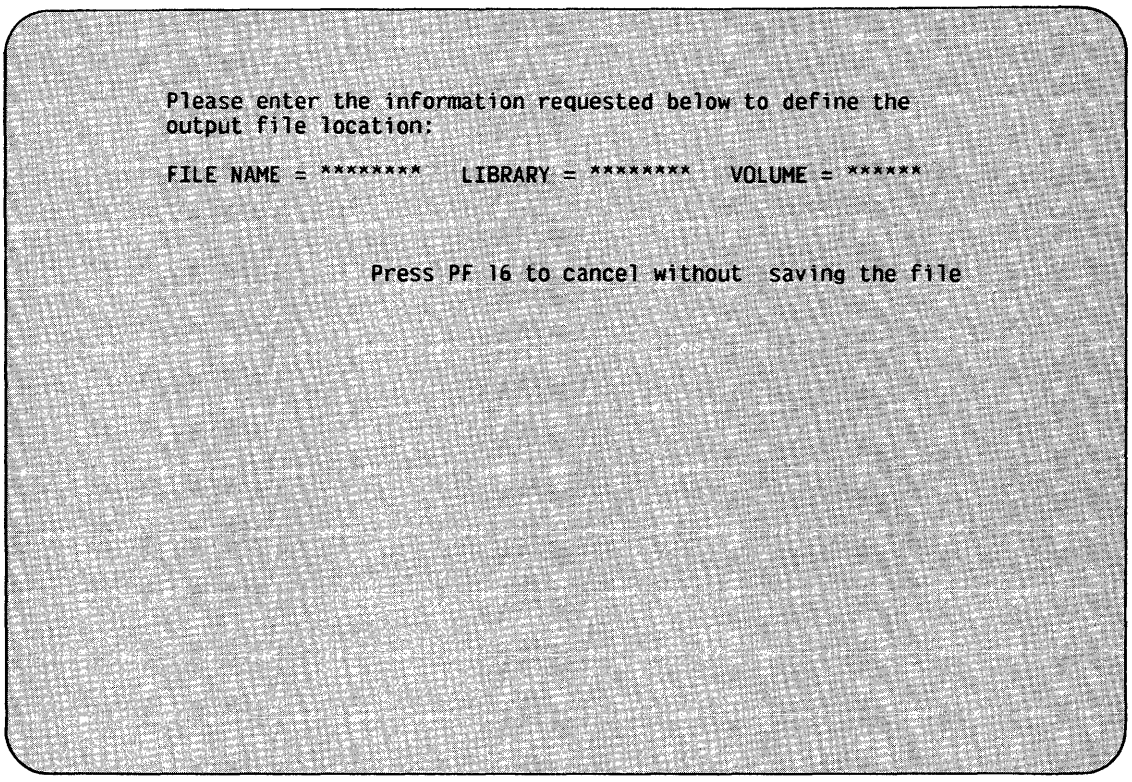


Figure 23-9. The STABLEMT Output Definition Screen

After specifying the output file, press ENTER. STABLEMT then stores the translation tables in the output file and returns you to the Input Definition screen. You can press PF16 to return to the Input Definition screen without saving the translation tables.

APPENDIX A
SYSTEM UTILITY GETPARMS

A.1 INTRODUCTION TO GETPARMS

The VS Operating System supports a supervisor call routine (SVC), the GETPARM SVC, that solicits and accepts runtime parameter information, and displays and awaits acknowledgement of runtime messages. GETPARM-generated prompts are displayed on the workstation screen during normal execution. These prompts solicit parameter information from you or from a controlling procedure. Values entered from either source are verified for validity. If the values entered are not acceptable, the GETPARM SVC responds with an error message.

GETPARM processing is distinguished from other methods of obtaining runtime information primarily because it can interface with a procedure. (Refer to the VS Procedure Language Reference.) A procedure is the preferred source of information for a GETPARM request; GETPARM prompts never appear on the workstation screen when they are satisfied by a Procedure language ENTER statement. Therefore, the procedure writer can precisely control the interaction between a user and an executing program. GETPARM requests are used wherever possible by the VS system programs to solicit parameter information. GETPARM processing enables you to run system programs with little or no user interaction by supplying most or all of the required program parameters from procedures.

A.2 THE STRUCTURE OF A GETPARM

Each GETPARM request in a program is identified with a parameter reference name (prname). The prname for each request is, in general, unique within that program. System programs generally observe certain conventions when identifying GETPARM requests with prnames; for example, a GETPARM request soliciting information for an input file is usually identified with the prname INPUT, while a GETPARM soliciting parameters for an output file is named OUTPUT.

Many GETPARM requests for information contain one or more modifiable fields into which you or a procedure can enter information. Each field is labeled with a keyword. When a GETPARM request is displayed, the keyword for each field provides a description of the information to be supplied for that field. Certain conventions are commonly used in keyword naming. For example, a request for a file name often uses the keyword FILE. Also, many GETPARM requests solicit a PF key response (such as 16 = Exit Program). No keyword is associated with a PF key choice; only the PF key number itself is specified.

COMPRESS-IN-PLACE (CIP)

Prname	Keyword	Length	Options	Defaults
INPUT	VOLUME	6		
	PF Keys*		16 = End Processing	
MOUNT	DEVICE	3		
	PF Keys*		1 = Return to Specify Input	

* The keyword is not required.

COPY

Prname	Keyword	Length	Options	Defaults
INPUT	COPY	7	FILE, LIBRARY, VOLUME	FILE
	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	MODE	6	INPUT, SHARED	INPUT
LOCK	LOCK	3	YES, NO	YES
	TIMEOUT	3	0-255	10
	BYPASS	3	YES, NO	NO
OPTIONS	FILEORG	1	C, I, R	Input value
	LENGTH	1	F, V	Input value
	COMPRESS	1	Y, N	Input value
	REORG	3	YES, NO	NO
	KEYLEN	3	1-999	Input value
	KEYPOS	3	1-99999	Input value
	IPACK	3	1-100	Input value
	DPACK	3	1-100	Input value
	RECBLK	1	A, N, U	A for DMS/TX files, else N
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	RETAIN	3		
	RELEASE	3	YES, NO	YES for consecutive file; NO for indexed files.
	FILECLAS	1	A-Z, #, ¢, @, $	For indexed file, same as input.
	DEVICE	11	DISK, PRINTER	DISK
	PRTCLASS	1	A-Z	A
	FORM#	3	0-255	000
	COPIES	3	1-32, 767	1
OPTIONS*	OPTION	1	N, S, R, C	N
	NEWNAME	8		
EOJ	PF Keys**		1 = Rerun COPY 16 = End processing	

* For duplicate file names

** The keyword is not required.

Prname	Keyword	Length	Options	Defaults
MENU	PF Keys		1 = Initialize diskette 2 = Copy File from VS to diskette 4 = Copy/Convert OIS files to VS 5 = Create a diskette Catalog listing 7 = Rename a file on diskette 8 = Delete a file from diskette 9 = Assign a diskette Volume/File password 12 = Display the file conversion listing (with PF4) 14 = Display the diskette catalog listing (with PF5) 16 = End Processing	

Initialize Diskette -- PF1

VOLUME	VOLUME	8	OIS Volume Name	
	PASSWORD	8	Legal OIS password	
	PF Keys*		↵ = Continue 1 = Return to Main Menu files	
MOUNT (if not already mounted)	VOLUME	6	VS Volume Name	
	DEVICE	8	None	DISKETTE
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		↵ = Continue 1 = Return to previous screen	

Copy File from VS to Diskette -- PF2

VSFILE	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	PF Keys*		↵ = Continue 1 = Terminate this file copy 9 = Multiple file copy	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
OISFILE (1 for each file)	VOLUME	8		
	FILE	63		OIS file name
	PF Keys*		⍉ = Continue 1 = Respecify input	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	DISK or DISKETTE	
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		⍉ = Continue 1 = Return to previous screen	

Copy/Convert OIS file to VS -- PF4

INPUT	VOLUME	8	OIS or VS Volume Name	User's INVOL
	FILE	63	VS, OIS	VS
	DEVICE	3		
	CONVERT	3	YES or NO	YES
	PF Keys*		⍉ = Continue 1 = Return to main menu 9 = Multiple file copy/convert	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	DISK or DISKETTE	
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		⍉ = Continue 1 = Return to previous screen	
OPTIONS	FORMAT	8	FIXED, VARIABLE, KEYED	FIXED
	LENGTH	4		
	NUMERICS	8	INTEGER, FLOATING, PACKED	
	DECIMALS	2	Decimal Alignment for Packed Numeric Format Only	
	KEYFIELD	3	YES or NO	
	KEYSTART	4	Keyfield Start Position	
	PF Keys*		⍉ = Continue 1 = Respecify input 13 = instructions	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
DATAFILE	VOLUME	8		Invol, outvol or input file volume
	FILE	63		
	DEVICE	3	OIS or VS	
	PF Keys*		∅ = Continue 1 = Return to main menu	
OUTPUT	FILE	8		
	LIBRARY	8		
	VOLUME	6		User's OUTVOL
	PF Keys*		∅ = Continue 1 = Respecify input menu	

Create a Diskette Catalog Listing -- PF5

CATALOG	VOLUME	8		
	PF Keys*		∅ = Continue 1 = Return to main menu	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	None	DISKETTE
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		∅ = Continue 1 = Return to previous screen	

Rename a Diskette File -- PF7

RENAME	VOLUME	8		
	FILE	63		
	PF Keys*		∅ = Continue 1 = Return to previous screen	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	None	DISKETTE
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		∅ = Continue 1 = Return to previous screen	

*The keyword is not required.

Prname	Keyword	Length	Options	Defaults
NEWNAME	FILE	63		
	PF Keys*		↵ = Continue 1 = Return to previous screen	
			Delete a file from Diskette -- PF8	
DELETE	VOLUME	8		
	FILE	63		
	PF Keys*		↵ = Continue 1 = Return to previous screen	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	None	DISKETTE
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		↵ = Continue 1 = Return to previous screen	
			Assign a Diskette Volume/File Password -- PF9	
PASSWORD	VOLUME	8		
	NAME	63		
	PASSWORD	8		
	PF Keys*		↵ = Continue 1 = Return to main menu	
MOUNT (if not already mounted)	VOLUME	6	OIS Volume Name	
	DEVICE	8	None	DISKETTE
	UNIT	3		
	PASSWORD	8	when applicable	
	PF Keys*		↵ = Continue 1 = Return to previous screen	

*The keyword is not required.

Prname	Keyword	Length	Options	Defaults
ACTION	PF Keys*		1 = Create VS files from files on a 2200 diskette or in a VS image file 2 = Create a 2200 diskette or VS image file from VS files 9 = Create a VS image file from a diskette(s) 10 = Create a diskette(s) from a VS image file 14 = Dismount the diskette most recently mounted through COPY2200 (only if a diskette was mounted) 16 = End processing	

Create VS Files from Files on a 2200 Diskette or in a VS Image File -- PF1

INPUT	DISKETTE	6		
	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	STARTER	1	A-Z, 0-9, $, @	V
	FILLER	1	A-Z, 0-9, #, $, @	#
	DEVICE#	3		
	PF Keys*		∅ = Continue 1 = Return to main menu 4 = Mount input volume	
COPYMODE	MODE	1	A,D,P,L	
	PF KEYS*		∅ = Continue 1 = Return to Input Definition Screen	
FILELIST	(Keywords are names of files	1	Blank, nonblank	blank
	PF Keys*		∅ = Continue 1 = Return to Copy Mode screen	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
PROGOUT	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	FILECLAS	1	A-Z,#,ϕ,$,@	User's FILECLAS
	CONVERT	3	YES, NO	YES
	COMPRESS	3	YES, NO	YES
DATAOUT	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	FILECLAS	1	A-Z,#,ϕ,$,@	Users FILECLAS
	TYPE	1	F,V,T,X,E,S,N	
	COMPRESS	3	YES,NO	YES
	RECSIZE	4	0-2048 (required only if TYPE = F,V,T or X)	
	TRANSL	3	YES,NO (required only if TYPE = T or X)	NO
	PF Keys*		ϕ = Continue 14 = Display Information	
DATAINFO	PF Keys*		ϕ = Display other information screen 1 = Return to Data Files Output screen	
NUMERICS	TYPE	1	D,F,B,H,P,N	
	LENGTH	2	1-16	
	PLACES	2	0-99	

* The keyword is not required.

Create a 2200 Diskette or VS Image File from VS Files -- PF2

Prname	Keyword	Length	Options	Defaults
OUTPUT	DISKETTE	6		
	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	FILECLAS	1	A-Z,#,ϕ,$,@	User's FILECLAS
	COMPRESS	3	YES,NO	YES
	SECTORS	5	1-32767, ALL	1024
	INDEX	3	1-255	16
	DEVICE#	3		
	PF Keys*		ϕ = Continue 1 = Return to main menu 4 = Mount output volume	
CONTINUE	CONTINUE	3	YES, anything else	
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	DEVICE#	3		
	PF Keys*		ϕ = Continue specifying files 4 = Mount input volume 16 = Copy no more files	
FILETYPE	TYPE	1	T,X,E,S	
	FILENAME	8		Input file name
	RECSIZE	4	1-9999	
	TRANSL	3	YES,NO	NO

* The keyword is not required.

Create a VS Image File from a Diskette(s) -- PF 9

Prname	Keyword	Length	Options	Defaults
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	FILECLAS	1	A-Z,#,ϕ,$,@	User's FILECLAS
	COMPRESS	3	YES,NO	YES
	TYPE	1	T,N	T
	MULTIPLE	3	YES,NO	NO
	SECTORS	5	USED,ALL,1-32767	ALL
	DEVICE#	3		
	PF Keys*		ϕ = Continue 1 = Return to main menu 4 = Mount output volume	
INPUT	DISKETTE	6		
	SECTORS	4	ALL, 1-3874 (Keyword appears only if MULTIPLE = YES)	
	DEVICE#	3	ϕ = Continue	
	PF Keys*		1 = Return to Output Definition screen 4 = Mount diskette	

Create a Diskette(s) from a VS Image File -- PF 10

INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	TYPE	1	T,N	T
	MULTIPLE	3	YES, NO	NO
	DEVICE#	3		
	PF Keys*		ϕ = Continue 1 = Return to main menu 4 = Mount input volume	
OUTPUT	DISKETTE	6		
	SECTORS	4	ALL,1-3874 (Keyword appears only if MULTIPLE = YES)	
	DEVICE#	3		
	PF Keys*		ϕ = Continue 1 = Return to Input Definition Screen 4 = Mount diskette	
CONTINUE	CONTINUE	3	YES, anything else	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
FUNCTION	PF Keys*		1 = Copy Single Document 3 = Copy Document Library 4 = Delete Single Document 6 = Delete Document Library 7 = Rename Single Document 9 = Rename Document Library 10 = Reorganize Single Document 12 = Reorganize Document Library 13 = Convert Document to VS File 14 = Convert VS File to Document 15 = Document Merge 16 = Terminate Processing	
INTERNAT	DATE	1	A, E	A if the system is generated with American dates; E if the system is generated with European dates.
	DECALIGN	1	'.', ','	'.' if the system is generated with American dates; ',' if the system is generated with European dates.
	CURRENCY	1	Local currency symbols	$
	REQSPACE	2	Any hexadecimal value in the ASCII character set	5C
	DEVCHARS	3	YES, NO	YES
	Copy or Rename Single Document -- PF1 or PF7			
INPUT	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		↵ = Continue 1 = Return to main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
PASSWORD	PASSWORD	6		
	PF Keys*		↵ = Continue 1 = Return to main menu	
OUTPUT	DOCUMENT	5	Valid document IDs, including next and NEXT	
	VOLUME	6		Document library's standard volume if WP is installed)
	PF Keys*		↵ = Continue 1 = Return to main menu	

Copy or Rename Document Library -- PF3 or PF9

INPUT	LIBRARY	1		
	VOLUME	6		Library's standard volume (if WP is installed)
	PF Keys*		↵ = Continue 1 = Return to main menu	
OUTPUT	LIBRARY	1		
	VOLUME	6		Library's standard volume (if WP is installed)
	RENUMBER	3	YES, NO	NO
	DOCUMENT	4	Four digits	0001
	DUPFILES	6	PROMPT, NOCOPY, DELETE	PROMPT
	PF Keys*		↵ = Continue 1 = Return to main menu	
PASSWORD	PASSWORD	6		
	PF Keys*		↵ = Continue 1 = Return to main menu 2 = Bypass current document	
SAMEFILE	OPTION	1	N, D, R, C,	N
	NEWDOCID	4	Four digits, next, NEXT	
	PF Keys*		↵ = Continue 1 = Return to main menu	
ERROR	PF Keys*		1 = Return to main menu 2 = Skip current document (not available for some error conditions)	

* The keyword is not required.

Delete Single Document -- PF4

Prname	Keyword	Length	Options	Defaults
INPUT	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	ERASE	3	YES, NO	NO
	PF Keys*		⍉ = Continue 1 = Return to main menu	
PASSWORD	PASSWORD	6		
	PF Keys*		⍉ = Continue 1 = Return to main menu	

Delete Document Library -- PF6

INPUT	LIBRARY	1		
	VOLUME	6		Library's standard volume (if WP is installed)
	ERASE	3	YES, NO	NO
	PF Keys*		⍉ = Continue 1 = Return to main menu	
PASSWORD	PASSWORD	6		
	PF Keys*		⍉ = Continue 1 = Return to main menu 2 = Bypass current document	
ERROR	PF Keys*		1 = Return to main menu 2 = Skip current document (not available for some error conditions)	

Reorganize Single Document - PF10

INPUT	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		⍉ = Continue 1 = Return to main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
PASSWORD	PASSWORD	6		
	PF Keys*		↵ = Continue 1 = Return to main menu	
			Reorganize Document Library -- PF12	
INPUT	LIBRARY	1		
	VOLUME	6		Library's standard volume (if WP is installed)
	RENUMBER	3	YES, NO	NO
	DOCUMENT	4	Four digits, excluding next and NEXT	0001
	PF Keys*		↵ = Continue 1 = Return to main menu	
ERROR	PF Keys*		1 = Return to main menu 2 = Skip current document (not available for some error conditions)	
			Convert Document to VS File -- PF13	
INPUT	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		↵ = Continue 1 = Return to main menu	
PASSWORD	PASSWORD	6		
	PF Keys*		↵ = Continue 1 = Return to main menu	
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	TYPE	6	DATA, PRINT, SOURCE, TC	
	PF Keys*		↵ = Continue 1 = Return to main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
PRINT (for TYPE = PRINT only)	START	3		Value from last document print request
	FINISH	3		Total number of pages in document
	NUMBER	4		Value from last document print request
	HEADER	3		Value from last document print request
	FOOTER	3		Value from last document print request
	LINE	2		Value from last document print request
	MARGIN	3		Value from last document print request
	FORMAT	11	JUSTIFIED, UNNUSTIFIED, NOTES	Value from last document print request
	STYLE	5	FINAL, DRAFT	Value from last document print request
	SUMMARY	5	PRINT, OMIT	Value from last document print request
	PF Keys*		⍉ = Continue 1 = Return to main menu	
			Convert VS File to Document - PF14	
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	PF Keys*		⍉ = Continue 1 = Return to main menu	
OUTPUT	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume if WP is installed)

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
	TITLE	25		Input file's document title (TC file only)
	OPERATOR	20		Input file's document operator (TC file only)
	AUTHOR	20		Input file's document author (TC file only)
	COMMENTS	20		Input file's document comments (TC file only)
	TYPE	6	SOURCE, IMAGE (required only for 80-byte record input files)	SOURCE
	TABS	3	YES, NO (required only for Print and Source input files)	YES
	PAGE	3	1-255 (required only for Source input files)	User's current LINES
	PF Keys*		⍉ = Continue 1 = Return to main menu	
OPTIONS (TYPE = IMAGE only)	TABS	59	Up to 15 sets of 1 to 3 digit numbers, separated by commas or spaces	
	LENGTH	3		The smaller value of either the file's maximum record length plus 1, or 158
	RETURN	3	YES, NO	YES
	TRUNCATE	3	YES, NO	YES
	PF Keys*		⍉ = Continue 1 = Return to main menu	
			Document Merge -- PF15	
PRIMARY	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		⍉ = Continue 1 = Return to main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
SECOND	DOCUMENT	5	Valid document IDs, excluding next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		⍉ = Continue 1 = Return to main menu	
PASSWORD	PASSWORD	6		
	PF Keys*		⍉ = Continue 1 = Return to main menu	
OUTPUT	DOCUMENT	5	Valid document IDs, including next and NEXT	
	VOLUME	6		Document library's standard volume (if WP is installed)
	PF Keys*		⍉ = Continue 1 = Return to main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
FUNCTION	FUNCTION	10	INITIALIZE, REFORMAT RELABEL, VERIFY, REMOVE	VERIFY
	VOLUME	6		
MOUNT	DEVICE	3		
	PLATTER	9	REMOVABLE, FIXED	REMOVABLE
			INITIALIZE VOLUME	
ACTION (ONLY if input volume is NL)	PF Keys*		0 = Continue 1 = Go to EOJ	
INPUT	NEWVOL	6		Volume name specified on Function menu
	LABEL	2	SL,NL	SL
	TOLERATE	5	NONE, CRASH, MEDIA	NONE
	VTOCSIZE	3		Standard size of VTOC for input disk volume type
	OWNER	14		
	PASSES	6	BRIEF, NORMAL	BRIEF
	DUMPFILE	3	YES, NO	NO
	PAGEPOOL	3	YES, NO	NO
	PF Keys*		0 = Continue 1 = Go to EOJ	
EOJ	PF Keys*		1 = Rerun DISKINIT 16 = End processing	
			REFORMAT VOLUME	
ACTION (only if input volume is NL)	PF Keys*		0 = Continue 1 = Go to EOJ	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
INPUT	NEWVOL	6		Volume name specified on Function menu
	TOLERATE	5	NONE, CRASH, MEDIA	NONE
	VTOCSIZE	4		Standard size of
	OWNER	14		VTOC for input disk volume type
	DUMPFILE	3	YES, NO	YES if volume has a dump file; NO if not
	PAGEPOOL	3	YES, NO	YES if volume has a page block or pool; NO if not
EOJ	PF Keys*	14	1 = Rerun DISKINIT 16 = End processing	
			RELABEL VOLUME	
INPUT	NEWVOL	6		
	PAGEPOOL	3	YES, NO	YES if volume has a page block or pool; NO if not
	PF Keys*		ϕ = Continue 1 = Go to EOJ	
EOJ	PF Keys*	6	1 = Rerun DISKINIT 16 = End processing	
			VERIFY VOLUME	
ACTION (only if input volume is NL)	PF Keys*		ϕ = Continue 1 = Go to EOJ	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
INPUT	PF Keys*		∅ = Continue 1 = Go to EOJ	
EOJ	PF Keys*		1 = Rerun DISKINIT 16 = End processing	
			REMOVE BLOCK FROM VOLUME	
ACTION (only if input volume is NL)	PF Keys*		∅ = Continue 1 = Go to EOJ	
INPUT	BLOCK PF Keys*	6	∅ = Continue 1 = Go to EOJ	
EOJ	PF Keys*		1 = Rerun DISKINIT 16 = End processing	
			ALLOCATE DUMP FILE	
DUMPFILE	SIZE	5	0-65536; entry should correspond to size of main memory	Size of the current dump file if it exists
			ALLOCATE PAGE BLOCK OR POOL	
PAGEPOOL	SIZE	5	0-65536	Size of the current page block or pool if it exists
	LOCATION	1	0-9	Location of the current page block or pool if it exists
POOLSIZE	PF Keys*		∅ = Continue	

* The keyword is not required.

DISPLAY

Prname	Keyword	Length	Options	Defaults
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	ACCESS	6	RECORD, BLOCK, PRINT	RECORD
	MODE	6	INPUT, SHARED	INPUT
LOCK	LOCK	3	YES, NO	NO
	TIMEOUT	3	0-255	10
	BYPASS	3	YES, NO	NO
EOJ	PF Keys*		1 = Display another file 16 = End processing	
PRINT**	(see PRINT under DEFAULT GETPARMS)			

* The keyword is not required.

** Default GETPARM

Prname	Keyword	Length	Options	Defaults
OPTIONS	LANGUAGE	10	ASSEMBLER, A, BASIC, B, COBOL, C, D, MENU, M, RPG, R	
	PROCEDUR	3	YES, NO	YES
	PF Keys*		2 = Define new screen format 3 = Modify existing screen format	
INSCREEN	FILE	8		
	LIBRARY	8		User ID + SAVE
	VOLUME	6		User's INVOL
FUNCTION	PF Keys*		2 = Save only generated output 3 = Save only screen contents 4 = Save screen contents and generated output 16 = Exit without saving any files	
SCREEN	FILE	8		
	LIBRARY	8		User ID + SAVE
	VOLUME	6		User's OUTVOL
SOURCE (Menu option only)	PF Keys*		2 = Assembly language 3 = RPG II	
SOURCE (Data Entry option only)	PF Keys*		1 = COBOL 3 = RPGII 16 = Return	
CONTROL (Data Entry option only)	FILE	8		
	LIBRARY	8		User ID + CTL
	VOLUME	6		User's INVOL
	PF Keys*		0 = Continue 1 = Switch to COBOL option 2 = Invoke CONTROL 16 = Exit EZFORMAT	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
COPYLIBR	FILE	8		
	LIBRARY	8		User ID + COPY
	VOLUME	6		User's INVOL
PROGRAM	FILE	8		
	LIBRARY	8		
	VOLUME	6		User's OUTVOL

Prname	Keyword	Length	Options	Defaults
OPTIONS	OPTION	1	D, C, G	
	PF Keys*		∅ = Continue	
			16 = End processing	
			Duplicate Diskette -- D	
INPUT	VOLSER	6		
	LABEL	2	SL, NL	SL
	DEVICE	3		
OUTPUT	VOLSER	6		
	DEVICE	3		
EOJ	PF Keys*		1 = Rerun FLOPYDUP	
			2 = Generate another diskette with the same data	
			16 = End processing	
			Copy a Diskette to a Diskette Image File -- C	
INPUT	VOLSER	6		
	LABEL	2	SL, NL	SL
	DEVICE	3		
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	RECORDS	7		154
	RETAIN	3		
	RELEASE	3	YES, NO	YES
	FILECLAS	1		
	DEVICE	11	DISK	DISK
EOJ	PF Keys*		1 = Rerun FLOPYDUP	
			16 = End processing	

* The keyword is not required.

Generate a Diskette from a Diskette Image File -- G

Prname	Keyword	Length	Options	Defaults
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	DEVICE	11	DISK, NONE	DISK
OUTPUT	VOLSER	6		
	DEVICE	3		
EOJ	PF Keys*		1 = Rerun FLOPYDUP 2 = Generate another diskette with the same data 16 = End processing	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
FUNCTION	PF Keys*		1 = Copy files from IBM to VS 2 = Copy files from VS to IBM 3 = Initialize diskette to IBM format 4 = Mount IBM diskette 5 = Display an IBM diskette directory 16 = End processing	
			IBM to VS Copy -- PF 1	
OPTIONS	COPY	6	VOLUME, SELECT, FILE	VOLUME
	VOLUME	6		
	FILE	8	Valid IBM file name	
	PF Keys*		⍉ = Continue 1 = Return to the main menu	
MOUNT	DEVICE	3		
	PF Keys*		⍉ = Continue 1 = Return to IBM to VS Input Definition screen	
INPUT (COPY = SELCT only)	FILE0001 through FILE0020	1	A nonblank character selects the file	blank
	PF Keys*		⍉ = Continue 1 = Return to IBM to VS Input Definition screen 16 = End input definition	
OUTPUT	LIBRARY	8		
	VOLUME	6		
	TRANSL	3	YES, NO	YES
	PF Keys*		⍉ = Continue 1 = Return to IBM to VS Input Definition screen 2 = Specify output options separately for each file 16 = Return to the main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
IOUTPUT	FILE	8		
	LIBRARY	8		
	VOLUME	6		
	TRANSL	3	YES, NO	YES
	RECORDS	7	1 to 9999999	Number of input file records
	RELEASE	3	YES, NO	YES
	FILEORG	1	C, I	C
	COMPRESS	3	YES, NO	YES
	FILECLAS	1	A-Z, $, #, @, ⍉	User's FILECLAS
	KEYLEN	3		
	KEYPOS	5		
	IPACK	3	0-100	100
	DPACK	3	0-100	100
	PF Keys*		⍉ = Continue 16 = Cancel and go to EOJ screen	
NEWFILE	FILE	8		
	PF Keys*		⍉ = Continue 2 = Skip the current file 16 = Cancel and go to the EOJ screen	
NEWVOL	VOLUME	6		
	PF Keys*		⍉ = Continue 2 = Skip the current file 16 = Cancel and go to the EOJ screen	
BADIO	PF Keys*		⍉ = Acknowlege error and go to EOJ sreen	
CANCEL	PF Keys*		⍉ = Acknowlege error and go to EOJ screen	
EOJ	PF Keys*		1 = Return to IBM to VS Input Definition screen 2 = Return to the main menu 16 = End processing	

* The keyword is not required.

VS to IBM Copy -- PF2

Prname	Keyword	Length	Options	Defaults
OPTIONS	COPY	6	LIBRARY, SELECT, FILE	LIBRARY
	VOLUME	6		
	FILE	8	Valid IBM file name	
	PF Keys*		0 = Continue 1 = Return to the main menu	
INPUT (COPY = SELECT only)	FILE0001 through FILE0020**	1	A nonblank character selects the file	blank
	PF Keys*		0 = Continue 1 = Return to VS to IBM Input Definition screen 16 = End input definition	
OUTPUT	VOLUME	6		
	PADCHAR	1	blank, 0	blank
	TRANSL	3	YES, NO	YES
	PF Keys*		0 = Continue 1 = Return to VS to IBM Input Definition screen 2 = Specify output options separately for each file 16 = Return to EOJ screen	
MOUNT	DEVICE	3		
	PF Keys*		0 = Continue 1 = Return to Output Definition screen	

* The keyword is not required.

** These names are changed to the actual file names.

Prname	Keyword	Length	Options	Defaults
IOUTPUT	FILE	8		
	VOLUME	6		
	PADCHAR	1	blank, 0	blank
	TRANSL	3	YES, NO	YES
	PF Keys*		⌀ = Continue 16 = Go to EOJ screen	
CLEAROUT	NEWINDEX	3	YES, NO	NO
	PF Keys*		⌀ = Continue 16 = Cancel and go to EOJ screen	
BIGRECS	PF Keys*		⌀ = Continue 2 = Skip current file 16 = Cancel and go to EOJ screen	
ABORT	PF Keys*		⌀ = Acknowledge error and go to the EOJ screen	
FULLVOL	VOLUME	6		
	PF Keys*		⌀ = Continue 2 = Skip current file and copy 3 = Copy and create a multi-volume file if needed 16 = Cancel and go to the EOJ screen	
DUPLNAM	FILE	8		
	PF Keys*		⌀ = Continue 2 = Skip current file and copy 3 = Scratch existing file on output volume 16 = Cancel and go to the EOJ screen	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
NEWFILE	PF Keys*		2 = Skip current file 16 = Cancel and go to the EOJ screen	
NEWVOL	VOLUME	6		
	PF Keys*		⏎ = Continue 2 = Skip the current file 16 = Cancel and go to the EOJ screen	
BADIO	PF Keys*		⏎ = Acknowledge error and go to the EOJ screen	
CANCEL	PF Keys*		⏎ = Acknowledge error and go to the EOJ screen	
EOJ	PF Keys*		1 = Return to VS to IBM Input Definition screen 2 = Return to the main menu 16 = End processing	

Initialize Diskette to IBM Format -- PF 3

VOLUME	VOLUME	6		
	PF Keys*		⏎ = Continue 1 = Return to the main menu	
MOUNT	DEVICE	3		
	PF Keys*		⏎ = Continue 1 = Return to the Define Volume screen	
ONESIDED	PF Keys*		⏎ = Initialize diskette 1 = Return to the Define Volume screen	
TWOSIDED	FORMAT	5	BASIC, H	BASIC
	PF Keys*		⏎ = Initialize diskette 1 = Return to the Define Volume screen	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
BADIO	PF Keys*		⍉ = Acknowledge error and go to the EOJ screen	
CANCEL	PF Keys*		⍉ = Acknowledge error and go to the EOJ screen	
EOJ	PF Keys*		1 = Return to the Define Volume screen 2 = Return to the main menu 16 = End processing	
		Mount an IBM Diskette -- PF 4		
MOUNTCOM	VOLUME	6		
	DEVICE	3		
	PF Keys*		⍉ = Continue 1 = Return to the main menu	
BADIO	PF Keys*		⍉ = Acknowledge error and return to the main menu	
CANCEL	PF Keys*		⍉ = Acknowledge error and return to the main menu	
		Display or Print an IBM Diskette Directory -- PF5		
VOLUME	VOLUME	6		
	DEVICE	7	DISPLAY, PRINTER	DISPLAY
	PF Keys*		⍉ = Continue 1 = Return to the main menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
FUNCTION	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	PF keys*		1 = Analyze contents of I/O Error Log File 2 = Print contents of I/O Error Log File 13 = Help 16 = Terminate Processing	
RANGE	STARTDAY	8		Defaults to first and last records contained in the I/O Error Log file.
	STARTTIME	5		
	ENDDAY	8		
	ENDTIME	5		
	PF Key*		1 = Return to Function Selection Menu	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
INPUT	LOWDEV	3		System-generated
	HIGHDEV	3		System-generated
	IOSWTRAP	16	0-9, A-F, X	XXXXXXXXXXXXXXXX
	CIOTRAP	16	0-9, A-F, X	XXXXXXXXXXXXXXXX
	SIOTRAP	16	0-9, A-F, X	XXXXXXXXXXXXXXXX
	HIOTRAP	16	0-9, A-F, X	XXXXXXXXXXXXXXXX
	VOLUME	6		System volume
	TRAPREQ	3	0-999	1
	PF Keys*		16 = End processing	
CCTRAPS	SIODB	1	Y, N	N
	SIOIB	1	Y, N	N
	SIOIN	1	Y, N	N
	CIODB	1	Y, N	N
	CIOIB	1	Y, N	N
	CIOIN	1	Y, N	N
	HIODB	1	Y, N	N
	HIOIB	1	Y, N	N
	HIOIN	1	Y, N	N
	PF Keys*		1 = Return	

* The keyword is not required.

LISTVTOC

Prname	Keyword	Length	Options	Defaults
INPUT	VOLUME	6		
	FILES	3	YES, NO	NO
	LIBRARY	8	A library name or blank	Blank
	VTOCMAP	3	YES, NO	NO
	SCREEN	3	YES, NO	YES
	PRINT	2	4-99	45

* Default GETPARM

Prname	Keyword	Length	Options	Defaults
OPTIONS	FUNCTION	5	SORT, MERGE	SORT
	MEMORY	3		128
	ADDROUT	3	YES, NO	NO
	KEYOUT	3	YES, NO	NO
	STABLE	3	YES, NO	NO
INPUT (Up to 20 input files if MERGE or MOREFILE = YES)	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	SELECT	3	YES, NO	NO
	MOREFILE (only if FUNCTION=SORT)	3	YES, NO	NO
	SHARED	3	YES, NO	NO
	DEVICE	4	DISK, TAPE	DISK
	FILESEQ	4	1-9999	1
	RECORDS	6	1-999999	1000
LOCK (Only if SHARED = YES)	LOCK	3	YES, NO	YES
	TIMEOUT	3	0-255, NO	10
	BYPASS	3	YES, NO	NO
SELECT (Only if SELECT = YES in INPUT)	FLDPOS1 - FLDPOS4	4		
	LENGTH1 - LENGTH4	3		
	FLDTYP1 - FLDTYP4	1	B,C,D,L,P,Z	C
	TSTREL1 - TSTREL4	2	EQ,NE,GT,GE, LT,LE	
	VALUE1 - VALUE4	18	Either position in bytes or literal data inside quotes	
	CONECT1 - CONECT4	3	AND, OR, ∅	∅

Prname	Keyword	Length	Options	Defaults
SELECT2 (Only if CONECT4 ≠ ¢)	FLDPOS5 - FLDPOS8	4		
	LENGTH5 - LENGTH8	3		
	FLDTYP5 - FLDTYP8	1	B,C,D,L,P,Z	C
	TSTREL5 - TSTREL8	2	EQ,NE,GT,GE, LT,LE	
	VALUE5 - VALUE8	18	Either position in bytes or literal data inside quotes	
	CONECT5 - CONECT8	3	AND, OR, ¢	¢
SELECT3 (Only if CONECT8 ≠ ¢)	FLDPOS9 - FLDPOS12	4		
	LENGTH9 - LENGTH12	3		
	FLDTYP9 - FLDTYP12	1	B,C,D,L,P,Z	C
	TSTREL9 - TSTREL12	2	EQ,NE,GT,GE, LT,LE	
	VALUE9 - VALUE12	18	Either position in bytes or literal data inside quotes	
	CONECT9 - CONECT12	3	AND, OR, ¢	¢
SELECT4 (Only if CONECT12 ≠ ¢)	FLDPOS13 - FLDPOS16	4		
	LENGTH13 - LENGTH16	3		
	FLDTYP13 - FLDTYP16	1	B,C,D,L,P,Z	C
	TSTREL13 - TSTREL16	2	EQ,NE,GT,GE, LT,LE	
	VALUE13 - VALUE16	18	Either position in bytes or literal data inside quotes	
	CONECT13 - CONECT16	3	AND, OR, ¢	¢

Prname	Keyword	Length	Options	Defaults
SELECT5	FLDPOS17 -			
(Only if	FLDPOS20	4		
CONECT16	LENGTH17 -			
≠ ∅)	LENGTH20	3		
	FLDTYP17 -			
	FLDTYP20	1	B,C,D,L,P,Z	C
	TSTREL17 -			
	TSTREL20	2	EQ,NE,GT,GE, LT,LE	
	VALUE17 -			
	VALUE20	18	Either position in bytes or literal data inside quotes	
	CONECT17 -			
	CONECT20	3	AND, OR, ∅	∅
SELECT6	FLDPOS21 -			
(Only if	FLDPOS24	4		
CONECT20	LENGTH21 -			
≠ ∅)	LENGTH24	3		
	FLDTYP21 -			
	FLDTYP24	1	B,C,D,L,P,Z	C
	TSTREL21 -			
	TSTREL24	2	EQ,NE,GT,GE, LT,LE	
	VALUE21 -			
	VALUE24	18	Either position in bytes or literal data inside quotes	
	CONECT21 -			
	CONECT24	3	AND, OR, ∅	∅
SELECT7	FLDPOS25 -			
(Only if	FLDPOS28	4		
CONECT24	LENGTH25 -			
≠ ∅)	LENGTH28	3		
	FLDTYP25 -			
	FLDTYP28	1	B,C,D,L,P,Z	C
	TSTREL25 -			
	TSTREL28	2	EQ,NE,GT,GE, LT,LE	
	VALUE25 -			
	VALUE28	18	Either position in bytes or literal data inside quotes	
	CONECT25 -			
	CONECT28	3	AND, OR, ∅	∅

Prname	Keyword	Length	Options	Defaults
SELECT8 (Only if CONECT28 ≠ ¢)	FLDPOS29 - FLDPOS32	4		
	LENGTH29 - LENGTH32	3		
	FLDTYP29 - FLDTYP32	1	B,C,D,L,P,Z	C
	TSTREL29 - TSTREL32	2	EQ,NE,GT,GE, LT,LE	
	VALUE29 - VALUE32	18	Either position in bytes or literal data inside quotes	
	CONECT29 - CONECT31	3	AND, OR, ¢	¢
KEYS	KEYS	1	1-8	1
	POST1 - POST8	4		
	LENGTH1 - LENGTH8	3	1-256	
	TYPE1 - TYPE8	1	B,C,D,F,L,P,Z	C
	ORDER1 - ORDER8	1	A,D	A
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	REPLACE	3	YES, NO	NO
	COMPRESS	3	YES, NO	YES for variable-length input, NO for fixed-length input
	DEVICE	4	DISK, TAPE	DISK
	FILESEQ	4		1
WORKFILE*	(see WORK under DEFAULT GETPARMS)			
KEYFILE*	(see WORK under DEFAULT GETPARMS)			

* Default GETPARM

Prname	Keyword	Length	Options	Defaults
OPTIONS	FUNCTION	5	SORT, MERGE	SORT
	MEMORY	3		128
	ADDROUT	3	YES, NO	NO
	KEYOUT	3	YES, NO	NO
	STABLE	3	YES, NO	NO
	EXTERNAL	3	YES, NO	NO
	REFORMAT	3	YES, NO	NO
	SHARED	3	YES, NO	NO
INTABLE	FILE	8		
	LIBRARY	8		TRNTABLE
	VOLUME	6		User's INVOL
INPUT (Up to 20 input files if MOREFILE = YES)	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	SELECT	3	YES, NO	NO
	MOREFILE	3	YES, NO	NO
SELECT (Only if SELECT = YES in INPUT)	FLDPOS1 - FLDPOS4	4		
	LENGTH1 - LENGTH4	3		
	FLDTYP1 - FLDTYP4	1	B,C,D,L,P,Z	C
	TSTREL1 - TSTREL4	2	EQ,NE,GT,GE, LT,LE	
	VALUE1 - VALUE4	18	Either position in bytes or literal data inside quotes	
	CONECT1 - CONECT4	3	AND, OR, ∅	∅
SELECT2 (Only if CONECT4 ≠ ∅)	FLDPOS5 - FLDPOS8	4		
	LENGTH5 - LENGTH8	3		
	FLDTYP5 - FLDTYP8	1	B,C,D,L,P,Z	C
	TSTREL5 - TSTREL8	2	EQ,NE,GT,GE, LT,LE	
	VALUE5 - VALUE8	18	Either position in bytes or literal data inside quotes	
	CONECT5 - CONECT8	3	AND, OR, ∅	∅

Prname	Keyword	Length	Options	Defaults
SELECT3	FLDPOS9 -			
(Only if	FLDPOS12	4		
CONECT8	LENGTH9 -			
≠ ∅)	LENGTH12	3		
	FLDTYP9 -			
	FLDTYP12	1	B,C,D,L,P,Z	C
	TSTREL9 -			
	TSTREL12	2	EQ,NE,GT,GE, LT,LE	
	VALUE9 -			
	VALUE12	18	Either position in bytes or literal data inside quotes	
	CONECT9 -			
	CONECT12	3	AND, OR, ∅	∅
SELECT4	FLDPOS13 -			
(Only if	FLDPOS16	4		
CONECT12	LENGTH13 -			
≠ ∅)	LENGTH16	3		
	FLDTYP13 -			
	FLDTYP16	1	B,C,D,L,P,Z	C
	TSTREL13 -			
	TSTREL16	2	EQ,NE,GT,GE, LT,LE	
	VALUE13 -			
	VALUE16	18	Either position in bytes or literal data inside quotes	
	CONECT13 -			
	CONECT16	3	AND, OR, ∅	∅
SELECT5	FLDPOS17 -			
(Only if	FLDPOS20	4		
CONECT16	LENGTH17 -			
≠ ∅)	LENGTH20	3		
	FLDTYP17 -			
	FLDTYP20	1	B,C,D,L,P,Z	C
	TSTREL17 -			
	TSTREL20	2	EQ,NE,GT,GE, LT,LE	
	VALUE17 -			
	VALUE20	18	Either position in bytes or literal data inside quotes	
	CONECT17 -			
	CONECT20	3	AND, OR, ∅	∅

Prname	Keyword	Length	Options	Defaults
SELECT6	FLDPOS21 -			
(Only if	FLDPOS24	4		
CONECT20	LENGTH21 -			
≠ ∅)	LENGTH24	3		
	FLDTYP21 -			
	FLDTYP24	1	B,C,D,L,P,Z	C
	TSTREL21 -			
	TSTREL24	2	EQ,NE,GT,GE, LT,LE	
	VALUE21 -			
	VALUE24	18	Either position in bytes or literal data inside quotes	
	CONECT21 -			
	CONECT24	3	AND, OR, ∅	∅
SELECT7	FLDPOS25 -			
(Only if	FLDPOS28	4		
CONECT24	LENGTH25 -			
≠ ∅)	LENGTH28	3		
	FLDTYP25 -			
	FLDTYP28	1	B,C,D,L,P,Z	C
	TSTREL25 -			
	TSTREL28	2	EQ,NE,GT,GE, LT,LE	
	VALUE25 -			
	VALUE28	18	Either position in bytes or literal data inside quotes	
	CONECT25 -			
	CONECT28	3	AND, OR, ∅	∅
SELECT8	FLDPOS29 -			
(Only if	FLDPOS32	4		
CONECT28	LENGTH29 -			
≠ ∅)	LENGTH32	3		
	FLDTYP29 -			
	FLDTYP32	1	B,C,D,L,P,Z	C
	TSTREL29 -			
	TSTREL32	2	EQ,NE,GT,GE, LT,LE	
	VALUE29 -			
	VALUE32	18	Either position in bytes or literal data inside quotes	
	CONECT29 -			
	CONECT31	3	AND, OR, ∅	∅

Prname	Keyword	Length	Options	Defaults
KEYS	KEYS	1	1-8	1
	POST1 -			
	POST8	4		
	LENGTH1 -			
	LENGTH8	3	1-256	
	TYPE1 -			
	TYPE8	1	B,C,D,F,L,P,Z	C
	ORDER1 -			
	ORDER8	1	A,D	A
REFORMAT	LENGTH	4	1-2048	
	PAD	2	Any valid hexadecimal code	20
	INPOS1 -			
	INPOS12	4	1-2048	
	LENGTH1 -			
	LENGTH12	3	1-999	
	OUTPOS1 -			
	OUTPOS12	4	1-2048	
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	REPLACE	3	YES, NO	NO
	COMPRESS	3	YES, NO	YES for variable-length input, NO for fixed-length input

WORKFILE* (see WORK under DEFAULT GETPARMS)

KEYFILE* (see WORK under DEFAULT GETPARMS)

* Default GETPARM

Prname	Keyword	Length	Options	Defaults
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	DEVICE	8	DISK, TAPE	TAPE
	CONVERT	2	N,E,A,BA,AB,HA,AH	N
	MULTYPE	1	Y,N	N
	FSEQ	4	1-9999	1
	LABEL	2	AL, IL, NL	AL
	HEADER2	3	YES, NO	YES
MOUNT	DEVICE	3		
	USAGE	2	SH, EX	SH
	PF Keys*		∅ = Continue 1 = Respecify input	
TAPEFILE	RECFORM	1	F,V,I,U	Depends on label type
	RECSIZE	8	4-2048	Depends on input file
	BLOCKED	1	Y,N	Y
	BLKSIZE	8	18-3760	RECSIZE times 100
	COMPRESS	1	Y,N	N
TAPE7	DENSITY	3	556, 800	800
	PARITY	4	EVEN, ODD	EVEN
	IBM1401	3	YES, NO (Appears only if DEVICE = TAPE)	NO
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	DEVICE	8	TAPE, DISK	TAPE if input DEVICE = DISK; DISK if input DEVICE = TAPE
	FSEQ	4	1-9999	1
	LABEL	2	AL, IL, NL	AL

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
DISKFILE	NRECS	7	Specified only if input file is on tape	
	FILEORG	1	C, I, X, P	C
	RECFORM	1	F, V	Input file's format
	COMPRESS	1	Y,N	
	FILECLAS	1		
	RETAIN	3	YES, NO	
	KEYLEN	3	1-255	
	KEYPOS	5	1-2048	
	IPACK	3	1-100	100
	DPACK	3	1-100	100
TYPES (for multiple record type convers-ions only)	CHAR1-CHAR3	2		
	COLUMN1-COLUMN3	4		
	SWITCH1-SWITCH3	3	YES, NO	
	CHARSET	6	INPUT, OUTPUT	INPUT
OPTIONS (for multiple record type convers-ions only)	POSTO1-POSTO8	2		
	LENGTH	3		
	TYPE01-TYPE08	1		
	SWITCH01-SWITCH08	1	B,C,P,Z A-Z	
PAD (only when coverting a variable-length file to a fixed-length file)	PAD	1	⌀,0	⌀
EOJ	PF Keys*		1 = Rerun TAPECOPY 16 = End processing	

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
INPUT	VOLUME	6		
MOUNT	DEVICE	3		
TAPE (for 9-track tape)	LABEL	2	AL, IL, NL	AL
	OWNER	14		
	DENSITY	4	800,1600	1600
TAPE (for 7-track tape)	PARITY	4	EVEN, ODD	EVEN
	DENSITY	3	556,800	800
EOJ	PF Keys*		∅ = End processing 1 = Rerun TAPEINIT 16 = End processing	

* The keyword is not required.

PRNAME	KEYWORD	LENGTH	OPTIONS	DEFAULT
INPUT	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	TYPES	3	YES, NO	NO
	CODE	6	EBCDIC, ASCII, TABLE, NONE	EBCDIC
INTABLE	FILE	8		∅
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
TABLE	HEX00#0F			
	HEXF0#FF	35	Hex chars with embedded spaces at positions 9, 20, and 29	Defaults are taken from the Default Table or INTABLE.
EXPLAIN	PF Keys*		∅ = Continue	
OUTTABLE	FILE	8		
	LIBRARY	8		Defaults taken
	VOLUME	6		from INTABLE
TYPES	CHAR1-			
	CHAR3	2	Any character	∅
	COLUMN1-			
	COLUMN3	4	Any numeric	∅
	SWITCH1-			
	SWITCH3	3	YES, NO	YES
	CHARSET	6	INPUT, OUTPUT	INPUT
OPTIONS	POST01-			
	POST24	4		
	LENGTH01-			
	LENGTH24	4		
	TYPE01-			
	TYPE24	1	B,C,P,Z	
	SWITCH01-			
	SWITCH24	1	∅,I,D	∅
INDEXED	KEYLEN	3		Same as input
	KEYPOS	3		file

* The keyword is not required.

Prname	Keyword	Length	Options	Defaults
OUTPUT	FILE	8		
	LIBRARY	8		User's OUTLIB
	VOLUME	6		User's OUTVOL
	RECORDS	7		Size of input file
	RETAIN	3	∅	
	RELEASE	3	YES, NO	NO
	FILECLAS	1	A-Z, #, ∅, @, $	∅
	DEVICE	11	DISK	
EOJ	PF Keys*		1 = Restart TRANSL 16 = End processing	
WARNING	PF Keys*		∅ = Accept 16 = Restart field definition	
ERROR01	PF Keys*		1 = Restart field definition 16 = End processing	
ERROR02	PF Keys*		1 = Start record type again 16 = End processing	
ERROR03	PF Keys*		1 = Restart field definition 16 = End processing	
ERROR04	PF Keys*		1 = Start record type again 16 = End processing	
ERROR05	PF Keys*		∅ = Acknowledge	
ERROR06	PF Keys*		∅ = Acknowledge	

* The keyword is not required.

VERIFY

Prname	Keyword	Length	Options	Defaults
OPTIONS	RANGE	7	FILE, LIBRARY, VOLUME	FILE
	FILE	8		
	LIBRARY	8		User's INLIB
	VOLUME	6		User's INVOL
	VERIFY	7	ALL, PRIMARY	ALL
	DISPLAY	3	YES, NO	YES
MOUNT	DEVICE	3		
	PF Keys*		⍉ = Start mount 1 = Respecify mount	
NOSPACE	PF Keys*		⍉ = Continue without printed report 16 = End processing	
ENDOFJOB	PF Keys*		⍉ = End processing 1 = Rerun VERIFY 16 = End processing	
ERROR1	PF Keys*		⍉ = Continue	
ERROR2	PF Keys*		⍉ = Continue 16 = End file verification	
ERROR3	PF Keys*		⍉ = Continue 1 = Display error details 16 = End file verification	
SUMMARY	PF Keys*		⍉ = Continue	

PRINT (PRNAME for Summary Report)

ERPRT (PRNAME for Error Report)

* The keyword is not required.

A.5 DEFAULT GETPARMS

Default GETPARMS are screens that do not appear at runtime because they have been previously supplied through the Set Usage Constants function (PF2) of the Command Processor. To change default values, include the appropriate default prname, keywords, and values from the following list in a procedure.

Prname	Keyword	Length	Options	Defaults
PRINT	FILE	8		
	LIBRARY	8		# + User I.D. + PRT
	VOLUME	6		User's SPOOLVOL
	RECORDS	7		
	RETAIN	3		
	RELEASE	3	YES,NO	YES
	FILECLAS	1	A-Z,#,∅	#
	DEVICE	11	DISK, PRINTER	DISK
	FORM#	3	000-255	User's FORM#
	PRTCLASS	1	A-Z	User's PRTCLASS
	COPIES	5	1-32767	1
WORK	FILE	8		
	LIBRARY	8		# + User I.D. + WORK
	VOLUME	6		User's WORKVOL
	RETAIN	3		
	RELEASE	3	YES,NO	YES
	FILECLAS	1	A-Z,#,∅	#
	RECORDS	7		
	DEVICE	11	DISK, NONE	DISK

Within a procedure, each ENTER or DISPLAY statement supplies parameters for a single GETPARM request. To associate a given ENTER or DISPLAY statement with a specific GETPARM request, you must specify the prname of the request in the statement. (GETPARM requests issued by a user program can, of course, be assigned any prname desired by the programmer. For user-defined GETPARM requests, modifiable fields and the keywords identifying them are specified by the person issuing the GETPARM.)

When a procedure supplies parameters, keywords in the ENTER or DISPLAY statement associate the specified values with the fields to which they are to be assigned in the GETPARM request. Values associated with keywords in the procedure statement are passed to the corresponding keyword-identified fields in the GETPARM request. (Be sure to spell the keywords correctly in the procedure statement.) If the procedure does not assign new values to fields, they retain their default values.

An example of a procedure that runs the COPYWP utility follows. If you compare this procedure to the list of prnames and keywords for COPYWP, you can see that certain default keyword values are used, since those keywords are not listed (e.g., VOLUME for prname INPUT). Also, PF key options are selected for certain other prnames (e.g., for prname FUNCTION, PF1, Copy a Single Document, was selected).

```
PROC RUN COPYWP
RUN COPYWP
ENTER FUNCTION 1
ENTER INPUT DOCUMENT=0001A
ENTER OUTPUT DOCUMENT=0002A, VOLUME=SYSTEM
ENTER FUNCTION 16
RETURN
```

Refer to the VS Procedure Language Reference for more information on the Procedure language and the use of GETPARM requests. GETPARM requests for the VS File Management Utilities not documented in this manual are listed in the VS File Management Utilities Reference.

A.4 SYSTEM UTILITY GETPARM REQUESTS

Following is a list of the prnames, keywords, options, and default values used by VS System Utility GETPARM requests. The GETPARMs are listed by utility, and the utilities are organized alphabetically.

The GETPARM summaries also contain PF key options (if any) available for a given GETPARM request. The VS Procedure language does not define a screen's PF key option through a keyword (refer to Section A.2 and the VS Procedure Language Reference) but simply lists the PF key value following the prname. For example, the statement ENTER FUNCTION 1 is equivalent to pressing PF1 on a screen identified by the FUNCTION prname. However, the ENTER key cannot be explicitly specified, and is assumed if no other PF key value is specified in the statement. Listings of PF key options on the following pages refer to the ENTER key as ∅, meaning that no PF key value is specified in the statement.

APPENDIX B
COPYWP DATA TYPE CONVERSION FORMAT

B.1 INTRODUCTION

The format specification of the two record types produced through the TYPE=DATA option of the Convert Document to VS File function is described as follows.

B.2 FORMAT LINE RECORD DESCRIPTION

The Format Line record describes the format of the text records that follow it. Each occurrence of a Format Line record signals the occurrence of a new format line or a new page in the original document. The record length is entirely dependent upon the number of TAB (►) characters in the corresponding format line of the original document.

Field Number	Length*	Data Type	Field Description
1	1	Character	F
2	1	Character	0 - New format line 1 - New page
3	2	Binary	Current record length
4	1	Character	Line spacing Q - 1/4 Spacing H - 1/2 Spacing 1 - Single spacing W - 1 1/2 spacing 2 - Double spacing 3 - Triple spacing
5	1	Reserved	
6	2	Binary	Maximum length of data records (until next format record or end of file)
7	2	Binary	First tab position or 0 (no tabs)

* Length is specified in bytes

Field Number	Length*	Data Type	Field Description
8	2	Binary	Length from the first tab position to the second tab position, or if no additional tabs are in the format line, the length from the first tab position to the RETURN (◄) (the end of the format line). If no tabs are in the format line, the length of the entire format line. Repeat for each additional tab position.
5+2n	2	Binary	nth Tab position
6+2n	2	Binary	Length from the nth tab position to the (n+1)th tab position, or if no additional tabs are in the format line, the length from the nth tab position to the RETURN (◄) (the end of the format line).

B.3 TEXT RECORD DESCRIPTION

The Text record provides the actual text of the document, in screen format, without any formatting codes other than RETURN (◄) and NOTE (!!).

Field Number	Length*	Data Type	Field Description
1	1	Character	D
2	1	Reserved	
3	2	Binary	Current record length (n+4 Bytes).
4	n	Character	The document text formatted as it would approximately appear printed, unjustified, with a left margin of 0, delimited by a RETURN (◄) character (X'03'), if one is on that line.

*** Length is specified in bytes**

APPENDIX C
IBM DATA EXCHANGE FORMAT DISKETTES

IBM Data Exchange format diskettes are structured differently than VS diskettes. IBM diskettes are always soft-sectored and are available in the following three forms:

- Single-sided, single-density (Diskette 1)
- Double-sided, single-density (Diskette 2)
- Double-sided, double-density (Diskette 2D)

Cylinder 0 of all IBM diskettes is reserved as the Index Cylinder. Track 0 of the Index Cylinder is referred to as the Index Track and describes the format and contents of the diskette. The Index Track contains the following information:

- An error map indicating defective and alternate tracks and sectors.
- A volume label indicating the physical and logical structure of the diskette.
- Nineteen data set header labels. For double-sided diskettes, track 1 of the Index Cylinder contains additional data set header labels.

Double-density diskettes have sector sizes of 128 and 256 bytes. Single-density diskettes have sector sizes of 128 bytes. Track 0 of the Index Cylinder (the Index Track) is always recorded with 128 bytes for each sector, regardless of whether the diskette is single- or double-sided. Track 1 of the Index Cylinder is recorded with 128 bytes per sector for single-density diskettes, and with 256 bytes per sector for double-density diskettes.

All IBM diskette files are consecutively organized with fixed-length records. Records must be unblocked, with logical records contained entirely within a single physical sector. The file's record length can vary from 1 to 128 on single-density diskettes, and from 1 to 256 on double-density diskettes.

Each file is stored in a single, consecutive extent. To locate the file, the data set header label for the file in the Index track contains Beginning-of-Extent (BOE) and End-of-Extent (EOE) extent limits, and an End-of-Data (EOD) pointer. The End-of-Data pointer always points to the next physical sector to be used for record storage.

Empty files are denoted by a header label in which the EOD and BOE pointers point to the same physical sector. Multivolume files are allowed, with sequence numbers optionally contained in the data set header labels. The EBCDIC character set is normally used for all character storage.

The maximum number of files that can be written to an IBM Data Exchange format diskette depends on the type of diskette. While the actual number of files that can be written to any given diskette depends on the size of the files, the following list indicates the maximum number of files that can be written to each type of diskette:

Diskette Type	Maximum Number of Files
Single-sided, single-density	19
Double-sided, single-density	45
Double-sided, double-density	71

APPENDIX D
SYSTEM UTILITY RETURN CODES

All VS programs and some Procedure statements generate return codes during execution. These return codes indicate the reasons for a program failure or warn of conditions that could cause program failure. A successful program execution generates a return code of 0, which is transparent to the user. The return codes are listed as follows, according to program.

Program Name	Return Code	Meaning
COPY	100	No copy took place
	104	Some program privileges lost
	108	Disk error; run LISTVTOC
	112	No space in the output volume
	116	I/O error on output
	120	Boundary violation
	124	Key out of sequence
	128	Duplicate key
	132	The primary extent was exceeded
	136	DMS error
	140	Duplicate file name encountered
	144	Copy completed in IO mode after primary extent was exceeded
	148	O/P VTOC Full
	152	Input RAM Error
DISKINIT	4	Termination by user
	8	Insufficient space in I/O buffer
	12	Mount operation unsuccessful
	16	Bad disk sector encountered
	20	Bad MOUNT SVC return code
	24	Bad FREEBUF return code
	28	Bad XIO return code
	32	Disk I/O error
EZFORMAT	1-4	Warning; program will probably run
	5-7	Severe Error; program will not run correctly
	8-16	Fatal Error; object code not generated
LISTVTOC	4	Extent lost on disk
	8	Error on file label
	12	Error in library directory
	16	Error in VTOC

Program Name	Return Code	Meaning
SORT and SORTINT	4	No records to be sorted; input records did not meet selection criteria, or you specified an empty input file
	8	Insufficient space in stack or I/O buffer
	12	Record size is more than 2024 bytes long
	16	Invalid sort key
	20	Unexpected program check
	24	Input records out of order in file to be merged; program cannot proceed
	28	Input Record Count Problem
All compilers	1-4	Warning
	5-8	Severe error (program will not execute correctly)
	9-16	Fatal error (program will not execute)

Summary of Changes
for the Fourth Edition of the VS System Utilities Reference

Change	Description/New Feature	Affected Chapters
Manual	This manual has been completely revised, and the person has been changed from third to second. The utilities are no longer divided into sections. They are presented alphabetically.	
Introduction	The Introduction has been revised to reflect the format changes.	Chapter 1
COPY	Various editorial changes and file definition changes for DMS/TX files have been incorporated.	Chapter 2
COPYOIS	This new utility transfers OIS files to and from the VS and allows you to convert OIS BASIC source programs and data files to VS BASIC source and VS data file format.	Chapter 3
COPY2200	Editorial changes have been made to this chapter.	Chapter 4
COPYWP	The chapter has been slightly restructured, and a few minor technical changes have been made.	Chapter 5
DISKINIT	Editorial changes have been made, and a new feature has been added to allocate a permanent dump file at disk initialization.	Chapter 6
DISPLAY	Editorial changes were made, and the overview diagram was redrawn.	Chapter 7
EZFORMAT	This utility has been enhanced to support modifiable source and object names for RPG screen definitions. The Data Entry option of EZFORMAT, documented in the VS File Management Utilities Reference, has also been enhanced to create RPG data entry programs. Additional enhancements have been made to allow the creation of both Assembly language and RPG II menu programs.	Chapter 8

Summary of Changes (continued)

Change	Description/New Feature	Affected Chapters
FLOPYDUP	Editorial changes were made, and the overview diagram was redrawn.	Chapter 9
FORMCNTL	Editorial changes and a new list of printers supported by this utility have been added.	Chapter 10
IBMCOPY	Enhancements to this utility are: the ability to display or print the diskette directory; the ability to copy the contents of several IBM diskettes into a single VS File; to append data to an existing file; to return, as a default value, the device number of a diskette drive; and, when initializing an IBM diskette, to set the system code field in the diskette's volume label to WANGVS rather than blanks.	Chapter 11
IOELOG	New utility.	Chapter 12
LISTVTOC	Various editorial changes.	Chapter 13
SORT	Various editorial changes. Also, the ADDROUT output file example was corrected.	Chapter 14
TAPECOPY	Various editorial changes and a new overview diagram.	Chapter 15
TAPEINIT	Various editorial changes. Also, technical changes were made to reflect the newly introduced tape and tape drives.	Chapter 16
TRANSL	Various editorial changes.	Chapter 17
VERIFY	Various editorial changes.	Chapter 18
APPENDIX A	The GETPARM listings have been updated.	Appendix A
INDEX	The index has been updated and rewritten.	Index

INDEX

E

F

H

I

L

M

WANG

Customer Comment Form

Publication Number 800-1303-04.01

Title VS SYSTEM UTILITIES REFERENCE ADDENDUM

Help Us Help You . . .

We've worked hard to make this document useful, readable, and technically accurate. Did we succeed? Only you can tell us! Your comments and suggestions will help us improve our technical communications. Please take a few minutes to let us know how you feel.

How did you receive this publication?

- ☐ Support or Sales Rep
- ☐ Wang Supplies Division
- ☐ From another user
- ☐ Enclosed with equipment
- ☐ Don't know
- ☐ Other _____

How did you use this Publication?

- ☐ Introduction to the subject
- ☐ Classroom text (student)
- ☐ Classroom text (teacher)
- ☐ Self-study text
- ☐ Aid to advanced knowledge
- ☐ Guide to operating instructions
- ☐ As a reference manual
- ☐ Other _____

Please rate the quality of this publication in each of the following areas.

	EXCELLENT	GOOD	FAIR	POOR	VERY POOR
Technical Accuracy — Does the system work the way the manual says it does?	☐	☐	☐	☐	☐
Readability — Is the manual easy to read and understand?	☐	☐	☐	☐	☐
Clarity — Are the instructions easy to follow?	☐	☐	☐	☐	☐
Examples — Were they helpful, realistic? Were there enough of them?	☐	☐	☐	☐	☐
Organization — Was it logical? Was it easy to find what you needed to know?	☐	☐	☐	☐	☐
Illustrations — Were they clear and useful?	☐	☐	☐	☐	☐
Physical Attractiveness — What did you think of the printing, binding, etc?	☐	☐	☐	☐	☐

Were there any terms or concepts that were not defined properly? ☐ Y ☐ N If so, what were they? _____

After reading this document do you feel that you will be able to operate the equipment/software? ☐ Yes ☐ No ☐ Yes, with practice

What errors or faults did you find in the manual? (Please include page numbers) _____

Do you have any other comments or suggestions? _____

Name _____ **Street** _____

Title _____ **City** _____

Dept/Mail Stop _____ **State/Country** _____

Company _____ **Zip Code** _____ **Telephone** _____

Thank you for your help.

Printed in U.S.A. 14-3140 7-83-5C

WANG

Fold

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 16 LOWELL, MA

POSTAGE WILL BE PAID BY ADDRESSEE

WANG LABORATORIES, INC.
TECHNICAL PUBLICATIONS
ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851

Fold

Cut along dotted line.

WANG

ONE INDUSTRIAL AVENUE
LOWELL, MASSACHUSETTS 01851
TEL. (617) 459-5000
TWX 710-343-6769, TELEX 94-7421

Printed in U.S.A.
800-1303-04.01
1-84-15M