

PERKIN-ELMER

**OS/32
SYSTEM SUPPORT UTILITIES**

Reference Manual

48-031 F00 R01

The information in this document is subject to change without notice and should not be construed as a commitment by the Perkin-Elmer Corporation. The Perkin-Elmer Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license, and it can be used or copied only in a manner permitted by that license. Any copy of the described software must include the Perkin-Elmer copyright notice. Title to and ownership of the described software and any copies thereof shall remain in The Perkin-Elmer Corporation.

The Perkin-Elmer Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by Perkin-Elmer.

The Perkin-Elmer Corporation, Data Systems Group, 2 Crescent Place, Oceanport, New Jersey 07757

© 1977, 1980, 1983 by The Perkin-Elmer Corporation

Printed in the United States of America

TABLE OF CONTENTS

PREFACE		v
CHAPTERS		
1	SYSTEM SUPPORT UTILITIES	
1.1	INTRODUCTION	1-1
2	OS/32 SPOOLER	
2.1	INTRODUCTION	2-1
2.2	INPUT SPOOLING	2-2
2.3	OUTPUT SPOOLING	2-3
2.4	OPERATING INSTRUCTIONS	2-4
2.4.1	Starting the Spooler	2-5
2.4.2	Examples of the Spooler START Command	2-7
2.4.3	.SPL Command	2-9
2.5	MEMORY REQUIREMENTS	2-15
2.6	SPOOLER MESSAGES	2-16
3	DISK BACKUP UTILITY	
3.1	GENERAL DESCRIPTION	3-1
3.2	DISK BACKUP UTILITY MAGNETIC TAPE FORMAT	3-3
3.3	DISK BACKUP UTILITY REQUIREMENTS	3-5
3.4	OPERATING PROCEDURES	3-5
3.4.1	Multiple Disk Backup	3-16
3.5	MESSAGES	3-17

CHAPTERS (Continued)

4	OS/32 ACCOUNTING REPORTING UTILITY	
4.1	INTRODUCTION	4-1
4.1.1	Data Collection	4-3
4.1.2	Starting the Data Collection For an OS Without MTM	4-3
4.2	COMPILING, ESTABLISHING, AND STARTING THE ACCOUNTING REPORTING UTILITY	4-4
4.2.1	LOAD Command	4-5
4.2.2	START Command	4-6
4.3	ACCOUNTING REPORTING UTILITY COMMANDS	4-8
4.3.1	GET Command	4-9
4.3.2	SELECT Command	4-11
4.3.3	OPTION Command	4-13
4.3.4	CHARGE Command	4-17
4.3.5	REPORT Command	4-20
4.3.6	SAVE Command	4-30
4.3.7	PAUSE Command	4-31
4.3.8	END Command	4-32
4.4	ACCOUNTING REPORTING UTILITY TASK TERMINATION CODES	4-32
4.5	ACCOUNTING REPORTING UTILITY MESSAGE SUMMARY	4-32
5	ERROR REPORTING UTILITY	
5.1	GENERAL DESCRIPTION	5-1
5.2	DESCRIBING MEMORY TO INTERPRET ERRORS	5-1
5.3	LOADING AND STARTING THE ERROR REPORTING UTILITY	5-6
5.4	SPECIFYING THE INPUT FILE	5-6
5.5	DEFINING MEMORY CONFIGURATION	5-7
5.6	SELECTING SPECIFIC DATA	5-13
5.7	GENERATING AN ERROR REPORT	5-15
5.7.1	List Reports	5-15
5.7.1.1	Bulk Device Input/Output(I/O) Error Report	5-16
5.7.1.2	File Manager Detected Errors	5-17
5.7.1.3	System Detected Errors	5-20
5.7.1.4	System Milestones	5-22
5.7.1.5	Memory Errors	5-25

CHAPTERS (Continued)

5.7.2	Summary Reports	5-26	
5.7.2.1	Summary of System Errors and Milestones Report	5-27	
5.7.2.2	Summary of Device Errors	5-28	
5.7.2.3	Memory Errors Summary Report	5-30	
5.8	SAVING INFORMATION TO AN ARCHIVAL FILE	5-31	
5.9	PAUSING A TASK	5-32	
5.10	ENDING A TASK	5-32	
5.11	MESSAGES	5-32	
6	DISK DUMP UTILITY		
6.1	GENERAL DESCRIPTION	6-1	
6.2	DISK DUMP UTILITY REQUIREMENTS	6-1	
6.3	PERFORMANCE ESTIMATES	6-2	
6.4	OPERATING PROCEDURES	6-2	
7	DUMP PRINT UTILITY		
7.1	GENERAL DESCRIPTION	7-1	
7.2	USING A TEMPORARY OR PERMANENT FILE	7-2	
7.3	EXECUTING THE DUMP PRINT UTILITY	7-2	
7.4	DUMP PRINT EXAMPLE	7-5	
7.5	CORRUPT SYSTEM POINTER TABLE (SPT)	7-6	
7.6	DUMP PRINT STRUCTURE DISPLAY	7-6	
7.7	DUMP PRINT MESSAGES	7-7	
APPENDIXES			
A	DISK INITIALIZER UTILITY		
A.1	GENERAL DESCRIPTION	A-1	
A.2	DISK INITIALIZATION REQUIREMENTS	A-2	
A.3	OPERATING PROCEDURES	A-4	

APPENDIXES (Continued)

B	DISK INTEGRITY CHECK UTILITY	
	B.1 GENERAL DESCRIPTION	B-1
	B.2 DISK INTEGRITY CHECK UTILITY REQUIREMENTS	B-2
	B.3 OPERATING PROCEDURES	B-3
C	CONTENTS OF MAG TAPE PRODUCED BY A STAND-ALONE DUMP	C-1
D	CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP	D-1

FIGURES

4-1	The Accounting Reporting Utility Process	4-2
4-2	System Summary Report	4-23
4-3	Account Summary Report	4-25
4-4	Individual Account Summary Report	4-27
5-1	Memory Described as Blocks	5-2
5-2	Memory Described as Blocks and Banks	5-3
5-3	A Bank Broken into Modules	5-4
5-4	A Bank Broken into Storage Units	5-5

TABLES

5-1	STORAGE MODULE INFORMATION	5-3
5-2	MODULE TYPES AND SIZES	5-11
6-1	TIMINGS FOR BINARY DUMPS	6-2
A-1	INFORMATION REFERENCE TABLE FOR DISK DEVICES	A-3

INDEX

Ind-1

PREFACE

This manual contains descriptions of and procedures for using all Perkin-Elmer disk utilities and is intended for system operators, software maintenance personnel, and users of stand-alone systems. This version has been reissued to reflect changes in content and structure.

Chapter 1 contains a general description of the utilities. Chapter 2 describes the OS/32 Spooler, and Chapter 3 describes the Disk Backup Utility. The OS/32 Accounting Reporting Utility and the Error Reporting Utility are covered in Chapters 4 and 5, respectively. The Disk Dump Utility is covered in Chapter 6. The Dump Print Utility is presented in Chapter 7. Discussions of the Disk Initializer Utility and the Disk Integrity Check Utility are covered in Appendixes A and B, respectively. Appendix C contains the contents of a magnetic tape produced by a stand-alone dump. Appendix D contains the display of the Dump Print Utility output from a Model 3200MPS System.

This manual documents enhancements to the OS/32 Spooler Utility and includes two new file types in Chapter 3. The Accounting Reporting Utility has added information about data collection. The Error Reporting Utility has additional report types which are only applicable to the Model 3200MPS System. The Dump Print Utility contains an altered sequence of user prompts along with additional material which is supported by the R07.1 software release and higher. A sample dump print display from a Model 3200MPS System has been included in Appendix D. In addition, a rearrangement of chapters has occurred to increase user convenience.

This manual is intended for use with the OS/32 R06.2 software release or higher. Additional material specifically related to the Model 3200MPS System has also been included. These Model 3200MPS System features are supported by the OS/32 R07.1 software release and higher. Throughout the text these features are identified as applicable only to the Model 3200MPS System.

For information on the contents of all Perkin-Elmer 32-bit manuals, see the 32-Bit Systems User Documentation Summary.

CHAPTER 1

SYSTEM SUPPORT UTILITIES

1.1 INTRODUCTION

The System Support Utilities include the following:

- OS/32 Spooler
- Disk Backup
- Accounting Reporting Utility
- Error Reporting
- Disk Dump
- Dump Print
- Disk Initializer
- Disk Integrity Check

A brief overview of the capabilities of the System Support Utilities is represented below.

The OS/32 Spooler allows input/output (I/O) to a slow device, such as a printer, to be placed on mass storage devices to await transmission. The Disk Backup Utility provides a fast way to save files to magnetic tape or disk and restore files to disk. The Accounting Reporting Utility generates reports or archival files from the accounting information recorded by the accounting data collection facility. These reports and files are generated through operator commands. The Error Reporting Utility produces reports from the error log information recorded by the hardware error logger and stored on the error log file by the operating system. The Disk Dump Utility dumps information from a disk to magnetic tape, restores a previously dumped disk volume from magnetic tape, verifies that the data is correctly restored, and displays information contained on a disk volume in a format useful for debugging system routines that manage direct access volumes. The Dump Print Utility interprets and prints to a list device the contents of the memory dump previously copied from memory to magnetic tape by the panic dump program. The Disk Initializer Utility initializes a previously formatted disk pack to be used with OS/32. The Disk Integrity Check Utility provides a means of recovering open disk files following an operating

system failure and restoring the integrity of data on disk volumes.

| The Disk Initializer Utility and the Disk Integrity Check Utility
| are discussed in Appendix A and Appendix B, respectively. These
| utilities will only be supported for OS/32 R06.2 software release
| and lower. Future releases of the System Support Utilities
| Reference Manual will have the discussion of these two utilities
| removed entirely. These utilities have been replaced by the
| Fastchek Utility which incorporates the capabilities of both.
| See the OS/32 Fastchek Reference Manual for a complete discussion
| of the Fastchek Utility.

CAUTION

THE UTILITIES CAN BE RUN ONLY WITH THE
OPERATING SYSTEM THAT EACH WAS
PACKAGED/RELEASED WITH.

CHAPTER 2 OS/32 SPOOLER

2.1 INTRODUCTION

The OS/32 package (R06.2 or higher), now comes with two spooler tasks:

- the OS/32 Spooler, and
- the SPL/32 Spooler.

OS/32 Spooler is Perkin-Elmer's first generation spooler and, prior to this release, was the only spooler available with OS/32. As such, the OS/32 Spooler provides basic input/output (I/O) spooling services as discussed in the balance of this chapter.

The SPL/32 Spooler offers a more extensive range of features and capabilities than the OS/32 Spooler. The SPL/32 Spooler is the latest spooling product offered with the OS/32 operating system. For a detailed discussion of the capabilities of SPL/32, see the SPL/32 Spooler Administration Reference Manual.

The system administrator determines which spooler will be used on a system by selecting the appropriate system generation (sysgen) statement. Only one spooler can be active on the system at any given time. The System Generation/32 (SYSGEN/32) Reference Manual presents detailed information regarding the appropriate procedures for sysgening either spooler.

NOTE

The manner in which pseudo devices are specified and used in the spooling environment is different for the two spoolers. Pseudo devices created for the OS/32 Spooler are incompatible with pseudo devices created for the SPL/32 Spooler. Do not attempt to mix the various pseudo device types.

2.2 INPUT SPOOLING

The input spooling feature provides facilities for copying a batch stream of cards such as source programs, operator commands, command substitution system (CSS) files, or other user data to disk files for subsequent processing. Each card deck can be spooled to a unique user-specified file for convenient access. One or more card readers can be designated as spool devices in the START command to a spooler task. All readers so defined are exclusively assigned to the Spooler.

The Spooler periodically monitors the status of the card readers by issuing supervisor call 1 (SVC 1) reads to them. If status X'A0' is returned (device unavailable), the Spooler goes into a time wait state. The length of this time wait can be specified as an option in the START command. The default is 30 seconds. Another read is issued at the conclusion of the interval.

A reader control statement specifies the disk file to which the input is being spooled. Data cards to be copied must be preceded by a control card.

Format:

```
/@INPUT fd ,D
/@SUBMIT fd ,D
```

Parameters:

fd	is the file descriptor of the disk file in the form voln:filename.ext/actno. An index file is allocated with name fd. If volume name is omitted, the file is allocated on the default spool volume.
D	specifies that if a file with the same name already exists, the file is deleted and reallocated.

The /@SUBMIT control statement is used only in a multi-terminal monitor (MTM) environment. If the file already exists and D is not specified, or if the control card is syntactically wrong, the Spooler logs a message to the console. Cards are read until another /@INPUT or /@SUBMIT statement is found. In this way, stacked input is processed even if an intervening deck contains an error. The ending control statement that halts input spooling is defined as follows:

```
/@      (columns 1 and 2)
```

Data is copied from the cards to the disk file until a /@, /@INPUT, or /@SUBMIT statement is read. In the last two cases, another copy operation is performed and a warning message is logged. If the card reader returns a status X'A0' before an ending control card is read, the Spooler assumes an incomplete input and retries the read until encountering more data. Other Spooler functions, such as printing and punching to or reading from other spool devices, continue. Input spooling continues until the last control card is read and the next I/O returns device unavailable. The time wait is then reissued. The resulting disk files can now be explicitly assigned and read by the user in order to access the spooled information.

2.3 OUTPUT SPOOLING

Output spooling provides facilities for more than one task to be simultaneously assigned to one or more output devices. The output spooling feature supports print and card punch devices. Data to be punched or printed is written to disk files from where it is copied by the Spooler to the available output devices on a task priority basis. The user interface for card punches is compatible with the interface for output spooling to the line printer. Spool files are uniquely identified within the operating system by a commercial at (@) symbol as the first character of the filename.

To spool output, the user assigns a logical unit (lu) to a pseudo device, defined at sysgen time. There is no limit to the number of tasks or logical units that can be assigned to a pseudo device. The operating system automatically intercepts all assignments to a pseudo device and allocates a file on the spool volume. Subsequent SVC 1 write calls cause data to be written to this file and not to the pseudo device. The Spooler supports both image and formatted output.

Form feeds can be issued after a banner page or at the end of each printed file.

NOTE

If the logical record length of a physical output device is less than 110 characters wide, Spooler will output the banner page in a narrow format of only 55 characters. However, the narrow format will contain all the information that would be contained in the full-width format.

When the user closes the lu assigned to a spool file, the filename, task name, task priority, and number of records in the file are placed on the spool queue. The spool queue is a file on the spool volume maintained by the Spooler. Files are printed on the basis of task priority. If the PURGE option was entered when the Spooler was started, the filenames of files that do not exist on the spool volume are deleted from the spool queue. The operator commands .SPL PRINT and .SPL PUNCH can be used to invoke the Spooler to output a user file. It is up to the user to ensure that sufficient disk space is available to accommodate output spooling. The user task (u-task) generating the spooled output is responsible for handling, within its own standard I/O recovery routines, an end of medium (EOM) status occurring while writing to spool files.

If the user desires multiple copies of a file, data can be written to any user defined disk file. Subsequently, a .SPL PRINT or .SPL PUNCH command can be issued to print multiple copies of the file (see section 2.4.3). Unless OPTION NOHEADER has been invoked, a header page with the following information precedes the printing of each spool file:

- Taskid or filename
- User account number
- Time of day
- Date

The account number is 0 for all tasks executing in the foreground and background environments. See the OS/32 Multi-Terminal Monitor (MTM) Reference Manual for information on account numbers. Output to card punch devices is preceded by a header card that helps to identify the file. The information is the same as that given on the header page for print devices. All fields not used in the display of filename, account number, etc., contain asterisks (*). To facilitate separating any files that have been punched on cards, a trailing card is output. This card contains punches in every row from columns 1 to 66.

2.4 OPERATING INSTRUCTIONS

The Spooler is provided in object format with the OS/32 package. The Spooler must be built as a non-privileged u-task and must be loaded with the task identifier (taskid) .SPL.

Example:

```
LOAD .SPL,SPOOLER.TSK
```

2.4.1 Starting the Spooler

The START command specifies to the Spooler task which devices are to be assigned exclusively for the spooling operation.

The format for the START command is as follows:

Format:

$$\begin{array}{l}
 \text{START} \left[\text{,pseudo dev} = \left\{ \begin{array}{c} \text{fd} \\ (fd_1, fd_2, \dots, fd_n) \end{array} \right\} \right] \\
 \left[\text{,CARD} = \left\{ \begin{array}{c} \text{fd} \\ (fd_1, fd_2, \dots, fd_n) \end{array} \right\} \right] \left[\text{,T=sss} \right] \left[\text{,M=n} \right] \\
 \left[\text{,OPTION} \left\{ \begin{array}{c} \text{fd} \\ (fd_1, fd_2, \dots, fd_n) \end{array} \right\} \left[\begin{array}{l} \left\{ \text{NOHEADER} \right\} \\ \left\{ \text{HEADER} \right\} \end{array} \right] \left[\begin{array}{l} \left\{ \text{PROTECT} \right\} \\ \left\{ \text{NOPROTECT} \right\} \end{array} \right] \right. \\
 \left. \left[\begin{array}{l} \left\{ \text{NOBFF} \right\} \\ \left\{ \text{BFF} \right\} \end{array} \right] \left[\begin{array}{l} \left\{ \text{NOTFF} \right\} \\ \left\{ \text{TFF} \right\} \end{array} \right] \right. \\
 \left. \left[\begin{array}{l} \left\{ \text{AUTOCONTINUE} \right\} \\ \left\{ \text{NOAUTOCONTINUE} \right\} \end{array} \right] \right] \\
 \left[\text{,QVOL} = \left\{ \text{vol_name:} \right\} \right] \\
 \left[\text{,PURGE} \right]
 \end{array}$$

NOTE

Whenever two or more OPTION subparameters (NOH, H, P, NOP, NOB, NOT, AUTO, BFF, TFF, and NOA) are used, they must be enclosed within parentheses as indicated in Examples 2 and 3 below. If one OPTION subparameter is used, as shown in Example 1, no parentheses are required.

Parameters:

pseudo dev= specifies the device name to which files will be output for spooling. The name of the physical device on which these files will actually be printed or punched is fd. A physical device can be assigned to more than one pseudo device. When multiple physical devices are associated with a single pseudo device, the device attributes for the first physical device specified apply to all of the remaining devices. It is advisable to only associate devices with like device attributes with a common pseudo device.

CARD= fd is a required keyword for input spooling and specifies input devices to be used for spooling.

T= sss specifies the number of seconds between retries on input spooling. The default is 30 seconds.

M= n specifies the number of message buffers to be used by the Spooler. The default is 4.

OPTION specifies the options in effect for the real device specified by fd. Specifying a pseudo device results in a device error.

NOHEADER specifies that no banner page is to precede spooled output to the specified fd. If this parameter is omitted, HEADER is the default.

HEADER specifies that a banner page is to precede spooled output to the specified fd. This parameter is the default.

PROTECT specifies that only those files that are either assigned to that device or are specified in the DEVICES=fd statement in the PRINT or PUNCH parameter of the .SPL command are output to the specified device. Files are not spooled to a device simply because it is available.

NOPROTECT makes the specified device available for all print/punch requests.

NOBFF specifies that no form feed will be sent at the end of a banner page. Text will begin immediately after the date line of the banner.

BFF	specifies that a form feed will be sent to the printer at the end of the banner page. This parameter is the default.
NOTFF	specifies that no form feed will be sent at the end of a printed file.
TFF	specifies that a form feed will be sent to the printer at the end of a printed file. This parameter is the default.
AUTOCONTINUE	causes OS/32 Spooler to periodically poll a device in the 'not ready' state to see if it is ready to continue. When the device is ready, the Spooler will automatically continue output to that device.
NOAUTOCONTINUE	causes OS/32 Spooler to cease output to a device in the 'not ready' state until the user enters a CONTINUE command for that device. This parameter is the default.
QVOL=	vol_name: allows a user to specify the volume on which the spool queue is to reside. If this parameter is omitted, OS/32 Spooler will establish the spool queue on the current spool volume. This parameter must be followed by a colon.
PURGE	causes filenames of files that cannot be found on the spool volume by OS/32 Spooler to be purged from the spool queue. These filenames are purged when they reach the top of the spool queue.

2.4.2 Examples of the Spooler START Command

Example 1:

```
START,PR=PR1:,PR2=(TTY1:,TTY2:,TTY3:),CARD=CR:
      ,T=60,M=2,OPTION TTY1:,NOH
```

In this case, PR1:, TTY1:, TTY2:, TTY3:, and CR: are actual device names as specified at sysgen time. PR and PR2 are pseudo devices defined at sysgen to be used for output spooling. If a user program assigns an lu to PR, output is spooled, eventually to be output on device PR1:. If a user program assigns an lu to PR2, output is spooled, eventually to be printed on either TTY1:, TTY2:, or TTY3:. Header output on device TTY1: is disabled since option NOH has been used. This option applies to the physical device TTY1: If a user program assigns an lu to PR1: directly (assuming it is available; i.e., not assigned by the Spooler), output is not spooled and is sent directly to device PR1:. Device CR: is used for input spooling.

Example 2:

```
| START,PR=(PR1:,PR2:),OPT PR2:,(NOH,TFF),PURGE
```

In this case, PR1: and PR2: are actual device names specified at sysgen time. PR is a pseudo device defined at sysgen to be used for output spooling. If a user program assigns an lu to PR, output is spooled to be generated either on PR1: or PR2:. Files spooled to PR2: are printed without a banner page. A form feed will be sent to the printer at the end of the printed file. The PURGE parameter tells OS/32 Spooler to delete any filenames for files not on the spool volume from the spool queue when they reach the top of the queue. Since both time and message buffer options are omitted, the following defaults are taken and no input spooling is requested:

- T = 30 seconds
- M = 4 message buffers

Example 3:

```
| START ,SPR1=PR1:, SPR2=PR2:, SPR3=(PR1:,PR3:), CARD=CR:  
| ,PNCH=CRDP:, OPTION (PR1:,PR2:,PR3:), (NOB, NOT, AUTO)
```

In this case, the pseudo devices are SPR1, SPR2, SPR3, CARD, and PNCH. The associated device names are PR1:, PR2:, PR3:, CR:, and CRDP:. SPR1 causes output to be spooled and output on PR1:; SPR2 causes output to be spooled and output on PR2:; SPR3 causes output to be spooled and output on either PR1: or PR3:; PNCH causes output to be spooled and output on CRDP; and CARD causes input to be spooled from device CR:. The option subparameters that are in effect for the physical devices PR1:, PR2:, and PR3: are NOB, NOT and AUTO. Therefore, text will be printed immediately after the banner page, no form feed will occur at the end of the printed file, and when a device is in a not ready state, the spooling operation pauses and then automatically continues when that device is ready.

Use of multiple pseudo device names enables the user to direct output to a separate device or a group of devices. In the following example, output is being directed from multiple pseudo devices to one physical device.

```
| ST,PR=PRT:, PR1=PRT:...
```

2.4.3 .SPL Command

The .SPL command controls spooling operations. Input and output spooling is supported by the card reader, card punch, and printer. The spool volume specified at sysgen can be changed by the VOLUME operator command. See the OS/32 Operator Reference Manual.

The fifteen Spooler operations available for use with the .SPL command are:

Format:

```

.SPL {
  BACKWARD fd,n
  FORWARD fd,n
  CANCEL fd
  CONTINUE fd
  DELETE fd
  REWIND fd
  SUSPEND fd
  INQUIRE [TASK=ttttttt] [,DEV=pdv:] [ACCOUNT=aaaaa] [,LIST=fd2]
  PRINT fd [,COPIES=n] [,DEVICE=fdx] [,DELETE] [,VFC]
  PUNCH fd [,COPIES=n] [,DEVICE=fdx] [,DELETE] [,VFC]
  OPTION { fd } [ { NOHEADER } ] [ { PROTECT } ]
           { (fd1,fd2,...,fdn) } [ { HEADER } ] [ { NOPROTECT } ]
  PURGE fd
  STATUS
  END
  TERMINATE
}

```

NOTE

Whenever any two of the option subparameters (H or NOH and P or NOP) are used, they must be enclosed within parentheses.

Parameters:

fd is the file descriptor of the device used for output spooling.

n is a decimal number indicating the number of records to skip backward or forward.

BACKWARD backspaces the spool output by the number of records specified by n. This parameter is invalid if the device is currently printing or punching.

FORWARD forward spaces the spool output file to be spooled out by the number of records specified by n. This parameter is invalid if the device is currently printing or punching.

CANCEL cancels the spool file currently being output to the device specified by fd. The current spool file is removed from the spool queue, and output spooling resumes with the next file on the queue.

CONTINUE continues I/O to the specified fd. This parameter should be entered after I/O is paused because an I/O error occurred or the SUSPEND parameter was entered. This parameter is invalid if the device is currently printing or punching.

DELETE deletes the specified fd as the output spooling device after the current file is completed.

REWIND restarts spooling a file, beginning with the header page. This parameter is invalid if output is currently active.

SUSPEND suspends or pauses I/O to an output spool device after the currently active spool file has terminated I/O. This parameter allows the operator to change ribbon or paper between tasks. I/O is continued by the CONTINUE parameter.

INQUIRE displays on the system console all filenames currently on the spool queue and the pseudo device assignments. If PRINT or PUNCH has been specified without a request for a certain pseudo device, PRINT or PUNCH replaces the pseudo device name.

TASK= tttttttt specifies the task for which the spool queue is to be searched. '*' and '-' can be used to indicate generic task identifiers. These special characters are used exactly the same way as in file selection with BACKUP. See Section 3.4 for an explanation of how to use the special characters '*' and '-' to reduce the number of repetitious entries.

DEV= pdv: specifies the pseudo device for which the spool queue is checked for files being spooled to or from.

ACCOUNT= aaaaa specifies a 1- to 5-digit number for which the spool queue is to be searched for spool files with the same account number.

LIST= fd₂ specifies the file or device to which the inquiry list is output. If this parameter is omitted, the inquiry list will go to the console. If fd₂ cannot be found, a file will be allocated.

PRINT specifies that the contents of the file specified by fd are to be printed. Files are placed on the spool queue through the PRINT parameter and are spooled with a priority of two.

COPIES= n is the number of copies to be printed or punched. If this parameter is not specified, the default is one copy.

DEVICE= fd_x specifies the pseudo device to which the file (fd) is to be output. If fd_x is not specified, the output is sent to any available line printer or card punch device assigned to the spooler. If PRINT or PUNCH is specified to a device that does not support the requested function, the file is output in a form supported by the specified device; a PRINT to a card punch device results in punched output; a PUNCH to a line printer results in printed output.

DELETE If used with the PRINT or PUNCH parameters, it deletes the user file after all copies have been printed or punched. Files that are automatically spooled by the system are deleted by default after they are printed or punched.

NOTE

Files with non-zero keys or files residing on a restricted disk without RW privileges for account 0 cannot be deleted by the spooler and remain on the disk.

VFC indicates that the specified file will be printed/punched using vertical forms control characters.

PUNCH specifies that the contents of the file specified by fd are to be punched. Files are placed on the spool queue through the PUNCH parameter and are spooled with a priority of two.

OPTION is specified in conjunction with the HEADER, NOHEADER and the PROTECT, NOPROTECT options.

fd is a file descriptor that refers to a real device. If a pseudo device is specified, a device error is displayed.

NOHEADER specifies that printing of the banner page is to be suppressed.

HEADER specifies that the banner page is to be printed.

PROTECT specifies that only those files that are either assigned to the specified device (fd₁) or specified in the DEVICE=fd statement in the PRINT or PUNCH parameter of the .SPL command can be output to the device. Files are not spooled to a device simply because it is available.

NOPROTECT enables a print or punch request to send data to any available pseudo device assigned to the Spooler.

PURGE removes a specified filename currently waiting to be spooled from the spool queue.

STATUS	displays to the system console the current status of each input and output device. The message identifies the type of input device, names the device, and gives its current status. For output devices, the message informs the user whether the HEADER, PROTECT, and AUTOCONTINUE options are enabled. If the output device is a printing device, the type of form feed option (BFF or TFF) which may be in effect is displayed.
END	allows the user to bring OS/32 Spooler to a gradual halt with all devices suspended after the current operation is completed. When all devices are suspended, OS/32 Spooler executes an SVC 3 code 0 instruction.
TERMINATE	causes OS/32 Spooler to terminate all operations immediately and execute an SVC 3 code 4 instruction. This command prevents damage to the spool queue if the program is cancelled.

Functional Details:

If the Spooler is not active when a spool file is ready for output, the operator is informed that a filename was not entered on the queue. The file subsequently can be output by issuing the PRINT or PUNCH parameters from the system console.

Messages:

ACTV-ERR

indicates that the device specified by a REWIND, CONTINUE, CANCEL, FORWARD, or BACKWARD parameter was still spooling.

ERROR: DEVICE IS ALREADY SUSPENDED

indicates that a SUSPEND command was issued for a device on which a SUSPEND was already pending. The additional SUSPEND command is rejected. The device will stop at the end of the current file.

ERROR: DEVICE IS IDLE

indicates that the device was not spooling a file when the REWIND, CONTINUE, CANCEL, FORWARD, or BACKWARD parameter was specified.

| ERROR: DEVICE IS QUIESCED. RESTART IS NOT ALLOWED

| indicates that an END command was issued and the user
| tried to restart a device that was stopped at the end of
| a file. The command is rejected.

| ERROR: DEVICE IS NOT A REAL DEVICE

| indicates that the device specified with the REWIND,
| CONTINUE, FORWARD, or BACKWARD parameter is not an output
| spool device.

| ERROR: dddd DEVICE IS NOT FOUND

| indicates that the device specified was not assigned by
| the Spooler.

fd-BAD LINE COUNT

indicates that the limits of the file being spooled on
device fd were exceeded following the FORWARD or BACKWARD
parameters.

FILE fd NOT ENTERED ONTO PRINT QUEUE

indicates that the Spooler was not active at the time a
filename was to be added to the spool queue. The
operator should start the Spooler and enter the filename
on the spool queue using the PRINT or PUNCH parameter.

FILE filename NOT ON QUEUE

indicates that the filename specified with the PURGE
parameter is not on the spool queue.

FILE filename SPOOLING

indicates that the filename specified with the PURGE
parameter is currently being spooled out. Determine the
device to which the file is being spooled, and use the
CANCEL parameter to stop I/O and remove the filename from
the spool queue.

FORM-ERR

indicates that an invalid command syntax was encountered.

INVALID FD=fd

indicates that the filename specified is not a file but a device, or the device specified is not a valid pseudo device.

INVALID SYNTAX fd

indicates that an invalid file descriptor was specified with the PRINT or PUNCH parameter.

INVALID PARAMETER

indicates that a parameter used with the .SPL command is invalid.

SPL-ERR

indicates that spooling was not specified at sysgen time and is not supported.

SPL-SVC7 ERR=xx,fd

indicates that the fd specified could not be assigned by the Spooler. The status returned by the SVC 7 call is xx.

WARNING: pseudo-dev ATTRIBUTES INCOMPATIBLE WITH device

indicates the attributes defined by the pseudo device are beyond the hardware restrictions of the real device.

2.5 MEMORY REQUIREMENTS

The Spooler memory requirements are approximately 12.75kb plus:

- 76 bytes for each message buffer,
- 150 bytes for each device specified in the START command, and
- 140 bytes for each I/O buffer.

The Spooler requires only one I/O buffer regardless of the number of spool devices. However, the Spooler operates more efficiently with multiple I/O buffers. For every additional 140 bytes of memory available in the segment into which the task is loaded, the Spooler creates another I/O buffer. If more than one device or more than four message buffers are specified and/or more than one I/O buffer is desired, the operator should load the Spooler into a larger segment.

Example:

```
LOAD .SPL,SPOOLER,20
```

2.6 SPOOLER MESSAGES

These messages are output by the Spooler:

ALLOCATE/ASSIGN ERROR ON PRINT QUEUE

indicates that bad status was returned from SVC 7 during either allocation or assignment of a spool queue file.

BATCH NOT IN SYSTEM

indicates that a /@SUBMIT control statement was processed during input spooling, but TASK.MTM was not loaded and started.

BUFFER NOT AVAILABLE

indicates that a buffer into which the particular inquire parameters would be passed cannot be found.

fd-INVALID FIRST CONTROL CARD

indicates a syntax error on /@ starting control card (fd=device name).

filename NOT ADDED TO BATCH QUEUE

indicates that a file spooled via /@SUBMIT was not put on the batch queue.

FILE voln:filename.ext/acct NOT ENTERED ONTO PRINT QUEUE

indicates that a spool file was closed, but the Spooler task was not loaded and started. The file may be printed or punched by entering a .SPL PRINT or .SPL PUNCH command whenever the Spooler is started. This message is generated by the operating system.

INQUIRY LIST ASSIGN ERROR CODE = xx

the list file for an INQUIRE command could not be assigned. The status returned from SVC 7 is xx.

INQUIRY LIST WRITE ERROR STATUS = yyyy

an I/O error occurred while writing to the fd specified in the LIST = parameter. The status returned from SVC 1 is yyyy.

INVALID FD - voln:

indicates that the file descriptor specified with .SPL PRINT or .SPL PUNCH command is a device, not a file.

INVALID START COMMAND

indicates a syntax error in START command.

INVALID SYNTAX-voln:filename.ext/acct

indicates that an invalid file descriptor was specified with a .SPL PRINT or .SPL PUNCH command.

I/O ERROR xxxx ON PRINT QUEUE FILE

indicates that SVC 1 returned bad status while writing to or reading from the print queue (xxxx is the status halfword).

NOTE

If I/O error code 84XX appears during the OS/32 Spooler operation, please be aware that the error may be related to Network Printing using PENnet. See the PENnet System Administration Manual, Appendix 9 for the PEN-NP Error Code Summary (84XX).

MISSING /@ CONTROL CARD, FILE fd

indicates that the ending control card is missing on stacked input. The file being spooled to is fd. Processing continues with the next job.

NO ROOM FOR MESSAGE BUFFERS

indicates that insufficient memory is available to build message buffers. Reload the Spooler with a larger segment size increment field.

OPTION-ERR xxxx:

indicates that device xxxx specified in the option parameter of the START command is not a device name used by the Spooler.

UNABLE TO ASSIGN fd

indicates that the physical device specified by fd in the START command cannot be assigned.

xxxx:I/O ERROR-PLEASE CORRECT AND CONTINUE

indicates that a spool device returned non-zero status; e.g., printer offline, CR: hopper check, CR: read check. The physical device is xxxx.

| Example of SPL INQUIRE OUTPUT:

FILENAME	TASKID	DEV:	CPY	PRI	#RCS	LRCL	SIZE
MTM:SPOOLER.UP/50 #	.SPL	PR:		2 D	75	80	66
M67A:@7563219.001/0	.BG	PR1:	2	128 V	1234	132	66
M67A:7563218.001/0#	.BG	PUN:		128 P	117	80	

Explanations of the column headings which appear in the above example are as follows:

FILENAME	specifies name of the file on the spool queue.
TASKID	identifies the task being spooled.
DEV:	specifies physical device to which output is spooled.
CPY	indicates number of copies spooled.
PRI	specifies the spool task priority.
	# indicates spooling in progress
	D indicates the file is to be deleted
	P indicates a punch file
	V indicates the VFC option is in effect
#RCS	specifies size of the file in logical records.
LRCL	indicates logical record length of the file.
SIZE	specifies page size in lines.

CHAPTER 3 DISK BACKUP UTILITY

3.1 GENERAL DESCRIPTION

The Disk Backup Utility provides a fast method of saving files. Files can be transferred from disk to disk, disk to magnetic tape, or tape to disk. The starting parameters specified, date of backup, and the names of the files backed up are listed. All files or selected files can be saved and restored. Files changed since a given date can be saved, and files on an output device can be replaced. Optionally, the data on the backup device can be verified.

The file types supported by the Disk Backup Utility are:

- Indexed files
- Contiguous files
- Nonbuffered indexed files
- Extendable contiguous files

The primary difference between nonbuffered indexed files and indexed files is that in nonbuffered indexed files, data is moved directly between the user's buffer and the disk, avoiding the central processing unit (CPU) overhead and system space memory requirements of buffered indexed files. As a result, each logical record starts on a physical sector boundary. Some unused space might exist between the logical records.

Extendable contiguous files have essentially the same features as contiguous files, with one important exception: they are extendable up to the capacity limit of the disk. By making suitable choices of block sizes, random access performance of these files will be equivalent to that of contiguous files.

See the OS/32 Application Level Programmer Reference Manual for a full explanation of supported file types.

When transferring files from one disk to another, the Disk Backup Utility writes the files onto the destination disk in a contiguous manner as long as there are no bad sectors. This minimizes access time on the destination disk for indexed files and maximizes the amount of contiguous free space on the destination disk. The Disk Backup Utility can copy the files onto an empty destination disk or onto a disk already containing files.

The Disk Backup Utility performs the following functions:

- Directly transfers files from one disk to another; the output disk serves as a backup of the original
- Transfers files from an input disk to an intermediate magnetic tape device; the magnetic tapes are used as a backup
- Appends files from an input disk to a magnetic tape containing previously backed up files; backup can be requested to locate the end of previously backed up files on the tape or can be notified that the tape is already positioned for the current backup operation
- Restores the data from the intermediate device to an output disk
- Verifies data copied during the backup operation
- Verifies data that was copied during a previous execution of the utility
- Selectively dumps individual files from disk to disk or from disk to tape
- Modifies the account numbers of files dumped from various accounts to one single account number during a disk to disk or disk to tape backup procedure
- Selectively restores files from tape to disk
- Modifies the account numbers of selectively restored files from various accounts to one single account number during a disk to disk or tape to disk restore procedure
- Selectively dumps files changed since a given time
- Deletes and replaces files on an output disk
- Transfers files created under OS/32 to OS/16 media or vice versa
- Displays the starting parameter specified, date of the Disk Backup Utility operation, and names of files backed up

- For indexed and nonbuffered indexed files, data block size and index block size
- Date and time the file was created
- Date and time the file was last changed

Data indicates the disk block image of the data on the file

EOF indicates end of file

EOV indicates end of volume

NOTE

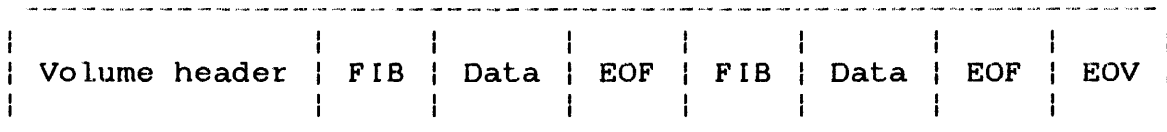
If a very large buffer size is specified in the START command, the user must ensure that the tape has a sufficient length of trailer following the end of tape marker or the tape might run off the reel in an attempt to write the last record.

For multivolume tapes; e.g., backup using two tapes, an EOV is written on the first tape and the following message is displayed:

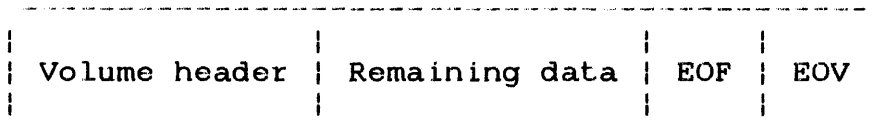
PLEASE MOUNT TAPE NUMBER xx

The volume header is written to the new tape and the remaining data is copied. The format is:

TAPE 1



TAPE 2



3.3 DISK BACKUP UTILITY REQUIREMENTS

The Disk Backup Utility requires:

- approximately 17kb of memory, plus additional memory required for buffers;
- a console device;
- at least one currently supported disk device and an additional disk or magnetic tape; and
- the version of the operating system with which the utility was released.

The Disk Backup Utility uses any additional memory available up to CTOP to expand its buffers. Utility execution times are a function of buffer size and decrease as buffer size increases.

3.4 OPERATING PROCEDURES

The Disk Backup Utility executes as a privileged user task (u-task) and must be built as such by using Link. Use of the SYSSPACE command ensures sufficient system space for copying files with large block sizes. See the OS/32 Link Reference Manual for further information on the SYSSPACE command.

If the user wishes to run the Backup Utility from the MTM terminal, their account must be authorized with bare disk privileges and file account privileges.

To reduce the possibility of errors in a data transfer to magnetic tape, the recommended blocking factors are:

TAPE RECORDING DENSITY	BLOCKING FACTOR
800 BPI	12.5K
1600 BPI	25.0K
6250 BPI	100.0K

These blocking factors refer to the size of the blocks on tape and are controlled by the SIZE parameter of the START command.

Using larger blocking factors than those recommended gains little additional storage space and results in an insignificant reduction in processing time. However, it does increase the probability of data transfer errors resulting in verify errors.

All disks used by the Disk Backup Utility must be marked online. The input disk can be marked online protected. When performing a backup from a fixed disk to a removable disk, the fixed disk must always be marked online protected, provided the disks are on the same disk drive. If the input disk is online protected, users can read from, but not write to, any files on the volume. If the input disk is not protected, users can read from and write to all files on the volume.

The Disk Backup Utility runs faster if the disks are marked on with the secondary directory option. If the secondary directory option is used on an output disk, an expansion factor should be used to ensure that no directory overflow occurs. (Default for the expansion factor is 100 files.)

If the Disk Backup Utility attempts to copy a file that is currently assigned with write privileges, a message is output indicating the file cannot be copied. If option SKIP is in effect, the program skips to the next file without pausing. If option SKIP is not in effect, the Disk Backup Utility pauses after logging the message. At this point, the condition can be corrected by closing the file. When the utility is continued, it attempts to copy the same file.

The integrity of all files is assumed. To guarantee successful execution of the program, either the output disk must be initialized prior to executing the Disk Backup Utility or, if files are to be restored in selective mode, the disk must be in a valid state. Initialization ensures that any bad sectors on the disk are avoided during the operation and that all file entries are removed from the disk directory. The integrity of a disk is ensured by executing either the Fastchek Utility or the Disk Integrity Check Utility. See Appendix B in this manual or the Fastchek Reference Manual for more details on these utilities.

NOTE

The Disk Backup Utility does not save temporary, spool, SYSTEM.DIR, or PACKINFO.DIR files. All filenames are output to the list device as they are copied, which provides the operator with a log of the files contained on a given tape.

Follow the procedure listed below to load and start the Disk Backup Utility:

1. Load the Disk Backup Utility using the LOAD command as follows:

LOAD BACKUP,n

n is the segment size increment, which is the maximum space available to the task (in kb).

2. Select backup as the current task using the TASK command as follows:

TASK BACKUP

3. If an empty disk is used as an output device, it must be already initialized using the Fastchek Utility or the Disk Initializer Utility. See the OS/32 Fastchek Reference Manual or Appendix A of this manual for more details on these utilities.
4. Mark the disk used as input online, (optionally) protected using the MARK command as follows:

MARK dm:,ON

or

MARK dm:,ON,PROTECT

5. Use the MARK command to mark the output disk online:

MARK dm:,ON

Use of the secondary directory option (CD) to mark the input and output disks online will achieve improved performance. The format of the commands are as follows:

MARK dm:,ON,,CD

or

MARK dm:,ON,PROTECT,CD

These commands, using the secondary directory option (CD), are to be used in place of the other commands (listed in procedures 4 and 5 above) to achieve improved performance.

6. Start the Disk Backup Utility using the START command. Filenames are read until an end of data indicator (/ * or . /) is found or until the maximum number of files that can be selected in one operation (40 if size was not specified in the START command) is found.

Format:

```
START ,IN=dev: ,OUT=dev: ,LIST=fd [SIZE=n]
      [,COMMAND=fd] [,END] [,DELETE[NODATECHECK]]
      [,VERIFY[/COUNT]] [,VO [/COUNT]] [,ABORT] [,SKIP] [,ACCOUNT=act]
      [,SINCE={mon/dd/yy} ,hh:mm [:ss] ] [,SELECT=fd] [ { APPEND
      { dd/mon/yy} } { POSITION } ] [,NEWDATE]
      { NOREWIND } ]
```

Parameters:

IN= dev: is the device mnemonic of the input device (the device from which data is copied). This device is assigned for sharable read only to logical unit 1 (lu1).

OUT= dev: is the device mnemonic of the output device (the device to which data is copied). The utility assigns lu2 SRO for disk to disk copies; SRO for tape to disk copies; and ERW for disk to tape copies.

END indicates stop scanning parameters (not needed unless COMMAND=fd is being used).

LIST= fd is optional in the START command but a required assignment. fd is the file descriptor of the list device for filenames and messages. The list device may be preassigned by the user to lu7. If entered in the START command, the list device is assigned for sharable write only to lu7.

SIZE= n is the buffer size in kb requested for disk to tape operation. The default size is 13kb. n is a decimal number with optional decimal places; e.g., 16.50.

COMMAND= fd is the input device from which additional parameters are to be taken. This allows the user to continue entering more arguments after the START parameter line is filled. COMMAND=fd may appear anywhere in the START parameter list. All parameters in the list are processed. After processing the START parameter list, additional parameters are read from the specified fd. The parameters are in the same format as the START parameter list and are processed until the parameter END is encountered.

DELETE/
NODATECHECK If DELETE is specified, the file on the output disk is deleted and replaced, only if the last written date indicates it is an older version than the file on the input medium. The additional parameter NODATECHECK overrides this provision.

If DELETE is specified and the file on the output disk is not an older version than the one on the input medium, BACKUP will display an error message and pause. Specification of DELETE and SKIP will cause the file to be skipped unless other errors are encountered.

The list of backed up files output during the Disk Backup Utility operation indicates if a file was deleted and replaced. It also indicates if a file was not deleted due to a date check. If DELETE and SKIP are specified, files already existing on the output disk are deleted. Files are only skipped if other errors are encountered.

NOTE

When the DELETE parameter is specified with the START command, program operation is slower.

VERIFY indicates that data on the input and output devices is verified after all files have been copied. If the data does not verify, the nonverifying records from both files are output to the list device along with an error message.

COUNT is a decimal number that indicates the number of records in a file that must fail to verify for Backup to skip the remainder of the file. If omitted, 5 is the default.

VO indicates data on the input and output devices is verified only. No copy operation is performed. Any records that do not verify are output to the list device.

ABORT terminates program if non-zero status is returned following an I/O operation or when allocating or assigning a file. If ABORT or SKIP is not specified, the task pauses.

SKIP any files that cannot be successfully assigned on the output disk by the Disk Backup Utility are not transferred. The files are identified in an error message and the program skips to the next file instead of pausing. If any files were skipped during the copy operation, a message is generated and verify is not performed. If neither SKIP nor ABORT is specified, the task is paused.

If SKIP is in effect and an I/O error is encountered on the tape during a tape to disk restore, the current file being restored is closed as is, and the restore is continued at the next file. If verify is the current operation, the file is not verified.

ACCOUNT= specifies the account number to which all files being backed up or restored are to be changed. Backup reads the file from the input device, changes the account number to the specified account number, and sends the file to the output device.

SINCE mon/dd/yy is the name of the month, day, and year. hh:mm:ss is the hour, minutes, and seconds. This option enables the user to back up or restore files changed since the specified date. The first three letters in the name of the month must be entered; the complete name can be entered. The month mnemonic should be followed by a slash and a two digit number for the day. The day should be followed by a slash and a two digit number for the year. The date can be entered with the month or day first. If the SINCE option is not specified, then no check is made of the date when a file was last changed. If the SINCE option is used in conjunction with the SELECT option, a file must have been changed since the given date and must match a SELECT entry in order to be backed up or restored.

SELECT= fd selectively copies or restores, and/or verifies, files from disk to disk, disk to tape, or tape to disk. fd is the file or device from which filenames to be restored or verified are specified or entered. The Disk Backup Utility assigns this fd to lu5.

The number of select entries that can be entered in an operation is limited by the segment size. Up to 40 select entries can be entered without additional memory from the segment size increment. Each additional select file in excess of 40 requires 16 additional bytes of memory. The SELECT option can be used in conjunction with the SINCE option, in which case the file must match a SELECT entry and must have been changed since the given date.

APPEND specifies that during a disk to magnetic tape backup operation, Backup is notified that the magnetic tape output device contains backup format data and that additional backup files are to be added. Backup scans the tape for the end of volume mark. It removes the mark and begins the current backup operation. The end of volume mark is rewritten after the last file is backed up. This option disables the VERIFY option.

POSITION specifies that during a disk to tape backup operation, Backup is notified that the magnetic tape output device contains backup format data, the files are to be appended to the tape, and the tape is already positioned at the end of volume mark. The tape is backspaced one record, the end of volume mark is removed, and the current backup operation begins. The end of volume mark is rewritten after the last file is backed up. This option disables the VERIFY option.

NOREWIND

specifies that during a disk to tape backup or restore operation, Backup is notified that the magnetic tape output device is positioned at the point where the current backup operation is to begin. If writing to the tape, backup will write a backup format volume header followed by the files to be backed up. If Backup reads from the tape, the first record encountered on the tape must be a backup format volume header. This option disables the VERIFY option.

CAUTION

BECAUSE BACKUP ASSUMES THAT THE TAPE IS PREPOSITIONED WHEN THE POSITION OR NOREWIND PARAMETERS ARE ENTERED, CAUTION SHOULD BE EXERCISED WHEN USING THESE OPTIONS. IF THE TAPE IS NOT CORRECTLY POSITIONED, PREVIOUSLY BACKED UP FILES COULD BE OVERWRITTEN AND LOST.

NEWDATE

specifies that the date created and date last written for each backed up file are updated to make them current dates.

Functional Details:

When started, the Disk Backup Utility prints the message:

```
PERKIN-ELMER OS/32 BACKUP 03-153 Rxx-yy
```

where xx and yy identify the revision level of the Disk Backup Utility.

| If the SELECT=fd parameter is specified in the START statement,
| a message displaying the maximum number of SELECT entries is
| output to the console and list device. Filenames (to be
| selected) are then read from the specified file or device (fd),
| until an end of data indicator (/ * or ./) is found, or until the
| maximum number of files that can be selected in one operation
| (default = 40 files) is reached. More than one select filename
| can be specified per 80-byte input record by separating the fds
| by commas (,) or by semicolons (;). After all filenames have
| been entered, the Disk Backup Utility starts the requested
| operation.

This header is output on the list device:

```
BACKUP xx-yy DATE RUN datetime Volume xxxx Size xx.xx
```

On a disk to disk or disk to tape operation, the volume name of the input device is displayed. If a tape restore is being done, the volume name displayed is the name of the output disk.

Backup then proceeds with the requested operation. Upon successful completion this message is printed:

```
END OF TASK CODE= 0
```

If any errors (other than a verify error) occur, this message is logged on the system console and printed on the list device:

```
END OF TASK CODE = 1
```

If an error occurs during a verify operation, this message is logged on the system console and printed on the list device:

```
END OF TASK CODE = 2
```

When a tape restore is successful, the following message is generated using information supplied from the original disk to tape operation.

```
BACKUP INFORMATION FROM TAPE:
```

```
BACKUP xx-yy DATE RUN date time VOLUME xxxx SIZE xx.xx
```

CAUTION

WHEN RESTORING FILES TO DISK FROM MAGNETIC TAPE, ALWAYS REMOVE THE WRITE RING FROM THE TAPE PRIOR TO MOUNTING THE TAPE. THIS PRECLUDES THE POSSIBILITY OF INADVERTENTLY WRITING ON AN ALREADY BACKED UP TAPE AND THE SUBSEQUENT LOSS OF FILES.

When selecting files for a backup or restore operation, it is possible to reduce the number of repetitive filename entries by using partial filenames. A hyphen (-) in the filename specifies that all files starting with the characters preceding the hyphen are to be backed up or restored, subject to any restrictions specified in the extension or account number fields.

Examples:

CAL32- selects for backup or restore operations all files whose first five characters are CAL32.

TESTPROG- selects for backup or restore operations all files named TESTPROG with any extensions.

The asterisk character (*) requests that all files with matching characters in the same positions as those entered be selected.

Examples:

CAL32*** selects for backup or restore operations all files between five and eight characters in length whose first five characters are CAL32.

***32.OBJ selects for backup or restore operations all files with a filename containing six characters whose fifth and sixth characters are 32 and whose extension is OBJ.

The characters * and - can be combined to further delimit selected files.

Example:

CAL**1- selects for backup or restore operations all files whose first three characters are CAL and whose sixth character is 1 with any extension.

When selectively restoring files from magnetic tape to disk, it is not necessary to read tapes prior to the tape containing the first file to be restored or verified. The program may be started with the first tape containing files to be restored or verified; whether that tape is the second, third, fourth, etc., tape of a set. Once the restore operation has begun, however, succeeding tapes must be mounted and read consecutively.

Examples:

```
START, IN=DSC2:, OUT=DSC1:, VERIFY, A, LIST=PR:
```

Copies DSC2: to DSC1:, verifies,
aborts on errors and sends listing
to PR:

```
ASSIGN 7, PR:
```

```
START, OUT=DSC1:, IN=DSC2:, VE
```

Preassigns list device to lu7;
copies DSC2 to DSC1, and verifies.

```
START , IN=DSC1:, OUT=MAG1:, LIST=PR:, SIZ=4.5, VE, A
```

Copies disk to tape, buffer
size = 4.5kb, verifies, aborts on
errors, sends listing to PR:.

```
START , IN=MAG1:, OUT=DSC2:, LIST=PR:, A, VE, DEL
```

Copies tape to disk, aborts on
errors. If filenames match,
restores only if the file from tape
has a more recent date.

```
START , IN=MAG1:, OUT=DSC2:, LI=PR:, VO
```

Verifies files from tape to disk but
does not copy files.

```
START , IN=DSC5:, OUT=MAG1:, L=PR:, SEL=CON:
```

Selectively backs up files from disk
to tape and reads filenames from the
console (CON:).

```
START , COMMAND=CON:, IN=DSC1:
```

```
BACKUP >OUT=MAG1:, LIST=PR:
```

```
BACKUP >SINCE=MAR/17/83, 12:30
```

```
BACKUP >VERI, END
```

Copies all files from disk changed
since March 17, 1983, at 12:30 to
tape.

3.4.1 Multiple Disk Backup

Backup of data from the fixed disk to the removable disk may require the use of multiple removable disks because the removable disk has a much smaller storage capacity than the fixed disk.

NOTE

Fixed disk to removable disk backup operations must be performed in a stand-alone environment with no other tasks running on the system. If other tasks are running while Backup is being performed, the system does not allow the user to mark the fixed disk off.

When Backup has filled a disk, the following message is displayed:

PLEASE MARK OFF THE INPUT DISK

PLEASE MARK OFF THE OUTPUT DISK AND MOUNT NEXT DISK VOLUME

TASK PAUSED

Follow this procedure to replace the removable disk and continue the backup operation:

1. Mark off the removable disk, using the MARK command.
2. Mark off the fixed disk, using the MARK command.
3. Power-down the drive.
4. Remove the removable disk and mount the next removable disk to be used.
5. Power-up the drive.
6. Mark the fixed disk on protect, using the MARK command.
7. Mark the removable disk on, using the MARK command.
8. Continue Backup, using the CONTINUE command.

Backup will not split a file between two disks. If backup cannot fit the entire file onto a disk, it will request that a new disk be mounted. Backup will then write the entire file onto the new disk. A file can be no larger than the total storage capacity of the output disk.

Disks that already have files residing on them can be used in Backup operations. Backup will not overwrite these files, but will use the remaining free space on the disk.

3.5 MESSAGES

ASSIGN ERROR FILE fn/message

indicates that bad status was encountered while trying to assign to a device through the START command or while trying to allocate or assign a file. This message specifies the type of error depending on returned SVC 7 status.

DATE-ERR

indicates invalid day, year, or month in START command.

DEVICE UNAVAILABLE FOR EXCLUSIVE USE:xxxx

indicates that device cannot be accessed exclusively; xxxx=device.

ENTER FILENAMES TO BE COPIED

indicates program request for filenames that are to be restored. If lu5 is assigned to the console, a prompt is output.

ENTER FILENAMES TO BE VERIFIED

indicates program request for filenames that are to be verified only. If lu5 is assigned to the console, a prompt is output.

FD-ERR

indicates invalid fd in START command.

FORM-ERR

indicates syntax error in START command.

INCORRECT NUMBER OF RECORDS TRANSFERRED

indicates that the number of data blocks written on the previous tape during a multivolume disk to tape operation is not equal to the number of data blocks read during tape to disk operation.

INPUT DISC CONTAINS NO FILES

indicates that no directory was found on input disk.

INPUT DISK MUST BE MARKED ON "PROTECT"

indicates that in an attempt to backup data from a fixed disk of a 10Mb disk to a removable disk of the same pack, the input was not marked online with a protected status. Backup pauses. The input disk must be marked OFF and then marked ON PROTECTED, and the task continued.

INSUFFICIENT MEMORY

indicates that not enough memory is available. Reload the program into a larger segment and restart.

INVALID DEVICE CODE

indicates use of an invalid device code or that a magnetic tape is specified as both the input and output device.

INVALID FILE TYPE, FILE fn NOT TRANSFERRED

indicates that file type of file fn is not contiguous or indexed.

INVALID TAPE VOLUME xxxx, EXPECTING xxxx

indicates that the currently mounted multivolume tape has not been created from the same input disk.

I/O ERROR LU=xx STATUS=yy ON fd:fn.ext

indicates that an I/O error was encountered during an SVC l read or write operation from a device or file. The lu is xx; yy is the error status.

LU xx UNASSIGNED

indicates that the input, output or list device was not assigned.

MARK INPUT DISC ON

indicates that input disk has not been marked on.

MARK OUTPUT DISC ON

indicates that the output disk is offline.

NON-VERIFY:FILE fn LOGICAL UNIT x: RECORD NUMBER xxxx

indicates that data in the file fn does not verify.

OPTION VERIFY

indicates that the program started a verify routine.

OS/32 BACKUP xx-yy

indicates that the program is operational. The program's revision level is xx; yy is the update level within the revision.

OS/32 Rxx-yy REQUIRED

indicates that backup is being run on an incompatible operating system. Revision xx-yy or higher of the operating system is required.

PLEASE MARK OFF THE INPUT DISC

PLEASE MARK OFF THE OUTPUT DISC AND MOUNT NEXT DISC VOLUME

indicates that the end of volume was reached before all files were copied and another volume must be mounted.

PLEASE MOUNT TAPE NUMBER xx

indicates that the end of a tape was reached before all files were copied or verified, or the tape currently mounted is not the first tape at the start of the verify routine.

SELECTED FILES EXCEED MAXIMUM

indicates that the maximum number of files allowed during selective restore/verify was exceeded.

SELECTED FILES NOT COPIED

fn

.
.
.

indicates that specified files were not found on the disk or tape after a selective restore operation. All filenames not processed are listed following this message.

SELECTED FILES NOT VERIFIED

fn

.
.
.

indicates that the specified files were not found on the disk or tape after a selective verify operation. The filenames follow this message.

SELECTIVE RESTORE MAXIMUM ENTRIES=xxx

indicates that the mode is selective restore. The maximum number of selected entries is xxx.

SELECTIVE VERIFY MAXIMUM ENTRIES=xxx

indicates that the mode is selective verify only. The maximum number of selected entries is xxx.

SKIP IN EFFECT VERIFY IGNORED

indicates that files were skipped during the copy operation. Verify cannot be performed.

SYNTAX ERROR fn

indicates invalid syntax in filename for selective restore.

TAPE OUT OF SEQUENCE, SEQU=xx

indicates that the currently mounted tape does not have the expected sequence number. The sequence number on the volume label of the currently mounted tape is displayed.

TIME-ERR

indicates invalid hours, minutes, or seconds in the START command.

***TAPE ERROR - FOLLOWING FILE PARTIALLY RESTORED

indicates that the file was partially restored during a tape to disk restore; the remaining part of the file was skipped.

CHAPTER 4
OS/32 ACCOUNTING REPORTING UTILITY

4.1 INTRODUCTION

The Accounting Reporting Utility processes accounting data collected at data collection time and generates reports or archival files through operator commands. However, please be aware that processor time and input/output (I/O) counts for tasks executed on an auxiliary processing unit (APU) of the Model 3200MPS System will not be reflected in the totals generated. The Accounting Reporting Utility provides selection of the:

- collected accounting data to be processed (input files),
- account numbers on which the reports or archives are generated,
- time period in which the accounting data was collected for the requested account numbers,
- disk devices with which additional file usage information is generated,
- cost factors used to calculate charges for system and disk usage,
- generation of reports, and
- generation of archives.

Collected accounting data to be processed can consist of these input files and devices:

- Accounting transaction files (ATFs)
- Authorized user file (AUF)
- Archival files
- Disk devices

See Figure 4-1 for an illustration of the Accounting Reporting Utility process.

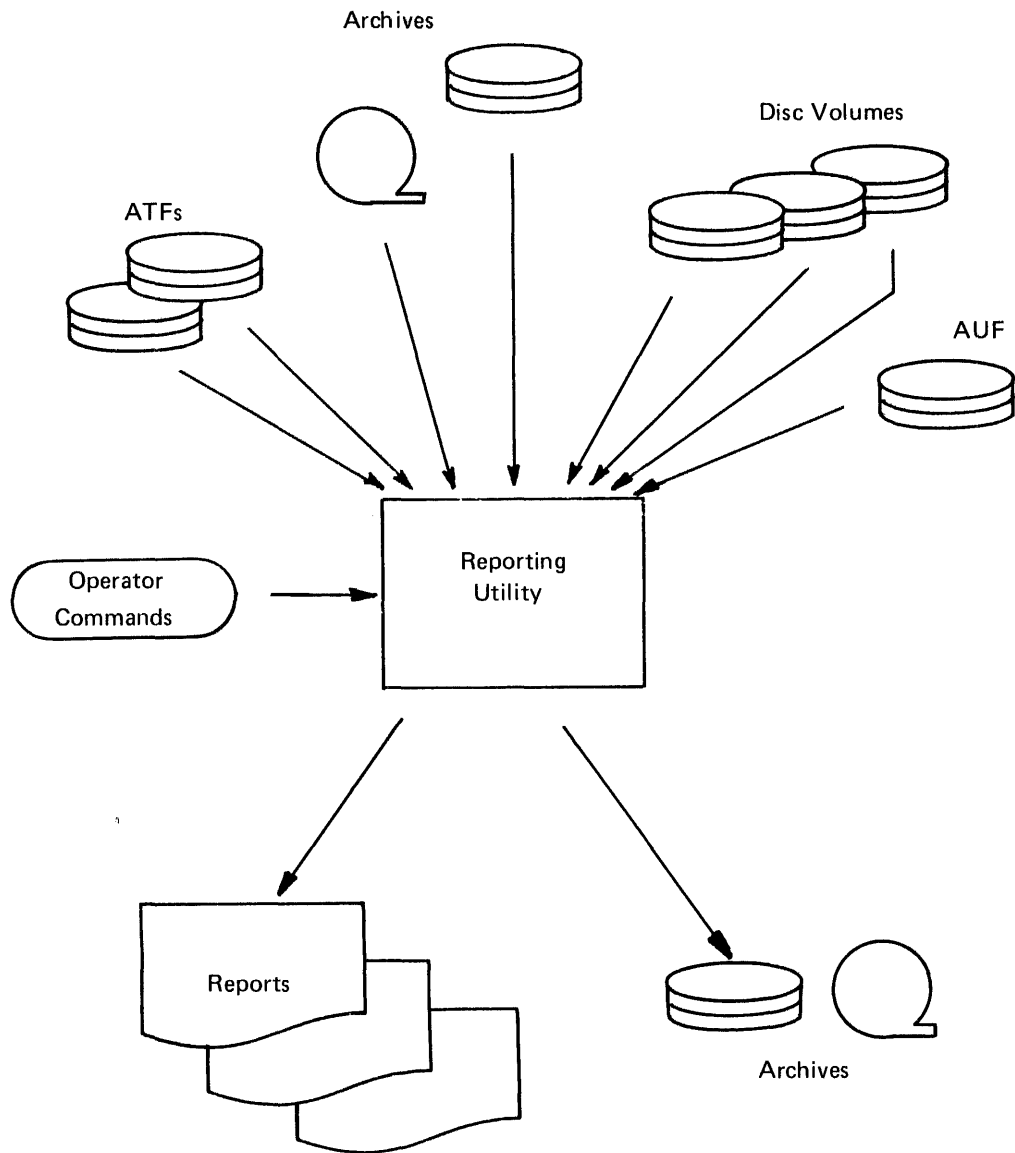


Figure 4-1 The Accounting Reporting Utility Process

Account numbers specify each account for which data is read from the input files. The time period selects the starting and ending dates for which data is read from the input files. Disk devices are the disk volumes to be scanned for file usage data. Cost factors specify the charge per unit of usage with which each customer is charged.

4.1.1 Data Collection

Data collection is performed by the operating system and multi-terminal monitor (MTM) through counting, collecting, and logging routines, which together collect and store the accounting information in a user-specified disk file. For systems with MTM, see the OS/32 Multi-Terminal Monitor (MTM) System Planning and Operator Reference Manual for the data collection start procedure.

4.1.2 Starting the Data Collection For an OS Without MTM

First, an OS must have been generated (sysgen) with accounting support. Next, the accounting facility data collection program (AFDCP), a "dummy MTM", can be loaded and started in the following manner.

```
LOAD .MTM,AFDCP
TASK .MTM
START,ATF=fd
```

where fd is the file descriptor of the accounting transaction file.

The AFDCP is able to accept two MTM commands, QUIESCE and ATF. The format of these commands are as follows:

```
.MTM QUIESCE
```

and

```
.MTM ATF fd
```

where fd is the new accounting transaction file.

The definitions for these commands are listed below:

QUIESCE terminates AFDCP.

ATF changes the accounting transaction file. The current ATF is closed and a new ATF is allocated (if necessary) and assigned.

4.2 COMPILING, ESTABLISHING, AND STARTING THE ACCOUNTING REPORTING UTILITY

| The command sequences below can be used as examples for creating
| an Accounting Reporting Utility task. Certain parameters will
| vary depending upon system configuration and library file
| location and names.

The following commands compile the Accounting Reporting Utility FORTRAN source:

```
LOAD F7D,.BG;TASK.BG
ASSIGN 1,ACCTF.FTN,SRO
XALLOCATE ACCTF.OBJ,IN
ASSIGN 2,ACCTF.OBJ
ASSIGN 3,CON:
ASSIGN 6,F7D.ERR/S
START,HOLL,BATCH,NRXT
```

Program common data areas are defined in the file ACCTF.FTN. This file is copied into each FORTRAN module through the FORTRAN INCLUDE option and must be on the default volume. The following commands assemble the assembly language modules:

```
LOAD CAL32;TASK,BG
XALLOCATE ACCTC.OBJ,IN
ASSIGN 1,ACCTC.CAL,SRO
ASSIGN 2,ACCTC.OBJ
ASSIGN 3,CON:
TEMPFILE 4,IN,80
TEMPFILE 5,IN,256
START,CROSS,BATCH,SQUEZ=99
```

The following commands establish the Accounting Reporting Utility as a task:

```
LOAD LINK,.B
START
LOG CON:
TITLE ACCT
ESTABLISH TASK
INCLUDE ACCTF.OBJ
INCLUDE ACCTC.OBJ
LIBRARY F7RTL/S
BUILD ACCT
MAP CON:
END
```


4.2.1 LOAD Command

The LOAD command loads the Accounting Reporting Utility into memory.

Format:

LOAD ACCT

Functional Details:

The MTM account under which the Accounting Reporting Utility is invoked has to be established with the no key check and file access by account number privileges, if OPTION AUF=fd is going to be specified. See the OPTION command in Section 4.3.3 for further information.

If disk volumes are to be scanned for current file allocation to be included in the report, the MTM account has to be established with bare disk assignment privilege.

START

4.2.2 START Command

The START command starts the Accounting Reporting Utility.

Format:

```
START [ ,COMMAND=fd1 ] [ ,LIST=fd2 ] [ ,LOG=fd3 ]
```

Parameters:

- COMMAND=fd₁ specifies the input device from which commands are to be entered. If this parameter is omitted, the default is the device (CON:). If the command input device is interactive and the LOG parameter is omitted, all commands entered and error messages generated are sent to the command input device.
- LIST=fd₂ specifies the output device to which reports are sent. If this parameter is omitted, the default is the device (PR:). If the list output device is a disk file, it must have been previously allocated. The list output file can be changed by the REPORT command.
- LOG=fd₃ specifies the output device to which all commands entered and error messages generated are recorded. If this parameter is omitted and the command input device is interactive, all commands entered and error messages generated are sent to the command input device. If this parameter is omitted and the command input device is batch, all commands entered and error messages generated are sent to the default log device (PR:). If the log output device is a disk file, it must have been previously allocated.

Messages:

INVALID START OPTION

The command, list, or log device entered as a start option is invalid.

COMMAND DEVICE ERROR

The syntax of the device or file entered as a start option is invalid.

LIST DEVICE ERROR

The device or file specified as the list device is invalid.

LOG DEVICE ERROR

The device or file specified as the log device is invalid.

DUPLICATE START OPTION

One of the start options was entered more than once.

SYNTAX ERROR

The syntax of the start options is invalid.

 COMMAND
UNABLE TO ASSIGN LIST
 LOG

The device or file specified as a start option cannot be assigned.

4.3 ACCOUNTING REPORTING UTILITY COMMANDS

The following commands, listed in their logical order of use, execute the Accounting Reporting Utility:

- GET
- SELECT
- OPTION
- CHARGE
- REPORT
- SAVE
- PAUSE
- END

All commands except PAUSE and END can be continued on succeeding lines by entering a comma after the last parameter specified on a line. A command is terminated by a carriage return (CR) or end of record.

The temporary volume must be mounted and online to the system during execution of the Accounting Reporting Utility. The program requires single and double precision floating point support. Messages for these commands are presented in Section 4.5.

4.3.1 GET Command

The GET command specifies the input files containing the collected accounting data and the disk volumes to be processed by the Accounting Reporting Utility.

Format:

$$\text{GET} \left[\begin{array}{l} \text{ATF=} \left\{ \begin{array}{l} \text{fd}_1 \\ (\text{fd}_1, \text{fd}_2 [\dots, \text{fd}_n]) \end{array} \right\} \\ \text{,ARCHIVE=} \left\{ \begin{array}{l} \text{fd}_1 \\ (\text{fd}_1, \text{fd}_2 [\dots, \text{fd}_n]) \end{array} \right\} \\ \text{,DISC=} \left\{ \begin{array}{l} \text{volid}_1 \\ (\text{volid}_1, \text{volid}_2 [\dots, \text{volid}_n]) \end{array} \right\} \end{array} \right]$$

Parameters:

ATF=fd specifies the file descriptors of the ATFs created by the MTM START or .MTM commands to be processed by the Accounting Reporting Utility. The ATF input files contain the accounting data collected at data collection time. If more than one fd is specified, parentheses must be used. When this parameter is specified, all ATFs are assigned with exclusive read-write (ERW) access privileges. The maximum number of ATF and archival files that can be specified in the GET command is 10.

NOTE

Once assigned to data collection, the ATFs cannot be simultaneously used by the Accounting Reporting Utility.

ARCHIVE=fd specifies the archival files (disk or tape) created by a previous execution of the Accounting Reporting Utility through the SAVE command. These files are to be processed by the Accounting Reporting Utility. If more than one fd is specified, parentheses must be used. The maximum number of fds that can be specified by the GET command, including ATF and archival files, is 10. When this parameter is specified, all archival files are assigned with ERW access privileges.

DISC=valid specifies the disk volumes to be scanned for current file allocation data. If a specified volume is not online, a message requesting that volume to be marked online is displayed. The maximum number of volumes that can be specified is 10.

Functional Details:

These parameters can be specified separately by entering multiple GET commands or combined in one GET command. The GET command must be entered before the REPORT or SAVE commands. If a subsequent GET command is entered after the REPORT or SAVE commands are entered, the files specified by the previous GET command are no longer input files to the Accounting Reporting Utility. The fds specified by the subsequent GET command are the new input files to the Accounting Reporting Utility. Specification of the DISK parameter includes all files which were active during the report period. This data is copied to a temporary file to be included in the generation of the final report.

Examples:

```
GE ATF=M67A:ATF04.E
```

```
GE ATF=(M300:ATF01.B,M300:ATF02.B),  
ARC=MAG4: ,  
DI=(M300,M301,M67B)
```

4.3.2 SELECT Command

The SELECT command selects from the input files specified by the GET command a particular subset of accounting data to be processed by the Accounting Reporting Utility. This subset of accounting data is selected by specifying account numbers and a time period for which the report or archive is to be generated.

Format:

$$\text{SELECT} \left[\text{FROM} = \left\{ \begin{array}{l} \text{mon/dd/yy} \\ \text{dd/mon/yy} \end{array} \right\} \right] \left[\text{TO} = \left\{ \begin{array}{l} \text{mon/dd/yy} \\ \text{dd/mon/yy} \end{array} \right\} \right]$$

$$\left[\text{ACCOUNT} = \left\{ \begin{array}{l} \text{actno}_1 \text{ } [-\text{actno}_n] \\ (\text{actno}_1, \text{actno}_2 \text{ } [, \dots, \text{actno}_n]) \\ \text{ALL} \end{array} \right\} \right]$$

Parameters:

FROM= mon/dd/yy or dd/mon/yy identifies the starting date of the particular time period for which a subset of accounting data is selected. All records containing a date within the specified starting and ending dates are to be processed by the Accounting Reporting Utility. If this parameter is omitted, the time period begins with the earliest date recorded in the ATFs or archival files. The month must be specified by at least three alphabetic characters.

TO= mon/dd/yy or dd/mon/yy specifies the ending date of the particular time period for which a subset of accounting data is selected. All records containing a date within the specified starting and ending dates are to be processed by the Accounting Reporting Utility. If this parameter is omitted, the time period ends with the latest date recorded in the ATFs or archival files. The month must be specified by at least three alphabetic characters.

ACCOUNT= actno specifies the account numbers within the specified time period for which accounting data is to be processed by the Accounting Reporting Utility. Account numbers can be specified as a list or as a range. The account numbers that can be specified by the SELECT command range from 0 through 65,535 (excluding 255). Account number 255 is reserved for the AUF. Account number 0 is for system files and is the default for all operator commands.

ALL specifies that records for all account numbers within the specified time period are processed by the Accounting Reporting Utility. If the ACCOUNT parameter is omitted, the default is all account numbers which were recorded in the input files within the specified time period.

Functional Details:

Multiple reports can be generated from the particular subset of accounting data processed by the Accounting Reporting Utility. If the SELECT command is not specified, all data in the files and devices specified by the GET command are processed by the Accounting Reporting Utility. If the SELECT command is entered, only records containing dates and account numbers specified by the SELECT command are processed from the input files. If a subsequent SELECT command is entered after the REPORT and SAVE commands are entered, the account numbers and dates specified by the previous SELECT command are no longer selected to be processed by the Accounting Reporting Utility. The account numbers and dates specified by the subsequent SELECT command are the new account numbers to be processed by the Accounting Reporting Utility.

Examples:

```
SEL FR=SEP/1/82 ,TO=OCT/1/82 ,
  AC=ALL
```

```
SEL FR=AUG/1/82
SEL TO=DEC/1/82
SEL AC=100-150
```

```
SEL FR=JUN/15/82 ,
  TO=AUG/15/82 ,
  AC=(45,107,145,118,155)
```


4.3.3 OPTION Command

The OPTION command includes any of these optional features in the generated report:

- AUF information
- Date
- Message
- Signature
- Titles

Format:

$$\begin{aligned}
 & \text{OPTION} \left[\text{AUF} = \left\{ \begin{array}{c} \text{fd} \\ * \end{array} \right\} \right] \left[\text{,DATE} = \left\{ \begin{array}{c} \text{'date'} \\ * \end{array} \right\} \right] \left[\text{,MESSAGE} = \left\{ \begin{array}{c} \text{'message'} \\ \text{fd} \\ * \end{array} \right\} \right] \\
 & \left[\text{,SIGNATURE} = \left\{ \begin{array}{c} \text{'name'} \\ * \end{array} \right\} \right] \left[\text{,HEADER} = \left\{ \begin{array}{c} \text{'title'} \\ \text{fd} \\ * \end{array} \right\} \right] \\
 & \left[\text{,TACCOUNT} = \left\{ \begin{array}{c} \text{'title'} \\ \text{fd} \\ * \end{array} \right\} \right] \left[\text{,TSUMMARY} = \left\{ \begin{array}{c} \text{'title'} \\ \text{fd} \\ * \end{array} \right\} \right] \\
 & \left[\text{,TSYSUSE} = \left\{ \begin{array}{c} \text{'title'} \\ \text{fd} \\ * \end{array} \right\} \right]
 \end{aligned}$$

Parameters:

- AUF=** fd is the file descriptor of the AUF, created by the account utility, that is to be included as an input file to the Accounting Reporting Utility. The user identifier (userid), group account number, total signon time, and user time left are included in the generated report for each account. When this parameter is specified, the AUF is assigned with shared read only (SRO) access privileges. If this parameter is omitted, the accounting data in the AUF is not included in the generated report.
- * closes the previously assigned AUF, excluding it as an input file to the Accounting Reporting Utility.
- DATE=** date is the date to be included as the report date in the generated report. The date is a 1- to 20-character alphanumeric string. If this parameter is omitted, the current date is included.
- * specifies that the current date is to be included in the generated report.
- MESSAGE=** message is a 1- to 80-character alphanumeric string specifying a message to be included at the end of the individual account or account summary report. If this parameter is omitted, no message is included.
- fd is the file descriptor containing the message to be included at the end of the individual account or account summary report.
- * specifies that the previously specified message is not to be repeated at the end of the individual account or account summary report.
- SIGNATURE=** name is a 1- to 40-character alphanumeric string specifying a name to be included in the generated report. If this parameter is omitted, no signature is output.
- * specifies that the previously specified name is not to be included in the generated report.

HEADER=

title is a 1- to 80-character alphanumeric string specifying the main title to be included at the top of each report. If this parameter is omitted, the main title PERKIN-ELMER OS/32 ACCOUNTING REPORT is the default.

fd is the file descriptor containing the main title to be included at the top of each report.

* specifies that the main title to be included at the top of each report:

PERKIN-ELMER OS/32 ACCOUNTING REPORT

TACCOUNT=

title is a 1- to 80-character alphanumeric string specifying the subtitle to be included at the beginning of each report generated for individual accounts. If this parameter is omitted, the subtitle ACCOUNT NUMBER: actno is the default.

fd is the file descriptor containing the subtitle to be included at the beginning of each report generated for individual accounts.

* specifies that the subtitle to be included in the beginning of each report generated for individual accounts is:

ACCOUNT NUMBER: actno

actno is an individual account number for which the report is being generated.

TSUMMARY=

title is a 1- to 80-character alphanumeric string specifying the subtitle to be included at the beginning of each report generated for account summaries. If this parameter is omitted, the subtitle ACCOUNT SUMMARY REPORT is the default.

fd is the file descriptor containing the subtitle to be included at the beginning of each report generated for account summaries.

* specifies that the subtitle to be included at the beginning of each report generated for account summaries is:

ACCOUNT SUMMARY REPORT

TSYSUSE= title is a 1- to 80-character alphanumeric string specifying the subtitle to be included at the beginning of each report generated for system usage summaries.

If this parameter is omitted, the subtitle SYSTEM SUMMARY REPORT is the default.

fd is the file descriptor containing the subtitle to be included at the beginning of each report generated for system usage summaries.

* specifies that the subtitle to be included in the beginning of each report generated for system usage summaries is:

SYSTEM SUMMARY REPORT

Functional Details:

If an OPTION parameter is entered more than once, the last parameters specified are used. If this command is entered with a syntax error, the default values are used. If the parameters exceed one line, close the last parameter's text on the first line with a single quotation mark followed by a comma and a CR. Continue the parameter's text on the next line with a single quotation mark followed by text and a closing single quotation mark.

Examples:

```
OP AUF=AUF.20C,DA=*,MES=MSG$.ACT,  
  SIG='M.PICKELL',HE=*,TA=*
```

```
OP AUF=*,DA='10/15/79',MES='DEPARTMENT NO. 6052',  
  SIG='I.SCHIEL',HE='SUTTON CO. ACCOUNTING REPORT',  
  TSU=*
```

4.3.4 CHARGE Command

The CHARGE command specifies the cost factors used to calculate charges for system usage. This command is used in conjunction with the REPORT command.

Format:

```

CHARGE [ CPUTIME= { (O/dlrs.cts,U/dlrs.cts)
                   NULL
                   dlrs.cts } ]
        [ ,IOCOUNT= { (classno1/dlrs.cts [, ..., classnon/dlrs.cts] )
                       NULL
                       dlrs.cts } ]
        [ ,MEMORY= { [I/dlrs.cts] [,P/dlrs.cts] [,S/dlrs.cts]
                     NULL
                     dlrs.cts } ]
        [ ,SECTORS= { NULL
                     dlrs.cts } ]
  
```

Parameters:

CPUTIME= O/dlrs.cts specifies the charge in dollars and cents (7 digits) for each second of operating system central processing unit (CPU) time used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the time used is included.

U/dlrs.cts specifies the charge in dollars and cents for each second of user CPU time used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the time used is included.

NULL specifies that no charges for CPU time used are to be included in the generated report.

dhrs.cts specifies the charge in dollars and cents (7 digits) for each second of operating system and user CPU time used.

IOCOUNT=

classno/dhrs.cts specifies the charge in dollars and cents (7 digits) for each I/O transfer executed for up to 10 device or file classes. If this parameter is omitted, no charges are calculated or included in the generated report. However, the total count of I/O transfers executed for all device or file classes is included. The maximum number of I/O classes that is currently processed by the Accounting Reporting Utility is 10.

NULL specifies that no charges for counts of I/O transfers are to be included in the generated report.

dhrs.cts specifies the charge in dollars and cents (7 digits) for each I/O transfer executed by the device or file classes in the system.

NOTE

When tasks are executed on an APU, the processor time spent on the APU of the Model 3200MPS System will not be included in the totals generated for those specific tasks. The time spent by the task on the CPU and all operating system service time is charged normally.

MEMORY=

I/dhrs.cts specifies the charge in dollars and cents (7 digits) for each .25kb segment of impure memory used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the number of .25kb segments of impure memory used is included.

P/dhrs.cts specifies the charge in dollars and cents (7 digits) for each .25kb segment of pure memory used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the number of .25kb segments of pure memory used is included.

S/dlrs.cts specifies the charge in dollars and cents (7 digits) for each .25kb segment of system space used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the number of .25kb segments of system space used is included.

NULL specifies that no charges for number of .25kb segments of impure memory, pure memory, or system space used is to be included in the generated report.

dlrs.cts specifies the charge in dollars and cents (7 digits) for each .25kb segment of impure memory, pure memory, and system space used.

SECTORS= NULL specifies that no charges or number of days on which disk sectors were used is to be included in the generated report.

dlrs.cts specifies the charge in dollars and cents (7 digits) for each day disk sectors were used. If a sector is used for only a part of a day, sector usage is charged at the full day rate.

Examples:

```
CHA CP=0/01.50,U/0.75,  
    IOC=0/1.00,1/1.10,2/1.15,3/1.25,  
    MEM=I/0.90,P/1.15,S/1.50,  
    SEC=1.05
```

```
CHA CP=1.15,  
    IOC=1.00,  
    MEM=1.00
```

```
CHA SEC=NULL
```

REPORT

4.3.5 REPORT Command

The REPORT command generates reports from the accounting data specified by the GET and SELECT commands and sends the reports to the list device specified in the START or REPORT command.

Format:

$$\text{REPORT} [\text{LIST}=\text{fd}] \left[\text{,FROM}=\left\{ \begin{array}{l} \text{mon/dd/yy} \\ \text{dd/mon/yy} \end{array} \right\} \right] \left[\text{,TO}=\left\{ \begin{array}{l} \text{mon/dd/yy} \\ \text{dd/mon/yy} \end{array} \right\} \right] \\
 \left[\text{,ACCOUNT}=\left\{ \begin{array}{l} \text{actno}_1 \text{ } [\text{actno}_n] \\ (\text{actno}_1, \text{actno}_2 \text{ } [, \dots, \text{actno}_n]) \\ \text{ALL} \end{array} \right\} \right] \\
 \left[\text{,SUMMARY}=\left\{ \begin{array}{l} \text{actno}_1 \text{ } [-\text{actno}_n] \\ (\text{actno}_1, \text{actno}_2 \text{ } [, \dots, \text{actno}_n]) \\ \text{ALL} \end{array} \right\} \right] \\
 [\text{,SYSUSE}]$$

Parameters:

- LIST=fd specifies the file descriptor to which the generated report is sent. If the specified fd is a disk file, the file must have been previously allocated. If this parameter is omitted, PR: is the default device.
- FROM= mon/dd/yy or dd/mon/yy specifies the starting date for the report period. If the SELECT command was entered, the date must be within the time period specified by the SELECT command. If this parameter is omitted, the starting date specified by the SELECT command or the earliest date in the input files is used. The month must be specified by at least three alphabetic characters.

TO= mon/dd/yy or dd/mon/yy specifies the ending date for the report period. If the SELECT command was entered, the date must be within the time period specified in the SELECT command. If this parameter is omitted, the ending date specified by the SELECT command or the latest date in the input files is used. The month must be specified by at least three alphabetic characters.

ACCOUNT= actno specifies the account numbers for which individual accounting reports are to be generated. These account numbers must be within the range specified by the SELECT command or within the range collected in the input files.

ALL specifies individual accounting reports are to be generated for all account numbers specified by the SELECT command or collected in the input files. If the ACCOUNT= parameter is omitted, ALL is the default.

SUMMARY= actno specifies the account numbers for which summary reports are to be generated. These account numbers must be within the range specified by the SELECT command.

ALL specifies all accounts are to be included in one account summary report. The account numbers included in the summary report are those specified by the SELECT command or all accounts collected in the input files (if the SELECT command was not entered). If this parameter is omitted, no account summary reports are generated.

SYSUSE specifies that a system summary report is to be generated.

Functional Details:

All parameters in a REPORT command that are required to generate a set of reports must be entered in one REPORT command. If the REPORT command is entered and no parameters are specified, the defaults are: ACCOUNT=ALL and the list device specified in a previous REPORT or START command. Any number of reports can be generated from input files by specifying various date periods and account numbers.

If the parameters exceed one line, enter a comma as the last character on that line and continue the remaining parameters on the next line.

Examples:

REP LI=PR1: ,FR=JUL/15/82,TO=OCT/15/82,
AC=(45,32,100,147,121)

REP LI=PR: ,FR=SEP/1/82,TO=OCT/1/82,
SUM=118-122

REP LI=M67B:SYSACT.40E,FROM=JAN/1/82,
TO=MAR/31/82,SYSUSE

REPORT

Sample Reports:

Examples of a system summary report, account summary report, and an individual account report are presented in Figures 4-2, 4-3, and 4-4, respectively.

PERKIN-ELMER OS/32 ACCOUNTING REPORT

SYSTEM SUMMARY REPORT
REPORT PERIOD: AUG/1/82 TO OCT/12/82

REPORT DATE: OCT/12/82

NUMBER OF TASKS LOADED	3.
OS CPU TIME	0: 0: 5.784
USER CPU TIME	0: 1: 1.629
WAIT TIME	0: 0: 0.
ROLL TIME	0: 9:28.320
ROLL CPU TIME	0: 0: .524
ROLL COUNT	3.
IMPURE MEMORY	63.75kb
PURE MEMORY	26.75kb
SYSTEM SPACE USED	28.50kb
NO. I/Os CLASS 0	1764.
BYTES TRANSFERRED	232405.
NO. I/Os CLASS 1	28.
BYTES TRANSFERRED	701440.

Figure 4-2 System Summary Report

| This sample report summarizes how the computer time was used by
| the three user tasks (u-tasks) from August 1, 1982, through
| October 12, 1982. The report shows that:

- | ● There are three tasks.
- | ● The operating system occupied the CPU for 5.784 seconds, and
| the u-task occupied the CPU for 1:1.629 minutes.
- | ● The tasks were rolled out in 9:28.320 minutes.
- | ● The amount of CPU time used to roll tasks in and out of memory
| is .524 seconds.
- | ● The tasks were rolled out of and into memory 3 times.
- | ● The system occupied 63.75kb of impure and 26.75kb of pure
| memory, and 28.50kb of system space.
- | ● There were 1,764 system I/Os, transferring 232,405 bytes.
- | ● There were 28 u-task I/Os, transferring 701,440 bytes.

PERKIN-ELMER OS/32 ACCOUNTING REPORT

ACCOUNT SUMMARY REPORT

REPORT PERIOD: AUG/1/82 to OCT/12/82

REPORT DATE: OCT/12/82

ACCOUNT NUMBER :0-200

CPU USAGE:

OS TIME: 0: 0: 5.784

USER TIME: 0: 1: 1.629

MEMORY USAGE:

SYSTEM SPACE: 28.50kb

IMPURE MEMORY: 63.75kb

PURE MEMORY: 26.75kb

I/O USAGE

CLASS 0:

BYTES TRANSFERRED: 0.232405D+06

NO. OF I/Os: 1764.;

CLASS 1:

BYTES TRANSFERRED: 0.701440D+06

NO. OF I/Os: 28.;

Figure 4-3 Account Summary Report

| This sample report summarizes how account number 200 utilized the
| CPU and memory from August 1, 1982, through October 12, 1982.
| The report shows that:

- | ● The operating system occupied the CPU for 5.784 seconds.
- | ● The u-task occupied the CPU for 1:1.629 minutes.
- | ● System space occupied 28.50kb of memory; impure memory
| occupied 63.75kb; and pure memory occupied 26.75kb.
- | ● There were 1,764. class 0 I/Os, transferring 0.232405D+6
| bytes.
- | ● There were 28. class 1 I/Os, transferring 0.701440D+06 bytes.

PERKIN-ELMER OS/32 ACCOUNTING REPORT

ACCOUNT NUMBER :86

REPORT PERIOD: AUG/1/82 TO OCT/12/82

REPORT DATE: OCT/12/82

ACCOUNT NUMBER :86

CPU USAGE:

OS TIME: 0: 0: 5.784

USER TIME: 0: 1: 1.629

MEMORY USAGE:

SYSTEM SPACE: 28.50kb

IMPURE MEMORY: 63.75kb

PURE MEMORY: 26.75kb

I/O USAGE

CLASS 0:

BYTES TRANSFERRED: 0.232405D+06

NO. OF I/Os: 1764.;

CLASS 1:

BYTES TRANSFERRED: 0.701440D+06

NO. OF I/Os: 28.;

CLASS 3:

BYTES TRANSFERRED: 0.116960D+05

NO. OF I/Os: 0.860000D+02

@ 0.0070/IO

81.87

Figure 4-4 Individual Account Summary Report

CLASS 5:

BYTES TRANSFERRED: 0.143798D+06

NO. OF I/Os: 0.140380D+07

@ 0.0070/IO 1006.59

CLASS 6:

BYTES TRANSFERRED: 0.296032D+06

NO. OF I/Os: 0.119900D+04

@ 0.0070/IO 2072.23

DISK USAGE:

IS : SECTOR DAYS 1403804.0 ;

@ 0.0040/SECTOR DAY 5615.22

PIC : SECTOR DAYS 143798.00 ;

@ 0.0040/SECTOR DAY 575.19

TOTAL DISK USAGE 6190.41

TOTAL CHARGES FOR THIS PERIOD 11956.50

MESSAGE MESSAGE MESSAGE MESSAGE.....

J. DOE

Figure 4-4 Individual Account Summary Report (Continued)

This sample report summarizes the amount of time an individual user, John Doe, used the computer. It also shows the cost of user I/Os and prints out the total charges for the period August 1, 1982, through October 10, 1982. This report shows that:

- The operating system occupied the CPU for 5.784 seconds, and the u-task occupied the CPU for 1:1.629 minutes.
- Memory was divided into 28.50kb of system space; 63.75kb of impure memory; and 26.75kb of pure memory.
- There were 1,764. class 0 I/Os, transferring 0.232405D+06 bytes.
- There were 28. class 1 I/Os, transferring 0.701440D+06 bytes.
- There were 0.860000D+02 class 3 I/Os, transferring 0.116960D+05 bytes charged at 0.0070 per I/O for a total of \$81.87.
- There were 0.140380D+07 class 5 I/Os, transferring 0.143798D+06 bytes charged at \$0.0070 per I/O transaction for a total of \$1,006.59.
- There were 0.119900D+04 class 6 I/Os, transferring 0.296032D+06 bytes charged at 0.0070 per I/O transmission for a total of \$2,072.23.
- The charges for the disk usage were 1403804.0 sector days at 0.0040 per sector day for a total of \$5,615.22; 143798.00 sector days at 0.0040 per sector day for a total of \$575.19.
- The total charge for disk use was \$6,190.41.
- The total charges against John Doe for the period are \$11,956.50.

SAVE

4.3.6 SAVE Command

The SAVE command generates archival files from the accounting data specified by the GET and SELECT commands and sends the archival files to the fd specified in the SAVE command.

Format:

SAVE fd [,NEW]

Parameters:

fd	is the file descriptor to which the generated archival file is sent. The specified fd can be a magnetic tape or an indexed file.
NEW	If the specified fd is a magnetic tape device and this parameter is entered, the accounting data is copied to the beginning of the magnetic tape. If the specified fd is a disk file and this parameter is entered, an indexed file with a record length of 1,024 and a block size of 1 (1,024/1) is allocated. The accounting data is then copied to the beginning of the disk file. If this parameter is omitted for magnetic tape or disk devices, the accounting data is appended to the previously created archives on the specified fd.

Functional Details:

If the specified fd is an existing disk file and the NEW parameter is entered, the accounting data is not copied to the specified fd and a message indicating the file cannot be allocated is displayed to the list device.

When the NEW parameter is omitted for disk files, the disk file is positioned so that accounting data can be appended to the file. When the NEW parameter is omitted for magnetic tape files, a double filemark is searched and a backspace filemark operation is performed to position the tape so that accounting data can be appended to the tape.

All accounting data is copied to tape or disk in compressed format, with the first record containing header information. The first record on each magnetic tape volume is an 80-byte volume identifier. The OS/32 Copy conventions for multivolume magnetic tape files apply to archival files stored on magnetic tape.

4.3.7 PAUSE Command

The PAUSE command pauses execution of the Accounting Reporting Utility and returns control to the operating system.

Format:

PAUSE

Functional Details:

The CONTINUE command can be used to continue the Accounting Reporting Utility to resume processing.

END

4.3.8 END Command

The END command terminates execution of the Accounting Reporting Utility.

Format:

END

Functional Details:

When the Accounting Reporting Utility terminates, an end of task code is displayed.

4.4 ACCOUNTING REPORTING UTILITY TASK TERMINATION CODES

There are three possible task termination codes that can be issued at the termination of the Accounting Reporting Utility task.

1. A task termination code of 0 indicates a normal termination.
2. A task termination code of 1 indicates an invalid start option was used (see Section 4.2.2).
3. A task termination code of 2 indicates an error occurred while the task was executing in batch mode.

4.5 ACCOUNTING REPORTING UTILITY MESSAGE SUMMARY

ACCOUNT INACTIVE:n

The account number had no recorded transactions.

ACCOUNT NOT SELECTED:n

An account number was specified in the REPORT command and was not specified in the SELECT command.

COMMAND DEVICE ERROR

The syntax of the command device or file entered as a start option is invalid.

COMMAND NOT RECOGNIZED

An invalid command was entered.

DATE NOT SELECTED

The dates specified in the REPORT command are not within the time period specified in the SELECT command.

DUPLICATE NAME: fd

An ATF, archival filename, or disk volume was entered twice.

DUPLICATE START OPTION

One of the start options was entered more than once.

FILE DESCRIPTOR ERROR: fd

The file descriptor was invalid or was omitted.

FILE ERROR ON: fd 'ASSIGNMENT ERROR'

lu is already assigned or device is offline.

FILE ERROR ON: fd 'BUFFER ERROR'

Insufficient system space for file control block (FCB) and/or buffers.

FILE ERROR ON: fd 'FILE DESCRIPTOR ERROR'

The format of the file descriptor is incorrect

FILE ERROR ON: fd 'I/O ERROR'

An SVC 1 I/O error has occurred while accessing the disk. See the OS/32 Application Level Programmer Reference Manual.

| FILE ERROR ON: fd 'NAME ERROR'

| If trying to allocate or rename, the file name already
| exists on the specified volume. If attempting to assign
| or delete, the named file does not exist on the specified
| volume.

| FILE ERROR ON: fd 'PRIVILEGE ERROR'

| File is already assigned for exclusive access; file may
| be assigned to another task.

| FILE ERROR ON: fd 'PROTECT ERROR'

| Read/write protection keys do not match; invalid
| protection keys.

| FILE ERROR ON: fd 'SIZE ERROR'

| Insufficient space exists on specified volume to allocate
| a file of the specified size.

| FILE ERROR ON: fd 'TYPE ERROR'

| Nondirect access device or device is marked offline.

| FILE ERROR ON: fd 'VOLUME ERROR'

| The volume specified or defaulted in fd does not exist in
| the system; volume specified is not mounted.

INVALID ACCOUNT NUMBER:n

The account number entered was not in the allowable
range.

INVALID ARGUMENT

An argument was invalid or was entered twice.

INVALID DATE:date

The data entered was in an incorrect format or was
omitted. If a 'to' date was an earlier date than the
'from' date, the default dates are reset.

INVALID DATES ON: fd,fd

The dates recorded in the two input files specified by fd,fd contain records with overlapping dates.

INVALID DECIMAL PARAMETER:n

The decimal number entered was incorrect or was omitted.

INVALID FILE OR DEVICE: fd

A file specified in the GET or SAVE commands has an invalid file type or record length or the specified fd is an invalid device.

INVALID RANGE:range

The first account number specified in the range was higher than the second account number (end of range).

INVALID SEPARATOR

One of these required separators was missing or incorrect:

Parentheses	()
Comma	,
Equal sign	=
Quotes	' '
Slash	/

INVALID START OPTION

The command, list, or log device or file entered as a start option is invalid.

INVALID VOLUME NAME:volid

The volid specified in the DISK parameter of the GET command is an invalid volume name.

INVALID REPORT DATES

The report dates are out of chronological order.

INVALID SELECT DATES

The select dates are out of chronological order.

I/O ERROR ON fd 'DEVICE UNAVAILABLE'

The specified fd was unavailable causing the program to pause. To retry the I/O, enter the CONTINUE command.

I/O ERROR ON fd 'END OF FILE'

An end of file condition occurred while processing the specified fd causing the program to pause. To retry the I/O, enter the CONTINUE command.

I/O ERROR ON fd 'END OF MEDIUM'

An end of medium condition occurred while processing the specified fd causing the program to pause. To retry the I/O, enter the CONTINUE command.

I/O ERROR ON fd 'ILLEGAL/UNASSIGNED LU'

An illegal lu was specified, or a required lu was not assigned causing the program to pause. To retry the I/O, enter the CONTINUE command.

I/O ERROR ON fd 'PARITY/RECOVER. ERROR'

A parity or recoverable error occurred on the specified fd causing the program to pause. To retry the I/O, enter the CONTINUE command.

I/O ERROR ON fd 'UNRECOVERABLE ERROR'

An unrecoverable error occurred on the specified fd causing the program to pause. To retry the I/O, enter the CONTINUE command.

LIST DEVICE ERROR

The device or file specified as the list device is invalid.

LIST DEVICE UNASSIGNED

The list device assigned when the program started was closed by the user.

LOG DEVICE ERROR

The device or file specified as the log device is invalid.

MOUNT NEXT TAPE ON fd

An end of volume condition exists on the magnetic tape specified by the fd causing the reporting utility to pause. Mount a new tape and continue the utility with the CONTINUE command.

NO INPUT SPECIFIED

A REPORT or SAVE command was issued before a GET command was specified.

NO MORE AVAILABLE MEMORY: fd

The list of ATFs, archives, or volume names entered in the GET command exceeds the maximum number the program can accept.

SYNTAX ERROR

The syntax of the start options is invalid.

TEXT TOO LONG

The contents specified in a DATE, MESSAGE, SIGNATURE, HEADER, TACCOUNT, TSUMMARY, or TSYSUSE parameter in the OPTION command exceed the maximum allowable length.

VOLUME NOT ON-LINE:volid

A disk volume specified in the GET command is not online to the system causing the reporting utility to pause. Mark the specified disk online and continue the utility with the CONTINUE command.

VOLUME OUT OF SEQUENCE ON: fd

The tape volume containing parts of an archive was mounted in the wrong order.

CHAPTER 5 ERROR REPORTING UTILITY

5.1 GENERAL DESCRIPTION

The Error Reporting Utility (ERROR.TSK) produces reports from the error log information recorded by the hardware error logger and stored on the error recording file by the operating system. These reports contain diagnostic information for memory errors, I/O errors, system errors, and system milestones. The Error Reporting Utility commands allow the operator to:

- specify an error recording file produced by the system error recording routines or a previously created archival file,
- select a subset of the errors to be included in the report by specifying the starting and ending dates of the time period in which those errors occurred,
- build a memory configuration definition (MCD) file that can be used to interpret memory errors,
- output a summary and optionally an itemized list of errors,
- output a memory error report that reports logical addresses as well as physical locations of memory errors, and
- save error logger information on an archival file.

5.2 DESCRIBING MEMORY TO INTERPRET ERRORS

To describe memory to interpret errors, a user must know how their installation's hardware is configured. See your customer service representative for this information.

If error recording is included in the system, the current memory hardware configuration must be described to the operating system through the MCONFIG macro at system generation (sysgen) time. See the System Generation/32 (SYSGEN/32) Reference Manual.

Memory is physically broken into:

- blocks,
- banks, and
- modules.

Memory is also logically broken into storage units.

A block is an area in memory to which a specific number of megabytes (Mbs) is assigned by the user. The smallest block allowed is 1Mb, and the largest block allowed is 16Mb, with the total size of all blocks combined not exceeding 16Mb. Each block must be aligned on a megabyte boundary and must be described starting with block 0 and not exceeding block 15. If shared memory is included in the system, it must be described following local memory. Shared and local memory cannot share the same block. (See Figure 5-1.) For more information on shared memory, see the Perkin-Elmer Models 3220, 3240, and 3250 Processors Shared Memory Systems Installation and Maintenance Manuals.

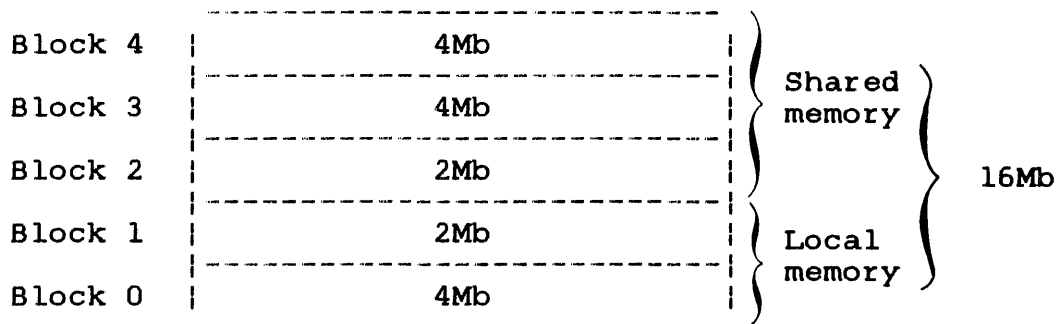


Figure 5-1 Memory Described as Blocks

If interleaving of memory within a block is desired, that block can be broken into smaller areas called banks. A block can be:

- noninterleaving (1 bank),
- 2-way interleaving (2 banks), or
- 4-way interleaving (4 banks).

Each bank within a block is equal in size to one another and has its own memory controller. The blocks shown in Figure 5-1 are further defined in Figure 5-2 to show blocks broken into banks.

	Bank 0	Bank 1	Bank 2	Bank 3	
Block 4	1Mb	1Mb	1Mb	1Mb	4-way interleaving
Block 3	1Mb	1Mb	1Mb	1Mb	4-way interleaving
Block 2	1Mb	1Mb			2-way interleaving
Block 1	1Mb	1Mb			2-way interleaving
Block 0	4Mb				noninterleaved

} 16Mb

Figure 5-2 Memory Described as Blocks and Banks

Banks are physically broken into smaller areas called modules which are also controlled by the memory controller belonging to that bank. There are two general types of modules:

- Single density storage module (SDSTM)
- Double density storage module (DDSTM)

These two types of modules cannot be intermixed on the same machine. Table 5-1 identifies the STMs.

TABLE 5-1 STORAGE MODULE (STM) INFORMATION

FUNCTIONAL VARIATION	STM DEFINITIONS
F00* (256kb)	SDSTM (fully populated with 16k chips)
F01** (256kb)	DDSTM (half populated with 16k chips)
F02** (512kb)	DDSTM (fully populated with 16k chips)
F03** (1024kb or 1Mb)	DDSTM (half populated with 64k chips)
F04** (2048kb or 2Mb)	DDSTM (fully populated with 64k chips)

* Used on Perkin-Elmer Models 3220 and 3240 processors

** Used on Perkin-Elmer Models 3210 and 3230 processors

The size of a bank must equal the total size of all modules within that bank.

Block 0, bank 0 in Figure 5-2 is extracted and magnified in Figure 5-3 to show banks broken into modules. The high density modules occupy low-address space.

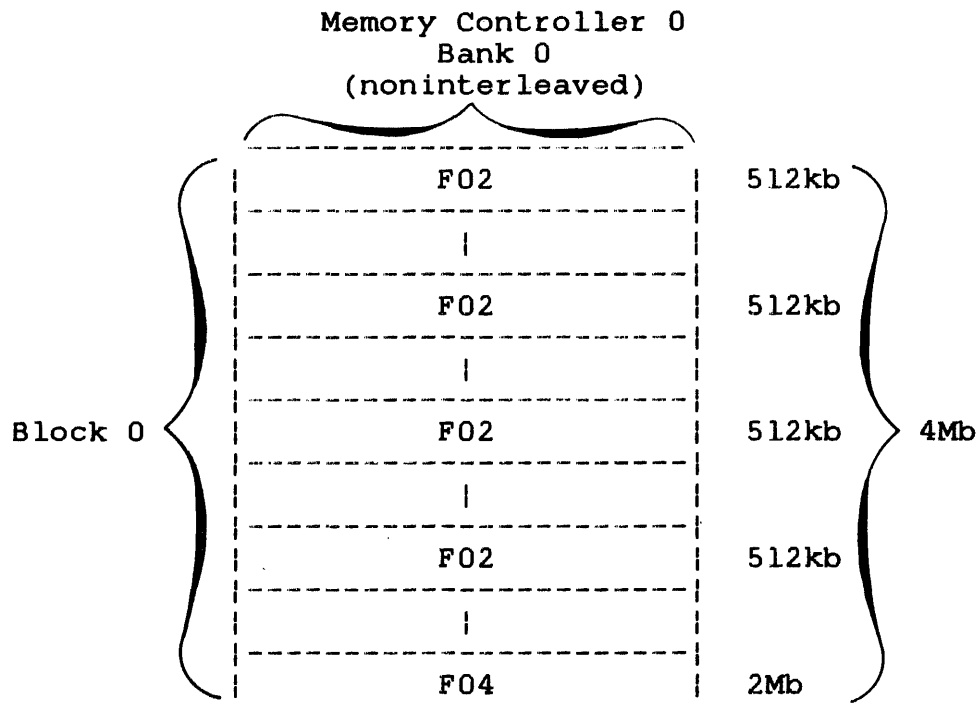


Figure 5-3 A Bank Broken into Modules

Banks are also logically broken into smaller areas called storage units, which are determined by the size of the bank.

If the bank size within a block is less than or equal to 4Mb, those banks are broken into 256kb storage units. The number of storage units can range from 1 through 16, depending on the size of the bank. If the bank size within a block is greater than 4Mb, those banks are broken into 1Mb storage units. The number of storage units can range from 5 through 16, depending on the size of the bank. See Figure 5-4.

Memory Controller 0
 Bank 0
 (noninterleaved)

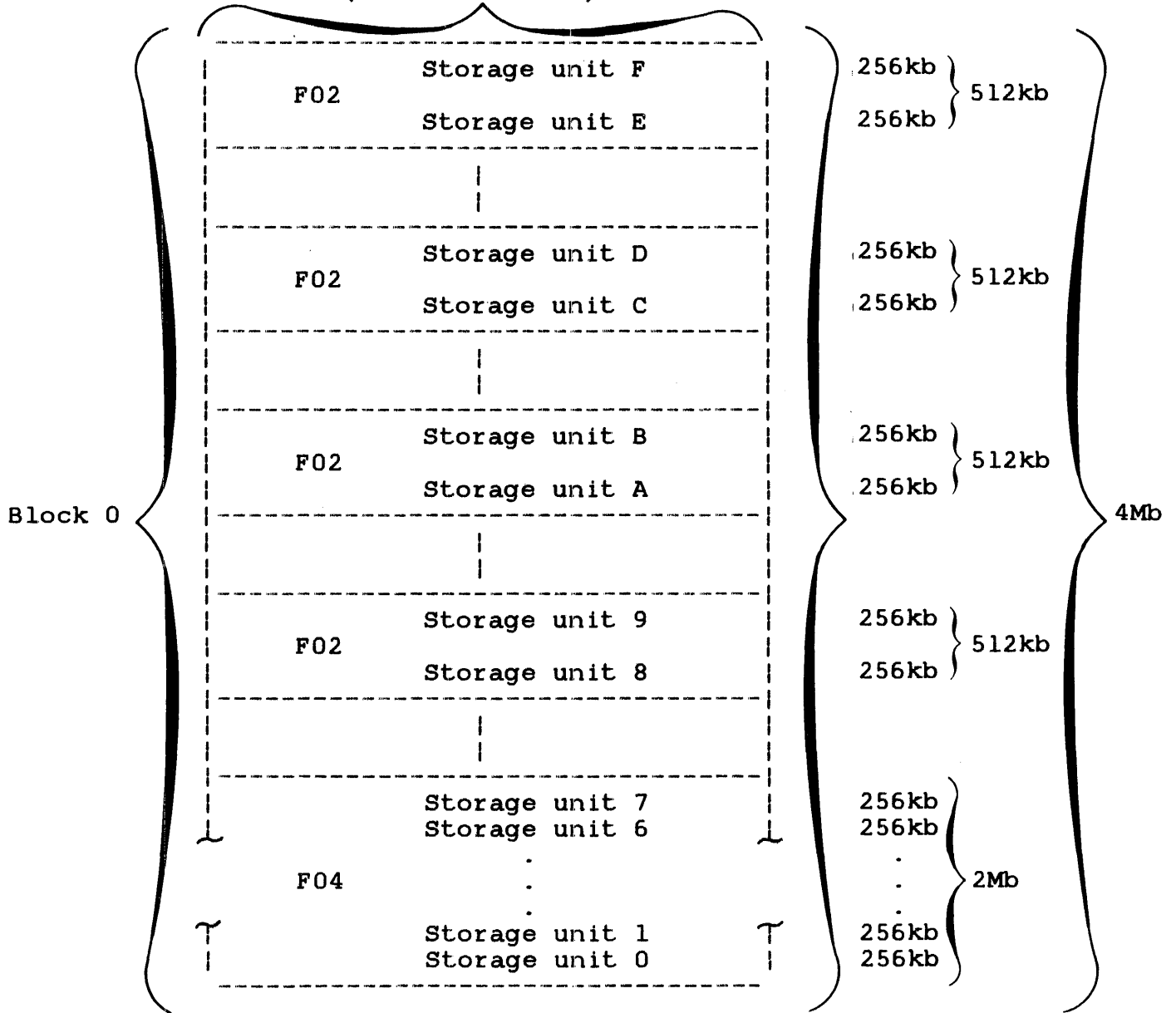


Figure 5-4 A Bank Broken into Storage Units

5.3 LOADING AND STARTING THE ERROR REPORTING UTILITY

To load and start the Error Reporting Utility, enter the following commands:

Formats:

LOAD ERROR

START [, [COMMAND=fd₁] [, LOG=fd₂]]

Parameters:

COMMAND= fd₁ specifies the file descriptor of the command input device. If this parameter is omitted, the console device (CON:) is the default. If fd₁ is not an interactive device, a log device must also be specified.

LOG= fd₂ specifies the file descriptor of the log device. If the command input device is not interactive, this parameter must be specified.

Messages:

COMMAND DEVICE ERROR

indicates that the specified command device could not be assigned.

INVALID STARTING PARAMETER

indicates that a starting parameter other than COMMAND= or LOG= was specified.

LOG DEVICE ERROR

indicates that the specified log device could not be assigned.

5.4 SPECIFYING THE INPUT FILE

The GET command identifies the file that contains the error records from which an error report is generated. This file is created when the system operator enables the Error Recording facility on the system. See the OS/32 Operator Reference Manual.

Format:

GET fd

Parameter:

fd is the file descriptor of the file used by the system error recording routines to store error records. The specified fd is assigned exclusive read only (ERO) privileges.

Functional Details:

The GET command must be specified before any other error reporting commands are specified. The specified file must not currently be in use for system error recording. The optimum procedure is to turn the error recording facility off for the specified file; then turn the error recording facility on with a new fd specification. The previous fd is now available for use by the Error Reporting Utility. It is possible to specify a tape file in the GET command. If the tape file spans more than one magnetic tape volume, the following message is displayed for each additional tape needed:

MOUNT NEXT TAPE

The task pauses, allowing the operator to mount the next tape. After mounting the tape, the operator continues execution by entering the CONTINUE command. The request to mount a new tape occurs after the Error Reporting Utility has begun processing the error file entries as requested via a REPORT command.

5.5 DEFINING MEMORY CONFIGURATION

The DEFINE command allows a user to define system memory configuration. This enables the Error Reporting Utility to identify the location of memory errors. With this command, the user can:

- initially create a memory configuration definition (MCD) file for the system via a sequence of prompts issued by the Error Reporting Utility;
- display the memory configuration of the system, as defined by a previously created MCD file, to a log or list device, or
- specify an MCD file that the Error Reporting Utility is to use to interpret and locate memory errors.

Format:

DEFINE fd [{ CONFIG }
{ INTERPRET }]

Parameters:

fd is the file descriptor of the MCD to be created or, of a previously created MCD. If the CONFIG or INTERPRET parameters are omitted, the Error Reporting Utility issues a series of prompts to allow the user to create a new MCD file.

CONFIG specifies that the content of the MCD file specified is to be displayed to the appropriate device. No prompt sequence is issued.

INTERPRET specifies that the data in the specified file is used to interpret subsequent memory errors. No prompt sequence is issued and the specified MCD file must exist on disk.

Functional Details:

When the DEFINE command is issued without the CONFIG or INTERPRET parameters, a prompt sequence is issued. Explanations of each prompt and valid responses are provided. When the user enters an invalid response, the error is identified and the prompt sequence returns to the prompt preceding the error. The prompt sequence must be restarted if the user enters incorrect information, such as the wrong STM type since the automatic recovery procedure will not consider this an invalid entry. The prompt sequence consists of the following prompts:

- Prompt: HOW MANY BLOCKS?

Response: n

n is a decimal number from 1 through 16. A block consists of from 1 to 16Mb and must be aligned on a megabyte boundary. When the response is entered, the program labels the blocks beginning with block 0.

NOTES

1. Unless otherwise specified via an MCD file, the Error Reporting Utility will assume that the memory is configured using F01 STMs as the default and will identify the location of memory errors according to the layout of that STM. If a system contains STMs of a type other than F01, the resulting error locations may be misleading. It is, therefore, advisable to specify an MCD before reporting memory errors if there is any doubt about the type of STMs in use.

2. When defining memory configuration, total memory must end on a full megabyte boundary. Therefore, when actual total memory does not end on a full megabyte boundary; e.g., 2 1/4Mb, 8 1/2Mb, it is necessary to raise the total memory size to the next full megabyte value above the actual memory size before dividing into blocks. For example:
 - 2 1/4Mb becomes 3Mb
 - 8 1/2Mb becomes 9Mb

This is necessary for the following two reasons:

1. The memory configuration procedure does not allow the entry of block sizes of less than a whole megabyte.

2. For the purpose of error reporting, all memory must be represented as contiguous. If a system has local and shared memory, and local memory does not end on a full megabyte boundary, there will be a gap between the end of local memory and the start of shared memory. Shared memory always begins on a full megabyte boundary.

Therefore, to insure that no gap is present between local and shared memory, local memory should be raised to the next full megabyte value and the appropriate module types added to account for the size of the gap.

Error Message: MEMORY EXCEEDS 16Mb

Recovery: The program reissues the prompt.

- Prompt: HOW MANY BANKS IN BLOCK n?

Response: $\left\{ \begin{array}{l} 1 \\ 2 \\ 4 \end{array} \right\}$

Banks are contained within blocks and are of equal length to each other. A block has more than one bank only if it is interleaved. The valid responses have the following meanings:

- 1 = noninterleaving
- 2 = 2-way interleaving
- 4 = 4-way interleaving

Error Message: NUMBER OF BANKS INVALID

Recovery: The program reissues the prompt.

- Prompt: HOW MANY MODULES IN BANK n?

Response: n

n is a decimal number from 1 through 64.

Error Message: INVALID NUMBER OF MODULES

Recovery: The program reissues the prompt.

- Prompt: ENTER MODULE TYPES

Response: module type1 ,module type2,...,module typen

Table 5-2 shows the variable module types and the memory and chip size of each.

TABLE 5-2 MODULE TYPES AND SIZES

MODULE TYPE	SIZE OF MEMORY	CAPACITY OF CHIPS	DENSITY
F00	256kb	16k	Single
F01	256kb	16k	Double
F02	512kb	16k	Double
F03	1024kb or 1Mb	64k	Double
F04	2048kb or 2Mb	64k	Double

WARNING

DO NOT ATTEMPT TO MIX SDSTMS AND DDSTMS.
 THE ERROR REPORTING UTILITY WILL NOT
 ACCEPT THEM AND WILL ISSUE THE FOLLOWING
 ERROR MESSAGE:

PRECEDING MODULE INCOMPATIBLE

The module types must be entered in ascending order. The total size of all modules entered must not exceed 16Mb.

Error Message: INVALID MODULE TYPE

Recovery: The program reissues the prompt.

Error Message: TOO MANY MODULES

Recovery: The program reissues the prompt; the previous entries are retained.

Error Message: PRECEDING MODULE INCOMPATIBLE

Recovery: The program restarts the prompts at the module definition sequence.

Error Message: CONFLICTING BANK LENGTH IN BLOCK

Recovery: The program reissues the prompt starting at the definition of the block.

Error Message: BLOCK NOT ON MB BOUNDARY

Recovery: The program reissues the prompt starting at the definition of the block.

When specifying the MCD file, new commands cannot be entered until all prompts are answered. To return control to the command level, enter the following word:

REDEFINE

The following is a sample prompt session. It designates MEMDEF.FIL as the fd of the memory configuration file to be built. The defined memory has two blocks, and both blocks have two banks. An error occurs when BLOCK 1, BANK 0 is being defined. The user enters too many modules. The prompt sequence recovers by reissuing the request to redefine BLOCK 1, BANK 0.

Example:

```
DEF MEMDEF.FIL

HOW MANY BLOCKS?
2
HOW MANY BANKS IN BLOCK 0?
2
HOW MANY MODULES IN BANK 0?
2
ENTER MODULE TYPES
F02, F02
HOW MANY MODULES IN BANK 1?
2
ENTER MODULE TYPES
F02
ENTER MODULE TYPES
F02
HOW MANY BANKS IN BLOCK 1?
2
HOW MANY MODULES IN BANK 0?
4
ENTER MODULE TYPES
F01,F01,F01,F01,F01
TOO MANY MODULES
ENTER MODULE TYPES
F01,F01,F01,F01
HOW MANY MODULES IN BANK 1?
4
ENTER MODULE TYPES
F01,F01,F01,F01
```

The following example displays the configuration data from the file MEMDEF.FIL.

Example:

```
DEF MEMDEF.FIL,CONFIG

MEMORY CONFIGURATION DEFINITION ON MEMDEF.FIL

BLOCK 0      2 WAY INTERLEAVED
  BANK 0
    F02
    F02
  BANK 1
    F02
    F02
BLOCK 1      2 WAY INTERLEAVED
  BANK 0
    F01
    F01
    F01
    F01
  BANK 1
    F01
    F01
    F01
    F01
```

The following command notifies the Error Reporting Utility that file MEMDEF.FIL is to be used to interpret memory errors during report generation:

Example:

```
DEFINE MEMDEF.FIL,INTERPRET
```

5.6 SELECTING SPECIFIC DATA

The SELECT command selects by time of occurrence the errors to report or save to an archival file.

Format:

```
SELECT [ FROM= { mon/dd/yy } [ , hh [ :mm [ :ss ] ] ] ]
      [ , TO= { mon/dd/yy } [ , hh [ :mm [ :ss ] ] ] ]
```

Parameters:

mon	is the name of the month and can be abbreviated to three alphabetic characters.
dd	is the decimal number from 1 through 31 specifying the day of the month.
yy	is a decimal number from 00 through 99 specifying the year.
hh	is a decimal number from 00 through 23 specifying the hours.
mm	is a decimal number from 00 through 59 specifying the minutes.
ss	is a decimal number from 00 through 59 specifying the seconds.

Functional Details:

The first date format is valid only if the U.S. format option was selected at sysgen. The second date format is valid only if the European format option was selected at sysgen.

The date and time in the FROM parameter specify the beginning of the desired interval; the date and time in the TO parameter specify the end of the desired interval. The errors that occurred within the specified interval are then used to generate reports and can be saved to an archival file.

If the FROM parameter is omitted, the interval begins with the date and time of the earliest entry in the error file. If the date is given but the time is omitted, the time defaults to 00:00:00. If the TO parameter is omitted, the interval ends with the date and time of the latest entry in the error file. If the date is given but the time is omitted, the time defaults to 23:59:59.

If the time is partially specified in both the FROM and TO parameters, the defaults for minutes and seconds are 0. If SELECT is not entered, all errors in the error file are available for use by the REPORT or SAVE command.

5.7 GENERATING AN ERROR REPORT

The REPORT command outputs error reports.

Format:

```
REPORT fd [,LIST]
```

Parameters:

fd	specifies the device to which all selected error information in the error recording file is output. Error information is automatically output in the summary format. See Section 5.7.2 for a description of the summary format.
LIST	specifies that all selected errors are output in a list format, in addition to being included in the summary report. See Section 5.7.1 for a description of the list format.

5.7.1 List Reports

The categories of the list reports are:

- Bulk device input/output (I/O) error report
- File manager detected error reports
- System detected error reports
- System milestone reports
- Memory errors

All of these reports are produced when a REPORT command is issued, provided at least one error of each type is present in the error file being reported. If no errors of a particular type occurred during the reporting period, then that category of report is not generated.

The following examples show formats for list and summary reports.

5.7.1.1 Bulk Device Input/Output (I/O) Error Report

Format:

6688

```
mon/dd/yy   hh:mm:ss   BULK DEVICE I/O ERROR
DEVICE      : ADDRESS  = bbbbbbbb ; STATUS = ssss NAME =xxxx
CONTROLLER : ADDRESS  = cccccccc ; STATUS = nnnn
SELCH       : ADDRESS  = dddddddd ; STATUS = mmmm
SECTOR = rrrrrrrr HEAD = kkkkkkkk CYLINDER = yyyyyyyy
DRIVER ENTRY COUNTER = vvvvvvvv SVC FC=gg; USER STATUS=pppp
BUFF START = aaaa BUFF END = bbbb SELCH END = cccc
CURRENT ESR = qqqqqqqq CURRENT INTERRUPT HANDLER = hhhhhhhh
MMDLOG LINK = zzzzzzzz INFO/SECONDARY LINK = iiiiii
```

Fields:

DEVICE information includes:

ADDRESS = bbbbbbbb is the hexadecimal address of the device.

STATUS = ssss is a hexadecimal number indicating the status of the device.

NAME = xxxx is the device name.

CONTROLLER information includes:

ADDRESS = cccccccc is the hexadecimal address of the controller.

STATUS = nnnn is a hexadecimal number indicating the status of the controller.

SELCH information includes:

ADDRESS = dddddddd is the hexadecimal address of the selector channel (SELCH).

STATUS = mmmm is a hexadecimal number indicating the status of the SELCH.

SECTOR = rrrrrrrr is the hexadecimal address of the sector where the error occurred.

HEAD = kkkkkkkk is the hexadecimal address of the track where the error occurred.

CYLINDER = yyyyyyyy is the hexadecimal address of the cylinder where the error occurred.

DRIVER ENTRY COUNTER = vvvvvvvv is a hexadecimal number indicating the sequence number of the I/O request.

SVC FC = gg is the supervisor call function code that was used to send the specific type of I/O.

USER STATUS = pppp is the SVC status returned to the user.

BUFF START = aaaa is the hexadecimal address of the start of transfer.

BUFF END = bbbb is the hexadecimal address of the planned end of transfer.

SELCH END = cccc is the hexadecimal address of the actual end address of transfer.

CURRENT ESR = qqqqqqqq is the address of the event service routine executing I/O at the time the error occurred.

CURRENT INTERRUPT HANDLER = hhhhhhhh is the hexadecimal address of the interrupt handler.

MMDLOG LINK = zzzzzzzz is a hexadecimal number of the location in the OS/32 driver that logged the entry indicating the precise driver activity at the time the error was detected.

INFO/SECONDARY LINK = iiiiiiiii is a hexadecimal number indicating the location in the OS/32 driver of the higher level subroutine call further clarifying driver activity at the time the error was detected. This field also can contain additional information provided by the primary call, MMDLOG LINK.

5.7.1.2 File Manager Detected Errors

The following are five types of reports generated when the file manager detects an error.

Report Type 1 - Bit map inconsistency when allocating an already allocated disk sector.

Format:

6689

```
mon/dd/yy hh/mm/ss BIT MAP ERROR VOL = voln SECTOR = nnnnnnnn
TYPE = type ; START SECTOR = xxxxxxxx LENGTH = yyyy
```

Report Type 2 - Bit map inconsistency when deallocating an already deallocated disk sector.

Format:

6690

```
mon/dd/yy hh:mm:ss BIT MAP ERROR VOL = voln SECTOR = nnnnnnnn
TYPE = TYPE ; START SECTOR = xxxxxxxx LENGTH = yyyy
```

Fields:

VOL = voln is the name of the volume where the error occurred.

SECTOR = nnnnnnnn is a hexadecimal number indicating the number of the physical sector where the error occurred.

TYPE = TYPE indicates whether the disk sector is being allocated or deallocated.

START SECTOR = xxxxxxxx is a hexadecimal number indicating the starting sector number of the block to be allocated or deallocated.

LENGTH = yyyy is a hexadecimal number indicating the number of sectors to be allocated or deallocated.

Report Type 3 - Directory block pointer to next block beyond disk range.

Format:

6691

```
mon/dd/yy hh:mm:ss DIRECTORY BLOCK ERROR ON voln
SECTOR = nnnnnnnn POINTER = yyyyyyyy
```

Fields:

voln is the name of the volume on which the error occurred.

SECTOR = nnnnnnnn is a hexadecimal number indicating the number of the physical sector where the error occurred.

POINTER = yyyyyyyy is the invalid hexadecimal number pointing to the next directory sector.

Report Type 4 - Indexed block pointer to next or previous block or index data sector beyond disk range.

Format:

6692

```
mon/dd/yy hh:mm:ss INDEX/DATA BLOCK ERROR IN voln:fd/actno
SECTOR = nnnnnnnn POINTER = yyyyyyyy
```

Report Type 5 - Directory entry for first and/or last logical block beyond disk range.

Format:

6693

```
mon/dd/yy hh:mm:ss DIRECTORY ENTRY ERROR FOR voln:fd/actno
SECTOR = nnnnnnnn POINTER = yyyyyyyy
```

Fields:

voln is the name of the volume on which the error occurred.

fd/actno is the filename and account number of the file with illegal sector number.

SECTOR = nnnnnnnn is a hexadecimal number indicating the number of the physical sector where the error occurred.

POINTER = yyyyyyyy is an invalid hexadecimal number pointing to the next sector, previous sector, index data sector, or first or last logical block number.

5.7.1.3 System Detected Errors

The following six report types are generated when the system detects an error.

Report Type 1 - Task queue overflow.

Format:

6694

```
mon/dd/yy hh:mm:ss TASK QUEUE OVERFLOW FOR taskname
TCB TASK STATUS = nnnnnnnn TSW = xxxxxxxx CODE = yyyyyyyy
```

Fields:

taskname is the name of the task for which the system was processing when the error occurred.

TCB TASK STATUS = nnnnnnnn is the task status in the task control block when the overflow occurred.

TSW = xxxxxxxx is the task status word indicating the state of the task when the overflow occurred.

CODE = yyyyyyyy is the fullword reason code of the entry that was to be added to the specified task queue.

Report Type 2 - System queue full; might overflow.

Format:

6695

```
mon/dd/yy hh:mm:ss SYSTEM QUEUE FULL - POSSIBLE OVERFLOW
```

Report Type 3 - System queue service handler processes leaf with unspecified system queue service (SQS) routine.

Format:

6696

```
mon/dd/yy hh:mm:ss UNSPECIFIED SQS ROUTINE; SERVICE FOR taskname
LEAF ADDRESS = xxxxxxxx DCB ADDRESS = yyyyyyyy
```

Report Type 4 - SQS handler processes leaf for device driver with unspecified event service routine (ESR).

Format:

6697

```
mon/dd/yy hh:mm:ss UNSPECIFIED ESR ROUTINE; SERVICE FOR taskname  
LEAF ADDRESS = xxxxxxxx DCB ADDRESS = YYYYYYYY
```

Fields:

taskname is the name of the task for which the system was processing at the time the error occurred. This may be blank.

LEAF ADDRESS = xxxxxxxx is the address of the leaf which was queued with the unspecified routine address.

DCB ADDRESS = yyyyyyyy is the address of the device control block connected to the leaf.

Report Type 5 - A level 0 interrupt was ignored.

Format:

6698

```
mon/dd/yy/ hh:mm:ss LEVEL 0 INTERRUPT IGNORED  
PSW STATUS = xxxxxxxx PSW LOCATION = YYYYYYYY
```

Fields:

PSW STATUS = xxxxxxxx is the portion of the program status word indicating the status when the level 0 interrupt occurred.

PSW LOCATION = yyyyyyyy is the portion of the program status word indicating the location at which the level 0 interrupt occurred.

The following report type is supported by the R07.1 software release and higher. It is only applicable to the operation of the Model 3200MPS System.

Report Type 6 - APU signal queue full, possibility of overflow.

Format:

6699

```
mon/dd/yy hh:mm:ss APU SIGNAL QUEUE FULL - POSSIBLE OVERFLOW
```

5.7.1.4 System Milestones

| The following are six types of reports generated when a system
| milestone occurs:

Report Type 1A - Marking on a bulk device.

Format:

6700

```
mon/dd/yy hh:mm:ss DEVICE xxxx MARKED ON ; VOLUME NAME = yyyy  
BLOCKSIZE = nnnn EXPANSION = xxxx  
-WRITE PROTECT - DIRECTORY -SYSTEM
```

Fields:

DEVICE xxxx is the mnemonic of the device being
 marked online.

VOLUME NAME = yyyy is the volume name of the device
 involved.

BLOCKSIZE = nnnn is the blocksize specified or defaulted
 to when marking a nonbulk device online.

EXPANSION = xxxx is the expansion specified or defaulted
 to when marking a bulk device online.

Report Type 1B - Marking on a nonbulk device.

Format:

6701

```
mon/dd/yy hh:mm:ss DEVICE xxxx MARKED ON; DEVICE NUMBER = YYYYYYYY
```

Fields:

DEVICE xxxx is the mnemonic of the device being
 marked online.

DEVICE NUMBER yyyyyyyy is a decimal number specifying the
 device being marked online.

Report Type 2A - Marking off a bulk device.

Format:

6702

```
mon/dd/yy hh:mm:ss DEVICE xxxx MARKED OFF; VOLUME NAME = yyyy  
TOTAL I/O = nnnnnnnn TOTAL ERRORS = mmmmmmmmm
```

Fields:

DEVICE xxxx is the mnemonic of the device being marked offline.

VOLUME NAME = yyyy is the volume name of the device involved.

TOTAL I/O = nnnnnnnn is the total number of times the driver was entered before the device was marked offline.

TOTAL ERRORS = mmmmmmmmm is a decimal number specifying the total number of errors that occurred on the device before it was marked offline.

Report Type 2B - Marking off a nonbulk device.

Format:

6703

```
mon/dd/yy hh:mm:ss DEVICE xxxx MARKED OFF; DEVICE NUMBER = YYYYYYYY
```

Fields:

DEVICE xxxx is the mnemonic of the device being marked offline.

DEVICE NUMBER = YYYYYYYY is a decimal number specifying the device being marked offline.

Report Type 3 - Volume change for system, spool, roll, or temporary volume.

Format:

6704

```
mon/dd/yy hh:mm:ss xxxx DESIGNATED AS yyyy VOLUME
```

Fields:

xxxx xxxx is the volume name of the device.

yyyy yyyy is either temp, roll, spool, or system volume.

Report Type 4 - No system space for error recording; third allocated system buffer full; error recording file not allocated; I/O error when writing to error recording disk file.

Format:

```

6705                    mon/dd/yy hh:mm:ss ERROR RECORDING RESUMED
                         TOTAL ERRORS LOST                    bbbbbbbb
                         MEMORY ERRORS                        dddddddd
                         BULK DEVICE I/O ERRORS                mmmmmmmm
                         FILE MANAGER DETECTED ERRORS           nnnnnnnn
                         SYSTEM DETECTED ERRORS                xxxxxxxx
                         SYSTEM MILESTONES                      yyyyyyyy

```

Fields:

TOTAL ERRORS LOST bbbbbbbb is a decimal number specifying the total number of errors not recorded due to any of the events mentioned in the definition of Report Type 4. The totals presented below represent total numbers including the errors lost.

MEMORY ERRORS dddddddd is the total number of memory errors.

BULK DEVICE I/O ERRORS mmmmmmmm is the total number of device I/O errors.

FILE MANAGER DETECTED ERRORS nnnnnnnn is the total number of file manager detected errors.

SYSTEM DETECTED ERRORS xxxxxxxx is the total number of system detected errors.

SYSTEM MILESTONES yyyyyyyy is the total number of system milestones.

Report Type 5 - System queue reached its depth.

Format:

6706

```
mon/dd/yy hh:mm:ss SYSTEM QUEUE REACHED DEPTH n
```

Fields:

SYSTEM QUEUE REACHED DEPTH n is the depth of the system queue at the time of error recording. An initial depth of 5 used and is updated if it is exceeded.

The following report type, which is generated when a system milestone occurs, is supported by the R07.1 software release and higher. It is only applicable to the operation of the Model 3200MPS System.

Report Type 6 - APU signal queue reached its depth.

Format:

6707

```
mon/dd/yy/ hh:mm:ss APU SIGNAL QUEUE REACHED DEPTH n
```

Fields:

APU SIGNAL QUEUE REACHED DEPTH n is the depth of the APU signal queue at the time of error recording. An initial depth of 11 is used and is updated if it is exceeded.

5.7.1.5 Memory Errors

The following is the memory report generated when a memory error occurs. The module data is omitted if a memory configuration definition file was not specified.

Format:

6708

```
mon/dd/yy hh:mm:ss MEMORY ERROR BLOCK=mS, BANK=n, SU=x,  
CHIP=ccc-ch MODULE y-zzz
```

Fields:

BLOCK	m is the block number in hexadecimal (0-F) where the error occurred. S is included if the block is shared memory.
BANK	n is the bank number (0-3) where the error occurred.
SU	x is the storage unit in memory where the error occurred.
CHIP	ccc is the alphanumeric column identifier where the error occurred. This identifier is identical to those printed on each memory board (SDSTM - A or B; DDSTMS - AA, AB, AC, or AD). ch is the chip number (0-38) where the error occurred. This field can also contain asterisks (**) indicating a multiple error occurred. Individual errors follow immediately on the report.
MODULE	y is the module count using the order in which modules were defined in the bank. zzz is the functional variation. (F01, F02, F03, F04).

5.7.2 Summary Reports

There are three summary reports:

1. Summary of system errors and milestones report
2. Device errors summary report
3. Memory errors summary report

5.7.2.1 Summary of System Errors and Milestones Report

Error types itemized by taskname will be itemized for a maximum of nine tasknames. All additional errors of the same type, but different tasknames, are listed under the category ALL OTHER TASKS.

Format:

6709

SYSTEM QUEUE FULL - POSSIBLE OVERFLOW	bbb
APU SIGNAL QUEUE FULL - POSSIBLE OVERFLOW	ccc
ERROR RECORDING INTERRUPTED	ddd
MAXIMUM SYSTEM QUEUE DEPTH REACHED	ggg
MAXIMUM APU SIGNAL QUEUE DEPTH REACHED	hhh
TASK QUEUE OVERFLOWS	
taskname 1	nnnnnn
taskname 2	nnnnnn
taskname 3	nnnnnn
.	
taskname 9	nnnnnn
ALL OTHER TASKS	YYYYYY
UNSPECIFIED SQS ROUTINE	
taskname 1	nnnnnn
taskname 2	nnnnnn
taskname 3	nnnnnn
.	
taskname 9	nnnnnn
ALL OTHER TASKS	YYYYYY
UNSPECIFIED ESR ROUTINE	
taskname 1	nnnnnn
taskname 2	nnnnnn
taskname 3	nnnnnn
.	
taskname 9	nnnnnn
ALL OTHER TASKS	YYYYYY

Fields:

SYSTEM QUEUE
FULL-
POSSIBLE
OVERFLOW

bbb is a decimal number indicating the total number of times the system queue was full.

| APU SIGNAL
| QUEUE FULL-
| POSSIBLE
| OVERFLOW

ccc is a decimal number indicating the total number of times the APU signal queue was full.

ERROR
RECORDING
INTERRUPTED

ddd is a decimal number indicating the total number of times error recording was interrupted.

MAXIMUM SYSTEM
QUEUE DEPTH
REACHED

ggg is a decimal number indicating the maximum system queue depth reached.

| MAXIMUM APU
| SIGNAL QUEUE
| DEPTH REACHED

hhh is a decimal number indicating the maximum APU signal queue depth reached.

taskname1...
taskname9

nnnnnn is a decimal number indicating the total errors.

ALL OTHER
TASKS

yyyyyy is a decimal number indicating the total number of errors for all tasks not reported individually for each type of system error.

5.7.2.2 Summary of Device Errors

Device error summaries are itemized for a maximum of nine devices and nine device addresses. Additional errors are included under the category ALL OTHER DEVICES. Additional I/O errors are included under the category ALL OTHER I/O ERRORS.

Example:

6710

```
DEVICE device name1
  BIT MAP ALLOCATION ERRORS          nnnnn
  BIT MAP DEALLOCATE ERRORS         nnnnn
  DIRECTORY BLOCK ERRORS            nnnnn
  DIRECTORY ENTRY ERRORS            nnnnn
  INDEX/DATA BLOCK ERRORS           nnnnn

DEVICE device name9
  BIT MAP ALLOCATION ERRORS          nnnnn
  BIT MAP DEALLOCATE ERRORS         nnnnn
  DIRECTORY BLOCK ERRORS            nnnnn
  DIRECTORY FILE ERRORS             nnnnn
  INDEX/DATA BLOCK ERRORS           nnnnn
  .
  .
  .

ALL OTHER DEVICES
  BIT MAP ALLOCATION ERRORS          mmmmm
  BIT MAP DEALLOCATE ERRORS         mmmmm
  DIRECTORY BLOCK ERRORS            mmmmm
  DIRECTORY ENTRY ERRORS            mmmmm
  INDEX/DATA BLOCK ERRORS           mmmmm

I/O ERRORS
  DEVICE NAME      ADDRESS      I/O ERRORS
  xxxx            YYYYYYYY      bbbbbbbb
  xxxx            YYYYYYYY      bbbbbbbb
  xxxx            YYYYYYYY      bbbbbbbb
  .                .                .
  .                .                .
  .                .                .
  xxxx            YYYYYYYY      88888888
  ALL OTHER I/O ERRORS
```

Fields:

device name1... are the names of up to nine devices on
device name9 which the errors occurred.

nnnnn is a decimal number indicating the total
number of errors occurring in five
categories for the specific device.

ALL OTHER DEVICES mmmm is a decimal number indicating the total number of errors occurring on additional devices.

DEVICE NAME xxxx are the names of up to nine devices on which the I/O errors occurred.

ADDRESS yyyyyyyy is the address of the device where the error occurred.

I/O ERRORS bbbbbbbb is a decimal number indicating the total number of I/O errors occurring on the specific device.

ALL OTHER I/O ERRORS ssssssss is a decimal number indicating total number of additional I/O errors.

5.7.2.3 Memory Errors Summary Report

Memory errors are broken down by module and functional variation, only if an MCD file was specified.

Example:

6711

BLOCK	BANK	STORAGE UNIT	PHYSICAL MODULE	CHIP	NUMBER OF FAILURES
mS	n	x	x-Fnn	ccc-ch	nnnn
.
.
.

Fields:

BLOCK m is the block number in hexadecimal (0-F) where the error occurred. S is included if the block is shared memory.

BANK n is the bank (0-3) where the error occurred.

STORAGE UNIT x is the storage unit in memory where the error occurred. This information is not displayed if module information was defined in an MCD file.

PHYSICAL MODULE x is the physical module number within the bank where the error occurred.

 Fnn is the functional variation; e.g., F01, F02, F03, F04, of the module where the error occurred.

CHIP ccc is the alphanumeric column identifier where the error occurred. This identifier is identical to those printed on each memory board (SDSTM - A or B; DDSTM - AA, AB, AC, or AD). An M preceding the column id indicates multiple errors.

ch is the chip number (0-38) where the error occurred. This field also can contain asterisks (**) indicating a multiple error. Individual errors follow immediately on the report.

NUMBER nnnn is a decimal number indicating the total
OF FAILURES number of memory errors occurring within the
 blocks displayed.

5.8 SAVING INFORMATION TO AN ARCHIVAL FILE

The SAVE command saves the error report to an archival file.

Format:

SAVE fd [,NEW]

Parameters:

fd is the file descriptor of the device to which the report is saved. This fd must be specified as a magnetic tape device or an indexed file. The report is appended to existing data in the specified fd unless the NEW parameter is specified.

NEW specifies that the report is to be saved at the beginning of the file or tape.

Functional Details:

If NEW is not specified and if a magnetic tape device is specified as the fd, the selected error information is appended to the existing archive by writing over the first filemark encountered on the tape. When appending to a multiple volume file, only the last tape should be mounted. If an indexed file is specified as the fd, the selected information is appended to the file.

If NEW is specified and an indexed file is specified, the indexed file is allocated with 256-byte records and an index block size and data block size of 1. For both magnetic tape and indexed files, header information is written to the first record in the archival file.

If the specified fd is a magnetic tape and multiple volumes are required, an end of tape label is written and the message MOUNT NEXT TAPE is output to the console. The task pauses, allowing the operator to mount the next tape. After mounting the next tape, the operator continues execution by entering the command CONTINUE. Multiple tape handling formats are compatible with OS/32 Copy multiple tape processing.

5.9 PAUSING A TASK

The PAUSE command pauses execution of the Error Reporting Utility and returns control to the operating system.

Format:

PAUSE

5.10 ENDING A TASK

The END command terminates execution of the Error Reporting Utility.

Format:

END

5.11 MESSAGES

CANNOT ALLOCATE fd

indicates that allocation of the file named in a SAVE or DEFINE command failed because the volume was incorrect or protected.

CANNOT ASSIGN fd

indicates that requested access privileges could not be granted or the file had non-zero protection keys.

fd NOT FOUND

indicates that the specified file does not exist.

fd NOT INDEXED FILE OR MAGNETIC TAPE

indicates that the argument of a SAVE command is not an indexed file or a magnetic tape.

FILE fd ALREADY EXISTS

indicates that a file named in a SAVE command as NEW already exists,

or

indicates that the file named in the DEFINE command entered already exists.

INVALID ARGUMENT SEPARATOR

indicates that the second argument was not preceded by a comma.

INVALID COMMAND

indicates that the last command read was not recognized as a valid command.

INVALID HEADER ON fd

indicates that pointers in the header do not point to valid records or are not zero.

INVALID KEYWORD

indicates that an argument of the last command was not a valid keyword for that command or was used more than once in the same command.

INVALID OR MISSING DATA

indicates that the argument of a FROM or TO parameter in the last SELECT command was invalid or missing.

INVALID OR MISSING FILE DESCRIPTOR

indicates that the first parameter entered was not valid.

INVALID TIME INTERVAL

indicates that the last FROM date is later than the last TO date at the time of a REPORT or SAVE command.

NO INPUT FILE

indicates that a SELECT, REPORT, or SAVE command cannot be processed because a previous GET command was not issued.

NOT ENOUGH SYSTEM SPACE

indicates that during allocation or assignment, the task exceeded its system space limit or there was not enough system space available.

MEMORY ERROR/MCD CONFLICT

indicates a memory error that does not fit the defined memory configuration is encountered.

CHAPTER 6 DISK DUMP UTILITY

6.1 GENERAL DESCRIPTION

The Disk Dump Utility provides facilities for:

- dumping the information from a disk volume to a magnetic tape and verifying that the data is correctly dumped,
- restoring a previously dumped disk volume from magnetic tape to a disk volume and verifying that the data is correctly restored, and
- displaying the information contained on a disk volume in a format useful for debugging system routines that manage direct access volumes.

CAUTION

DISK DUMP IS NOT SUITABLE FOR BACKING UP A DISK BECAUSE IT WILL DUMP TO A BAD SECTOR IF INSTRUCTED TO DO SO. ANY DEFECTIVE SECTORS ON A DISK PACK CAN RESULT IN LOSS OF DATA. IT IS RECOMMENDED THAT THE BACKUP UTILITY BE USED FOR SAVING FILES.

6.2 DISK DUMP UTILITY REQUIREMENTS

The Disk Dump Utility requires:

- 9kb of memory. When using the DUL and UDL commands, an additional buffer is necessary. The size of the buffer depends on the record length as specified in the RECORD command, and
- any currently supported disk device.

The Disk Dump Utility is provided for the user in object format. It must be built as a privileged user task (u-task) using Link.

The Disk Dump Utility requires $n \times 256$ bytes of memory for use by the UDL and DUL commands. The value of n is set by the RECORD command. The default for n is 48. Therefore, at a minimum, Disk Dump requires 48×256 bytes of workspace if the DUL or UDL commands are to be used and twice that amount of buffer space for the verify option.

6.3 PERFORMANCE ESTIMATES

Timings for binary dumps of an entire disk (sector/block=48) are listed in Table 6-1.

TABLE 6-1 TIMINGS FOR BINARY DUMPS

COMMAND USED	DISK USED	OUTPUT DEVICE USED	TIMING (MINUTES: SECONDS)
DUL/UDL	2.5Mb	45 ips mag tape	1:40
DUL/UDL	5Mb	45 ips mag tape	3:00
DUL/UDL	5Mb	5Mb disk	1:05
DUS/UDS	2.5Mb	45 ips mag tape	13:10
DUS/UDS	5Mb	45 ips mag tape	16:45

6.4 OPERATING PROCEDURES

To execute the Disk Dump Utility, follow this procedure:

1. Load the Disk Dump Utility using the LOAD command.
2. Select the Disk Dump Utility as the current task using the TASK command.
3. Assign logical units (lu) using the ASSIGN command as follows:
 - disk device to lu1
 - binary input/output (I/O) device to lu2
 - ASCII output device; e.g., printer, to lu3
 - command input device to lu5

4. Start the Disk Dump Utility using the START command.

Format:

START

When the program is started, the following message is displayed on the console:

OS/32 DISCDUMP Rxx-yy

The current revision level of the program is xx; yy is the number of updates within this revision.

NOTE

Earlier versions of the Disk Dump Utility and Dump Print Utility are compatible with current versions of the OS/32 system. For example, if you are executing a 5.2 revision level panic dump print of the disk file, you would use the appropriate 5.2 revision level panic dump task on the 6.2 or higher revision level OS/32 system.

5. The user should be familiar with the file manager to interpret the output of the different types of dumps. The following commands dump various file management data to the ASCII output device (lu3):

Format:

VOLUME

Information displayed:

VOLN:	Volume name of disk
ATRB:	Attributes
FDP:	Pointer to first directory block
MAP:	Pointer to bit map

Format:

DIRECTORY

Information displayed:

FNM:	Filename
EXT:	Extension
ACT:	Account number
FLBA:	First logical block address of file
LLBA:	Last logical block address of file
LENGTH:	Record length (indexed or nonbuffered indexed file) or size of file (contiguous or extendable contiguous file)
KEYS:	Read/write keys
RCNT:	Read count
WCNT:	Write count
ATRB:	File type (attributes)
BKSZ:	Block size (index/data block size for index files)
CSEC:	Logical record length for indexed files, or pointer to last record accessed on contiguous files
TIME ALLOC:	Date and time of allocation
TIME WRITTEN:	Date and time of last change

Format:

BIT

Information displayed:

BIT MAP DUMP: Free and allocated sectors in bit map

Format:

FILES fd

Information displayed:

SECTORS: The sectors occupied by the specified file are displayed. If fd is omitted, all files are dumped including sectors occupied by the volume descriptor, bit map, and directory blocks.

6. This command dumps disk data by filename or sector.

Format:

DUH [{ SECTOR=st adr [,end adr] }]
DUA [{ FD=filename.ext }]
DUS [*]
DUL []

Parameters:

DUH dumps hexadecimal and ASCII to lu3, the list device.

DUA dumps ASCII to lu3, the list device.

DUS dumps 256 bytes (sector) binary to lu2, the binary output device.

DUL dumps multi-sector binary to lu2, the binary output device.

st adr is the starting hexadecimal sector address. The default is 0.

end adr is the ending hexadecimal sector address. If this parameter is omitted, only one sector is dumped.

* dumps only those sectors that are allocated in the bit map. The * option dumps only those blocks of sectors with at least one sector allocated, where the block is defined by the RECORD command. The default is 48 (one cylinder for a 2.5Mb or 5Mb disk).

filename.ext is the file descriptor (fd) of a contiguous or indexed file. The program dumps only the sectors occupied by that file.

Examples:

DUH SECTOR=0,10 dumps sectors 0 through 10 in hexadecimal and ASCII to lu3.

DUH SECTOR=1 dumps sector 1 in hexadecimal and ASCII.

DUA F=RUN.CSS dumps to lu3, in ASCII, all sectors occupied by file RUN.CSS.

DUA dumps all sectors in ASCII to lu3.

DUS * dumps binary to lu2 sectors marked as allocated in the bit map.

DUL dumps all blocks of sectors to lu2.

DUL * dumps to lu2 only those blocks of sectors with at least one sector allocated.

7. The UNDUMP command is the inverse of the DUMP command. Records are read from lu2 and written to the disk as specified.

Format:

UDS [SECTOR=st adr [,end adr]]
UDL [*]

Parameters:

UDS dumps 256 bytes (sector) binary from lu2, the binary output device.

UDL dumps multi-sector binary from lu2, the binary output device.

st adr is the starting hexadecimal sector address. The default is 0.

end adr is the ending hexadecimal sector address. If this parameter is omitted, only one sector is dumped.

Functional Details:

A disk should only be dumped to and restored from the same type of disk. That is, a 2.5Mb disk should not be dumped to a 5Mb or 40Mb disk, and vice versa.

Examples:

UDS SEC=0 read a sector from lu2 and write it to the disk as sector 0.

UDS * causes the first record to be read from lu2 and written, starting at sector 0. It becomes the volume descriptor for the pack. The bit map is read sequentially from lu2, and all sectors marked as allocated are read sequentially and written to the disk.

UDL * causes the first record to be read from lu2 and written, starting at sector 0. The bit map is searched for blocks of sectors with at least 1 sector allocated. Such blocks are sequentially read from lu2 and written onto the disk. The number of sectors per block can be specified by the RECORD command. The default assumed is 48.

8. The VERIFY command is used with the DUMP/UNDUMP commands. The VERIFY command should be used following all binary DUL and UDL operations. Any sectors between lu1 and lu2 that do not match are printed in hexadecimal format.

Format:

VERIFY *

Parameters:

- * verifies all allocated sectors between two logical units. This option is available after DUL * or UDL * operations only.

Functional Details:

The lu assignments are:

- lu1 indicates disk
- lu2 indicates disk or magnetic tape

If lu2 is assigned to a magnetic tape, the tape must be precisely positioned. The following commands are provided to support the positioning of the magnetic tape from within the program:

COMMAND	FUNCTION
<u>REWIND</u> lu	Rewind
<u>RW</u> lu	Rewind
<u>WFILE</u> lu	Write filemark
<u>FFILE</u> lu	Forward to end of file
<u>BFILE</u> lu	Backspace to beginning of file
<u>BRECORD</u> lu	Backspace record
<u>ERECORD</u> lu	Forward space record

9. The following commands can be used with the Disk Dump Utility:

COMMAND	FUNCTION
<u>WIDTH</u> n	sets width of print line to n, where n is a decimal number between 10 and 132. The default width is determined by a fetch attribute of lu3. If this fetch returns a zero record length or one larger than 132, the width is set to 72. Otherwise, the default width used is the logical record length returned from the fetch attributes call. This command is invalid if output is spooled. See Chapter 2 for information on OS/32 Spooler.

RECORD n sets the record length for the UDL, DUL, and VERIFY commands, where n is the number of sectors per record. The default value of n is 48. The allowable values for n are between 2 and 4096, inclusive. A buffer of n*256 bytes is required for the operation of the UDL and DUL commands. Twice the buffer is needed for VERIFY.

PAUSE pauses the Disk Dump Utility.

END terminates the Disk Dump Utility by a supervisor call (SVC) 3 code 0.

Messages:

COMMAND ERROR

indicates that an invalid operator command, option, or operand is read from the command input device (lu5). Another command is then read from lu5.

DEVICE NOT DISC

indicates that the device assigned to lu1 is not a disk device. The program pauses.

DISCDUMP Rxx-yy

indicates that the program is operational and ready for command input. The current revision level is xx; yy is the update level.

FILE NOT WRITTEN ON YET

indicates that a dump is requested for an indexed file on which no data has been written. Another command is read from lu5.

IO ERROR ssdd

indicates that non-zero status is returned on an I/O operation to a nondisk device. The device independent status is ss; dd is the device dependent status. The program pauses.

IO ERROR ssdd LBA =nnnnnn

indicates that non-zero status is returned on a disk I/O operation. The SVC 1 device independent status is ss; dd is the SVC 1 device dependent status; nnnnnn is the logical sector address. If the status is C0 (illegal function), A0 (device unavailable), or 81 (unassigned lu), the task pauses. When the task continues, the operation is retried. If the status is 90 (end of medium), the command is terminated and another command is read from lu5. For any other SVC 1 status, the I/O operation is retried 10 times before the message is logged; another command is then read from lu5.

MEM-FUT.L

indicates insufficient memory to set up a buffer used by the DUL and UDL commands. The program reads the next command.

OS/32Rxx-yy REQUIRED

indicates that Disk Dump is being run on an incompatible operating system. Revision xx-yy or higher is required.

CHAPTER 7 DUMP PRINT UTILITY

7.1 GENERAL DESCRIPTION

The Dump Print Utility interprets and prints to a list device the contents of the memory dump previously copied from memory to magnetic tape by the panic dump program. The Dump Print Utility prints:

- system data structures (STRUCS),
- system journal entries, and
- contents of memory.

The Dump Print Utility requires these system resources:

- OS/32,
- 18.25kb of memory in addition to that of the operating system,
- a disk device, and
- a list device (high speed line printer is recommended).

The Dump Print Utility can print the contents of a memory dump from magnetic tape or a disk device by copying it to a temporary file and then to a list device, or copying it to a permanent file and then to a list device.

NOTE

The dump print task relates to the particular OS/32 system from which you are printing. For example, if you are executing a 5.2 revision level panic dump print of the disk file, you would use the appropriate 5.2 revision level panic dump task on the 6.2 or higher revision level OS/32 system.

7.2 USING A TEMPORARY OR PERMANENT FILE

If a temporary file is used, it is allocated by the utility and automatically deleted at the end of task. However, if a permanent file is used, it must be allocated by the user as an indexed file with 256-byte records and assigned to logical unit 2 (lu2).

Copying the contents of the memory dump to a permanent file allows the user to subsequently restart the utility to print multiple copies or to modify a corrupted area so it can be interpreted and printed.

7.3 EXECUTING THE DUMP PRINT UTILITY

To print the contents of a memory dump from magnetic tape, enter the following sequence of commands:

```
LOAD dump print utility
TASK dump print utility
START
```

| To print the contents of a memory dump from a disk, the Dump
| Print Utility will accept any file descriptor (fd) as a response
| to the input device request.

The program prompts the user for the files and options required to produce the dump. A syntax or fd error entered as a response causes the prompt to be repeated.

Prompt: ENTER MAGTAPE (INPUT) FD:>

Response: fd

 is the file descriptor of the magnetic tape or
 disk file containing the memory dump to be
 printed; e.g., MAG1:. This fd is
 automatically assigned to lul.

NOTE

This prompt is not issued if lul was preassigned to a permanent disk file.

Prompt: ENTER LIST DEVICE FD:>

Response: fd

is the file descriptor of the list device on which the memory dump is to be printed; e.g., PR:. This fd is automatically assigned to lu3.

Prompt: DUMP ALL?>

Response: { YES }
{ NO }

If the user response is YES, the program displays the system journal, all operating system structures, and all memory to the list device. If the user response is NO, these prompts are displayed:

Prompt: DUMP STRUCS?>

Response: { YES }
{ NO }

If the user response is YES, the operating system structures are interpreted and dumped. If the user response is NO, no operating system structure dump is produced. Regardless of whether the answer to this prompt is YES or NO, the following prompts are displayed:

Prompt: DUMP JOURNAL?>

Response: { YES }
{ NO }

If the user response is YES, the operating system journal is formatted and dumped (if present). If the user response is NO, the journal dump is not produced.

Prompt: DUMP MEMORY?>

Response: { YES }
{ NO }

If the user response is NO, the previously requested data is printed on the list device. If the user response is YES, this prompt is displayed:

Prompt: DUMP MORE MEMORY?>

Response: { YES }
{ NO }

If the user response is NO, the Dump Print Utility terminates. If the user response is YES, the prompt sequence starting with ENTER PRINT RANGE (LOLIM,HILIM):> is displayed again.

7.4 DUMP PRINT EXAMPLE

The following example dialogue executes the Dump Print Utility:

```
*LOAD .BG,DUMPRINT
*TASK .BG
*START
OS/32 DUMPPRINT nn-nnn Rxx-yy
ENTER MAG TAPE (INPUT) FD:>MAG1:
ENTER LIST DEVICE FD:>PR:
DUMP ALL?>NO
DUMP STRUCS?>NO
DUMP JOURNAL?>YES
DUMP MEMORY?>YES
ENTER PRINT RANGE (LOLIM,HILIM):>OS
DUMP MORE MEMORY?>YES
ENTER PRINT RANGE (LOLIM,HILIM):>SYS
DUMP MORE MEMORY?>YES
ENTER PRINT RANGE (LOLIM,HILIM):>2C200,3EEEF
DUMP MORE MEMORY?>NO
.BG - END OF TASK CODE = 0 CPU TIME = n.nnn/n.nnn
*
```

Appendix C contains a sample display from the Dump Print Utility in effect for release R06.2 corresponding to the dialogue above. The sample display in Appendix C contains only the first two memory pages. Another example of a Dump Print Utility dialogue follows:

```
*LOAD .BG,DUMPRINT
*TASK .BG
*START
OS/32 DUMPRINT nn-nnn Rxx-yy
ENTER MAGTAPE (INPUT) FD:>MAG1:
ENTER LIST DEVICE FD:>M300: CRSHDUMP.TXT
DUMP ALL?>YES
END OF TASK
.BG - END OF TASK CODE= 0 CPU TIME = n.nnn/n.nnn
*
```

| Appendix D contains fragments of a sample display from the Dump
| Print Utility, in effect for software release R07.1 and higher,
| corresponding to the dialogue above. This dump print display
| contained in Appendix D is from a Model 3200MPS System.

7.5 CORRUPT SYSTEM POINTER TABLE (SPT)

If the halfword address at location X'62', which points to the SPT, or the SPT itself is corrupt, the following message is displayed:

```
POINTER TO SPT INVALID - X'62'  
ENTER A (SPT.INIT) - ELSE "NO">
```

The user is given the option to enter the valid address of the SPT, which is labeled SPT.INIT in the OS map. If the SPT is corrupt, the user can enter NO to continue. If NO is entered, the task control block (TCB) table and segment control list are not produced.

| 7.6 DUMP PRINT STRUCTURE DISPLAY

| The following sequence of structure display is in effect for the
| R06.2 release for the Dump Print Utility.

- | ● SPT
- | ● segment descriptor entries (SDEs) for pure segments
- | ● all TCBs for all TCBs in the TCB table
- | ● all device control blocks (DCBs)
- | ● volume mnemonic table (VMT)
- | ● coordination nodes (EVNs)

| The following sequence of structure display is in effect for the
| Dump Print Utility, release R07.1 and higher, for all processors.
| Please note difference for the Model 3200MPS System.

- | ● SPT
- | ● a TCB followed by its context block (CTX), for each task
| assigned to the central processing unit (CPU)

- for each auxiliary processing unit (APU) in the system:
 - an auxiliary processor block (APB), followed by a TCB and its associated CTX for each task assigned to the APU
- TCBs and their associated CTXs for all tasks in the TCB table but not already output
- SDEs for any pure segments in the segment control list (SCL)
- a DCB followed by its associated CTX (if any), for each device in the system
- VMT
- EVNs

NOTE

For systems, other than the Model 3200MPS, no APBs will be output. All TCBs are assigned to the CPU; therefore, all TCBs will be output following output of the SPT.

The display of structure output includes both hexadecimal and ASCII representation. The ASCII representation will be displayed on the far right of the report with a maximum of 32 characters.

7.7 DUMP PRINT MESSAGES

ADDRESS OUT OF RANGE: address

indicates that an invalid address pointer was encountered while processing the system STRUCS.

ASGN-ERR

indicates that lu is already assigned or offline.

BUFF-ERR

indicates there is no room in the system for file control block (FCB) and/or buffers.

FD-ERR

indicates a file descriptor error.

ILFN-ERR

indicates an illegal function, illegal file type.

INVALID TAPE FORMAT

indicates that the magnetic tape was created by an operating system not compatible with the current version of the Dump Print Utility Program.

LU-ERR

indicates an illegal lu.

NAME-ERR

indicates that the specified filename and extension do not exist.

POINTER TO SPT INVALID-CANNOT DUMP OS

A dump of OS memory was requested but the memory range of the OS cannot be determined because the SPT is corrupted.

POINTER TO SPT INVALID-CANNOT DUMP SYSTEM SPACE

A dump of system space memory was requested, but the memory range of system space cannot be determined because the SPT is corrupted.

PROT-ERR

indicates invalid protection keys.

SIZE-ERR

indicates invalid logical record length or not enough space on disk for the file.

SYNTAX ERROR

indicates a syntax error in fd.

TYPE-ERR

indicates a nondirect access device or device is marked offline.

VOL-ERR

indicates a volume error; no such volume or device exists in the system.

******ssdd I/O ERROR**

indicates that an I/O error occurred. ss is the device independent status byte from the standard SVC 1 parameter block; dd is the device dependent status byte. See the OS/32 Application Level Programmer Reference Manual.

APPENDIX A
DISK INITIALIZER UTILITY

A.1 GENERAL DESCRIPTION

NOTE

The Disk Initializer Utility and the Disk Integrity Utility will only be supported for OS/32 R06.2 software release and lower. Future releases of this manual will have the discussion of these two utilities removed entirely. These utilities have been replaced by the Fastchek utility which incorporates the capabilities of both. See the OS/32 Fastchek Reference Manual for a complete discussion of the Fastchek Utility.

The Disk Initializer Utility initializes a previously formatted disk pack for use with OS/32. Initialization includes placing the volume name and pointers to the bit map and directory in the volume descriptor, which is on the first sector of the disk. The volume name consists of one to four characters, the first of which must be alphabetic. This name identifies the disk to the system. The Disk Initializer allows a disk pack to be named and renamed. The directory describes all files on the pack. Filenames and starting and ending sector addresses identify each file on the disk.

The bit map contains one bit for every sector on the pack. If a bit is set, the sector it represents is allocated. Files are allocated on free sectors, and these sectors are then marked as used in the bit map. The Disk Initializer allows the user to clear the directory and bit map in order to delete all files on the disk. When initializing and clearing a new disk pack, the Disk Initializer also provides a facility to preallocate a fast access directory and to fill the rest of the disk with a specified data pattern.

The three options available with the Disk Initializer Utility are:

- CLEAR
- BLOCKS
- FILL

Whenever option CLEAR is specified, a readcheck operation checks the disk for bad sectors that have been flagged by the disk format/test program. These sectors are marked as unavailable in the bit map.

If the first sector on the disk fails the readcheck operation, the pack cannot be used and a message to that effect is printed. Because media degradation might occur at any time, there are instances where sectors that are not flagged at format time are flagged as bad sectors during initialization. In such instances, it is recommended that the disk pack be backed up, reformatted, and reinitialized. The data can then be restored. All files are deleted whenever CLEAR is specified.

The BLOCKS option preallocates a contiguous area on the disk for the directory. This speeds up access to the directory, which is normally noncontiguous.

For additional information regarding the relationship between the block parameter and other directory related features of OS/32, see the OS/32 Operator Reference Manual.

The FILL option specifies that all the sectors on the disk, except sector 0, the bit map sectors, and the preallocated directory sectors, are initially to be filled with the specified byte. The FILL option can be specified whenever option CLEAR is specified.

A.2 DISK INITIALIZATION REQUIREMENTS

The Disk Initializer Utility requires:

- memory that is 6kb above the operating system plus a buffer for the readcheck operation,
- a console device, and
- a currently supported disk device.

The Disk Initializer Utility is provided for the user in both object and image format. If an object version is to be used, it must be built as a privileged user task (u-task) using Link.

For the CLEAR option, a readcheck of the entire disk is performed by reading as many sectors into memory as can be accommodated into a buffer. Therefore, for optimum efficiency, a segment size increment field should be large enough to read one cylinder of the disk if option CLEAR is selected.

The Disk Initializer Utility reads as many sectors as possible, up to the number of sectors in one cylinder. Sectors are read into a buffer made available to the program by the segment size increment specified at load time.

Table A-1 serves as a guideline for calculating the optimum segment size increment for each disk whenever a readcheck is to be performed. The segment size increments for the disk devices are also contained in Table A-1.

TABLE A-1 INFORMATION REFERENCE TABLE FOR DISK DEVICES

DEVICE CODE	DEVICE TYPE	SEGMENT SIZE INCREMENT KB	SECTORS/ CYLINDER	CYLINDERS ON DISK
44	300Mb fixed			
45	1.5Mb HPT	160	640	9.6
46	160Mb	160	640	821
47	1.5Mb HPT	640	2560	2.4
48	2.5Mb fixed	12	48	203
49	2.5Mb removable	12	48	203
50	5Mb fixed	12	48	408
51	5Mb removable	12	48	408
52	40Mb removable	100	400	406
53	67Mb removable	80	320	823
54	256Mb removable	304	1216	823
55	Floppy	4	13	77
56	68.5Mb fixed	80	320	836.2
57	1.5Mb HPT	80	320	19.2
58	67Mb fixed	80	320	820
59	16Mb removable	16	64	823
60	16Mb fixed	16	64	821
61	48Mb fixed	48	192	821
62	80Mb fixed	80	320	821
63	675Mb fixed	640	2560	821

LEGEND

HPT = head per track

A.3 OPERATING PROCEDURES

The Disk Initializer Utility operates and must be established as a privileged u-task. To execute the Disk Initializer:

1. Load the Disk Initializer Utility provided with the operating system package using the LOAD command with a segment size increment field if the CLEAR option is specified. See Table A-1 for optimum segment size increments.
2. Select the Disk Initializer Utility as the current task by using the TASK command.
3. Mark the disk to be initialized offline by using the MARK command.
4. Start the Disk Initializer Utility by using the START command.

Format:

```
START ,DISC=dev:,VOLUME=voln [[,CLEAR],BLOCKS=n/m]
    [,FILL=pattern]
```

Parameters:

DISC=	dev: is the name of the disk device.
VOLUME=	voln is the volume name to be given to the disk pack.
CLEAR	specifies a clear disk and read check operation. When option CLEAR is specified, all files are deleted from the disk. A readcheck of the entire disk is performed and bad sectors are flagged as allocated in the bit map. Up to 20 bad sectors are identified in a message by their hexadecimal sector addresses on the disk. The total number of bad sectors is listed when the readcheck is completed. The clear option execution times for the various disks are listed as follows:
	<ul style="list-style-type: none">● 20 seconds for 1.5Mb disk● 20 seconds for 2.5Mb disk● 35 seconds for 5Mb disk

- 5 minutes for 40Mb disk
- 6 minutes for 67Mb disk
- 6 minutes for 68.5Mb disk
- 20 minutes for 256Mb disk

BLOCKS=

n is a decimal number specifying the number of directory blocks to be allocated. One directory block can describe five files. Each directory block is one physical sector and contains space for five directory entries (filenames and associated directory information). For optimal performance, n should be no larger than the maximum number of files that will be on the disk, divided by five. For example, a disk which is planned to contain 5000 files should be initialized with blocks=1000.

m is a decimal number specifying the cylinder number where the directory is to start. The default is m=1. Table A-1 contains the number of cylinders on a given disk. This option is allowed whenever the CLEAR option is specified. This option provides a facility to allocate a fast access directory. The directory blocks are allocated for optimal search time.

NOTE

If the directory starts on a cylinder other than 0 when initializing floppy disks, the maximum number of blocks that can be allocated is 6. However, if the directory starts on cylinder 0, the maximum number of blocks that can be allocated is 5.

If any sectors are found to be defective within the proposed directory, the program relocates the first directory block to the next track on the disk. The bit map is located immediately following the last preallocated directory sector. If the directory is not preallocated, the bit map is located on sector 1.

For optimal search time, the directory sectors are allocated as follows:

- every second sector for floppy disks
- every thirty-second sector for 1.5Mb disks
- every fourth sector for 2.5Mb disks
- every sixth sector for 5Mb disks
- every fifth sector for 40Mb disks
- every thirty-second sector for all other disks

FILL= pattern is a byte (two hexadecimal digits), halfword (four hexadecimal digits), or fullword (eight hexadecimal digits). This option specifies that all available sectors are initially to be filled with the specified pattern. This option also requires specifying the **CLEAR** option; otherwise, the program terminates and a message is displayed.

Examples:

ST ,DISC=DSC1:,VOLUME=PACK,CLEAR

DSC1: is cleared, readchecked, and named PACK. No directory blocks are preallocated.

ST ,D=DSC1:,V=TEST

DSC1: is renamed TEST. No file deletion takes place.

ST ,V=FIXD,D=DSC2:,CL,BL=0

DSC2: is cleared, readchecked, and named FIXD. No directory blocks are preallocated.

ST ,CL,D=DSC2:,V=TEST,BL=24/0,FI=A55A6996

DSC2: is cleared, readchecked, and named TEST. Starting at cylinder 0, 24 directory blocks are allocated. All available sectors are filled with words containing the hexadecimal value A55A6996.

Messages:

ARG-ERR

indicates that BLOCKS and/or FILL option was specified in the START command without a CLEAR option.

ASSN-ERR

indicates that non-zero status was returned when an attempt was made to assign the device name of the disk to logical unit 1 (lu1). The disk can be assigned by another task or marked on protected.

BAD SECTOR LBA=nnnnnn

indicates that a bad sector was found during a readcheck operation (CLEAR option specified). The sector is flagged as allocated in the bit map. This message is printed for up to 20 bad sectors, after which it is suppressed. The hexadecimal sector address on the disk is nnnnnn.

DIR-ERR

indicates that non-zero status was returned when trying to read, write, or allocate a directory block.

DISCINIT-Rxx-yy

informs the user that the program is operational. The revision level is xx; the update level is yy.

END OF TASK 0

indicates that the Disk Initializer Utility has successfully completed.

END OF TASK 1

indicates that the Disk Initializer Utility issues a supervisor call (SVC) 3 with an end of task code of 1 when it cannot perform the requested operation due to one of the error conditions. The user should correct the condition and reissue the START command.

FD-ERR

indicates that the device name of the disk is illegal.

FORM-ERR

indicates that a syntax error or illegal option was found in the argument list.

IO ERROR ssdd LBA=nnnnnn FD=fd

indicates that a non-zero status was returned on a disk input/output (I/O) operation; ss is the SVC 1 device independent status; dd is the SVC 1 device dependent status; nnnnnn is the logical sector address requested; and fd is the file descriptor of the disk.

n DIRECTORY BLOCKS PREALLOCATED

indicates that a number other than 5 or 6 was specified in the BLOCKS option when preallocating directory blocks on a floppy disk. If the directory starts on any cylinder other than zero, n=6. If the directory starts on cylinder zero, n=5.

NO SECTORS FLAGGED

indicates no bad sectors were found during the readcheck operation.

NOPR-ERR

indicates no operands were entered with the START command.

NUMBER OF BAD SECTORS = nnnnnn

indicates that after a readcheck operation a number of bad sectors were found.

OS/32 Rxx-yy REQUIRED

indicates that this version of Disk Initializer is being run on an incompatible operating system. Revision xx-yy or higher of the operating system is required.

PACK-ERR

indicates that the disk could not be initialized for one of the following reasons:

- I/O error was encountered.
- Sector 0 is bad.
- Not enough contiguous good sectors exist to allocate a bit map.
- Disk is not in ready state.

STAT-ERR

indicates that the disk was marked on.

APPENDIX B DISK INTEGRITY CHECK UTILITY

B.1 GENERAL DESCRIPTION

NOTE

The Disk Initializer Utility and the Disk Integrity Utility will only be supported for OS/32 R06.2 software release and lower. Future releases of this manual will have the discussion of these two utilities removed entirely. These utilities have been replaced by the Fastchek Utility which incorporates the capabilities of both. See the OS/32 Fastchek Reference Manual for a complete discussion of the Fastchek Utility.

The Disk Integrity Check Utility provides a means of recovering open disk files following an operating system failure. This utility is also used to restore the integrity of data on disk volumes. This condition can occur if the disk is dismounted without being marked offline. A system failure has the effect of dismounting online disk volumes. The program rebuilds the bit map and validates file pointers of indexed and contiguous files. Directory blocks with no active entries are deleted unless they were preallocated by the Disk Initializer Utility.

All temporary files are deleted. Spool files that had not been closed and sent to the Spooler are deleted. A spool file is also deleted if it was created more than 24 hours before the current date. If there are spool files more than a day old that are to be kept on the disk, an earlier date can be entered via the operating system SET TIME command before running the Disk Integrity Check Utility. Spool files with a creation date later than the current date are not deleted.

If a disk volume is inadvertently dismounted without being properly marked offline, it can only be marked online in a write protected mode; e.g., MARK D1:,ON,PROTECT. The Disk Integrity Check Utility must be executed before the MARK command can be entered without the PROTECT option.

The Disk Integrity Check Utility closes all files that are assigned, sets the date last written if the file was open for write, and validates all control information on the disk. The latter function is performed in case bad data was written to the volume during a system failure. Complete volume recovery is not always possible because bad data might have been written to the volume prior to the failure. The program's output messages explain the status of individual files or the entire disk and describe what actions have been taken or attempted.

WARNING

IT IS IMPERATIVE TO RUN THE DISK INTEGRITY CHECK UTILITY WHENEVER THE INTEGRITY OF A DISK IS IN QUESTION. FAILURE TO DO SO IMMEDIATELY CAN RESULT IN THE UNNECESSARY LOSS OF DATA AND FILES.

Systems without direct access devices need only restore the operating system environment that existed prior to the system failure. No further action is required. Systems with direct access devices that did not have any direct access devices marked online at the time of the system failure can be recovered using the procedure for systems without direct access devices. It is not always possible to determine if any files were assigned; i.e., a program might have made an assignment using supervisor call 7 (SVC 7). Therefore, it is recommended that the Disk Integrity Check Utility be used on all systems with online direct access devices. Failure to execute this utility after a system failure can leave direct access volumes in a state where files can be neither assigned nor deleted.

B.2 DISK INTEGRITY CHECK UTILITY REQUIREMENTS

The Disk Integrity Check Utility requires:

- memory of 7.75kb above the operating system size, plus an optional buffer for the read check operation;
- a console device;
- a list device (can be the console device); and
- any currently supported disk device.

The Disk Integrity Check Utility is provided for the user in both object and image format. If an object version is to be used, it must be established as a privileged user task (u-task) using Link.

If the READCHECK option is selected, as many sectors up to one cylinder that can be accommodated into a buffer between UTOP and CTOP are read. Therefore, when READCHECK is selected, the speed of the program increases if sufficient memory is available to read one entire cylinder. See Table A-1 for optimum segment size increments.

B.3 OPERATING PROCEDURES

The following six procedures are recommended after an operating system failure for systems configured with direct access devices:

1. Reload the operating system.
2. Mark the system volume on, protected.
3. Load the Disk Integrity Check Utility using the LOAD command.
4. Select the Disk Integrity Check Utility as the current task using the TASK command.
5. Mark the disk to be checked offline.
6. Start the Disk Integrity Check Utility by using the START command.

Format:

$$\text{START ,dev: , [list fd] } \left[\begin{array}{c} \left\{ \begin{array}{l} \text{CLOSE} \\ \text{NOREADCHECK} \\ \text{READCHECK} \end{array} \right\} \end{array} \right]$$

Parameters:

dev: is the name of the device.

list fd is the file descriptor of the device or file to which the Disk Integrity Check messages are displayed.

CLOSE specifies that all open files assigned for read only are closed. A message is printed and the files are closed, but no data is lost. An indexed file open for write causes a message to be printed; the file is not closed. In this case, the Disk Integrity Check Utility must be rerun without the CLOSE option.

NOREADCHECK specifies that no readcheck is performed. When the Disk Integrity Check Utility clears the bit map, all sectors previously flagged defective are freed. This option should never be specified if a disk is known or suspected to have bad sectors. If this parameter is not specified, READCHECK is the default.

READCHECK specifies that the program is to search for bad sectors. Any bad sectors are marked as allocated in the bit map, and a message is output. The program reads as many sectors as it can, up to one cylinder, into a buffer. If zero status is returned, the next group of sectors is read. This process continues until the entire disk is checked. If non-zero status is returned, a sector by sector read is performed until the bad sectors are located.

Examples:

ST ,DSC1:,CON:	Normal program with read check
ST ,DSC1:	Normal program with read check (list device was preassigned)
ST ,DSC1:,PR:	Normal program with read check
ST ,DSC1:,CON:,R	Normal program with read check
ST ,DSC1:,CON:	Normal program
ST ,DSC1:,CON:,CLOSE	Close files only

Messages:

ASSIGN ERROR CODE xx

indicates that the program attempted to assign either logical unit 1 (lu1) or lu3 to their respective file descriptors. The returned SVC 7 status is xx.

BAD DIRECTORY-CHAIN BROKEN-(reason n)

| indicates that a pointer to a directory block is not
| valid because the directory itself is no longer valid.
The directory chain is broken; that is, directory entries
further down the directory chain are no longer accessible
to any program, and the files that they defined are lost.
To obtain a list of the valid files, use the DISPLAY
FILES command.

Because OS/32 R06.2 contains two new file types, nonbuffered index and extendable contiguous, the files displayed by the DISPLAY FILES command might have files displayed under an NB or EC heading. See Chapter 3 for an explanation of the two new file types supported.

- Invalid pointer to directory block (reason 2).
- Unable to mark a directory block as allocated in the bit map (reason 1).

BAD FILENAME filename

indicates that a filename does not conform to the operating system naming conventions. The file is deleted.

BAD PACK-REINITIALIZE

indicates that the disk cannot be checked because an input/output (I/O) error (other than write-protect) was encountered.

BAD SECTOR, LBA=nnnnnn

indicates that a bad sector was found during a read check operation. The sector is marked as allocated in the bit map. The hexadecimal logical block address is nnnnnn.

CHAIN BROKEN ON FILE filename

indicates that either:

- when examining an indexed file, the forward pointer of the last index block was not zero. This causes the last forward pointer to be reset to zero. The first sector of the last index block is rewritten, or
- the backward pointer of an index block did not point to the previous index block. The FILE filename DELETED message is then printed.

DATA POINTERS FOLLOWING LAST POINTER NOT ZERO, FILE filename

indicates that the data block pointers following the last calculated pointer in an indexed file were non-zero.

DEVICE NOT DISC

indicates that lul is not assigned to a disk, or the first fd in the START command is not a disk fd.

DISCCHECK xx-yy

indicates that the program is operational. The current revision level of the Disk Integrity Check Utility is xx; yy is the update level within the revision.

FD-ERR

indicates that an invalid fd or volume name was issued in the START command.

FILE filename ASSIGNED FOR WRITE, COUNTS NOT RESET

indicates that non-zero write counts were found in an indexed file during execution of the CLOSE option. The integrity of the disk has not been restored and the program must be re-executed without the CLOSE option.

FILE filename DELETED-(reason n)

indicates that a previously active directory entry was marked inactive. This message is issued for one of the following seven reasons:

- An invalid directory pointer (reason 2)
- Spool file not closed (reason 6)
- Spool file more than a day old (reason 7)
- An invalid file type
- Unable to mark a used sector as allocated in the bit map (reason 1)
- A contiguous file with the last logical block address less than the first logical block address (reason 5)
- Temporary file (reason 9)

This message is output for an indexed file if:

- An invalid block size was encountered
- The index or data pointers are invalid (reason 2)
- The calculated number of data blocks does not agree with the actual number of data blocks between the first and last logical block addresses
- Unable to read index block (reason 4)
- The first data block address (FLBA) is 0, but the number of logical records is nonzero (reason 3)
- The last data pointer is not contained in the last index block (reason 8)
- A backward pointer of an index block does not point to the previous index block

FORM-ERR

indicates the options in the START command do not conform to specifications.

INCORRECT BLOCK COUNT ON FILE filename

indicates that the calculated block count does not equal the actual number of data blocks for an indexed file. This message is followed by message FILE filename DELETED.

INVALID BLOCKSIZE OF ZERO ON FILE filename

indicates that the blocksize field in the directory is zero for an indexed file. The file is deleted.

INVALID FILE TYPE, FILE filename

indicates that the file type field in the directory is not contiguous or indexed. The file is deleted.

IO ERROR ssdd

indicates that a non-zero status was received. The program is then paused. The device independent status is ss; dd is the device dependent status.

IO ERROR ssdd LBA=nnnnnn

indicates that a non-zero status was received while trying to read or write a sector on the disk. The device independent status is ss; dd is the device dependent status; nnnnnn is the hexadecimal logical block address.

OS/32 Rxx-yy REQUIRED

indicates that this version of the Disk Integrity Check Utility is being run on an incompatible operating system. Revision xx-yy or higher of the operating system is required.

PACK IS HARDWARE PROTECTED

indicates that the disk cannot be checked because the pack is hardware protected.

POTENTIAL LOST DATA ON FILE filename

indicates that a file was restored to a valid state and closed by the program. If the file was open for write, the date last written is set to the current date. There could be a potential loss of data.

APPENDIX C
CONTENTS OF MAG TAPE PRODUCED
BY STAND-ALONE DUMP

The following pages are a sample of a stand-alone dump formatted by the OS/32 Dump Print Utility effective for release R06.2.

CRASH CODE = 102

AT 18:38:33 ON 1/04/83

PROCESSOR = 8/32

OS32MT06-02

628C.C22

DUMP OF GENERAL PURPOSE REGISTER SETS

	SET F	SET 0	SET 1	SET 2	SET 3	SET 4	SET 5	SET 6
R0	00000050	00007001	00000000	00000000	00000000	00000000	00000009	FFFFFFB50
R1	00000150	0001A4B0	00000000	00000000	00000000	00000000	000072E0	00000001
R2	00000017	00000000	00000000	00000000	00000000	00000000	0000AF74	000D6788
R3	000C34E0	00000004	00000000	00000000	00000000	00000000	00007330	00000000
R4	000006BC	00000560	00000000	00000000	00000000	00000000	0001CACE	000D4D10
R5	00000710	00000000	00000000	00000000	00000000	00000000	0001C65A	00061058
R6	00000000	00000007	00000000	00000000	00000000	00000000	000072E0	00000050
R7	00000000	00002002	00000000	00000000	00000000	00000000	0000AF74	000FC508
R8	00000006	00024756	00000000	00000000	00000000	00000000	0001DOAC	00027398
R9	00000064	000D7060	00000000	00000000	00000000	00000000	000D6788	000D69A4
RA	000011CC	30302C30	00000000	00000000	00000000	00000000	00000000	00000001
RB	00000558	000FD330	00000000	00000000	00000000	04000000	00000000	00000000
RC	000040E0	430080B6	00000000	00000000	00000000	00000000	00000000	0002CCF0
RD	00000448	430080BE	00000000	00000000	00000000	00000000	0000B700	0000002A
RE	00001670	00007002	00000000	00000000	00000000	00000000	00004080	00007000
RF	000C3FE4	000080B6	00000000	00000000	00000000	00037B8C	00000000	00007062

DUMP OF HARDWARE FLOATING POINT REGISTERS

F0	00000000	D0	00000000	00000000
F2	00000000	D2	00000000	00000000
F4	00000000	D4	00000000	00000000
F6	00000000	D6	00000000	00000000
F8	00000000	D8	00000000	00000000
FA	00000000	DA	00000000	00000000
FC	00000000	DC	00000000	00000000
FE	00000000	DE	00000000	00000000

CURRENT TASK		D7060	IRDR
UT REGISTERS OWNER		D6788	LEE
RS REGISTERS OWNER		D6788	LEE

DUMP OF TCB TABLE

ID	TCB NAME	ADDRESS	TASK FILE NAME	MID	GID	NLU	START	END	SIZE	SHSZ	OPTION	STATUS	WAIT
001	.CSL	57C8		000	001	12	16A64	16A64	0	0	00108008	00000000	00000400
002	.CMDP	5AA8		000	001	12	17394	17394	0	0	00108008	00000000	00000400
003	.MTM	D9378	MTM :MTM8CC22.TSK/00000	000	001	254	41C00	59A00	17E00	0	02388038	00000000	00000400
004	.MTMASST	D7D80	MTM :MTMASST .TSK/00000	003	003	1	59A00	59C00	200	0	00208028	00000000	00000400
005	.SPL	D7620	MTM :SPOOLER .TSK/00000	000	001	15	59C00	5FE00	6200	0	0230082D	00000000	00000400
006	DATE	D8148	MTM :TIME .TSK/00000	000	002	15	5FE00	60300	500	0	00012021	00000000	00000080
007	IRDR	D7060	MTM :IRDR .TSK/00000	000	002	1	60300	60B00	800	0	00000CB1	00001000	00000000
008	LEE	D6788	MTM :EDIT32 .TSK/00000	003	003	15	60B00	63800	2D00	7A00	0C200CA1	00004000	00010000
009	MTM	D59A0	MTM :CAL32 .TSK/00000	003	003	15	6BB00	8B800	1FD00	9C00	0C200CA1	00001000	00000000
00A	LAURIE	CE140	MTM :TEXT3 .TSK/00000	003	003	15	9B300	A0500	5200	D900	0C200CA1	00000000	00000040

C-4

SEGMENT CONTROL LIST

ADDRESS	NAME		START	END	SIZE	USE	ROLL	TYPE	FLGS	SREG	KEY	ROLL DCB	SECTOR	SSTP	PRIV
FC4A0	HELPRO6		3A200	3A600	400	2	0	SHARED	C8		00				RE
FC468	EDIT32S		3A600	41C00	7600	2	0	SHARED	C8	C	00				RE
D9378	.MTM		41C00	59A00	17E00	0	0	IMPURE	48		00				RWE
D7D80	.MTMASST		59A00	59C00	200	0	0	IMPURE	48		00				RWE
D7620	.SPL		59C00	5FE00	6200	0	0	IMPURE	48		00				RWE
D8148	DATE		5FE00	60300	500	0	0	IMPURE	48		00				RWE
D7060	IRDR		60300	60B00	800	0	0	IMPURE	48		00	FF007740	A48		RWE
D6788	LEE		60B00	63800	2D00	0	0	IMPURE	48		00	FF007740	2210		RWE
FREE			63800	6BB00	8300										
D59A0	MTM		6BB00	8B800	1FD00	0	0	IMPURE	48		00	FF007740	4COB		RWE
FD390	MTM CAL32	TSK	8B800	95400	9C00	1	0	PURE	49	5	00	FF007740	53B8		RE
FCC88	MTM TEXT3	TSK	95400	9B300	5F00	1	0	PURE	49	D	00	FF007740	4AB7		RE
CE140	LAURIE		9B300	A0500	5200	0	0	IMPURE	48		00	FF007740	4748		RWE
FREE			A0500	B5000	14B00										

48-031 F00 R01

DCB	FCB	SIZE	FILENAME	TYPE	WCNT	RCNT	FLGS
69C0	FCEA8	490	M300:20426991.001/00016	IN	FFFF	1	C0000000
69C0	C71A0	C90	M300:OS3230 .OBJ/00010	IN	0	4646	84589040
69C0	D61F0	02020	M300:SLIB3220.OBJ/00125	IN	9529	0036	6555045
69C0	EB2C8	128	M300:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000
6E50	D11F0	2090	M301:MTMMAIN .CAL/00096	IN	0	1	C0000000
6E50	DA860	128	M301:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000
72E0	D4D10	C90	M67A:20427110.001/00082	IN	FFFF	1	C4400000
72E0	CEA40	490	M67A:G .CSS/00096	IN	0	1	C0000000
72E0	CEED0	1690	M67A:MTMMAIN .LST/00096	IN	FFFF	2	C4000000
72E0	D0560	C90	M67A:MTMMAIN .OBJ/00096	IN	FFFF	1	E1000000
72E0	D3280	C90	M67A:MTM .CSS/00096	IN	0	1	C0000000
72E0	D5D60	490	M67A:MTM .LOG/00096	IN	1	0	C4400000
72E0	D6B48	490	M67A:MTM .JOB/00096	IN	FFFF	FFFF	C0000000
72E0	F8630	128	M67A:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000
7740	FC6A8	490	MTM :SPL .QUE/00000	IN	FFFF	1	C4400000
7740	FC220	128	MTM :PAGE . /00255	CO	FFFF	FFFF	40400000
7740	DA3D0	490	MTM :BATFIL . /00255	IN	FFFF	FFFF	C4400000
7740	D86E8	C90	MTM :AUFIL . /00255	IN	1	1	EC400000
7740	FCB38	128	MTM :ERROR .LOG/00000	CO	1	1	40400000
7740	FD990	128	MTM :SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000
7F40	FBD98	128	FIXD:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000

JOURNAL DUMP

TASKID	MODULENAME	REGISTER.C	REGISTER.D	REGISTER.E	REGISTER.F
003	SVC1	00043E6C	00043E6C	000072F0	00043E6A
003	SQS	0001C6AC	00000000	000D9378	0000B2E4
003	THRDISP	000D9544	00001000	000072F0	00043E6A
003	SVC1	0005345C	0005345C	000072F0	00043E00
003	THRDISP	000D9544	00001000	000072F0	00043E00
003	SVC9	00000000	000423A4	000072F4	00042D40
003	THUCHN	C8015E30	00001000	000072F0	00042D40
008	THRDISP	000D6954	00001000	000077F0	000C4040
008	SVC2	000C0650	0003AC50	000077F2	000C04EA
008	THRSIN	000A70F0	0003AC50	000077F2	000C04EA
008	THRDISP	000D6954	00001000	000077F0	000C04EA
008	SVC2	00000534	00061034	000077F2	000C0514
008	THRSIN	000070F0	00061034	000077F2	000C0514
008	SQS	0001C6AC	00000000	000D6788	0000B2E4
008	THREHW	00000006	00000400	000D6788	8001CF60
008	THCHN	00000006	00000400	000D6788	8001CF60
003	SQS	0001C6AC	00000000	0000B2E4	0000B2E4
003	THRDISP	000D9544	00001000	000072F0	00042CE2
003	SVC9	00000000	000423A4	000072F4	00042D40
003	THUCHN	C8015E30	00001000	000072F0	00042D40
008	THRD.SYS	000D69A4	00000000	00007260	000207B8
008	THRDISP	000D6954	00001000	000077F0	000C0514
008	SVC2	0000050C	0006100C	000077F2	000C3526
008	THRSIN	000070F0	0006100C	000077F2	000C3526
008	THRDISP	000D6954	00001000	000077F8	000C3526
008	SVC1	00000884	00061384	000077F0	000C4040
008	THRSAIN	000FD868	000D687C	000D4D10	000FD820
008	EVQCOM	80025BF2	000D687C	000D4D10	000D4DC4
008	SVC1	000D4E14	000D4E14	00007260	000263EC
008	THUCHN	000D687C	00008000	00004080	000D6788
009	SQS	0001C6AC	00000000	000D6788	000AF74
009	THRDISP	000D5B6C	00001000	000077F2	0005651E
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	THRDISP	000D5B6C	00001000	000077F0	00056182
009	SVC1	00000174	0006BC74	000077F0	00051206
009	THRSAIN	000FCDB0	000D5A94	000D11F0	000FCDB0
009	EVQCOM	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353636	000D17E8	000D12A4
009	THRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	THREHW	000000FF	00008000	000000EC	00004080
009	THCHN	000000FF	00008000	000000EC	00004080
008	THRD.SYS	000FD868	00000000	00007260	000263EC
008	SVC1	000D4E7C	000D4E7C	00007260	0002689A
008	THUCHN	000D687C	00008000	00004080	00000000
009	SQS	0001C6AC	00000000	000D6788	000AF74
009	THRDISP	000D5B6C	00001000	000077F0	00055164
009	SVC1	00000174	0006BC74	000077F0	00051206
009	THRSAIN	000FCDB0	000D5A94	000D11F0	000FCDB0
009	EVQCOM	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353637	000D1838	000D12A4
009	THRDISP	000D5B6C	00001000	000077F0	00051206
009	SVC1	00000174	0006BC74	000077F0	00051206
009	THRSAIN	000FCDB0	000D5A94	000D11F0	000FCDB0
009	EVQCOM	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353638	000D1888	000D12A4
009	THRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	0000B234
009	THREHW	00000006	00000400	000D59A0	8001CF60
009	THCHN	00000006	00000400	000D59A0	8001CF60
005	THRDISP	000D77EC	00001000	000077F0	00000C96
005	SVC2	00003594	0005D194	000077F9	00000D3A
005	THNSOUT	000D77EC	00001000	000077F9	00000D3A
005	SVC1	000044R4	0005E0B4	000077F0	000011D6
005	THRSAIN	000FD908	000D7714	000FCEA8	000FD8C0
005	EVQCOM	80025BF2	000D7714	000FCEA8	000FCF5C
005	EVDIS	00025F7C	00000020	000FD2F3	000FCFFC
005	THRDISP	000D77FC	00001000	000077F0	000011D6
005	SVC1	000044B4	0005E0B4	000077F0	0000121A
005	SQS	0001C6AC	00000000	000D7620	0000R234
005	THRDISP	000D77EC	00001000	000077F0	0000121A
005	SVC9	000FD6F8	0005D408	000077F1	00000C96
009	THUCHN	DE001600	00001000	000077F0	00000C96
009	THRDISP	000D5B6C	00001000	000077F2	0005654E
009	SVC1	00000174	0006BC74	000077F0	00051206
009	THRSAIN	000FCDB0	000D5A94	000D11F0	000FCDB0
009	EVQCOM	80025BF2	000D5A94	000D11F0	000D12A4
009	SQS	0001C6AC	00000000	000D59A0	0000B234
009	THRD.SYS	000FCDB0	00000000	00007262	0002B02A
009	EVDIS	00025F7C	31353639	000D18D8	000D12A4
009	THRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	THRDISP	000D5B6C	00001000	000077F0	00051944
009	SVC1	00000174	0006BC74	000077F0	00051206
009	THRSAIN	000FCDB0	000D5A94	000D11F0	000FCDB0
009	EVQCOM	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353730	000D1928	000D12A4
009	THRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	THREHW	000000FF	00008000	000000EC	00004080
009	THCHN	000000FF	00008000	000000EC	00004080
008	THRD.SYS	000FD868	00000000	00007260	0002689A
008	EVDIS	000D4E14	20202020	000D4FE8	000D4DC4
008	THRDISP	000D6954	00001000	000077F0	000C4040
008	SVC7	000FD670	00061058	000077F2	000C3FF4
008	THRSIN	000FD670	00061058	000077F2	000C3FF4
008	THUCHN	000D69A4	00000000	000D6788	000C3FF4
009	THREHW	000D8658	00000400	000FD330	0002B944
009	THCHN	000D8658	00000400	000FD330	0002B944
007	I/H	430080R6	430080BE	00007002	000080B6

48-031 F00 R01

```

000000: FFFFFFFF 0001B186 0000FFFF 00000000 00000000 000D86BC 00000000 00000000 * .....1.....<..... *
000020: 88018801 88018801 00000002 FFFF0000 00000000 00037B74 00000000 0001A930 * .....@.....T.....) *
000040: 88018801 88018801 00007000 0001A85A D50000CF 43000080 010201FF 00621313 * .....P....(ZU..OC.....B.. *
000060: 23025590 43004003 61240000 00000000 43004002 DCF40000 1399ED35 EBFA0C22 * #.U.C.@.AS.....C.@.\T...M5KZ." *
000080: 0000B700 30100D88 00007050 0001AF7A 00000000 0001A6C4 00007000 2D102D1A * ..7.0.....PP.../Z.....ED.P.-.- *
0000A0: 2D5C2DA2 2DB42DD4 2E042E4E 2D102EDE 2F022D10 2D102D10 2F642F76 00000000 * -\--4-T...N-..~/.-.-./D/V... *
0000C0: 00000000 00000000 00007000 0001AA6C 3018302C 0501302C 302C302C 302C302C * .....P....*L0...0,0,0,0,0, *
0000E0: 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000100: 0871302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C * .Q0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000120: 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 0561302C 05A105BD * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000140: 302C302C 302C302C 05E105FD 302C302C 302C302C 0621063D 302C302C 0661302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000160: 302C302C 302C302C 06A106BD 302C302C 302C302C 302C302C 302C302C 302C302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000180: 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
0001A0: 302C302C 302C302C 30203026 302C302C 302C302C 06E106FD 0721073D 302C302C * 0,0,0,0,0 0&0,0,0,0,0,0,0,0,0,0, *
0001C0: 302C302C 0761077D 07A107BD 07E1302C 302C302C 302C302C 302C302C 302C302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
0001E0: 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C * 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000200-0002FF ***SAME AS ABOVE ***
000300: 02C60B10 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * .F..... *
000320: 00000000 00000000 00000000 00000000 0753A630 00000000 0033A230 00000000 * .....S&0.....3"0..... *
000340: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000360-0003FF ***SAME AS ABOVE ***
000400: 00000000 00000019 00000000 00000000 00000000 26103000 000069C0 00000000 * .....&.0.....I@.... *
000420: 00000000 00000058 00000000 00000000 00000000 26103000 00006E50 00000000 * .....X.....&.0...NP.... *
000440: 00000000 00000000 00000000 00000000 00000000 26103000 000072E0 00000000 * .....&.0...R.... *
000460: 00000000 00000177 00000000 00000000 00000000 26103000 00007740 00000000 * .....W.....&.0...W@.... *
000480: 00000015 00000000 FFFF0001 00000000 44000000 25080000 00007F40 00000000 * .....D...%.....@.... *
0004A0: 00000000 00000000 00000000 00000000 00000000 00000000 000082E0 00000000 * ..... *
0004C0: 00000000 00000000 00000000 00000000 00000000 26B00000 000084A0 00000000 * .....&0..... *
0004E0: 00000000 00000000 00000000 00000000 00000000 26B00000 00008660 00000000 * .....&0..... *
000500: FF82FFB1 000065CC 0076FFFE 0000BAA2 0000BAB0 0FF06480 00008820 00000000 * ...1..EL.V...:"...:0.PD.... *
000520: FF00FFB1 0007AD5F 000029DE 6000FF01 0000E6F0 29100000 000088C0 00000000 * ...1..-(.)^.....FP).....@.... *
000540: FF000000 00000000 00000000 00000000 0000F8A4 2B640000 00008960 00000000 * .....XS+D..... *
000560: FF010000 00000000 00000000 00000000 0000C55A 1A500000 00008AD0 00000000 * .....EZ.P.....P.... *
000580: 00000000 00000000 00000000 0000F8A4 302C0000 00008AD0 0000C5DC 00000000 * .....XS0,.....P..E\.... *
0005A0: FF01FF89 00055A73 00000000 00000000 0000C55A 124E0500 00008CF0 FF0C0001 * .....ZS.....EZ.N.....P.... *
0005C0: 0005253A 00FD0000 00008EFD 0000F8A4 302C0000 00008CF0 0000C5DC FF061444 * ...%:.....XS0,.....P..E\...D *
0005E0: FF01FF89 00055673 FF971A36 00000000 0000C55A 124E0500 00008F10 FF0C0001 * .....VS...6.....EZ.N..... *
000600: 0005253A 00C60000 0000911D 0000F8A4 302C0000 00008F10 0000C5DC FF061444 * ...%:F.....XS0,.....P..E\...D *
000620: FF01FF89 00055373 43981D1A 00000000 0000C55A 124E0500 00009130 FF0C0001 * ...%:SSC.....EZ.N.....0.... *
000640: 00052530 003C0000 0000933D 0000F8A4 302C0000 00009130 0000C5DC FF061444 * ...%0.<.....=..XS0,.....0..E\...D *
000660: FF010000 00000000 00000000 00000000 0000C55A 1A500000 00009350 00000000 * .....EZ.P.....P.... *
000680: 00000000 00000000 00000000 0000F8A4 302C0000 00009350 0000C5DC 00000000 * .....XS0,.....P..F\.... *
0006A0: FF01FF89 00055B73 FFD61D1A 0000FFFF 0000C55A 124E0500 00009570 FF0C0001 * .....[S.V.....EZ.N.....P.... *
0006C0: 0005253A 00090000 0000977D 0000F8A4 302C0000 00009570 0000C5DC FF86233A * ...%:.....XS0,.....P..E\...#: *
0006E0: FF01FF89 00055BF3 FFB41A36 00000000 0000C55A 124E0500 00009790 FF0C0001 * .....[S.4.6.....EZ.N..... *
000700: 0005253A 004E0000 0000999D 0000F8A4 302C0000 00009790 0000C5DC FF86233A * ...%:N.....XS0,.....P..E\...#: *
000720: FF01FF89 000556F3 000020FC 00000000 0000C55A 124E0500 000099B0 FF0C0001 * .....VS... ..EZ.N.....0.... *
000740: 0005253A 00490000 000099BD 0000F8A4 302C0000 000099B0 0000C5DC FF061444 * ...%:I.....=..XS0,.....0..E\...D *
000760: FF01FF89 000558F3 FF9E1A36 0000FFFF 0000C55A 124E0500 000099BD FF0C0001 * .....XS...6.....EZ.N.....P.... *
000780: 0005253A 00AC0000 000099DD 0000F8A4 302C0000 000099BD 0000C5DC FF86233A * ...%:.....]..XS0,.....P..E\...#: *
0007A0: FF01FFB1 00054483 FF8A20FC 00000000 0000C55A 124E0500 00009DF0 FF0C0001 * ...1..D.....EZ.N.....P.... *
0007C0: 00052535 00A30000 00009FFD 0000F8A4 302C0000 00009DF0 0000C5DC FF061444 * ...%5.#.....XS0,.....P..E\...D *
0007E0: FF02FFB1 00055ECB 00340001 00000000 0002EBEC 32140001 0000A010 00055E7C * ...1..^K.4.....KL2.....^.. *
000800: 00000000 00000000 10020000 00000000 0000007E 00000000 00000000 00000000 * ..... *
000820: 00000000 FF060001 0005253A 00200001 00000000 0002EBEC 394A0001 0000A010 * .....%:.....KL9J..... *
000840: 00052537 38CC0000 00000000 10040000 00004000 000C007F 00000000 0000A178 * ...%78L.....!X *
000860: 00000000 00000000 00000000 00000000 FF000001 00000000 00000001 00000000 * ..... *

```

C-7

C-8

```

000880: 0002EBEC 30BEOC00 0000A1D0 00000000 00000000 00000000 10110000 00000000 * ..KLO>...!P..... *
0008A0: FFFF0018 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
0008C0: 00000000 0002EBEC 00000000 0000A1D0 00000000 00000000 00000000 00000000 * .....KL.....!P..... *
0008E0: 00000000 00000019 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000900: 00000000 00000000 00000000 00000000 00000000 00000000 0000A300 00000000 * .....#..... *
000920: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000940: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000A300 * .....#..... *
000960: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000980- 00099F ***SAME AS ABOVE *** * - SAME - *
0009A0: 00000000 00000000 0000A440 00000000 00000000 00000000 00000000 00000000 * .....$@..... *
0009C0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
0009E0: 00000000 00000000 00000000 0000A440 00000000 00000000 00000000 00000000 * .....$@..... *
000A00: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000A20: 00000000 00000000 00000000 00000000 00000000 00000000 0000A580 00000000 * .....%..... *
000A40: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000A60: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000A580 * .....%..... *
000A80: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000AA0- 000ABF ***SAME AS ABOVE *** * - SAME - *
000AC0: 00000000 00000000 0000A6C0 00000000 00000000 00000000 00000000 00000000 * .....&@..... *
000AE0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000B00: 00000000 00000000 00000000 0000A6C0 00000000 00000000 00000000 00000000 * .....&@..... *
000B20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000B40: 00000000 00000000 00000000 00000000 00000000 00000000 0000A800 00000000 * .....(..... *
000B60: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000B80: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000A800 * .....(..... *
000BA0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000BC0- 000BDF ***SAME AS ABOVE *** * - SAME - *
000BE0: 00000000 00000000 0000A940 00000000 00000000 00000000 00000000 00000000 * .....)@..... *
000C00: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000C20: 00000000 00000000 00000000 0000A940 00000000 00000000 00000000 00000000 * .....)@..... *
000C40: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000C60: 00000000 00000000 00000000 00000000 00000000 00000000 0000AA80 00000000 * ..... *
000C80: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000CA0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000AA80 * ..... *
000CC0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000CE0- 000CFF ***SAME AS ABOVE *** * - SAME - *
000D00: 00000000 00000000 0000ABCO 00000000 00000000 00000000 00000000 00000000 * .....+@..... *
000D20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000D40: 00000000 00000000 00000000 0000ABCO 00000000 00000000 00000000 00000000 * .....+@..... *
000D60: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000D80: 00000000 00000000 00000050 00000150 00000017 000C34E0 000006BC 00000710 * .....P...P.....4...<... *
000DA0: 00000000 00000000 00000006 00000064 000011CC 00000558 000040E0 00000448 * .....D...L...X...@...H *
000DC0: 00001670 000C3FE4 00007001 0001A4B0 00000000 00000004 00000560 00000000 * ...P...?D...P...$O... *
000DE0: 00000007 00002002 00024756 000D7060 30302C30 000FD330 430080B6 430080RE * .....GV...P 00,0..SOC..6C..> *
000E0C: 00007002 000080B6 00000000 00000000 00000000 00000000 00000000 00000000 * ..P...6..... *
000E20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000E40- 000EDF ***SAME AS ABOVE *** * - SAME - *
000EE0: 00000000 00000000 00000000 00000000 00000000 04000000 00000000 00000000 * ..... *
000F00: 00000000 00037B8C 00000009 000072E0 0000AF74 00007330 0001CACE 0001C65A * .....R.../T..SO..JN..FZ *
000F20: 000072E0 0000AF74 0001DOAC 000D6788 00000000 00000000 00000000 0000B700 * ..R.../T..P,..G.....7. *
000F40: 00004080 00000000 FFFFFB50 00000001 000D6788 00000000 000D4D10 00061058 * ..@.....P.....G.....M...X *
000F60: 00000050 000FC508 00027398 000D69A4 00000001 00000000 0002CCF0 0000002A * ...P...E...S...IS.....LP... *
000F80: 00007000 00007062 00000000 00000000 00000000 00000000 00000000 00000000 * ..P...PB..... *
000FA0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000FC0- 000FDF ***SAME AS ABOVE *** * - SAME - *
000FE0: 00000000 00000000 00000000 00000000 4220804C 42108100 58540018 48740016 * .....B..LB...XT...HT... *
001000: 9D234270 80F20733 4B340002 21322631 1132C370 00802333 CA30003C 4035002C * ..#BP.R..3K4...12&1.2CP...#3JO.<@5., *

```

48-031 F00 R01

48-031 F00 R01

001020:	C3700020	2338C860	10604064	0014C370	00802134	2468756#	00009477	9E271800	* CP. #8H . @D..CP..!4SHUD...W.'.. *
001040:	24697464	00162138	C860FF8C	40640000	C8600058	9E26C860	10604064	00141800	* SITD..!8H ..@D..H .X.&H . @D.... *
001060:	212F4210	80929D23	4270808C	9B25E354	00004220	8022C550	000D233F	18002468	* !/B....#BP...%CT..B ."EP...?#..SH *
001080:	76640000	E6608006	40640014	18002458	76540016	42308076	24305854	0018D235	* VD..F ..@D....\$[VT..RO.VSOXT..R5 *
0010A0:	00724835	002C2339	25314035	002C5855	00046450	4000B700	C85000C4	9E251800	* .RH5.,#9%1a5.,XU..DPa.7.HP.D.%.. *
0010C0:	C3300020	42308028	C5300008	2335C830	00844300	FFC45854	00182439	74340016	* CO. BO.(FO...#5HO..C..DXT..S9"4.. *
0010E0:	23384865	002E2761	208D4065	002E1800	C8300082	4300FFA2	C3300001	2135C530	* #8HE..*A .@E...HO..C..."CO..!5E0 *
001100:	00044230	FFBAC830	00A04300	FF8C2450	58640018	4556002C	4330FF9C	58560038	* ..BO.:HO. C...SPXD..EV...CO...XV.8 *
001120:	40540002	25524054	000A0755	4B540002	CA50003C	4056002C	C8500FF0	40540014	* @T..%R@T...UKT..JP.<@V..HP.P@T.. *
001140:	C850FF82	40540000	C8500064	9E251800	245B7554	00162305	C430007F	E3340000	* HP..@T..HP.D.%\$[UT..#.DO..C4.. *
001160:	25524054	000A4300	FEDE5854	00184864	00025565	00382334	27614064	00021800	* %R@T..C..^XT..HD..UE.8#4'A@D.... *
001180:	43004000	11504300	40001158	43004000	116A0000	00000000	00000000	00000000	* C.@..PC.@..XC.@..J..... *
0011A0:	DE20929E	DE20929E	58540018	4830929A	4035002C	24317435	01004230	804CD335	* ^ ..^ ..XT..HO..@5.,S1T5..RO.LS5 *
0011C0:	0070C330	00014230	8040C330	0020233F	41708642	7335009A	4873000A	4220802A	* .PCO..BO.@CO. #?AP.BS5..HS..B . *
0011E0:	48730016	4230813A	43008E76	E665009D	D3360000	C5300018	23394170	8618E665	* HS..BO.:C..VEE..S6..E0..#9AP..FE *
001200:	009D2470	24334300	81089B26	E670803E	D3350070	C3300001	2333E670	80824074	* ..\$PS3C...&FP.>S5.PCO..#3FP..@T *
001220:	00142435	74350100	42308FE4	9D23C330	00014230	8FDA2466	74650100	2335C330	* ..S5T5..BO.D.#CO..RO.ZSFE..#5CO *
001240:	00024230	888AC330	00080337	18009B26	08334230	8852C460	007FC560	00202383	* ..BO..CO...7...&3BO.RD .E .# . *
001260:	E7640010	417085AE	41708402	E3640000	42608058	18005854	00182432	74350100	* GD..AP..AP..CD..B .X..XT..S2"5.. *
001280:	42308C82	4300FFDC	417083A4	25314035	00C01800	4170857E	DA25009F	4300FF6A	* BO..C..^AP.S%1a5.@..AP..Z%..C..J *
0012A0:	9B260833	42308800	C460007F	41708566	24767475	00F62335	C560000D	43308828	* .&3BO..D ..AP.FSVTU.V#5E ..CO.(*
0012C0:	9A26E364	00004260	88221800	07335854	0018D235	00FA2461	76650100	586500EC	* .&CD..B ."...3XT..R5.ZSAVE..XF.L *
0012E0:	D3750104	D2760000	487500F4	26710B67	4300801E	58540018	24715175	00A45865	* SU..RV..HU.TEQ.GC...XT..SQU..SXE *
001300:	00ECD375	0102D276	00002761	487500F4	0B677345	009A4034	00164074	00A5064	* .LSU..RV..'AHU.T.GSE..@4..@T..PD *
001320:	000C7345	0098E660	800C4064	00147345	009A4300	80D69B26	C3300001	42308ED0	* ..SE..F ..@D..SE..C..V.&CO..BO.P *
001340:	C3300002	42308788	18009B26	0833203C	C460007F	58540018	C560007F	43308044	* CO..BO.....&3 <D ..XT..F ..CO.D *
001360:	C5600020	238D0876	11715A74	00104877	00001171	F5700000	12764230	80264875	* E .#..V.QZT..HW...QUP...VRO.&HU *
001380:	01102671	C570000F	238E4075	01104875	01142771	2312247D	40750114	D2654700	* ..EQEP..#.aU..HU..'Q#\$.S.aU..REG. *
0013A0:	01161800	E660FF8E	40640014	18002721	DE20908E	58540018	24667465	01002334	* ..F ..@D...!'^ ..XT..SFE..#4 *
0013C0:	9D234220	870A2463	746500F6	233E9D23	232C4170	849E2302	230F0766	40650072	* .#B ..SCTE.V#>..#.#,AP..#.F@F.R *
0013E0:	50650038	430087BC	07664B64	00021062	26644065	002C9B26	C460007F	C5600014	* PE.8C.<.FKD...B&D@E.,&D .E . *
001400:	43308194	C5600013	4330818C	2621DE20	9032E660	80184064	00140834	73450098	* CO..E ..CO..&1^ .2F ..@D...4SE.. *
001420:	24657465	01004230	8DE60843	9D230833	23321800	58540018	24377435	01002337	* SETE..BO.F.C.#.3#2..XT..S7T5..#7 *
001440:	41708450	24377635	01001800	E660802E	40640014	24687564	0000246E	74640000	* AP.PS7V5...F ..@D...SHUD..SWP.. *
001460:	23394864	00025A64	0004D336	0000E734	0010E364	00004260	80449A26	18004220	* #9HD..ZD..S6..G4..CD..B .D.&.B *
001480:	803E2335	08334230	861E1800	58540018	24617665	00AC4330	86524875	00P04075	* .>#5.3BO...XT..SAVE..CO.RHU.P@U *
0014A0:	00ACD375	00F2D275	00AE4170	83E62478	76740000	41704000	F1244300	862E9A26	* .,SU.RRU..AP.FSXVT..AP@.QSC>...& *
0014C0:	48640016	42308022	58540018	D3750070	C3700001	2137247E	76740000	4874000A	* HD..BO.."XT..SU.PCP..!7S.VT..HT.. *
0014E0:	23242478	76740000	1800C860	FF0C4064	00004170	4000F124	24624564	00164330	* #SSXVT...H ..@D..AP@.QSSBFD..CO *
001500:	87EC5854	00182721	E6604000	302C4064	00140834	73450098	24643563	00164330	* .LXT..'!F @.0,@D...4SE..SCEC..CO *
001520:	FCE84300	FD6E29B26	2621C460	007F5854	0018C560	000E4330	88445660	00144330	* .HC..B.&&!D ..XT..E ..CO.DE ..CO *
001540:	80A2C560	00134330	809AC560	001B4330	85680833	42308550	C560000F	21352461	* ."E ..CO..E ..CO.H.3BO.PE ..!5SA *
001560:	77650100	18002470	74750100	2337C560	000D4330	808A4300	8514C560	00124330	* WE...SPTU..#7E ..CO..C..E ..CO *
001580:	807EC560	00114330	80764855	002CC550	7FFF4330	84F81800	C8707FFF	4075002C	* .E ..CO.VHU',EP..CO.X..HP..aU., *
0015A0:	73450098	41704000	F1249B26	08334230	84F6C460	007FC560	001B4330	84FCC560	* SE..AP@.QS.&.3BO.VD .E ..CO..E *
0015C0:	00122335	C5600011	4230849A	E660FF56	40640014	58540018	4170803E	7345009A	* ..#5E ..BO..F .V@d..XT..AP.>SF.. *
0015E0:	4300FE28	DE208E5E	C8607FFF	E67086BE	58540018	4835002C	03374065	002C0307	* C..(^ .^H ..FP.>XT..H5..7@F... *
001600:	DE208E40	E67086A6	58540018	24607665	0100E660	FF104064	00145854	00187345	* ^ .@FP.&XT..S VE..F ..@D..XT..SE *
001620:	009A0766	4B640002	10622664	4300FFC0	58540018	24327435	0100E630	88821333	* ...FKD...B&D@..@XT..S2T5..FO.2!3 *
001640:	E630FC0A	40340014	7345009A	E6304000	302C4034	0014DE20	8DEC7345	00980207	* FO..@4..SE..FO@.0.@4..@.LSF... *
001660:	58540018	585500E8	D3550001	23022551	48340016	211BC560	007F2338	C5600020	* XT..XU.HSU..#.Q%4..!.E ..#8E . *
001680:	2389C560	00094330	80405854	00182531	03070855	21172531	C3507F00	21359A25	* #.E ..CO.@XT..%1...!%1CP..!5.% *
0016A0:	23039A26	24315854	00185135	00B45835	00B45535	00B02183	503500B0	453500A2	* #..&S1XT..QS.4X5.4U5.0!P5.OF5." *
0016C0:	23837635	01620733	03074054	000E5854	00182430	743500AC	4330FFAE	48340014	* #.V5.B.3..@T..XT..SOT5..CO..H4.. *
0016E0:	403500C6	F5300000	134A2335	E630FC46	40340014	4074000C	DE208D48	4834000E	* @5.FUO...J#5FO.F@4..@T..^ .HH4.. *
001700:	42108076	587500E8	D3670001	11684065	00C4C330	7F004330	801E5575	00E0213D	* B..VXU.HSG...H@E.DCO..CO..!!U. != *
001720:	D36740FF	FFFFC660	20004065	00C4DA27	40FFFFFD	4170808A	583500B4	27315035	* S@a...F .@E.DZ.'@...AP..X5.4'1P5 *

C-9

001740:	00B4DA25	00C44535	00A22387	74350162	21344170	806C220F	D33500C5	08332335	* .4Z%.DE5."#.T5.BI4AP.L".S5.E.3#5 *
001760:	4170805E	DA2500C5	487500C6	40740014	7374000C	24692431	03075835	00B44535	* AP.^Z%.EHU.FaT..ST..SIS1..X5.4E5 *
001780:	00A22383	75350162	C8600020	9A265835	00B42631	503500B4	553500B0	21835035	* ."#.U5.BH . .EX5.4E1P5.4U5.0I.P5 *
0017A0:	00B04535	00A24380	FFBE7635	01625875	00D87437	00004230	FFAE4170	80044300	* .0E5."C.>V5.BXU.XT7..B0..AP..C. *
0017C0:	FFC64074	000A7345	009A7364	00144065	00BC4170	4000F124	C5300008	22355854	* .FaT..SE..SE..SE..@E.<AP@.QSF0.. "5XT *
0017E0:	00187365	00BC4064	00147345	00982721	7374000A	03074074	000A4170	4000F124	* ..SE.<ad..SD..!ST....@T..APa.QS *
001800:	9B260833	423082A0	C460007F	58540018	7374000A	03075854	00182437	76350100	* .E.3B0. D ..XT..ST....XT...S7V5.. *
001820:	03372435	74350100	02374074	000AD264	00087345	009A2621	DE208C08	246A9A26	* .7S5T5...7aT..RD..SE..E!^ .S.SJ.E *
001840:	41704000	F124C530	00082235	DE208BF6	E6604000	302C4064	00145854	00182461	* APa.QSE0.."5^ .VF @.0.@D..YT..SA *
001860:	516500A4	73450098	2721D364	00087374	000A0307	D3650070	C4600021	C5600021	* QE.SSE..!SD..ST....SE.PD .!F .!
001880:	42370002	5835006C	24617463	00144337	00020307	48640000	40640020	40740022	* B7..X5.LSATC..C7...HD..ad. @T." *
0018A0:	58540018	D3650103	D37500F8	C3700040	2133D365	01022470	24337635	00AC4230	* XT..SE..SU.XCP.@13SE..SPS3V5.,B0 *
0018C0:	80365875	00A4D365	01042439	76350100	42308024	D3650102	587500A4	26715575	* .6XU.SSE..S9V5..R0.SSE..XU.S&QUU *
0018E0:	0090218B	58350090	23382470	24307435	001C2133	D3650103	507500A4	587500EC	* ..I.X5..#8SPS0T5..!3SE..PU.SXU.L *
001900:	D2670300	483500F4	26314034	000A0B73	5074000C	C870FF8C	40740000	D37500F9	* RG..H5.TE1a4...SPT..HP..aT..SU.Y *
001920:	08774330	8062C460	007FC560	000A4230	8056D365	00FA2661	D26500FA	05764380	* .WCO.BD ..E ..BO.VSE.Z&ARE.Z.VC. *
001940:	80462460	75650100	48608AFC	4170FCA0	9D23238B	24687664	00004170	4000F124	* .FS UE..H ..AP..##.SHVD..APa.OS *
001960:	24687564	0000220B	24679A26	DE208AD6	41704000	F1244210	812E5854	00182461	* \$HUD.."SG.E^ .VAPa.E^ .YAPa.OSB..YT..SA *
001980:	D26500FA	43008028	24640B63	10624A65	002C4170	FC5A9D23	2185E364	00002168	* RE.ZC..(SD.C.BJE.,AP.Z.#!CD..!H *
0019A0:	9A264170	4000F124	421080FC	23029A26	58540018	D36500AE	27612665	00AE4220	* .EAPa.QSB...#.EXT..SE..!AR^..R *
0019C0:	FF12E660	FAB84064	00145865	00ECD375	0104D276	00004875	00742671	0B675064	* ..F Z8aD..XE.LSU..RV..HU.TEQ.GPD *
0019E0:	000C4074	000A4864	0020C660	00804064	00004874	00220307	E630F93A	40340014	* ..aT..HD. F ..ad..HT..T.FQ:44.. *
001A00:	DE208A40	C6200001	9D23C330	00084230	80204834	00025A34	0004D363	0000E364	* ^ .aF ...#CO..B0. H4..Z4..SC..CD *
001A20:	00004170	FC482316	586500EC	D3660001	9A264170	FD8C4834	00024534	00084230	* ..AP.H#.XE.LSF...EAP..H4..E4..B0 *
001A40:	FFD42479	76750100	42308218	4300F838	9B26C330	00014230	87B6C330	00202334	* .TSYVU..R0..C.XB.ECO..B0.6CO. #4 *
001A60:	08664330	802E2531	C460007F	C560000E	4330830A	C560000F	43308376	C560001B	* .FCO..%1D ..E ..CO..E ..CO.VF .. *
001A80:	233AC560	00112184	C5600015	21832467	9A261800	58540018	24717475	00F62036	* #:E ..!E ..!SG..E^..XT..SQTU.V 6 *
001AA0:	247A7575	01001800	C3300001	42308760	C3300020	233E0866	213C5854	00182461	* SZUU....CO..B0. CO. #>F!<XT..SA *
001ACC:	746500F6	4230FA20	C8608200	43008046	C8608400	C42003FE	9D274220	8038C330	* TE.VBOZ H ..C..FH ..D ...!B .8CO *
001AE0:	00084330	FFD41800	F3640000	07665854	0018D375	0070C370	0001213E	487500AC	* ..CO.T..CD...FXT..SU.PCP..>HU., *
001B00:	C470C000	C570C000	23372477	75750100	2471D275	00AE5854	00184065	00720866	* DPa.EPa.#7SWUU..SORU..XT..aE.R.F *
001B20:	233DD235	00730733	D23500FA	24307635	01002334	24377635	01000866	42308064	* #R5.S.3R5.ZSOV5..#4S7V5...FR0.D *
001B40:	24667565	01004170	FD2A2303	43008054	4170822E	24607465	00F64330	80462561	* SFUE..AP.#.C..TAP..S TE.VCO.F7A *
001B60:	9A264170	4000F124	58540018	24667665	01002463	75650100	24607665	01002463	* .EAPa.QSXT..SFVE..SVE..S VE..SC *
001B80:	4170FA6C	C42003FE	E6604000	302C4062	420000D0	40624200	00D2DE20	88A5DE20	* APZLD ..F @.0.@BB..P@BB..R^ .% ^ *
001BA0:	8BA31800	C5608400	4330FFC6	4835002C	213ED335	0070C330	00404330	801E5835	* .#.E ..CO.FH5.,!>S5.PCO.aCO..X5 *
001BC0:	000CF330	000C0000	213A2306	58650004	64604000	B7002531	4035002C	C42003FE	* ..S0...!:#.XE..D @.7.%1a5.,D .. *
001BE0:	E6304000	302C4032	420000D2	E630FE60	73450098	40340014	26414042	420000D0	* F0a.0.@2B..RF0. SE..a4..Ea@BB..P *
001C00:	DE20883E	9B262621	DE20883A	18005854	00184865	003A4874	00024565	002E2134	* ^ .>.E&E!^ .:..XT..HE.:HT..EE..!4 *
001C20:	05674330	FE684065	002E4064	00024835	00F84034	00160733	503500B4	503500B0	* .GC0.HaE..ad..H5.X@a4...3P5.4P5.0 *
001C40:	24327635	01002431	4300F6A8	58540018	48740002	4975002E	23162479	75750100	* \$2V5..S1C.V(XT..HT..IU..#.SYUU.. *
001C60:	430083FE	246DE364	00004300	F65E2470	23022571	58540018	48640002	4565003A	* C...SNCD..C.V^SP.#QXT..HD..EE.: *
001C80:	4330FE0A	4965002E	21174065	002E6175	002E5175	00B02761	40640002	5A640004	* CO..IE..!@E..AU..QU.0^AaD..ZD.. *
001CA0:	0877D376	00002115	08674170	F9B21800	C8300020	D2360000	0867C850	F7004170	* .WSV..!..GAPY2..HO. R6..GHP..AP *
001CC0:	F9AE203A	583500E4	D2630000	586500E0	D3360000	27610777	0B732431	4300F632	* Y. :X5.DRC..XE. S6..'A.W.SS1C.V2 *
001CE0:	48340016	4210FDA6	24324300	F6065854	00187345	00984864	00024875	003A0976	* H4..B..E&S2C.VXT..SE..HD..HU..:V *
001D00:	4330F584	40640008	40740002	07775075	00B44300	FCE24170	FADCU170	F9124300	* COU.ad..aT..T..WPU.4C..BAPZ\APY.C. *
001D20:	F5545854	00182462	74650100	4230FD5E	07667465	00F84230	FD547764	00161800	* UTXT..SBTE..BO.^.FTE.XBO.TWD... *
001D40:	4170FAB2	D265009D	4300F53C	48640002	58540018	4965002E	4310FD32	5A640004	* APZ2RE..C.UKHD..XT..IE..C..2ZD.. *
001D60:	D3660000	4170F90E	E3640000	48640002	4965000E	21142462	76650100	1800E670	* SF..APY.CD..HD..IE..!SBVE...FF *
001D80:	FFFA5854	0018D365	00FBD265	009D2468	583500E8	D2630001	D2630003	D36500F8	* .ZXT..SE..RF..SHX5.HRC..RC..SE.X *
001DA0:	C4600004	0A365035	00E00766	D26500F9	D33500F8	C4300001	21382436	D4350105	* D ...6P5. .FRE.YS5.XD0..!8S675.. *
001DC0:	2184D335	01052631	0B634065	00FA4261	76650100	03074170	FA1C9A26	4300F4A8	* !.S5..E1.CaE.TSAVE...APZ..eC.T(*
001DE0:	4170FA12	D26500F9	0766D265	00FA4300	F4965854	00182461	77650100	23372631	* APZ.RE.Y.FRE.ZC.T.XT..SAWE..#7E1 *
001E00:	23354170	FA10DA25	009F1800	48740016	4210FC7A	58540018	48640002	4565003A	* #5APZ.Z%...HT..B..ZXT..HD..EE.: *
001E20:	4330FC6A	4170F9CE	406500C0	48740016	4210FC5A	58540018	48640002	4565003A	* CO.JAPYN@aHT..B..ZXT..HD..EE.: *
001E40:	4330FC4A	4965002E	21134065	002EE660	F4E44064	00147345	009A5660	80084064	* CO.JIE..!@E..F TDaD..SE..F ..ad *

C-10

48-031 F00 R01

48-031 F00 R01

001E60:	0014DE20	85DE0833	4230FC3C	58540018	73450098	48640000	4565003A	4330F682	* ..^ .^3BO.<XT..SE..HD..FE.:COV. *
001E80:	27614064	00025A64	0004D366	00004170	F7CE4875	00C02114	0567233D	23047460	* 'AaD..ZD..SF..APNNU.?!..G#=#.T *
001EA0:	85902339	08332316	586500EC	D3660001	9A261800	58540018	7345009A	07664064	* ..#9.3#.XE.LSF...&.XT..SE...FAD *
001EC0:	00160833	4210F63A	4300F61E	48640016	4210FBBA	58540018	48640002	4965002E	* ...3B.V:C.V.HD..B...:XT..HD..IE.. *
001EE0:	4310FBAA	24627765	01004170	F7421800	9B260833	4230FB80	C460007F	C5600020	* C...SBWE..APWB...&.3BO.OD ..E *
001F00:	2383E764	00105854	00182430	50350110	40350114	50350124	243D4035	0128D265	* #.GD..XT..SOP5..a5..P5.\$S=a5.(RE *
001F20:	4300012A	48340002	26314034	00082431	6135002E	E630F412	40340014	7335009A	* C...*H4..&1a4..s1A5..FOT.a4..S5.. *
001F40:	E670800A	40730014	DE2084F8	23090833	4230FB54	27215854	00187345	00984865	* FP..aS..^ .X#..3BO.T'!XT..SF..HE *
001F60:	01262761	2312246D	40650126	D3654600	012A4874	00025A74	0004D337	00004875	* .&'A#.sM@a.E&SEF..*HT..ZT..S7..HU *
001F80:	01282771	2312247D	40750128	D2354700	012A4170	F6D8E364	00004260	F32E4874	* .('Q#..S.aU.(R5G...*APVXCD..B.S.HT *
001FA0:	00024575	002E2335	08334210	FFFA1800	7345009A	E670800A	40740014	08332218	* ..EU..#5.3B..Z..SE..FP..aT...3". *
001FC0:	23070833	4230FAE0	27215854	00187345	00984864	00024564	0008233D	27614064	* #..3BOZ '!XT..SE..HD..ED..#='AaD *
001FE0:	00025A64	0004D366	00004170	F6724210	FEB61800	48750110	4330F28C	58350110	* ..ZD..SF..APVRB..6..HU..COR.X5.. *
002000:	50350124	58350114	50350128	58350118	5035012C	5835011C	50350130	58350120	* P5.SX5..P5.(X5..P5..X5..P5.OX5. *
002020:	50350134	24305035	01104035	01144834	00020A37	40340008	6175002E	4320FEFC	* P5.4SOP5..a5..H4...7a4..AU..C .. *
002040:	4875002E	27714330	FEF24835	01280A37	40350128	273E4210	FEE24035	01284300	* HU..*QCO.RH5.(.7a5.('B..B@a5.(C. *
002060:	FEDA4874	00025854	00184865	002E0976	4310FA1A	24327635	01004834	00162319	* .ZHT..XT..HE...VC.Z.S2V5..H4..#.. *
002080:	40640002	24797675	01004230	FBD61800	40640008	4300F960	48740016	4210F9EE	* aD..sYVU..BO.V..aD..C.Y HT..B.YN *
0020A0:	58540018	24727475	01004230	F9E04875	01104310	F9D84170	F75CE670	F2784074	* XT..SRTU..BOY HU..C.YXAPW\FPRXaT *
0020C0:	0014DE20	837E2571	23034875	00C02671	407500C0	D3654700	0111C560	000D4330	* ..^ ..%Q#..HU.a&qQaU.aSEG...E ..CO *
0020E0:	F1A6E364	0000216D	4170F582	23165865	00ECD366	00019A26	4170F6C6	4300FFCA	* Q&CD..!MAPU.#.XE.LSF...&APVFC..J *
002100:	25716174	00022467	9A264170	F6B44300	F1764864	00164210	F9745854	00184864	* %QAT..SG.&APV4C.QVHD..B.YTXT..HD *
002120:	00024875	002E0967	4310F962	4575003A	4330F95A	277114075	002E0967	21142432	* ..HU...GC.YBEU..COYZ'QaU...G1.S2 *
002140:	76350100	E630F1EE	40340014	48340002	40340008	7345009A	E6308010	40340014	* V5..FOQNa4..H4..a4..SE..FO..a4.. *
002160:	DE2082E0	24304035	01102306	08334230	F9365854	00187345	00984874	00024575	* ^ .s@a5..#..3BOY6XT..SE..HT..EU *
002180:	002E4330	801E2671	40740002	5A740004	D3670000	D26740FF	FFFF4170	F4D04210	* ..CO..&q@aT..ZT..SG..R@a...APTpa. *
0021A0:	FD061800	5A740004	C8600020	D2670000	25315875	00B05B75	00B44320	FD25075	* ...ZT..H .RG..%1XU.0IU.H.C.RPU *
0021C0:	00B8C860	00204170	F4A44170	F5F45875	00B45575	00B0208A	C8600020	4170F480	* .8H . APTSAPUTXU.4UU.0.4C . APT. *
0021E0:	25715175	00B82324	4170F5D6	220A5875	00B45075	00B00733	4300FDB4	58540018	* %QQU.#\$SAPUV".XU.4PU.0.3C..4XT.. *
002200:	24727475	00F64330	F06CC830	C0002303	C8304000	73640014	4064000C	E6608040	* SRTU.VCOPLHOa.#.HOa.SD..aD..F .a *
002220:	40640014	73700062	486700A6	4063000E	9B265854	00182460	70650100	24267565	* aD..SP.BHG.&aD...&XT..S VE..SEUE *
002240:	01004865	009CC460	3FFF0663	4065009C	DE2081EE	DE2081EE	DE25009C	4300FB1E	* ..HE..D ?..CaE..^ .N^ .N%^..C... *
002260:	9B260873	C8304000	C370000B	4230FFB4	08774230	802AC460	007FC560	000D4230	* .&.SHOa.CP..RO.4.WB0.*D ..E .BO *
002280:	801E7364	0004C064	0014DE20	81865854	00182465	76650100	C860002A	9A261800	* ..SD..aD..^ .6XT..SEVE..H .*&.. *
0022A0:	73700062	486700A6	4564000E	22374064	000E5854	00184865	009CF660	000F0000	* SP.BHG.&ED..7aD..XT..HE..V *
0022C0:	CB604000	4065009C	DE25009C	18005854	0018D265	01042471	74750100	233C2477	* K a.aE..^%....XT..RE..SQTU..#<SW *
0022E0:	75750100	2471D275	00AE2471	767500AC	4300806C	24797575	01004864	00022661	* UU..SQRU..SQVU..C..LSYUU..HD..EA *
002300:	506500A8	43008028	58540018	D2650102	24707475	00AC233D	48640002	26615065	* PE.(C..(XT..RE..SPTU..#HD..&APE *
002320:	00A8246D	24717475	01004230	FFB09A26	2471D275	00AE4170	F55AE364	00004260	* .(SMSQTU..BO.O.&SQRU..APUZCD..B *
002340:	80201800	58540018	D2650103	24737575	00AC4300	FFBAE364	00002163	9A261800	* ..XT..RE..SSUU..C..:CD..!C.E.. *
002360:	9A260733	73740014	03075854	00182470	747500AC	223F2478	74750100	4330FFCC	* .&.3ST...XT..SPTU..%SXTU..CO.L *
002380:	48640002	26615065	00A8246D	24717475	01004230	FFA89A26	2471D275	00AE4170	* HD..&APE.(SMSQTU..BO.H.&SQRT..AP *
0023A0:	F4F25865	00DC5875	00A4A475	00A04746	00004230	FF84220F	58540018	24707475	* TRXE.\XU.SJU. TV..BO.."XT..SPTU *
0023C0:	00AC4330	FF904874	00025B75	00A84074	00204874	00144074	0022C860	0020E334	* ..CO..HT..IU.(aT..HT..aT..H .C4 *
0023E0:	00004260	FF7AC860	00209A26	48740020	26714074	00204575	00A2F380	FF605865	* ..B.ZH . &HT. &q@aT. EU."C.. XE *
002400:	00D87476	00004230	FF542571	517500A8	24787674	00004170	4000F124	58540018	* XTV..BO.T%QU.(SXVT..APa.QSXT.. *
002420:	48740022	40740014	24787574	00004300	FFB40000	00000000	FFC07FFF	FFE07FFF	* HT."aT..SXUT..C..4.....a..... *
002440:	FFE061A1	6383A323	012C000F	00000000	58540018	4865002C	233BD235	00732561	* .AIC.#.#.,.....XT..HE.,#;R5.S%A *
002460:	4065002C	58650004	64604000	B700E660	4000302C	40624200	00D01800	58540018	* aE.,XE..D a.7.F a.O,aBB..P..XT.. *
002480:	4835009C	4825001A	24744075	002C2641	40434300	00D02741	E670806C	40740014	* H5..H%.STaU.,&a@aCC..P'AFP.LaT.. *
0024A0:	DE308090	DA350039	D835003A	DA350079	D835007A	4865009E	4874000A	43308034	* ^O..Z5.9X5.:Z5.YX5.ZHE..HT..CO.4 *
0024C0:	D8240006	DA640003	24717474	0010233A	9577C470	B7FF9557	DF608054	DE308051	* XS..ZD..SQT..#:WDP7..W^ .T^O.Q *
0024E0:	18009577	C470B7FF	9557DE60	8044DE30	80411800	DA640003	D3740005	11744674	* ...WDP7..W^ .D^O.A..ZD..ST..!ZFT *
002500:	00069867	4300FFC0	58540018	4525009C	4230FF40	DE20801C	4824000A	21321800	* ..GC..aXT..E%..BO.a^'.HS...!2.. *
002520:	4825009E	9D23C330	00804230	FF261800	41304210	48002000	00000000	00000000	* H%...#CO..BO.&..AOB.H. *
002540:	58540018	4865002C	433080F6	403503C6	C3300001	423080FC	430080D6	42000000	* XT..HE..CO.V@a5.FCO..BO..C..VB... *
002560:	58540018	4835009C	4865001A	487503F4	4075002C	9D674075	03C6C370	007B4230	* XT..H5..HE..HU.TaU..,GaU.FCP..BO *

C-11

C-12

48-031 F00 R01

002580:	80D2E674	00014073	430000D0	E6708060	40740014	DE3080C2	DA3503C9	D83503CA	* .RFT..@SC..PFP. @T..^0.BZ5.IX5.J *
0025A0:	DA3503CD	D83503CE	D37503E2	C3700040	2337D374	0016C370	000C2332	9E672470	* Z5.MX5.NSU.BCP.@#7ST..CP..#2.GSP *
0025C0:	9E27DA25	03DDD375	03DF9867	117A4675	03E09827	D37503E2	10761171	48678076	* .%Z%.JSU.<.G.ZFU. .'SU.B.V.QHG.V *
0025E0:	9577C470	B7FF9557	9E269466	9E361800	58540018	4865002C	43308046	403503D6	* .WDP7..W.E.F.6..XT.F..HE.,CO.F@5.V *
002600:	DE208056	E6608008	40640014	43008038	58540018	4865002C	43308026	403503D2	* ^ .VF ..@D..C..8XT..HE.,CO.E@5.R *
002620:	C3300002	4330802C	4835001A	9D364065	03C62561	4065002C	58650004	64604000	* CO..CO.,H5...6@E.FX@E.,XE..D @.
002640:	B700C860	00C09E26	F6604000	302C4062	420000D0	18001442	34414C00	42000000	* 7.H .@.EF @.0,@BB..P...B4AL.R... *
002660:	41308210	4835009C	DE3081D0	41704000	F12A9577	C470B7FF	9567DE25	00F6DE25	* 'AO..H5..^O.PAP@.Q*.WDP7..G^V.V% *
002680:	00FBDE30	81B44170	4000F124	41704000	F1184525	009C2332	2209DE20	819E4865	* ..^O.4AP@.QSAP@.Q.E%..#2".^ ..HE *
0026A0:	001A9D63	C3300011	213E4170	4000F124	41704000	F1184525	001A2134	C3300011	* ...CCO..!>AP@.QSAP@.Q.E%..!4CO... *
0026C0:	21321800	D2350073	4825001A	DE2500F8	41708194	4130819C	4835009C	DE30815C	* !2..R5.SH%..^%.XAP..AO..H5..^O.\ *
0026E0:	41704000	F1529577	C470B7FF	95672469	746500E4	23367665	00E4C860	004B9E26	* AP@.QR.WDP7..GSTE.D#6VE.DH .K.E *
002700:	DE2500F6	DE25C0FD	DE30812F	4160817E	41704000	F1244170	4000F118	4525009C	* ^%.V^%.^O./A ..AP@.QSAP@.Q.E%.. *
002720:	23322209	4865001A	9D63C330	0011213E	41704000	F1244170	4000F118	4525001A	* #2".HE...CCO..!>AP@.QSAP@.Q.E%.. *
002740:	2134C330	00112132	18004825	001AD235	0073DE25	00F84170	810E4170	80E04170	* !4CO..!2..H%..R5.S^%.XAP..AP. AP *
002760:	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	00F84170	80EA4170	* @.QSAP@.Q.CO..!2..R5.S^%.XAP.JAP *
002780:	80D24170	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	00F84170	* .RAP@.QSAP@.Q.CO..!2..R5.S^%.XAP *
0027A0:	80C64170	80984170	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	* .FAP..AP@.QSAP@.Q.CO..!2..R5.S^% *
0027C0:	00F84170	80A24170	80744170	4000F124	41704000	F118C330	00112132	1800D235	* .XAP."AP.TAP@.QSAP@.Q.CO..!2..R5 *
0027E0:	0073DE25	00F84170	807E4170	80504160	809C4170	4000F124	41704000	F118D235	* .S^%.XAP..AP.PA ..AP@.QSAP@.Q.R5 *
002800:	0073C330	00112132	1800DE25	00F84170	80564170	80284170	4000F124	41704000	* .SCO..!2..^%.XAP.VAP.(AP@.QSAP@. *
002820:	F118C330	00132132	1800D235	0073DE25	00F84170	8032E1E2	20C03414	48005854	* Q.CO..!2..R5.S^%.XAP.2AB @4.H.XT *
002840:	0018DE25	00F64865	00EE4065	002CDE25	00F40307	58540018	DE2500F8	24614065	* ..^%.VHE.N@E.,^%.T..XT..^%.VSA@E *
002860:	002CDE25	00F40307	58650004	64604000	B7001800	58540018	24605875	00785B75	* .,^%.T..XE..D @.7...XT..S XU.XIU *
002880:	00745D65	01BC2679	4075002C	0303D375	0019C570	00440386	9D27C370	00100336	* .T]E.<EY@U.,..SU..EP.D...^CP..6 *
0028A0:	24777575	00E42471	4075002C	58540018	DE2500F8	24614065	002C4170	4000F124	* SWUU.DSQ@U.,XT..^%.XSA@F..AP@.Q5 *
0028C0:	41704000	F1181800	58540018	C8607080	4D6500E2	26724075	002CC330	0010213D	* AP@.Q...XT..H P.HE.B@R@U.,CO..! = *
0028E0:	DE2500F6	41704000	F1244170	4000F118	C3300011	21321800	40350072	DE2500F8	* ^%.VAP@.QSAP@.Q.CO..!2..@5.R^%.X *
002900:	4170FF64	00000000	00000000	00000000	58540018	C860001E	4065002C	C5300004	* AP.D.....^%.XT..H .@E.,EO... *
002920:	43308042	C3300021	4230803A	C3300010	213FC860	00489E26	9D23C330	00102136	* CO.BCO.!BO.:CO..!?H .H.E.#CO..!6 *
002940:	E670FFD8	40740014	1800DE20	80167374	000A4074	00144874	000E4074	0000DE24	* FP.X@T.....^ ..ST..@T..HT..@T..^S *
002960:	000C1800	C800C870	00A02303	C8700084	58540018	D2750072	43008024	E670800A	*H.HP. #.HP..XT..RU.RC..SFP.. *
002980:	40740014	24787674	00009D23	C33000C1	4230FFD8	C3300002	21349B26	9B261800	* @T..SXVT...#CO.ABO.XCO..!4.E.E.. *
0029A0:	58540018	4875002C	23392571	4070002C	58750004	64704000	B700DE20	FFA61800	* XT..HU.,#9Q@P.,XU..DP@.7.^ .E.. *
0029C0:	C33000D1	4230FFA4	C3300008	0337C530	00084330	FF961800	4170FFE4	23030833	* CO.QBO.SCO...7EO..CO...AP.D#..3 *
0029E0:	20349B25	9B260855	43308044	24770475	43308040	73774700	2A7A4330	80502774	* 4.%E.#CO.DSWG.UCO.@SWG.^%CO.P'T *
002A00:	C360003C	42308046	C4500038	94669062	C56000FF	43808036	06560657	58740010	* C .<BO.FDP.8.F.BE ..C..6.V.WXT.. *
002A20:	D3754700	0000E374	00004220	FF4E1800	0866223B	0876C470	003C4330	FFCAC460	* SUG...CT..B .N...F"; VDP.<CO.JD *
002A40:	00039071	73778032	08774230	FFBAC870	00845854	0018D275	0072D375	00730877	* ...QSW.2.WBO.:HP..XT..RU.RSU.S.W *
002A60:	213A4874	00025A74	00045B75	00382671	D2750073	24704300	FFAC0000	00070006	* !:HT..ZT..IU.8&QRU.SSPC.,..... *
002A80:	00000005	00000000	00000004	00000000	00000000	00000000	00000000	00000833	*3 *
002AA0:	42308052	9B259156	9B260656	90544064	000AE354	00004220	FEC2E670	80064074	* BO.R.%V.E.V.T@D..CT..B .BFP..@T *
002AC0:	00141800	08334230	80349B26	7354000A	91560656	9052E354	00004220	FE9E9166	*3BO.4.EST...V.V.RCT..B ...F *
002AE0:	9B250656	E3540000	4220FE90	E670FFAE	40740014	18004170	FEC64300	FFA64170	* .%VCT..B .FP..@T...AP.FC..&AP *
002B00:	FEBE4300	FFC44170	FE62303	08332034	9B259B26	E3540000	4220FE60	E3640000	* .>C..DAP.6#..3 4.%E.CT..B .CD.. *
002B20:	4220FE58	18004220	FE524170	FE929B26	E3640000	4220FE44	18000000	00000000	* B .X..B .RAP..&CD..B .D..... *
002B40:	58540018	D2350073	C3300008	4330804C	C8700040	9E279D23	D2350073	23844170	* XT..R5.SCO..CO.LHP.@.'.#R5.#S.AF *
002B60:	4000F124	41704000	F118D235	0073C330	00084330	8020C530	0008213E	7375002E	* @.QSAP@.Q.R5.SCO..CO. EO..!>SU.. *
002B80:	26714575	00D22384	4075002E	1800C830	0082D235	0073C670	00C09E27	25714075	* &QEU.R#@U...HO..R5.S^%Q@U *
002BA0:	002C5875	00046470	4000B700	41704000	F1241800	58540018	DE208052	DA240005	* .,XU..DP@.7.AP@.QS..XT..^ .RZS.. *
002BC0:	D8240006	DA24000D	D824000E	9577C470	B7FF9567	4865001A	C87000C0	9E67DE20	* XS.ZS..XS..WDP7..GHE..HP.@.G^ *
002BE0:	802D4170	4000F124	41704000	F118DE20	801CD235	00732571	4075002C	58750004	* .-AP@.QSAP@.Q.^ ..R5.S^%Q@U.,XU.. *
002C00:	64704000	B7004170	4000F124	18004810	C860005F	C8700080	430080DE	C860007F	* DP@.7.AP@.QS..H.H .<HP..C..^H .. *
002C20:	C8700084	430080D2	C860001F	430080C6	C860001E	430080BE	C860001D	430080BE	* HP..C..PH .C..FH .C..>H .C..6 *
002C40:	C860001C	430080AE	C860001B	430080A6	C860001A	4300809E	C8600019	43008096	* H ..C...H .C..EH .C...H .C... *
002C60:	C8600018	4300808E	C8600017	43008086	C8600016	4300807E	C8600015	43008076	* H ..C...H .C...H .C...H .C..V *
002C80:	C8600014	4300808E	C8600013	43008066	C8600012	4300805E	C8600011	43008056	* H ..C..NH .C..FH .C...^H .C..V *

002CA0:	C8600010	4300804E	24674300	80482466	43008042	24654300	803C2464	43008036	* H ..C..NSGC..HSFC..BSEC..<SDC..6 *
002CC0:	24644300	80302462	4300802A	24614300	80242460	4300801E	246F230E	246E230C	* SDC..OSBC..*SAC..SS C...SO#.SN#.
002CE0:	2468230A	24692308	246A2306	246B2304	246C2302	246D0876	11725A74	001C5877	* SH#.SI#.SJ#.SK#.SL#.SM.V.RZT..XW *
002D00:	00000237	E2000000	05010000	00000000	41C08288	43004001	A5B2C4E0	FFF041C0	* ...7B.....A@..C.@.%2D .PA@ *
002D20:	82A658C9	005C4230	8022E6BA	00134180	4002118E	42A04001	A57A08CD	08DA4180	* .EXI.\BO."F:...A.@...B @.%Z.M.ZA. *
002D40:	40020FE0	00614300	4001BB44	41804002	CDB44180	4002CDEC	4300FFCE	41C08268	* @.. .AC.@.;DA.@.M4A.@.MLC..NA@.H *
002D60:	41804002	109A4280	4001A57A	D38A0001	27872134	58C90060	213A08CD	08DA4180	* A.@...B.@.%ZS...'.!4XI. !:..M.ZA. *
002D80:	40020FE0	00624300	400200F4	41804002	CDB44180	4002CDEC	41804002	109A4300	* @.. .BC.@..TA.@.M4A.@.MLA.@...C. *
002DA0:	FFD841C0	81F64180	40020FE0	00634300	40021334	41C08210	41804002	109A42A0	* .XA@.VA.@.. .CC.@..4A@..A.@...B *
002DC0:	4001A57A	08DA4180	40020FE0	00644300	40021A1E	41C081F0	E6BA000B	41804002	* @.%Z.ZA.@.. .DC.@...A@.PF:...A.@. *
002DE0:	118E42A0	4001A57A	08DA4180	40020FE0	006558A0	400056C8	41B04002	4F344300	* ..B @.%Z.ZA.@.. .EX @.VHA0@.04C. *
002E00:	40021BEA	41C081C0	58C90068	4230802E	E6BA002F	41804002	118E42A0	4001A57A	* @..JA@.@XI.HB0..F:./A.@...B @.%Z *
002E20:	08CD08DA	41804002	0FE00066	58A04000	56C841B0	40024F34	43004002	1EA04180	* .M.ZA.@.. .FX @.VHA0@.04C.@.. A. *
002E40:	4002CDB4	41804002	CDEC4300	FFC241C0	817658C9	006C4230	8074E6BA	000308DA	* @.M4A.@.MLC..BA@.VXI.LB0.TF:...Z *
002E60:	41804002	118E42A0	4001A57A	D38A0000	C58000FF	42308028	078848BA	0000D4B8	* A.@...B @.%ZS...E...BO.(.H:..T8 *
002E80:	4001A3C8	23362682	C5800018	20872309	D3B84001	A3C908AD	E6BA4B00	00002304	* @.#H#6&.E... .#.S8@.#I.-F:K...#. *
002EA0:	08ADE6BA	001B4180	4002118E	42A04001	A57A08DA	41804002	0FE00067	58A04000	* -F:...A.@...B @.%Z.ZA.@.. .GX @. *
002EC0:	56C841B0	40024F34	43004002	72144180	4002CDB4	41804002	CDEC4300	FF7C41C0	* VHA0@.04C.@.R.A.@.M4A.@.MLC...A@ *
002EE0:	80E6E6BA	00074180	4002118E	42804001	A57A08DA	41804002	0FE00069	43004002	* .FF:...A.@...B.@.%Z.ZA.@.. .IC.@. *
002F00:	19525890	400056AC	433080F8	5899026C	21334300	FDFA08DF	27D641C0	80AAF6BA	* .RX.@.V,CO.XX..L!3C..Z.<'VA@.*F: *
002F20:	00074180	4002118E	42A04001	A57A08CD	08DA4180	40020FE0	006A58A0	400056C8	* ..A.@...B @.%Z.M.ZA.@.. .JX @.VH *
002F40:	588901C8	50C80004	73C90080	E6CC4900	00000899	08CC08AA	41B04002	4F144300	* X..HPH..SI..FLI.....L.*A0@.O.C. *
002F60:	40022EE0	41C08034	41804002	0FE0006E	43004002	19FA41C0	804EE6BA	000B4180	* @.. A@.4A.@.. .NC.@..ZA@.NF:...A. *
002F80:	4002118E	42A04001	A57A08CD	08DA4180	40020FE0	006F4300	4003054C	C3F00001	* @...B @.%Z.M.ZA.@.. .OC.@..LCP.. *
002FA0:	42304001	A47C5890	400056AC	43308054	C3E00200	43304001	A5C45889	01C8D0E8	* B0@.S.X.@.V,CO.TC ..CO@.%DX..HPH *
002FC0:	000050D9	01A4030C	C3F00001	42304001	A47C5890	400056AC	43308028	C3E00200	* ..PY.S..CP..B0@.S.X.@.V,CO.(C .. *
002FE0:	43304001	A5C45889	01C8D0E8	000050D9	01A408AD	C3D00003	033C50A9	01AC4300	* C0@.%DX..HPH..PY.S.-CP...<P)..C. *
003000:	4001A57A	E2000000	01314300	4001AC60	00000000	00000000	43004001	B91E0000	* @.%ZB....1C.@.,C.@.9... *

APPENDIX D
CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

The following pages contain fragments of a sample display from the Dump Print Utility for release R07.1 and higher. It is from a Model 3200MPS System.

CRASH CODE = *NONE* AT 10:27:11 ON 2/14/83 PROCESSOR = 3200 OS32M07-00 3200.202

DUMP OF GENERAL PURPOSE REGISTER SETS

	SET F	SET 0	SET 1	SET 2	SET 3	SET 4	SET 5	SET 6
R0	00000000	00007FF1	00000000	00000000	00000000	00000000	00000006	00000035
R1	00010168	00005D7A	00000000	00000000	00000000	00010168	FFFFFFFF	0000001A
R2	00010168	0002BAB2	00000000	00000000	00000000	00010168	00000000	0000003C
R3	00000001	FFFFFFFF	00000000	00000000	00000000	00000001	0000000D	0001AA54
R4	00000001	00FFA030	00000000	00000000	00000000	00000001	00022666	00000008
R5	00000000	00000000	00000000	00000000	00000000	00000000	00000A00	00003320
R6	00000002	00007DF8	00000000	00000000	00000000	00000002	0004205A	00011DF4
R7	00000006	80FFE5B0	00000000	00000000	00000000	00000006	00047052	00000000
R8	00000003	0002C506	00000000	00000000	00000000	00000001	0002250C	0002CCBC
R9	00000000	00FFA030	00000000	00000000	00000000	00000000	0002C1C8	00003320
RA	00000002	00003ABC	00000000	00000000	00000000	00000001	00000000	00000000
RB	00000001	000002A3	00000000	00000000	00000000	00000001	0000029E	00000000
RC	00002534	00FFA3F0	00000000	00000000	00000000	0000219C	00000000	000035F0
RD	00000001	00000000	00000000	00000000	00000000	00000FDF	00008348	00000000
RE	E000371C	00007FF0	00000000	00000000	00000000	00000000	00FFA030	000472F0
RF	1900225C	00004D06	00000000	00000000	00000000	00000000	00007DF8	0001A916

DUMP OF HARDWARE FLOATING POINT REGISTERS

F0	00000000	D0	00000000	00000000
F2	00000000	D2	00000000	00000000
F4	00000000	D4	00000000	00000000
F6	00000000	D6	00000000	00000000
F8	41100000	D8	00000000	00000000
FA	3F1800A8	DA	00000000	00000000
FC	00000000	DC	00000000	00000000
FE	00000000	DE	00000000	00000000

CURRENT TASK	FFA030	A91
UT REGISTERS OWNER	FFA030	A91
RS REGISTERS OWNER		**NONE**

48-031 P00 R01

DUMP OF APB TABLE

ID	ADDR	STATUS	CTCB	CONTROL STATE	READY QUEUE	NO. QUEUED	WAIT TCB	MAPPING TCB	CONTROL TCB
1	8490	C000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
2	8880	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
3	8C70	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
4	9060	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
5	9450	C000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
6	9840	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
7	9C30	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
8	A020	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE
9	A410	0000	00000000	APU OFF	00000000	0	NONE	NONE	NONE

D-3

D-4

DUMP OF TCB TABLE

ID	TCB NAME	ADDRESS	TASK FILE NAME	MID	GID	NLU	START	END	SIZE	SHSZ	OPTION	STATUS	WAIT
001	.CSL	3320		000	001	12	1A124	1A124	0	0	30108008	00000000	00000400
002	.CMDP	36F0		000	001	12	1AC34	1AC34	0	0	30108008	00000000	00000400
003	APUMON	FAE650	MPS :APUMON .TSK/00000	000	002	15	3E800	43000	4800	0	002004A1	00000000	00000400
00D	A91	FFA030	OSEX:MAININF .TSK/00000	000	002	15	D1800	E2000	10800	4000	00202DA1	08000100	00000000
030	A85	FEDC10	OSEX:MAININF .TSK/00000	000	002	15	313000	323800	10800	4000	00202DA1	08000100	00000000
057	A210	FE0930	OSEX:MAININF .TSK/00000	000	002	15	596800	5A7000	10800	4000	00202DA1	08000100	00000000
05F	A111	FDDD30	OSEX:MAININF .TSK/00000	000	002	15	61A800	62B000	10800	4000	00202DA1	08000100	00000000
0E3	A725	FB0630	OSEX:MAININF .TSK/00000	000	002	15	E9C800	EAD000	10800	4000	00202DA1	08000100	00000000

48-031 P00 R01

48-031 F00 R01

SEGMENT CONTROL LIST

ADDRESS	NAME	START	END	SIZE	USE	ROLL	TYPE	FLGS	SREG	KEY	ROLL DCB	SECTOR	SSTP	PRIV
FAE68C	APUMON	3E800	43000	4800	0	0	IMPURE	48		00	FF007918	33DE		RWE
FREE		43000	5A000	17000										
FFE920	OSEXMAININF TSK	5A000	5E000	4000	5	0	PURE	49	5	00	FF007918	339E	900	RF
FREE		5E000	D1800	73800										
FFA06C	A91	D1800	E2000	10800	0	0	IMPURE	48		00	FF007918	26E0		RWE
FREE		E2000	313000	231000										
FEDC4C	A85	313000	323800	10800	0	0	IMPURE	48		00	FF007918	30B0		RWE
FREE		323800	596800	273000										
FE096C	A210	596800	5A7000	10800	0	0	IMPURE	48		00	FF007918	33E8		RWE
FREE		5A7000	61A800	73800										
FDDD6C	A111	61A800	62B000	10800	0	0	IMPURE	48		00	FF007918	33C28		RWE
FREE		62B000	E9C800	871800										
FB066C	A725	E9C800	EAD000	10800	0	0	IMPURE	48		00	FF007918	3C448		RWE
FREE		EAD000	F83000	D6000										

D-6

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:37:24 02/14/83

PAGE 5

DCB	FCB	SIZE	FILENAME	TYPE	WCNT	RCNT	FLGS
7570	FF0930	0	MPFX:ERROR .LOG/00000	CO	1	1	40400000
7570	FFF4D0	0	MPFX:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000
7918	FFEF40	0	OSEX:SYSTEM .DIR/00000	CO	FFFF	FFFF	40400000

48-031 F00 R01

DUMP OF SYSTEM DATA STRUCTURES

48-031 F00 R01

SPT 002D70

SPT.INIT 43004002 083C
SPT.CRSR 0000
SPT.FLV 00007DA0
SPT.LLV 00008218
SPT.MLBL 0050
SPT.CTSP 0004
SPT.CSLV 00000005
SPT.CSBF 00000052
SPT.CHBK 0032
SPT.ISPT 0900
SPT.CTOP 0003E160
SPT.UTOP 00F83000
SPT.UBOT 0003E800
SPT.PMEM 01000000
SPT.MTOP 01000000
SPT.OSID 4F533332 4D543037
SPT.IVT 00008320
SPT.TTAB 00002F1C
SPT.MTM 00
SPT.NTCB FF
SPT.60 0037
SPT.DMT 00005D40
SPT.VMT 00007D80
SPT.SVOL 4F534558
SPT.SPVL 4D504658
SPT.RVOL 4F534558
SPT.TVOL 4D504658
SPT.SNOD 00000000
SPT.JRNL 00003ABC
SPT.FREQ 0078
SPT.PIC 006C
SPT.LFC 006D
SPT.CPU 0C80
SPT.SOPT 97F6F000
SPT.VERN 33323030 2E323032
SPT.CLSS 0004
0000
SPT.SSTE 0A00
SPT.SSTS 0020
SPT.ADCK 000284D2
SPT.ADK1 000284E2
00000000 00000000 00000000 00000000

SPT.DATE
SPT.MNTH 0002
SPT.DAY 000E
SPT.YEAR 0053
SPT.TSL 0000
SPT.TIME 000092FF
SPT.DTHD 00005E10
SPT.TQHD 00000000
SPT.IQHD 80FFE5B0
SPT.RTLS 00000000
SPT.TCMS 00000000
SPT.SCTH 00000000

```

*C.@..<
*..
*..}.
*....
*.P
*..
*....
*...R
*.2
*..
*...`
*..0.
*....
*....
*....
*OS32MT07
*..
*../.
*.
*.
*.7
*..1@
*..}.
*OSEX
*MPEX
*OSEX
*MPEX
*....
*...:
*.x
*.1
*.m
*..
*....
*3200.202
*..
*..
*..
*.
*....
*....
*.....
*..
*..
*.S
*..
*...
*...
*...
*...
*...
*...
*...
*...
*...

```

D-7

D-8

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:37:24 02/14/83

PAGE 7

SPT.SCTT 00000000
 SPT.OSUP 2D303020
 SPT.PANC 00036C34
 SPT.PSV 0000B300 00007DA0
 SPT.RSV 00FFFFFFA0
 SPT.TSV 00000000
 SPT.AFSV 00000000 00000000 00000000 00000000 00000000
 SPT.SCL 00FAE68C
 SPT.FLST 00043000
 SPT.FSYP 00FFFA50
 00000000
 SPT.RINQ 0002C1C4
 SPT.RDCB 00000000
 SPT.SPCT 0000
 0000
 SPT.EDNA 00000000
 SPT.EMHD 00000000
 SPT.EMTL 00000000
 SPT.CTCB 00FFA030
 SPT.UTOW 00FFA3F0
 SPT.RSOW 00000000
 SPT.ESOW 00000000
 SPT.MCOW 00000000
 SPT.EFOW 00000000
 SPT.DFOW 00000000
 SPT.RSON 00047260
 SPT.RSOF 00047060
 SPT.RLIO 00000000
 00000000
 SPT.VALU 00000000
 SPT.HLDA 00000000
 SPT.HLDB 00000000
 SPT.PAGE 00000800
 SPT.SCLP 0000000B
 SPT.ADDM 00FFFFFFF
 SPT.OFFM 000007FF
 SPT.PAGM 0000F800
 SPT.SEGM 00FF0000
 SPT.CPID 00000000
 SPT.CPIE 292D36A8
 SPT.MISS 00000005
 SPT.NODE 20202020
 20202020
 SPT.PSDD 00000000
 SPT.PSDT 00000000
 SPT.QH 00000000
 SPT.TID 000000ED
 SPT.PID 0000
 SPT.IPID 0000
 SPT.ERBL 00005D28
 SPT.RDYQ 0002C1BC
 SPT.RCVQ 0000FFE0
 SPT.LPMT 0000A860
 00000000 00000000 00000000 00003320 000036F0

*....
 *-00
 *..14
 *.....}.
 *....
 *....
 *.....
 *.....
 *..0.
 *...P
 *....
 *....
 *....
 *..
 *....
 *....
 *....
 *...0
 *....
 *....
 *....
 *....
 *....
 *..r`
 *..p`
 *....
 *....
 *....
 *....
 *....
 *....
 *....
 *....
 *....
 *....
 *)-6.
 *....
 *
 *
 *....
 *....
 *....
 *....
 *
 *
 *..]
 *....
 *....
 *....
 *....
 *.....3 ..6.

48-031 F00 R01

48-031 F00 R01

```

TCB      003320
TCB.FPTR 00FFA030      *...0      *
TCB.BPTR 00FB0630      *...0      *
TCB.QPTR 00000000      *...      *
TCB.CTX  000035F0      *..5      *
TCB.FLGS 00000000      *...      *
TCB.RTCV 00000000      *...      *
TCB.ETA  00000000 00000000 *...      *
TCB.TPTR 00000000      *.....      *
TCB.PBLK
TCB.PRCA 00000000      *...      *
TCB.PRCB 00000000      *...      *
TCB.PRCC 00000000      *...      *
TCB.PRCD 00000000      *...      *
TCB.PRCE 00000000      *...      *
TCB.PRCF 00000000      *...      *
SDE.FPTR 00000000      *...      *
SDE.BPTR 00000000      *...      *
SDE.NAME 2E43534C 20202020 00000000 00000000 00000000 *...CSL ..... *
SDE.SADR 0001A124      *...$ ..... *
SDE.SIZE 00000000      *...      *
SDE.USE  0000          *..      *
SDE.ROLL 0000         *..      *
SDE.FLGS 00           *..      *
SDE.TYPE 00          *..      *
SDE.SREG
SDE.PREG 00          *..      *
SDE.KEY  00          *..      *
SDE.ASG  00000000     *...      *
SDE.FSEC 00000000     *...      *
SDE.SSTP 0000         *..      *
SDE.PRIV 00          *..      *
00                   *..      *

TCB.TID  01000000     *...      *
TCB.STAT 00000000     *...      *
ICB.EQ   00000000     *...      *
ICB.PQ   00000000     *...      *
ICB.PSW  00000000 00000000 *.....      *
ICB.FLIH 00000000     *...      *
ICB.IPCB 00000000     *...      *
ICB.LINK 00000000     *...      *
ICB.HEAD 00000000     *...      *
ICB.1CL  00000000     *...      *
ICB.27CL 00000000     *...      *
ICB.3CL  00000000     *...      *
ICB.6CL  00000000     *...      *
ICB.7CL  00000000     *...      *
ICB.6RX  00000000     *...      *
ICB.RCNT 0000         *..      *
ICB.FLGS 0000         *..      *

TCB.CLC  00000000     *...      *
TCB.TGD  00000000     *...      *

```

D-9

D-10

```

TCB.VOFF 0000          *..          *
TCB.RSV  0000          *..          *
TCB.ASV                   *..          *
TCB.RSAC 0000          *..          *
TCB.RCNT 0000          *..          *
TCB.PRI  01            *..          *
TCB.RPRI 01            *..          *
TCB.DPRI 01            *..          *
TCB.MPRI 01            *..          *
TCB.CTSW C80044C0      *..D..        *
TCB.SLOC 00000000      *..          *
TCB.CTOP 00000000      *..          *
TCB.UTOP 00000000      *..          *
TCB.TTOP 00000000      *..          *
TCB.OBOT 00000000      *..          *
TCB.OCB  00000000      *..          *
TCB.TEQH 00000000      *..          *
TCB.SEG  00000000      *..          *
TCB.LRA                   *..          *
TCB.PSTD 00000000      *..          *
TCB.SSTD 00000000      *..          *
TCB.ADCK 00000000      *..          *
TCB.MXSP FFFFFFFF      *..          *
TCB.USSP 00000000      *..          *
TCB.SYSP 00000000      *..          *
TCB.SHSZ 00000000      *..          *
TCB.NSHD 00            *..          *
TCB.MSEG 0000          *..          *
TCB.WAIT 00000400      *..          *
TCB.OPT  30108008      *0...        *
TCB.SOPT 00000000      *..          *
TCB.DLAY 00000000      *..          *
TCB.VOL  00000000      *..          *
TCB.SVAD 0001AAD4      *..          *
TCB.TMP1 00000000      *..          *
TCB.USER 00000000 00000000 *.....        *
          00000000 00000000 *.....        *
TCB.SYS  0001AB70      *..p         *
TCB.SYS1 00000000      *..          *
TCB.SYS2 00000000      *..          *
TCB.SYS3 00000000      *..          *
TCB.SYS4 00000000      *..          *
TCB.SYS5 00000000      *..          *
TCB.FD                   *..          *
TCB.VOLN 00000000      *..          *
TCB.FN   00000000 00000000 *.....        *
TCB.EXT  00000000      *..          *
TCB.DATE 00000000      *..          *
TCB.ACCT 0000          *..          *
TCB.RC   0000          *..          *
TCB.USRA 00000000 00000000 *.....        *
TCB.UACT 0000          *..          *
TCB.GACT 0000          *..          *
TCB.MID  00            *..          *
TCB.GID  01            *..          *
    
```

48-031 F00 R01

48-031 F00 R01

```

0000
TCB.CPLM 00000000      *..
TCB.CPU 001EB149 00028505 00000000 00000000      *....
TCB.ACUM      *...I.....
TCB.WTIM 623B65B9      *b;e.
TCB.RTIM 00000000      *....
TCB.HOLD      *bw.B
TCB.THWT 6277F742      *....
TCB.ROUT 00000000      *....
TCB.STIM 00000000      *..
TCB.TSL 0000          *..
TCB.RLSL 0000          *..
TCB.TIMR 0000          *..
TCB.LOAD 00           *..
00                   *..
TCB.XFRS 00000000      *....
TCB.IOC 0001          *..
TCB.IOAC 0000          *..
TCB.IOBL 000034DC      *..4.
TCB.CIOB 00005E60      *..^
IOB.NXT 00000000      *....
IOB.RFLG 0000          *..
IOB.PRI 01           *..
IOB.TYPE 01           *..
IOB.DONE 00000000      *....
IOB.DCB 00005E10      *..^
IOB.TCB 00003320      *..3
IOB.ESR 0000AC80      *....
IOB.UPBK 0001AAE4      *....
IOB.PBLK 0001AAE4      *....
IOB.FC 40             *@
IOB.LU 00             *..
IOB.STAT 00           *..
IOB.DDPS 00           *..
IOB.SADR 00005A25      *..Z%
IOB.EADR 00005A74      *..Zt
IOB.RAND 00000000      *..^
IOB.LUE EB005E10      *....
IOB.SV1X 00000000      *....
IOB.WCHN 00000000      *....
IOB.CYL 0000          *..
IOB.SECT 00           *..
IOB.LSEC 00           *..

TCB.UCTX 000035F0      *..5.

CTX.FPTR 00000000      *....
CTX.BPTR 000035F0      *..5.
CTX.TCB 00003320      *..3
CTX.TRCE 0001A916      *....
CTX.PSW 00047052 00029CE8      *..pb....
CTX.PSTD 00000000      *....
CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
00000000 00000000 00C00000 00000000 00000000 00000000 00000000 00000000
*.....
*.....
*

```

D-11

48-031 F00 R01

SDE.SADR	0001AC34	*...4	*
SDE.SIZE	00000000	*....	*
SDE.USE	0000	*..	*
SDE.ROLL	0000	*..	*
SDE.FLGS	00	*..	*
SDE.TYPE	00	*	*
SDE.SREG			
SDE.PREG	00	*.	*
SDE.KEY	00	*	*
SDE.ASG	00000000	*....	*
SDE.FSEC	00000000	*....	*
SDE.SSTP	0000	*..	*
SDE.PRIV	00	*	*
	00	*	*
TCB.TID	02000000	*....	*
TCB.STAT	00000000	*....	*
ICB.EQ	00000000	*....	*
ICB.PQ	00000000	*....	*
ICB.PSW	00000000 00000000	*.....	*
ICB.FLIH	00000000	*....	*
ICB.IPCB	00000000	*....	*
ICB.LINK	00000000	*....	*
ICB.HEAD	00000000	*....	*
ICB.1CL	00000000	*....	*
ICB.27CL	00000000	*....	*
ICB.3CL	00000000	*....	*
ICB.6CL	00000000	*....	*
ICB.7CL	00000000	*....	*
ICB.6RX	00000000	*....	*
ICB.RCNT	0000	*..	*
ICB.FLGS	0000	*..	*
TCB.CLC	00000000	*....	*
TCB.TGD	00000000	*....	*
TCB.VOFF	0000	*..	*
TCB.RSV	0000	*..	*
TCB.ASV			
TCB.RSAC	0000	*..	*
TCB.RCNT	0000	*..	*
TCB.PRI	02	*..	*
TCB.RPRI	02	*..	*
TCB.DPRI	02	*..	*
TCB.MPRI	02	*..	*
TCB.CTSW	C8004240	*..B@	*
TCB.SLOC	0001AF3C	*...<	*
TCB.CTOP	0001AC32	*...2	*
TCB.UTOP	00000000	*....	*
TCB.TTOP	00000000	*....	*
TCB.OBOT	00000000	*....	*
TCB.OCB	00000000	*....	*
TCB.TEQH	00000000	*....	*
TCB.SEG	00000000	*....	*
TCB.LRA			
TCB.PSTD	00000000	*....	*
TCB.SSTD	00000000	*....	*

D-13

48-031 P00 R01

```

APB          008490

TKQ.LOKS    0000
TKQ.CNT     0000
TKQ.FRNT    00000000

APB.CTCB    00000000
APB.FLGS    C0000001
APB.HRQP    0000FFE0
APB.SSTD    003E0012
APB.PRCA    00000000
APB.PFSP    00008530
APB.XOP     00000000
APB.XOL     00000000
APB.XDEV    00000000
APB.XSTA    00000000
                00000000 00000000
APB.XNL     00000000
APB.PFCB    00000000
APB.PRCC    00000000
APB.PRCD    00000000
APB.PRCE    00000000
APB.PRCF    00000000
APB.STRC    00000000
APB.MMNP    00000000
APB.MMNL    00000000
APB.TPTR    00000000
APB.STAT    0301
APB.OPT     0000
APB.CTLR    000A
APB.CCB     0F20
APB.ETID    0000
APB.WTID    0000
APB.MTID    0000
APB.CTID    0000
EVN.CORD    00000000
EVN.FLGS    0800
EVN.CLEV    00
EVN.TSIZ    00
EVN.SQS     0001017C
EVN.DCB     000087E0
EVN.TCB     00000000
EVN.CLC     00000000
EVN.PREV    00000000
EVN.NEXT    000088F4
EVN.TOP     00000000
EVN.BOT     00000000
EVN.CYL     0000
EVN.RDCT    00
                00

APB.PWRF    00000000 00000000 0001E800 00051BFA 00000000 00010000 00020000 00030000
                00040000 00050000 00060000 00070000 00080000 00090000 000A0000 000B0000
                000C0000 000D0000 000E0000 000F0000 00000011 00010011 00020011 00030011
                00040011 00050011 00060011 00070011 00080011 00090011 000A0011 000B0011

```

D-15

D-16

000C0011	000D0011	000E0011	000F0011	00000022	00010022	00020022	00030022	*....."....."
00040022	00050022	00060022	00070022	00080022	00090022	000A0022	000B0022	*...."....."....."
000C0022	000D0022	000E0022	000F0022	00000033	00010033	00020033	00030033	*....".....".....3...3...3...3*
00040033	00050033	00060033	00070033	00080033	00090033	000A0033	000B0033	*...3...3...3...3...3...3...3...3*
000C0033	000D0033	000E0033	000F0033	00000044	00010044	00020044	00030044	*...3...3...3...3...D...D...D...D*
00040044	00050044	00060044	00070044	00080044	00090044	000A0044	000B0044	*...D...D...D...D...D...D...D...D*
000C0044	000D0044	000E0044	000F0044	00000055	00010055	00020055	00030055	*...D...D...D...D...U...U...U...U*
00040055	00050055	00060055	00070055	00080055	00090055	000A0055	000B0055	*...U...U...U...U...U...U...U...U*
000C0055	000D0055	000E0055	000F0055	00000066	00010066	00020066	00030066	*...U...U...U...U...f...f...f...f*
00040066	00050066	00060066	00070066	00080066	00090066	000A0066	000B0066	*...f...f...f...f...f...f...f...f*
000C0066	000D0066	000E0066	000F0066	00000000	0000FF08	00000000	08001080	*...f...f...f...f.....
00000006	00000000	00010224	00000002	00000001	0000010D	00000000	00000000	*.....S.....
0002000C	0000003C	80000310	00050234	0000003D	0000003D	0000003D	0000003D	*.....<.....4...=...=...=...=*
0000003D	0000003D	0000003D	0000003D	80008490	00000000	8000003D	00000002	*...=...=...=...=.....=...=*
00051BF6	00010094	00007FF2	00051BFA	00000000	00000000	00000000	00000000	*.....
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....
00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....

APB 008880

TKQ.LOKS	0000							*..*
TKQ.CNT	0000							*..*
TKQ.FRNT	00000000							*....*
APB.CTCB	00000000							*....*
APB.FLGS	00000002							*....*
APB.HRQP	0000FF00							*....*
APB.SSTD	003E0012							*.>..*
APB.PRCA	00000000							*....*
APB.PFSP	00008920							*....*
APB.XOP	00000000							*....*
APB.XOL	00000000							*....*
APB.XDEV	00000000							*....*
APB.XSTA	00000000							*....*
	00000000	00000000	00000000					*.....*
APB.XNL	00000000							*....*
APB.PRCB	00000000							*....*
APB.PRCC	00000000							*....*
APB.PRCD	00000000							*....*
APB.PRCE	00000000							*....*
APB.PRCF	00000000							*....*
APB.STRC	00000000							*....*
APB.MMNP	00000000							*....*
APB.MMNL	00000000							*....*
APB.TPTR	00000000							*....*
APB.STAT	0301							*..*
APB.OPT	0000							*..*
APB.CTLR	0050							*.P*
APB.CCB	0F40							*.a*
APB.ETID	0000							*..*
APB.WTID	0000							*..*

48-031 F00 R01

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:37:24 02/14/83

PAGE 51

DCB.RCNT	FFFF	*..	*
DCB.FLGS	42000000	*B...	*
DCB.1INC	00000000	*....	*
DCB.7INC	00000000	*....	*
	00	*..	*
DCB.DCOD	27	*:	*
DCB.DN	0010	*..	*
DCB.ATRB	EB81	*..	*
DCB.RECL	0050	*.P	*
DCB.INIT	0000AC80	*....	*
DCB.FUNC	00000000	*....	*
DCB.TERM	0000B17A	*...z	*
DCB.TOUT	000C	*..	*
DCB.RTRY	FFB1	*..	*
DCB.KEYS			
DCB.WKEY	00	*.	*
DCB.RKEY	00	*.	*
DCB.ILVL	0000	*..	*
DCB.ERR1	00000000	*....	*
DCB.LLXF	FFFFFFFFB1	*....	*
DCB.TOCH	FFFFFFFF	*....	*
DCB.XFLG	0000	*..	*
DCB.CLAS	0018	*..	*
DCB.IOH	0000B300	*....	*
DCB.Q	000226D6	*..&	*
DCB.EDMA	00000000	*....	*
IOB.NXT	00000000	*....	*
IOB.RFLG	2800	*(.	*
IOB.PRI	01	*.	*
IOB.TYPE	03	*.	*
IOB.DONE	00000000	*....	*
IOB.DCB	00005E10	*..^	*
IOB.TCB	00003320	*..3	*
IOB.ESR	0000B17A	*...z	*
IOB.UPEK	0001AAE4	*....	*
IOB.PBLK	0001AAE4	*....	*
IOB.FC	40	*@	*
IOB.LU	00	*.	*
IOB.STAT	00	*.	*
IOB.DDPS	00	*.	*
IOB.SADR	00005A25	*..Z%	*
IOB.EADR	00005A74	*..Zt	*
IOB.RAND	00000000	*....	*
IOB.LUE	EB005E10	*..^	*
IOB.SV1X	00000000	*....	*
IOB.WCHN	00000000	*....	*
IOB.CYL	0000	*..	*
IOB.SECT	00	*.	*
IOB.LSEC	00	*.	*
DCB.SIZE	00000018	*....	*
DCB.VFC	00000000	*....	*
DCB.DVDP			

48-031 P00 R01

```

DCB.WAPB 00000000      *....      *
DCB.LRBA 0000          *..        *
DCB.LSCT 04B1         *..        *
DCB.FIRP 0077         *.W        *
DCB.FIRB 0720         *.         *
DCB.LASP 0078         *.X        *
DCB.LASB 0027         *..        *
DCB.LSEC 0006         *..        *
                   0000 *..        *
DCB.LALC 00000048     *...H     *
DCB.MAST 00000000     *....     *
DCB.NEXT 00000000     *....     *
DCB.CNT 0000          *..        *
DCB.NUM 0002          *..        *
DCB.BITB 00007D68     *..}h     *
DCB.IORC
DCB.WRTR 00000570     *...P     *
DCB.RDR 00001208     *....     *
DCB.IODC
DCB.WRTD 00000660     *...`     *
DCB.RDD 00010A58     *...X     *
DCB.GOBY 00000000 00000000 *.....     *
DCB.SEEK 000009B2     *...     *
DCB.XFNM 41502020 20202020 43535300 *AP      CSS. *
DCB.END

```

```

DCB APU1 0087E0

DCB.DMT 00000000      *....      *
DCB.LEAF 00008504     *....     *
DCB.WCNT 0000         *..        *
DCB.RCNT 0000         *..        *
DCB.FLGS 40804000     *@.@.     *
DCB.1INC 00000000     *....     *
DCB.7INC 00000000     *....     *
                   00 *..        *
DCB.DCOD 00          *..        *
DCB.DN 000A          *..        *
DCB.ATRB 0000         *..        *
DCB.RECL 0000         *..        *
DCB.INIT 00000000     *....     *
DCB.FUNC 0001048C     *....     *
DCB.TERM 00000000     *....     *
DCB.TOUT 7FFF         *..        *
DCB.RTRY 0000         *..        *
DCB.KEYS
DCB.WKEY 00          *..        *
DCB.RKEY 00          *..        *
DCB.IVLV 0000         *..        *
DCB.ERRL 00000000     *....     *
DCB.LLXF 00000000     *....     *
DCB.TOCH 00000000     *....     *
DCB.XFLG 0000         *..        *

```

D-21

D-22

```

DCB.CLAS 0000
DCB.IOH 00021AA4
DCB.Q 00000000
DCB.EDMA 000C0000
IOB.NXT 000C0000
IOB.RFLG 0000
IOB.PRI 00
IOB.TYPE 03
IOB.DONE 00000000
IOB.DCB 000087E0
IOB.TCB 00FAE650
IOB.ESR 00000000
IOB.UPBK 00000000
IOB.PBLK 00010000
IOB.FC 90
IOB.LU 00
IOB.STAT 00
IOB.DDPS 00
IOB.SADR 00000000
IOB.EADR 00000000
IOB.RAND 00000000
IOB.LUE 00000000
IOB.SV1X 00000000
IOB.WCHN 00000000
IOB.CYL 0000
IOB.SECT 00
IOB.LSEC 00

DCB.SIZE 00000000
DCB.VFC 00000000
DCB.DVDP

```

```

DCB APU2 008BD0

DCB.DMT 00000000
DCB.LEAF 000088F4
DCB.WCNT 0000
DCB.RCNT 0000
DCB.FLGS 40804000
DCB.1INC 00000000
DCB.7INC 00000000
00
DCB.DCOD 00
DCB.DN 0050
DCB.ATRB 0000
DCB.RECL 0000
DCB.INIT 00000000
DCB.FUNC 0001048C
DCB.TERM 00000000
DCB.TOUT 7FFF
DCB.RTRY 0000
DCB.KEYS

```

48-031 F00 R01

. . .

48-031 P00 R01

IOB.STAT 00
 IOB.DDPS 00
 IOB.SADR 00000000
 IOB.EADR 00000000
 IOB.RAND 00000000
 IOB.LUE 00000000
 IOB.SV1X 00000000
 IOB.WCHN 00000000
 IOB.CYL 0000
 IOB.SECT 00
 IOB.LSEC 00

 DCB.SIZE 00000000
 DCB.VFC 00000000
 DCB.DVDP

```

* .
* .
* . . . .
* . . . .
* . . . .
* . . . .
* . . . .
* . . . .
* . .
* .
* .
* . . . .
* . . . .

```

DCB APU9 00A760

DCB.DMT 00000000
 DCB.LEAF 0000A484
 DCB.WCNT 0000
 DCB.RCNT 0000
 DCB.FLGS 40804000
 DCB.1INC 00000000
 DCB.7INC 00000000
 00
 DCB.DCOD 00
 DCB.DN 005E
 DCB.ATRB 0000
 DCB.RECL 0000
 DCB.INIT 00000000
 DCB.FUNC 0001048C
 DCB.TERM 00000000
 DCB.TOUT 7FFF
 DCB.RTRY 0000
 DCB.KEYS
 DCB.WKEY 00
 DCB.RKEY 00
 DCB.IVLV 0000
 DCB.ERRL 00000000
 DCB.LLXF 00000000
 DCB.TOCH 00000000
 DCB.XFLG 0000
 DCB.CLAS 0000
 DCB.IOH 00021AA4
 DCB.Q 00000000
 DCB.EDMA 00000000
 IOB.NXT 00000000
 IOB.RFLG 0000
 IOB.PRI 00
 IOB.TYPE 03
 IOB.DONE 00000000
 IOB.DCB 0000A760

```

* . . . .
* . . . .
* . .
* . .
* @. @.
* . . . .
* . . . .
* .
* .
* . ^
* . .
* . .
* . . . .
* . . . .
* . . . .
* . . . .
* . .
* .
* .
* .
* . . . .
* . . . .
* . . . .
* . . . .
* . .
* .
* .
* .
* . . . .
* . . . .
* . . . .
* . . . .
* . .
* .
* .
* .
* . . . .
* . . . .

```

D-23

D-26

```

FCB.BAPB
FCB.IOBF 00000000
FCB.IOBL 00000000
FCB.IOCN 0000
FCB.CFLG 0003
FCB.EXIT 00000000
          00000000
FCB.SLU 00000C00
FCB.BAPT 00000000
FCB.COE
FCB.BBPT 00000000
FCB.RSAS 00FEE0D0 00009500 00000000
FCB.PBUF 00000040
FCB.NBLK FC3E51E0
FCB.CBLK 00000000
FCB.NLR FC0053E0
FCB.CLRL 00000000
FCB.COFF 0000
          00
FCB.CBUF 00
FCB.BBPE 00000000 00000000 00000000 00000000 00000000 D4800000
FCB.CHE
FCB.ECH
FCB.BUFA
FCB.BUFB
FCB.CINX 0005
          0000
FCB.CINB 00FF01B0
FCB.NINB 00000070
FCB.IBPB 00000000 00000000 00000000 00000000
FCB.SPD 00000000
FCB.SPSZ 0000
          0000
FCB.INE
FCB.IBUF
FCB.IBFA
FCB.IBFB

```

48-031 F00 R01

```

FCB      FFF4C0
FCB.VMT 00007D88
FCB.LEAF 0000808C
FCB.WCNT FFFF
FCB.RCNT FFFF
FCB.FLGS 40400000
FCB.1INC 00000000
FCB.7INC 00000000
FCB.INBS 00
FCB.DCOD 00
FCB.DN 0000
FCB.ATR 7F7F
FCB.RECL 0000
FCB.INIT 00000000

```

48-031 F00 R01

FCB.FUNC	00000000	*.....	*
FCB.TERM	0000B840	*...@	*
FCB.OFF	34	*4	*
FCB.BKSZ	00	*.	*
FCB.BLK	0000	*..	*
	0000	*..	*
	00000000	*.....	*
	0000	*..	*
FCB.LLXF	00000000	*.....	*
FCB.TOCH	00000000	*.....	*
FCB.XFLG	0000	*..	*
FCB.CLAS	000C	*..	*
FCB.IOH	0002D914	*.....	*
FCB.Q	00000000	*.....	*
	00000000	*.....	*
IOB.NXT	00000000	*.....	*
IOB.RFLG	0080	*..	*
IOB.PRI	02	*.	*
IOB.TYPE	04	*.	*
IOB.DONE	00000000	*.....	*
IOB.DCB	FF007570	*..up	*
IOB.TCB	000036F0	*..6.	*
IOB.ESR	00000000	*.....	*
IOB.UPBK	00000000	*.....	*
IOB.PBLK	00FFF618	*.....	*
IOB.FC	4C	*L	*
IOB.LU	00	*.	*
IOB.STAT	00	*.	*
IOB.DDPS	00	*.	*
IOB.SADR	00FFF650	*...P	*
IOB.EADR	00FFFA4F	*...0	*
IOB.RAND	00000000	*.....	*
IOB.LUE	7FFFF4D0	*.....	*
IOB.SV1X	00000000	*.....	*
IOB.WCHN	00000000	*.....	*
IOB.CYL	0000	*..	*
IOB.SECT	00	*.	*
IOB.LSEC	00	*.	*
FCB.NAME	53595354 454D2020	*SYSTEM	*
FCB.EXT	444952	*DIR	*
FCB.ACT	00	*.	*
FCB.DATE	03F725D3	*..%.	*
FCB.FLBA	0000000A	*.....	*
FCB.LLBA	00000015	*.....	*
FCB.DIR	00000030	*...0	*
FCB.FCB	00000000	*.....	*
FCB.CSEC	00000004	*.....	*
FCB.SWL			
	00000000 00000000 00000000	*.....	*
CTX.FPTR	00000000	*.....	*
CTX.BPTR	00000000	*.....	*
CTX.TCB	00000000	*.....	*
CTX.TRCE	00000000	*.....	*
CTX.PSW	00000000 00000000	*.....	*
CTX.PSTD	00000000	*.....	*

D-27

D-28

```

CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
          000000C0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
          *.....*

FCB.PFCB 00000000 *.....*
FCB.BAPB *.....*
FCB.IOBF 00000000 *.....*
FCB.IOBL 00000000 *.....*
FCB.IOCN 0000 *.....*
FCB.CFLG 0003 *.....*
FCB.EXIT 00000000 *.....*
          00000000 *.....*
FCB.SLU 00000000 *.....*
FCB.BAPT 00000000 *.....*
FCB.COE *.....*
FCB.BBPT 00000000 *.....*
FCB.RSAS 7FFFF4D0 EBO11DF4 4C000000 *.....L...*
FCB.PBUF 00FFF650 *...P*
FCB.NBLK 00FFFA4F *...O*
FCB.CBLK 00000000 *.....*
FCB.NLR 00000400 *.....*
FCB.CLRL 00000000 *.....*
FCB.COFF 3000 *0.*
          00 *.....*
FCB.CBUF 00 *.....*
FCB.BBPB 00000001 00000018 0000000C 00000000 00000044 00000036 *.....D...6*
FCB.CHE *.....*
FCB.ECH *.....*
FCB.BUFA *.....*
FCB.BUFB *.....*
FCB.CINX 0400 *...*
          0004 *...0*
FCB.CINB 00000030 *PACK*
FCB.NINB 5041434B *INFODIR.SYSTEM DIR.*
FCB.IBPB 494E464F 44495200 53595354 454D2020 44495200 *B*
FCB.SPD 42202020 *.....*
FCB.SPSZ 2020 *.....*
          2020 *.....*

FCB.INE *.....*
FCB.IBUF *.....*
FCB.IBFA *.....*
FCB.IBFB *.....*

FCB      FFEF30 *...}.*

FCB.VMT 00007D90 *...h*
FCB.LEAF 00008168 *...*
FCB.WCNT FFFF *...*
FCB.RCNT FFFF *@@..*
FCB.FLGS 40400000 *.....*
FCB.1INC 00000000 *.....*
FCB.7INC 00000000 *.....*
FCB.INBS 00 *.....*

```

48-031 F00 R01

48-031 F00 R01

```

FCB.DCOD 00 *
FCB.DN 0000 *
FCB.ATRB 7F7F *
FCB.RECL 0000 *
FCB.INIT 00000000 *
FCB.FUNC 00000000 *
FCB.TERM 0000C090 *
FCB.OFF 04 *
FCB.BKSZ 00 *
FCB.BLK 0000 *
      0000 *
      00000000 *
      0000 *
FCB.LLXF 00000000 *
FCB.TOCH 00000000 *
FCB.XFLG 0000 *
FCB.CLAS 000C *
FCB.IOH 0002D914 *
FCB.Q 00000000 *
      00000000 *
      00000000 *
      00000000 *
      0080 *
      02 *
      04 *
      00000000 *
      FF007918 *
      000036F0 *
      00000000 *
      00000000 *
      00FFF088 *
      4C *
      00 *
      00 *
      00 *
      00FFF0C0 *
      00FFF4BF *
      00000030 *
      7FFFEF40 *
      00000000 *
      00000000 *
      0000 *
      00 *
      00 *
      00 *
FCB.NAME 53595354 454D2020 *SYSTEM *
FCB.EXT 444952 *DIR *
FCB.ACT 00 *
FCB.DATE 03F825D3 *% *
FCB.FLBA 00000428 *.( *
FCB.LLBA 0000045F *.< *
FCB.DIR 00000082 * *
FCB.FCB 00000000 * *
FCB.CSEC 00000034 *.4 *
FCB.SWL *
      00000000 00000000 00000000 *
      00000000 *
      00000000 *

```

D-29

D-30

```

CTX.BPTR 00000000 *....*
CTX.TCB 00000000 *....*
CTX.TRCE 00000000 *....*
CTX.PSW 00000000 00000000 *.....*
CTX.PSTD 00000000 *....*
CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
          00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....*
          *.....*

```

```

FCB.PFCB 00000000 *....*
FCB.BAPB *....*
FCB.IOBF 00000000 *....*
FCB.IOBL 00000000 *....*
FCB.IOCN 0000 *..*
FCB.CFLG 0003 *..*
FCB.EXIT 00000000 *....*
          00000000 *....*
FCB.SLU 00000000 *....*
FCB.BAPT 00000000 *....*
FCB.COE *....*
FCB.BBPT 00000000 *....*
FCB.RSAS 7FFFEF40 EB011DF4 4C000000 *...@....L...*
FCB.PBUF 00FFF0C0 *....*
FCB.NBLK 00FFF4BF *....*
FCB.CBLK 00000030 *...0*
FCB.NLR 00000400 *....*
FCB.CLRL 00000000 *....*
FCB.COFF 2000 *..*
          00 *..*
FCB.CBUF 00 *..*
FCB.BBPB 000000C3 000000C3 00000038 0000002C 0000025C 00003295 *.....8.....\..2.*
FCB.CHE *....*
FCB.ECH *....*
FCB.BUFA *....*
FCB.BUFB *....*
FCB.CINX 5400 *d.*
          0004 *..*
FCB.CINB 0000D0CD *....*
FCB.NINB 00000000 *....*
FCB.IBPB 00000000 00000000 00000000 00000000 *.....*
FCB.SPD 00000000 *....*
FCB.SPSZ 0000 *..*
          0000 *..*

```

```

FCB.INE *....*
FCB.IBUF *....*
FCB.IBFA *....*
FCB.IBFB *....*

```

```

EVN 007DA0
EVN.CORD 00000000 *....*
EVN.FLGS 4000 *@.*
EVN.CLEV 00 *..*

```

48-031 F00 R01

48-031 F00 R01

```

EVN.TSIZ 01
EVN.SQS 0002256E
EVN.DCB 00005E10
EVN.TCB 00000000
EVN.CLC 00000000
EVN.PREV 00000000
EVN.NEXT 00000000
EVN.TOP 00000000
EVN.BOT 00000000
EVN.CYL 0000
EVN.RDCT 00
00

```

```

*
*..%n
*..^
*....
*....
*....
*....
*....
*....
*....
*..
*
*

```

```

EVN      007DCC

EVN.CORD 00000000
EVN.FLGS 4000
EVN.CLEV 00
EVN.TSIZ 01
EVN.SQS 0002256E
EVN.DCB 00000000
EVN.TCB 00000000
EVN.CLC 00000000
EVN.PREV 00000000
EVN.NEXT 00000000
EVN.TOP 00000000
EVN.BOT 00000000
EVN.CYL 0000
EVN.RDCT 00
00

```

```

*....
*@.
*
*..%n
*....
*....
*....
*....
*....
*....
*....
*..
*
*

```

```

EVN      007DF8

EVN.CORD 00000000
EVN.FLGS 4000
EVN.CLEV 00
EVN.TSIZ 01
EVN.SQS 00022548
EVN.DCB 00000000
EVN.TCB 00000000
EVN.CLC 00000000
EVN.PREV 00000000
EVN.NEXT 00000000
EVN.TOP 00000000
EVN.BOT 00000000
EVN.CYL 0000
EVN.RDCT 00

```

```

*....
*@.
*
*..%H
*....
*....
*....
*....
*....
*....
*....
*..
*
*

```

D-31

JOURNAL DUMP

TASKID	MODULENAME	REGISTER.C	REGISTER.D	REGISTER.E	REGISTER.F
003	SVC13	00002408	00040C08	000477F2	00004180
003	TMNSOUT	00FAF9D0	00000000	000477F2	00004180
003	SVC9	00000000	0003EB44	000477F4	00003FBE
003	TMSETW	C8003600	00000400	000477F0	00003FBE
00D	THRDISP	00FFA3F0	00000000	00007FF0	00000CF6
00D	SQS	000224F0	00000000	00FFA030	00007DF8
00D	THREMW	00000002	00000400	00000000	80FFE610
00D	THRDYENQ	00FAE650	00000400	00000000	80FFE610
00D	THRDYENQ	00FFA030	00000000	000477F0	0002C474
003	THRDISP	00FAF9D0	00000000	000477F0	00004300
003	SVC9	00000000	0003E870	000477F4	00004318
003	TMNSOUT	00FAF9D0	00000000	000477F0	00003FBE
003	SVC13	000022B4	00040A84	000477F2	00004180
003	TMNSOUT	00FAF9D0	00000000	000477F2	00004180
003	SVC9	00000000	0003EB44	000477F4	00003FBE
003	TMSETW	C8003600	00000400	000477F0	00003FBE
00D	THRDISP	00FFA3F0	00000000	00007FF0	000003C0
00D	SQS	000224F0	00000000	00FFA030	00007DF8
00D	THRDISP	00FFA3F0	00000000	00007FF1	00003530
00D	SQS	000224F0	00000000	00FFA030	00007DF8
00D	THREMW	00000002	00000400	00000000	80FFE4B0
00D	THRDYENQ	00FAE650	00000400	00000000	80FFE4B0
00D	THRDYENQ	00FFA030	00000000	00007FF1	0002C474
003	THRDISP	00FAF9D0	00000000	000477F0	00004300
003	SVC9	00000000	0003E870	000477F4	00004318
003	TMNSOUT	00FAF9D0	00000000	000477F0	00003FBE
003	SVC13	00001C10	00040410	000477F2	00004180
003	TMNSOUT	00FAF9D0	00000000	000477F2	00004180
003	SVC9	00000000	0003EB44	000477F4	00003FBE
003	TMSETW	C8003600	00000400	000477F0	00003FBE
00D	THRDISP	00FFA3F0	00000000	00007FF2	00004EA4
00D	SQS	000224F0	00000000	00FFA030	00007DF8
00D	THREMW	00000002	00000400	00000000	80FFE5F0
00D	THRDYENQ	00FAE650	00000400	00000000	80FFE5F0
00D	THRDYENQ	00FFA030	00000000	000477F0	0002C474
003	THRDISP	00FAF9D0	00000000	000477F0	00004300
003	SVC9	00000000	0003E870	000477F4	00004318
003	TMNSOUT	00FAF9D0	00000000	000477F0	00003FBE
003	SVC13	00002160	00040960	000477F2	00004180
003	TMNSOUT	00FAF9D0	00000000	000477F2	00004180
003	SVC9	00000000	0003EB44	000477F4	00003FBE
003	TMSETW	C8003600	00000400	000477F0	00003FBE
00D	THRDISP	00FFA3F0	00000000	00007FF2	00005C6A
00D	SQS	000224F0	00000000	00FFA030	00007DF8
00D	THREMW	00000002	00000400	00000000	80FFE5D0
00D	THRDYENQ	00FAE650	00000400	00000000	80FFE5D0
00D	THRDYENQ	00FFA030	00000000	000477F0	0002C474
003	THRDISP	00FAF9D0	00000000	000477F0	00004300
003	SVC9	00000000	0003E870	000477F4	00004318
003	TMNSOUT	00FAF9D0	00000000	000477F0	00003FBE
003	SVC13	0000200C	0004080C	000477F2	00004180
003	TMNSOUT	00FAF9D0	00000000	000477F2	00004180

. . .

D-32

48-031 F00 R01

D-34

48-031 F00 R01

000F60:	00000000	00000000	00380000	00000000	00000000	29800009	00008FC0	00000000	*8.....).....).....*
000F80:	00000000	00000000	00380000	00000000	00000000	29800009	000093B0	00000000	*8.....).....0.....*
000FA0:	00000000	00000000	00000000	00000000	00000000	29800008	000097A0	00000000	*8.....).....).....*
000FC0:	00000000	00000000	00380000	00000000	00000000	29800009	00009B90	00000000	*8.....).....).....*
000FE0:	00000000	00000000	00380000	00000000	00000000	29800009	00009F80	00000000	*8.....).....).....*
001000:	00000000	00000000	00380000	00000000	00000000	29800009	0000A370	00000000	*8.....).....#P.....*
001020:	00000000	00000000	00380000	00000000	00000000	29800009	0000A760	00000000	*8.....).....).....*
001040:	58540018	D2350073	C3300008	4330804C	C8700040	9E279D23	D2350073	23844170	* XT..R5.SCO..CO.LHP.@.'.#R5.S#.AP*
001060:	4000DAB4	41704000	DAA8D235	0073C330	00084330	8020C530	0008213E	7375002E	* @.Z4AP@.Z(R5.SCO..CO.EO..!>SU..*
001080:	26714575	00D22384	4075002E	1800C830	0082D235	0073C870	00C09E27	25714075	* @QEU.R#@U...HO..R5.SHP.@.'@QU*
0010A0:	002C5875	00046470	40008348	41704000	DAB41800	58540018	DE208052	DA240005	* .,XU..DP@..HAP@.Z4..XT..^ .RZS..*
0010C0:	D8240006	DA24000D	D824000E	9577C470	B7FF9567	4865001A	C87000C0	9E67DE20	* XS..ZS..XS...WDP7..GHE..HP.@.G^*
0010E0:	802D4170	4000DAB4	41704000	DAA8DE20	801CD235	00732571	4075002C	58750004	* .-AP@.Z4AP@.Z(^ .R5.S@QU..XU..*
001100:	64704000	83484170	4000DAB4	18004810	DE20929E	DE20929E	58540018	4830929A	* DP@..HAP@.Z4..H.^ .^ .XT..HO..*
001120:	4035002C	24317435	01004230	804CD335	0070C330	00014230	8040C330	0020233F	* @5..\$1T5..BO.LS5.PCO..BO.@CO..#?*
001140:	41708642	7335009A	4873000A	4220802A	48730016	4230813A	43008E76	E665009D	* AP.BS5..HS..B .*HS..BO.:C..VFE..*
001160:	D3360000	C5300018	23394170	8618E665	009D2470	24334300	81089B26	E670803E	* S6..E0..#9AP..FE..SPS3C...@FP.>*
001180:	D3350070	C3300001	2333E670	80824074	00142435	74350100	42308FE4	8020C330	* S5.PCO..#3FP..@T..S5T5..BO.D.#CO..*
0011A0:	00014230	8FDA2466	74650100	2335C330	00024230	888AC330	00080337	18009B26	* ..BO.ZSFTE..#5CO..BO..CO...7#&*
0011C0:	08334230	8852C460	007FC560	00202383	E7640010	417085AE	41708402	E3640000	* .3BO.RD .E . #.GD..AP..AP..CD..*
0011E0:	42608058	18005854	00182432	74350100	42308C82	4300FFDC	417083A4	25314035	* B .X..XT..\$2T5..BO..C..AP.S%1@5*
001200:	00C01800	4170857E	DA25009F	4300FF6A	9B260833	42308800	C460007F	41708566	* @..AP..Z%.C..J.E.3BO..D .AP.F*
001220:	24767475	00F62335	C560000D	43308828	9A26E364	00004260	88221800	07335854	* SVTU.V#5F .CO.(.@CD..B ."...3XT*
001240:	0018D235	00FA2461	76650100	586500EC	D3750104	D2760000	487500F4	26710B67	* ..R5.ZSAVE..XF.LSU..RV..HU."@Q.G*
001260:	4300801E	58540018	24715175	00A45865	00ECD375	0102D276	00002761	487500F4	* C...XT..SQUU.SXE.LSU..RV..'AHU.T*
001280:	0B677345	009A4034	00164074	000A5064	000C7345	0098E660	800C4064	00147345	* .GSE..@4 .@T..PD..SE..F ..@D..SE*
0012A0:	009A4300	80D69B26	C3300001	42308ED0	C3300002	42308788	18009B26	0833203C	* ..C..V.@CO..BO.PCO..RO.....@.3 <*
0012C0:	C460007F	58540018	C560007F	43308044	C5600020	238D0876	11715A74	00104877	* D ...XT..E .CO.DF . #..V.QZT..HW*
0012E0:	00001171	F5700000	11E64230	80264875	01102671	C570000F	238E4075	01104875	* ..QUP...FBO.&HU..@QEP..#@U..HU*
001300:	01142771	2312247D	40750114	D2654700	01161800	E660FF8E	40640014	18002721	* ..'Q#.S.@U..REG.....F ..@D.....'!*
001320:	DE20908E	58540018	24667465	01002334	9D234220	870A2463	746500F6	233E9D23	* ^ ..XT..SFTE..#4.#B ..SCTE.V#>.**
001340:	232C4170	849E2302	230F0766	40650072	50650038	430087BC	07664B64	00021062	* #,AP..#.#.#.F@E.RPE.8C.<.FKD..B*
001360:	26644065	002C9B26	C460007F	C5600014	43308194	C5600013	4330818C	2621DE20	* @D@E.,@D .E .CO..E .CO..@!^*
001380:	9032E660	80184064	00140834	73450098	24657465	01004230	8DE60843	9D230833	* .2F ..@D...4SE..SETE..BO.F.C.#.3*
0013A0:	23321800	58540018	24377435	01002337	41708450	24377635	01001800	E660802E	* #2..XT..\$7T5..#7AP.PS7V5...F ..*
0013C0:	40640014	24687564	0000246E	74640000	23394864	00025A64	0004D336	0000E734	* @D..SHUD..SNTD..#9HD..ZD..S6..G4*
0013E0:	0010E364	00004260	80449A26	18004220	803E2335	08334230	861E1800	58540018	* ..CD..B .D.@.B .>#5.3BO...XT..*
001400:	24617665	00AC4330	86524875	00F04075	00ACD375	00F2D275	00AE4170	83E62478	* SAVE.,CO.RHU.P@U..SU.RRU..AP.FSX*
001420:	76740000	41704000	DAB44300	862E9A26	48640016	42308022	58540018	D3750070	* VT..AP@.Z4C...&HD..BO."XT..SU.P*
001440:	C3700001	2137247E	76740000	4874000A	23242478	76740000	1800C860	FF0C4064	* CP..!7S.VT..HT..#SSXVT...H ..@D*
001460:	00004170	4000DAB4	24624564	00164330	87EC5854	00182721	E6604000	2D204064	* ..AP@.Z4SBED..CO.LXT..'!F @.-,@D*
001480:	00140834	73450098	24634563	00164330	FCE84300	FD629B26	2621C460	007F5854	* ..4SE..SCEC..CO.HC..B.@@ID ..XT*
0014A0:	0018C560	000E4330	8844C560	00144330	80A2C560	00134330	809AC560	001B4330	* ..E .CO.DE .CO."E .CO..E .CO.**
0014C0:	85680833	42308550	C560000F	21352461	77650100	18002470	74750100	2337C560	* .H.3BO.PE .15SAWE...SPTU..#7E*
0014E0:	000D4330	808A4300	8514C560	00124330	807EC560	00114330	80764855	002CC550	* ..CO..C...E .CO..E .CO.VHU..FP*
001500:	7FFF4330	84F81800	C8707FFF	4075002C	73450098	41704000	DAB49B26	08334230	* ..CO.X..HP.@U..SE..AP@.Z4.@.3BO*
001520:	84F6C460	007FC560	001B4330	84FCC560	00122335	C5600011	4230849A	E660FF56	* .VD .E .CO..E .CO..#5F .RO..#V*
001540:	40640014	58540018	4170803E	7345009A	4300FE28	DE208E5E	C8607FFF	E67086BE	* @D..XT..AP.>SE..C.(^ .^H .^P.>*
001560:	58540018	4835002C	03374065	002C0307	DE208E40	E67086A6	58540018	24607665	* XT..H5..7@E...^ .@FP.EXT..S VE*
001580:	0100E660	FF104064	00145854	00187345	009A0766	4B640002	10622664	4300FFC0	* ..F ..@D..XT..SE..FKD..B@DC..@*
0015A0:	58540018	24327435	0100E630	88B22133	E630FC0A	40340014	7345009A	E6304000	* XT..S2T5..FO.2!F0..@4..SE..FO@.*
0015C0:	2D2C4034	0014DE20	8DEC7345	00980307	58540018	585500F8	D3550001	23022551	* -@4..^ .LSE...XT..XU.HSU..#.%Q*
0015E0:	48340016	211BC560	007F2338	C5600020	2389C560	00094330	80405854	00182531	* H4..!..E ..#8E . #.E .CO.@X".%1*
001600:	03070855	21172531	C3507F00	21359A25	23039A26	24315854	00185135	00B45835	* ..U!.%1CP..15.%#&S1XT..Q5.4X5*
001620:	00B45535	00B02183	503500B0	453500A2	23837635	01620733	03074054	000E5854	* .4U5.0!P5.OF5."#.V5.B.3..@T..XT*
001640:	00182430	743500AC	4330FFAE	48340014	403500C6	F5300000	12BA2335	E630FC46	* ..SOT5..CO..H4..@5.FUO...:#5FO.F*

INDEX

A		D	
Accounting reporting utility		Data collection	
account summary report	4-25	starting without MTM	4-3
capabilities	4-1	DDSTM. See storage module.	
command sequences	4-4	DEFINE command	
commands	4-8	prompt sequence	5-8
compiling	4-4	Defining memory configuration	
data collection	4-3	banks	5-10
establishing	4-4	blocks	5-10
individual account		boundary requirements	5-9
summary report	4-27	DEFINE command	5-7
input	4-1	Device errors	
introduction	4-1	summary	5-28
LOAD command	4-5	Directory sector allocation	A-6
message summary	4-32	Disk backup utility	
process	4-2	buffer size	3-4
sample reports	4-22	capabilities	3-1
START command	4-6	end of task codes	3-13
starting	4-4	file types	3-1
system summary report	4-23	from the MTM terminal	3-5
task termination codes	4-32	functions	3-2
Accounting reporting utility		general description	3-1
commands	4-8	LOAD command	3-7
CHARGE	4-17	loading procedure	3-6
END	4-32	magnetic tape format	3-3
GET	4-9	MARK command	3-7
OPTION	4-13	messages	3-17
PAUSE	4-31	multiple disk backup	3-16
REPORT	4-20	operating procedures	3-5
SAVE	4-30	requirements	3-5
SELECT	4-11	secondary directory	
		option	3-6
		START command	3-8
		starting procedure	3-6
		TASK command	3-7
		using two tapes	3-4
		Disk device	7-1
		Disk dump utility	
		ASSIGN command	6-2
		capabilities	6-1
		DUMP command	6-5
		dumps of file management	
		data	6-3
		facilities	6-1
		general description	6-1
		LOAD command	6-2
		messages	6-9
		operating procedures	6-2
		pauses	6-9
		performance estimates	6-2
		requirements	6-1
		set the record length	6-9
		set width of print line	6-8
		START command	6-3
		TASK command	6-2
		terminates	6-9
		UNDUMP command	6-5
		VERIFY command	6-7
		Disk initializer options	
		blocks	A-1

Disk initializer options (Continued)		Error reporting utility (Continued)	
clear	A-1	pausing a task	5-32
fill	A-1	report command	5-15
Disk initializer utility		SAVE command	5-31
defective sectors	A-6	saving information to an archival file	5-31
execution times for various disks	A-4	SELECT command	5-13
general description	A-1	selecting data	5-13
initializing floppy disks	A-5	specifying the input file	5-6
LOAD command	A-4	starting	5-6
MARK command	A-4	summary reports	5-26
operating procedures	A-4	Error reports. See list reports.	
options	A-1 A-2		
replaced by Fastchek utility	A-1	F	
requirements	A-2	Fast access directory	A-1
START command	A-4	Fastchek utility	1-2
TASK command	A-4 A-1	File information block fields	3-3
Disk integrity check utility		File types	
capabilities	B-1	contiguous	3-1
deletion of files	B-1	extendable contiguous	3-1
general description	B-1	indexed	3-1
operating procedures	B-3	nonbuffered indexed	3-1
replaced by Fastchek utility	B-1	G	
volume recovery	B-2	Get command	4-9
Dump print utility		H	
capabilities	7-1	Header page	
command sequence	7-2	date	2-4
example	7-5	taskid or filename	2-4
executing	7-2	time of day	2-4
general description	7-1	user account number	2-4
input device default messages	7-4 7-7	I,J,K	
prompt sequence	7-2	Inactive directory entry	
requirements	7-1	causes	B-6
structure display	7-6	Initialization	A-1
Dump print utility files		Input spooling facilities	2-2
permanent	7-2	Interpreting memory errors during report generation	5-13
temporary	7-2	L	
Dump print utility sample display	D-1	List device	7-1
E		List reports	
Error reporting utility		bulk device input/output	5-16
capabilities	5-1	categories	5-15
DEFINE command	5-7	file manager detected	
describing memory to interpret errors	5-1	errors	5-17
ending a task	5-32	memory errors	5-25
error reports	5-15	system detected errors	5-20
general description	5-1	system milestones	5-22
generating an error report	5-15	Logical record length	2-3
GET command	5-6		
list reports	5-15		
loading	5-6		
memory configuration messages	5-9 5-32		
module types	5-11		
PAUSE command	5-32		

M			
Magnetic tape format			
blocking factors	3-5		
file information block	3-3		
using two tapes	3-4		
volume header	3-3		
MARK command			
format	3-7		
secondary directory			
option	3-7		
MCD file. See memory			
configuration definition.			
Memory			
banks	5-2		
	5-3		
	5-4		
	5-5		
blocks	5-2		
	5-3		
local	5-9		
logical	5-2		
modules	5-2		
	5-4		
physical	5-2		
shared	5-9		
storage units	5-4		
	5-5		
Memory configuration			
definition			
create	5-7		
display	5-7		
specify	5-7		
Memory configuration file			
configuration data	5-13		
sample prompt session	5-12		
Memory controller	5-3		
Model 3200MPS System			
APU I/O counts	4-1		
APU processor time	4-18		
APU processor time counts	4-1		
Module sizes	5-11		
Module types	5-11		
Multiple disk backup			
procedure	3-16		
N			
New disk pack			
clearing	A-1		
initializing	A-1		
Non-privileged u-task	2-4		
O			
Operating procedures			
disk backup utility	3-5		
disk dump utility	6-2		
disk initializer utility	A-4		
disk integrity check			
utility	3-2		
		Operating procedures	
		(Continued)	
		dump print utility	7-2
		spooler	2-4
		Operating system failure	B-1
		Operating system failure	
		recommended procedures	
		for recovery	B-3
		OPTION subparameters	
		.SPL command	2-9
		START command	2-5
		OS/32 spooler	2-1
		.SPL command	2-9
		input spooling	2-2
		memory requirements	2-15
		messages	2-16
		operating instructions	2-4
		output spooling	2-3
		starting	2-5
		Output spooling	
		facilities	2-3
P,Q			
		Panic dump program	7-1
		Permanent file	7-1
		Privileged user task	3-5
			A-4
		disk initializer	A-2
		disk integrity check	B-2
R			
		Readcheck operation	A-2
		Requirements	
		disk backup utility	3-5
		disk dump utility	6-1
		disk initializer utility	A-2
		disk integrity check	
		utility	B-2
		dump print utility	7-1
S			
		Sample of a stand-alone dump	C-1
		Sample reports	4-22
		account summary	4-25
		individual account	
		summary	4-27
		system summary	4-23
		SDSTM. See storage module.	
		Secondary directory option	3-7
		expansion factor	3-6
		Segment size increment	
		disk initializer utility	A-3
		Selecting files	
		in backup operation	3-14
		in restore operation	3-14
		use of special characters	3-14
		SPL/32 spooler	2-1
		Spool queue	2-4
		Spooler operations	2-9

PUBLICATION COMMENT FORM

Please use this postage-paid form to make any comments, suggestions, criticisms, etc. concerning this publication.

From _____ Date _____

Title _____ Publication Title _____

Company _____ Publication Number _____

Address _____

FOLD

FOLD

Check the appropriate item.

Error Page No. _____ Drawing No. _____

Addition Page No. _____ Drawing No. _____

Other Page No. _____ Drawing No. _____

Explanation:

FOLD

FOLD

CUT ALONG LINE

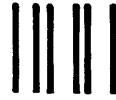
Fold and Staple
No postage necessary if mailed in U.S.A.

STAPLE

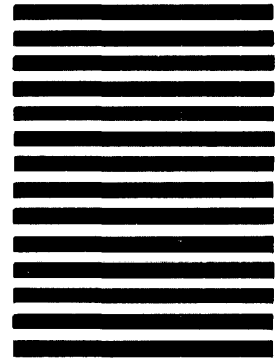
STAPLE

FOLD

FOLD



NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES



BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 22 OCEANPORT, N.J.

POSTAGE WILL BE PAID BY ADDRESSEE

PERKIN-ELMER

Computer Systems Division
2 Crescent Place
Oceanport, NJ 07757

TECH PUBLICATIONS DEPT. MS 322A

FOLD

FOLD

STAPLE

STAPLE