

**PERKIN-ELMER**

**OS/32  
SYSTEM SUPPORT UTILITIES**

**Reference Manual**

48-031 F00 R02

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, 1984 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	.SPL Command	2-11	
2.5	MEMORY REQUIREMENTS	2-17	
2.6	SPOOLER MESSAGES	2-18	
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 Back-Up	3-19	
3.5	MESSAGES	3-20	

## 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 Operating System Without Multi-Terminal Monitor (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	4-29
4.3.7	PAUSE Command	4-31
4.3.8	END Command	4-32
4.4	ACCOUNTING REPORTING UTILITY TASK TERMINATION CODES	4-33
4.5	ACCOUNTING REPORTING UTILITY MESSAGE SUMMARY	4-33
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	5-25
5.7.1.5	Memory Errors	5-28
5.7.2	Summary Reports	5-30

## CHAPTERS (Continued)

5.7.2.1	Summary of System Errors and Milestones Report	5-30	
5.7.2.2	Summary of Device Errors	5-33	
5.7.2.3	Memory Errors Summary Report	5-34	
5.8	SAVING INFORMATION TO AN ARCHIVAL FILE	5-36	
5.9	PAUSING A TASK	5-37	
5.10	ENDING A TASK	5-37	
5.11	MESSAGES	5-37	
6	DISK DUMP UTILITY		
6.1	GENERAL DESCRIPTION	6-1	
6.2	DISK DUMP UTILITY REQUIREMENTS	6-1	
6.3	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	AUXILIARY PROCESSING UNIT (APU) QUEUE PARAMETER BLOCK (QPB) SUMMARY DISPLAY	7-7	
7.8	AUXILIARY PROCESSING UNIT (APU) AUXILIARY PROCESSOR BLOCK (APB) SUMMARY DISPLAY	7-8	
7.9	DUMP PRINT MESSAGES	7-9	
8	MIRROR DISK SYNCHRONIZATION UTILITY		
8.1	MIRROR DISK FACILITY	8-1	
8.2	OVERVIEW	8-2	

## CHAPTERS (Continued)

	8.3	FUNCTION	8-2
	8.4	OPERATING PROCEDURES	8-3
	8.4.1	Preliminary Steps	8-3
	8.4.2	Loading DISCSYNC	8-4
	8.4.3	Starting DISCSYNC	8-4
	8.4.4	End of Task Codes	8-6
	8.4.5	Efficient Operation	8-6
	8.5	MESSAGES	8-8

## APPENDIXES

	A	CONTENTS OF MAGNETIC TAPE PRODUCED BY A STAND-ALONE DUMP	A-1
	B	CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP	B-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	STM INFORMATION	5-3
	5-2	MODULE TYPES AND SIZES	5-11
	8-1	DISCSYNC SEGMENT SIZES	8-7

## 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.

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. The Mirror Disk Synchronization Utility is described in Chapter 8. Appendix A contains the contents of a magnetic tape produced by a stand-alone dump from a uniprocessor. Appendix B contains the display of the Dump Print Utility output from a Model 3200MPS System.

This manual is intended for use with the OS/32 R07.2 software release or higher.

Revision 02 of this manual expands the Accounting Reporting Utility to include auxiliary processing unit (APU) user data as well as central processing unit (CPU) user data for the Model 3200MPS System. The Report Account formats have been realigned to make the accounting data more discernable. There are modifications to the OS/32 Spooler with the addition of four new START command options, five new run-time commands and two enhanced run-time commands. OS/32 Backup is also improved with the addition of eight new START command parameters. The Error Reporting Utility includes a new format for the output of memory error reports from the Model 3205 System. In addition, new system-detected error report types have been added that are applicable to the Model 3200MPS System. Information about the Synchronization Utility, a component of the Mirror Disk Facility, is provided in Chapter 8. Additionally, a more current sample display of the Dump Print Utility from a Model 3200MPS System has been included in Appendix B.

Discussions of the Disk Initializer Utility and the Disk Integrity Check Utility have been removed since these utilities are no longer supported for the OS/32 R07.1 release and higher. They 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.

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

System support utilities include the following:

- OS/32 Spooler
- Disk Backup
- Accounting Reporting
- Error Reporting
- Disk Dump
- Dump Print
- Mirror Disk Synchronization

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 Mirror Disk Synchronization Utility provides initial  
| synchronization and restoration of a synchronous state after  
| a disk failure, while still providing full availability of  
| data. It is a support component of the Mirror Disk Facility.

## CHAPTER 2 OS/32 SPOOLER

### 2.1 INTRODUCTION

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

- OS/32 Spooler
- SPL/32 Spooler

OS/32 Spooler is Perkin-Elmer's first generation spooler and, prior to the 6.2 release, was the only spooler available with OS/32. As such, the OS/32 Spooler provides the basic input/output (I/O) spooling services discussed in the remainder 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 (SVCL) 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'AO' 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 which it is copied by the spooler to the available output devices. 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 SVCL 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. 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 2.4.2). 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 nonprivileged u-task and must be loaded with the task identifier (taskid) .SPL. See the OS/32 Utilities Packaging Document.

### 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:

```

START [ ,pseudo dev= {          fd
                    ((fd1, fd2, ..., fdn)) } ]
      [ ,COMMAND=fd ]
      [ ,CARD= {          fd
                ((fd1, fd2, ..., fdn)) } ] [ ,T=sss ] [ ,M=n ]
      [ ,OPTION {          fd
                 ((fd1, fd2, ..., fdn)) }
        [ { NOHEADER } ] [ { PROTECT } ]
        [ { HEADER } ] [ { NOPROTECT } ]
        [ { NOBFF } ] [ { NOTFF } ]
        [ { BEE } ] [ { TEE } ]
        [ { AUTOCONTINUE } ]
        [ { NOAUTOCONTINUE } ]
      ]
      [ ,QVOL=voln: ]
      [ ,MAXDEVS=nnnn ]
      [ ,MXPSEUDO=mmmm ]
      [ ,QORDER= { FIFO
                  LIFO
                  PRIOSIZE
                  PRIORITY } ]
      [ ,PURGE ]
  
```

## 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 associate only devices with like device attributes with a common pseudo device.

**COMMAND=** fd specifies the input device or file from which additional parameters are to be taken. By using this parameter, a user may specify more parameters than can fit on a single command line. This parameter may appear anywhere in the parameter list. All parameters in the list are processed before any parameters are taken from the specified fd.

Any parameters contained in the command file must be in the same format as if they were issued from the START command line. The parameters within the specified fd are processed until end of file (EOF) is encountered.

**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 10.

**OPTION** specifies the options in effect for the device specified by fd. When a series of devices (fd) are specified, the OPTION subparameters apply to all devices, as in Example 3 below. In addition, different OPTION subparameters may also be specified for each real or pseudo device.

Whenever two or more OPTION subparameters 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 below, no parentheses are required.

**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 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= voln: 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.

MAXDEVS= nnnn specifies the maximum number of physical devices allowed. If this parameter is not specified, the default maximum number of real devices is as follows:

$$\frac{(n-1)}{2}$$

where n is the number of logical units specified when the task is built.

MXPSEUDO= mmmm specifies the maximum number of pseudo devices allowed. If this parameter is omitted, the maximum number of pseudo devices is the same as the number of real devices.

QORDER= defines the order in which entries are placed and maintained on the SPOOL queue file. This parameter will also determine how the entries are selected for output. The default is PRIORITY.

FIFO specifies first-in/first-out queue order.

LIFO specifies last-in/first-out queue order.

PRIOSIZE specifies that the queue is to be maintained in ascending order of task priority; entries of equal priority will be maintained in ascending order according to file size.

PRIORITY specifies that the queue is to be maintained in ascending order of task priority regardless of file size.

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.

### 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,QOR=FIFO
```

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. Entries on the spool queue file are placed and maintained in FIFO order. Since both time and message buffer options are omitted, the following defaults are taken and no input spooling is requested:

- T = 30 seconds
- M = 10 message buffers

### Example 3:

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

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:. CARD causes input to be spooled from device CR:. PNCH causes output to be spooled and output on CRDP:. The OPTION subparameters in effect for 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. The queue is to be maintained in ascending order of task priority. Tasks of equal priority will be maintained in ascending order of size.

### Example 4:

```
ST,COMM=SPL.CMD
```

The above example indicates the use of a user-specified command file (SPL.CMD) within the START command. This file may contain any of the spooler START command parameters. For convenience, separate command files can be established for individual spooler operations.

### Example 5:

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:,...
```

#### 2.4.2 .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 spooler operations available for use with the .SPL command follow.

Format:

.SPL {

ADD { dev: }  
      { pseudo-dev: }

DELETE { dev: }  
       { pseudo-dev: }

ATTACH pseudo-dev:,dev:

DETACH pseudo-dev:,dev:

HOLD fd

RELEASE fd

STATUS [ { dev: } ]  
       [ { pseudo-dev: } ]

BACKWARD fd,n

FORWARD fd,n

CANCEL fd

CONTINUE fd

REWIND fd

SUSPEND fd

INQUIRE [ TASK=ttttttt ] [,DEV=pdv] [ACCOUNT=aaaaa] [,LIST=fd<sub>2</sub>]

PRINT fd [,COPIES=n] [,DEVICE=fd<sub>x</sub>] [,DELETE] [,VFC]

PUNCH fd [,COPIES=n] [,DEVICE=fd<sub>x</sub>] [,DELETE] [,VFC]

OPTION { fd } [ { NOHEADER } ] [ { PROTECT } ]  
       { (fd<sub>1</sub>,fd<sub>2</sub>,...,fd<sub>n</sub>) } [ { HEADER } ] [ { NOPROTECT } ]  
                                  [ { NOBFF } ] [ { NOTFF } ]  
                                  [ { BFF } ] [ { TFF } ]  
                                  [ { AUTOCONTINUE } ]  
                                  [ { NOAUTOCONTINUE } ]

PURGE fd

END

TERMINATE

}

## Parameters:

ADD	defines a new device for the spooler. The device specified may be either a pseudo or a physical device.
dev:	specifies the name of a physical device.
pseudo-dev:	specifies the name by which a pseudo device is known to the spooler.
DELETE	deletes the specified device (pseudo or physical) as the output spooling device after the current file is completed.
ATTACH	associates a predefined pseudo device with a predefined physical device.
DETACH	disassociates the specified pseudo device from the specified physical device. Further output is prevented for the pseudo device.
HOLD	causes a file on the spool queue to be placed in a "hold" state until it is released.
RELEASE	causes a file on the spool queue to be released from the "hold" state and made available for output spooling.
STATUS	causes the status of the specified device (pseudo or physical) to be displayed to the system console. If the device parameter is not specified, this command displays to the system console the current status of each input, output and pseudo 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 which options are enabled.
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.
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.

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.

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.

TASK= tttttttt specifies the task for which the spool queue is to be searched. An asterisk (\*) and a dash (-) can be used to indicate generic task identifiers. These special characters are used in 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= fd 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 fd 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 specifies the pseudo device to which the file (fd) is to be output. If fd 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; i.e., a PRINT to a card punch device results in punched output, and a PUNCH to a line printer results in printed output.

**DELETE** If used with the PRINT or PUNCH parameters, this parameter 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 nonzero 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 or 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** specifies the options in effect for the real or pseudo device specified by fd. Whenever two or more of the OPTION subparameters are used, they must be enclosed within parentheses.



fd is a file descriptor that refers to a real or pseudo device.

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

HEADER specifies that the banner page is to be printed. This parameter is the default.

PROTECT specifies that only those files that are either assigned to the specified device (fd<sub>1</sub>) 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.

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.

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

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 SVC3 code 0 instruction.

TERMINATE causes OS/32 Spooler to terminate all operations immediately with EOT 4. 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.

#### 2.5 MEMORY REQUIREMENTS

The spooler memory requirements are approximately 20kb plus: |

- 76 bytes for each message buffer
- 180 bytes for each real device specified in the START command |
- 264 bytes for each I/O buffer |

If memory is not increased when the spooler is loaded, the additional memory designated by default (above the initial 20kb) is an additional 4kb.

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 264 bytes of memory available in the segment into which the task is loaded, the spooler creates another I/O buffer. If more than seven devices and more than ten message buffers are specified, the operator should load the spooler into a larger segment. If the number of real devices is not specified, the spooler will designate space in memory according to the number of logical units specified at link time. |

The space in memory taken by each pseudo device entry is related to the number of connected real devices. It is calculated by the following formula: |

$$(16+(8*n)) \text{ bytes} \quad |$$

where n is the number of connected real devices. |

#### Example:

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

## 2.6 SPOOLER MESSAGES

The following messages are output by the spooler.

### ALLOCATE/ASSIGN ERROR ON PRINT QUEUE

indicates that bad status was returned from SVC7 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 the free buffer into which the particular INQUIRE parameters would be passed is not available. The request is ignored, but can be repeated when a buffer is available.

### CURRENT OS DOES NOT SUPPORT THIS SPOOLER

indicates that the OS was sysgened either without spool support or with SPL/32 support.

### DEVICE xxxx NOT DEFINED FOR OPTION COMMAND

indicates that device xxxx specified in the OPTION parameter of the START or .SPL command is not a device name defined by the spooler at the time the OPTION parameter is processed. In the START command, specify the device xxxx as a pseudo or real device prior to specifying the OPTION parameter. In the .SPL command, use .SPL ADD xxxx prior to using the OPTION parameter.

### ERROR: xxxx ALREADY DEFINED

indicates that the device specified has already been ADDED or ATTACHED.

### ERROR: xxxx DEVICE ACTIVE

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

ERROR: xxxx 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: xxxx DEVICE IS IDLE

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

ERROR: xxxx DEVICE IS NOT A REAL DEVICE

indicates that the device xxxx specified is not a real device. The command entered requires a real device to be specified.

ERROR: xxxx DEVICE QUIESED. RESTART DISALLOWED.

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: xxxx IS NOT A PSEUDO-DEVICE

indicates that the device xxxx specified is not a pseudo device. The command entered requires a pseudo device to be specified.

ERROR: xxxx IS NOT CONNECTED TO ####

indicates that the pseudo device xxxx specified by the DETACH parameter is not connected to the real device ####.

FILE filename NOT ON QUEUE

indicates that the filename specified with the PURGE, HOLD or RELEASE 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.

INQUIRY LIST ASSIGN ERROR CODE=xx

indicates that the list file for an INQUIRE command could not be assigned. The status returned from SVC7 is xx.

INQUIRY LIST WRITE ERROR STATUS=yyyy

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

INSUFFICIENT LUS DEFINED FOR START COMMANDS

indicates that the number of logical units available is insufficient for the operation of the spooler as specified by the START command.

INSUFFICIENT MEMORY ALLOCATED FOR SPOOLER

indicates that insufficient space is available for the operation of the spooler. Reload the spooler with a larger segment size increment field.

INSUFFICIENT SPACE ALLOCATED FOR PSEUDO DEVICE TABLE

indicates that insufficient space has been allocated for the pseudo device table. Reload the spooler with a larger segment increment field or increase the maximum number of pseudo devices to be supported with the MXPSEUDO parameter of the START command.

INVALID fd

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

INVALID PARAMETER

indicates that the parameter is invalid since it does not match any of the acceptable parameter inputs for the command.

INVALID PARAMETER DELIMITER

indicates that a parameter delimiter such as a comma or a parenthesis is missing. See the command format representation.

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 fd PLEASE CORRECT AND CONTINUE

indicates that a spool device returned a nonzero status (xxxx), possibly because the printer is off-line or the card reader needs to be checked. The fd is the physical device.

I/O ERROR xxxx ON LU ##### fd

indicates that an I/O error has occurred on LU #####; xxxx is the status returned from SVCl.

I/O ERROR xxxx ON PRINT QUEUE FILE

indicates that SVCl 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).

MAXIMUM REAL DEVICES EXCEEDED

indicates that the number of real devices, as specified by the START or .SPL command, exceeds the maximum allowable amount that was specified by the MAXDEVS parameter.

MISSING /@ CONTROL CARD, FILE fd

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

NOT ADDED TO BATCH QUEUE

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

NOT ENOUGH MEMORY FOR 1 I/O BUFFER PER DEVICE

indicates that insufficient memory was designated to allow an I/O buffer for each connected real device. The spooler continues operation, but performance will be affected.

OS/32 REVISION IS LESS THAN R06-02

indicates that the version of OS/32 is earlier than R06-02.

PERKIN-ELMER OS/32 SPOOLER 03-152 R03-00

is an identification message that appears when OS/32 Spooler starts its operation.

PSEUDO-DEVICES EXCEEDS MAXIMUM

indicates that the number of pseudo devices specified by the START or .SPL command exceeds the maximum allowable amount as specified by the MXPSEUDO parameter.

QUEUE FILE NOT EMPTY - QORDER PARAMETER IGNORED

indicates that reordering of queue items can only be accomplished when the queue is empty. Spool out the items that are presently on the spool queue.

QUIESE IS PENDING

indicates that the spooler is completing its operation. The REWIND or CANCEL command will be ignored since the spooler has already been instructed to end.

WARNING: fd BAD LINE COUNT

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

WARNING: fd INVALID FIRST CONTROL CARD

indicates a syntax error on /@ starting control card. fd specifies the device name.

WARNING: fd SVC7 ERR=xx

indicates that the device specified by fd could not be assigned by the spooler. The status returned by the SVC7 call is xx.

WARNING - NO OUTPUT DEVICES DEFINED

indicates that no output spooling devices were defined by the START command.

WARNING: xxxx RECORD LENGTH LESS THAN nnnn

indicates that the record length defined for the real device xxxx is less than that defined for the associated pseudo device nnnn.



## 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, tape-to-tape, or tape-to-disk. The starting parameters specified, date of back-up and 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. Files on the input disk that have not been accessed since a specified date can be deleted after transferring them to the output device. Optionally, the data on the back-up 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 may 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.

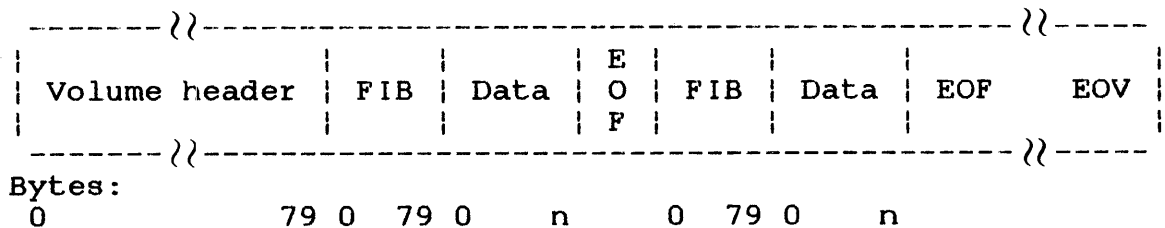
When transferring files from one disk to another, the Disk Backup Utility writes the files onto the destination disk in a contiguous manner as long as there are no bad sectors. This minimizes access time on the destination disk for indexed files and maximizes the amount of contiguous free space on the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

The Disk Backup Utility performs the following functions:

- Directly transfers files from one disk to another; the output disk serves as a back-up of the original.
- Transfers files from an input disk to an intermediate magnetic tape device; the magnetic tapes are used as a back-up.
- Appends files from an input disk to a magnetic tape containing previously backed up files; back-up 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 back-up operation.
- Restores the data from the intermediate device to an output disk.
- Verifies data copied during the back-up operation.
- Verifies data that was copied during a previous execution of the utility.
- | ● Selectively dumps individual files from an input device to an  
| output device.
- Modifies the account numbers of files dumped from various accounts to one single account number during a disk-to-disk, disk-to-tape, or tape-to-disk transfer procedure.
- Selectively restores files from tape to disk.
- Selectively dumps files changed since a given time.
- Deletes and replaces files on an output disk.
- | ● Copies files from an input tape to an output tape.
- Displays the starting parameter specified, date of the Disk Backup Utility operation, and names (and other data) of the files backed up.

### 3.2 DISK BACKUP UTILITY MAGNETIC TAPE FORMAT

When copying files onto magnetic tape, the output tape created by the Disk Backup Utility is in the following format:



#### Fields:

Volume header is 80 bytes long and contains the following fields:

- Disk volume (volume name of input disk)
- Sequence number of tape, starting with 1
- Number of blocks written on to the preceding tape
- Size of buffer used to transfer data
- Revision and update number of back-up used
- Start options specified
- Date files backed up

FIB (file information block) is 80 bytes long and provides information relative to the file. This parameter precedes the data of each file and contains the fields listed below.

- Filename, extension and account number
- File type
- Keys
- Logical record length
- Number of records

- 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., back-up 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

Volume header	FIB	Data	EOF	FIB	Data	EOF	EOV
---------------	-----	------	-----	-----	------	-----	-----

TAPE 2

Volume header	Remaining data	EOF	EOV
---------------	----------------	-----	-----

### 3.3 DISK BACKUP UTILITY REQUIREMENTS

The Disk Backup Utility requires:

- Approximately 25kb of memory, plus additional memory required for buffers
- A console device
- Two currently supported disk devices (for disk-to-disk operations); one currently supported disk and a magnetic tape device (for disk-to-tape operations); or two magnetic tape devices (for tape-to-tape operations)
- 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 dynamically increase its buffers in size and/or number. The amount of memory available will therefore determine execution times, particularly when using magnetic tapes or contiguous files in back-up operations.

### 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. To run the Backup Utility from the MTM terminal, the user's 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	12.5kb
1600 bpi	25.0kb
6250 bpi	100.0kb

These blocking factors refer to the size of the blocks on tape and are controlled by the SIZE parameter of the START command. The default blocking factor used by the Backup Utility is 12kb. 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 on-line. The input disk can be marked on-line protected. When performing a back-up from a fixed disk to a removable disk, the fixed disk must always be marked on-line protected, provided the disks are on the same disk drive. If the input disk is on-line 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. (The 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 the Fastchek Utility.

The Disk Backup Utility does not save temporary, 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
```

Where:

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

2. Select back-up 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 initialized using the Fastchek Utility prior to the back-up operation. See the OS/32 Fastchek Reference Manual for more details on this utility.
4. Mark the disk used as input on-line, (optionally) protected using the MARK command as follows:

MARK dn:,ON

or

MARK dn:,ON,PROTECT

5. Use the MARK command to mark the output disk on-line:

MARK dn:,ON

Using the secondary directory option (CD) to mark the input and output disks on-line will improve performance. The format of the commands follows:

MARK dn:,ON,,CD

or

MARK dn:,ON,PROTECT,CD

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 the size was not specified in the START command) is found.

**Format:**

START [ ,IN={devn:} ,OUT={devn:} ,LIST=fd ] [ ,SIZE=nn [.dd] ]  
[ ,COMMAND=fd ] [ ,END ] [ ,DELETE [/NODATECHECK] ]  
[ ,VERIFY [/COUNT] ] [ ,VOL [/COUNT] ] [ ,ABORT ] [ ,SKIP ] [ ,ACCOUNT=acct# ]  
[ ,SINCE={mon/dd/yy} ,hh:mm[:ss] ] [ ,SELECT=fd ] [ { APPEND } [ ,NEWDATE ]  
[ { POSITION } ] [ { NOREWIND } ] ]  
[ ,IOERR={ SKIP } [ ,ALTINPUT=fd ] [ ,ALTOUTPUT=fd ]  
[ { DELETE } ]  
[ { ABORT } ]  
[ { PAUSE } ] ]  
[ ,CLOSE={ REWIND } [ ,REPORTONLY ] [ ,SPOOL={ YES } ]  
[ { NOREWIND } ] [ { NO } ] ]  
[ ,ARCHIVE={mon/dd/yy} ,hh:mm[:ss] ]

**NOTE**

If the START command is entered without options, command entry will be interactive. A series of options may be entered. When a PROCESS command or an END command is entered, the back-up operation executes with all of the options that were previously specified in effect.

**Parameters:**

IN= devn: is the device mnemonic of the input device from which data is copied. This device is assigned for sharable read-only (SRO) to logical unit 1 (lul).  
voln: is the volume name for the volume from which data is to be copied. voln: may only be used for disk volumes.



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

voln: is the volume name for the volume to which data is to be copied. voln: may only be used for disk volumes. |  
|  
|

LIST= fd is the file descriptor of the device/file to which all list output is to be directed. 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 (SWO) to lu7.

SIZE= nn specifies the size in kilobytes of the output tape buffer requested for disk-to-tape operation. If the output device is not a magnetic tape, this parameter may be specified but will be ignored. The default size is 13kb. nn is a decimal number with optional decimal places (dd); 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.

When input is from a command file, the PROCESS option may be used. The PROCESS option causes Backup to execute all options specified since the beginning of the command file, or since the previous PROCESS option. Multiple back-ups can be performed in one program run by using this option embedded within a command file.

#### NOTE

When command entry is interactive, an on-line help guide is available for all options of the Backup START command. For example, when HELP SPOOL is entered, an explanation of the SPOOL option will appear on the terminal screen.

END

causes Backup to stop scanning parameters within a command file or when being entered interactively. Anything specified after END in the command file or in the START command will be ignored.

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. 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 1- to 5-digit number indicating the number of records in a file that must fail to verify before Backup will skip the remainder of the file. The number's value may be from 1 to 65,535. 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 a program if nonzero 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 indicates that 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.
- 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. The account number is a 1- to 5-digit number having a value from 0 to 65,535.

SINCE=

enables the user to back up or restore files changed since the specified date. mon/dd/yy is the name of the month, day and year. hh:mm:ss is the hour, minutes and seconds. 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, and/or verifies files from input device to output device. fd is the file or device from which filenames to be copied 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. All selected entries must reside in memory and use that area of memory between UTOP and CTOP. Memory used by select entries is not available to be used as I/O buffers. Each select file entry requires 14 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.

The list of files to be backed up can be augmented by embedding the !INCLUDE command or restricted by embedding the !EXCLUDE command within the select file. See examples of the use of embedded !INCLUDE and !EXCLUDE commands in the functional details section.

**APPEND**

specifies that during a disk-to-magnetic tape back-up operation, Backup is notified that the magnetic tape output device contains back-up format data and that additional back-up files are to be added. Backup scans the tape for the EOVS mark. It removes the mark and begins the current back-up operation. The EOVS mark is rewritten after the last file is backed up. This option conflicts with the VERIFY option.

**POSITION**

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

**NOREWIND**

specifies that during a disk-to-tape back-up or restore operation, Backup is notified that the magnetic tape output device is positioned at the point where the current back-up operation is to begin. If writing to the tape, Backup will write a back-up 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 back-up format volume header. This option conflicts with 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 the current date and time on the output device.
IOERR=	is used to specify the desired action in the event of an I/O error.
SKIP	specifies that the errant file is to be skipped if an I/O error occurs.
DELETE	specifies that any portion of the output file backed up prior to the I/O error is to be deleted.
ABORT	specifies that the back-up task is to be aborted if an I/O error occurs.
PAUSE	specifies that the back-up task is to be paused if an I/O error occurs.
ALTINPUT=	specifies an alternate input device for multiple volume input tapes. If IN= specifies a disk device, this parameter is ignored.
ALTOUTPUT=	specifies an alternate output device for multiple volume output tapes. If OUT= specifies a disk device, this parameter is ignored.
CLOSE=	specifies the action to be taken upon closing an input and/or output tape.
REWIND	indicates that the tape is to be rewound upon closing.
NOREWIND	indicates that the tape is not to be rewound upon closing. This is the default.
REPORTONLY	specifies that the input device is to be read and a report generated based on the selection criteria, but that no back-up is to take place.
SPOOL=	specifies whether spool files are to be transferred from the input device if they are found there. A spool file has a commercial at sign (@) as the first character of its file name. NO is the default.

ARCHIVE= enables the user to transfer files from a disk volume and then delete any transferred files that have not been accessed on or after the date specified by mon/dd/yy. Applies only to files that have been successfully transferred. The date can be entered with the month or day first.

#### Functional Details:

When started, the Disk Backup Utility prints this 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 is reached. More than one select filename can be specified per 80-byte input record by separating the fds with commas (,) or semicolons (;).

!INCLUDE allows a user to specify a file to be selected for transfer. Use of the keyword !INCLUDE is optional within the select file. !EXCLUDE allows a user to prevent a file (or set of files) from being transferred. Only files selected for inclusion are checked for possible exclusion.

The following is an example of a select file with an embedded !EXCLUDE. The processing of this file would result in the back-up of all files on accounts 118 and 119 that have an extension of .CAL. The !EXCLUDE would prevent the back-up of any file within the specified group that has TMP as the first 3 characters of the filename.

```
-.CAL/118  
-.CAL/119  
!EXCLUDE TMP-.CAL/-  
/*
```

The next example of a select file contains both !INCLUDE and !EXCLUDE. In this case, all files from account 118 with an extension of .043 are to be backed up. In addition, all files from account 119 are to be backed up, except for those files with an extension of .MEM, as specified by the !EXCLUDE. The exclusion of files with the .MEM extension is limited to those on account 119 because of the use of the !INCLUDE to make account 119 files members of the select file.

```
-.043/118
!INCLUDE -./119
!EXCLUDE -.MEM/119
/*
```

In this final example of the use of the !INCLUDE and !EXCLUDE commands, back-up is prevented for all files on accounts 73, 118 and 119 that have Z as the first character of the filename and an extension of .FTN.

```
-.FTN/73
!INCLUDE -.FTN/118
!INCLUDE -.FTN/119
!EXCLUDE Z -.FTN/-
/*
```

When selecting files for a back-up 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. The asterisk character (\*) requests that all files with any character in the same position as the asterisk be selected. The characters \* and - can be combined to delimit selected files.

The following example selects for back-up or restore operations all files whose first five characters are CAL32.

```
CAL32-
```

The next example selects for back-up or restore operations all files named TESTPROG with any extensions.

```
TESTPROG-
```



The next example selects for back-up or restore operations all files between five and eight characters in length whose first five characters are CAL32.

CAL32\*\*\*

The next example selects for back-up or restore operations all files with a filename containing six characters whose fifth and sixth characters are 32 and whose extension is OBJ.

\*\*\*32.OBJ

This final example selects for back-up or restore operations all files whose first three characters are CAL and whose sixth character is 1 with any extension.

CAL\*\*1-

End of task codes:

Upon successful completion of a Backup operation, the following message is printed:

END OF TASK CODE = 0

If there is an error in one of the START command parameters, the following 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, the following message is logged on the system console and printed on the list device:

END OF TASK CODE = 2

If the user specified ABORT for SVC7 or I/O errors or if Backup fails an internal consistency check, the following message is logged on the system console and printed on the list device:

END OF TASK=10

#### 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 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:

This example of the START command copies DSC2: to DSC1:, verifies, aborts on errors and sends the listing to PR:.

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

This example preassigns list device to lu7, copies DSC2 to DSC1, and verifies.

```
ASSIGN 7, PR:  
START,OUT=DSC1:,IN=DSC2:,VE
```

This example copies disk to tape, buffer size = 4.5kb, verifies, aborts on errors and sends a listing to PR:.

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

This example copies tape to disk and aborts on errors. If filenames match, it restores only if the file from tape has a more recent date.

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

This example verifies files from tape to disk but does not copy files.

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

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

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

This example copies all files from disk changed since March 17, 1984, at 12:30 to tape.

```
START,COMMAND=CON:,IN=DSC1:  
BACK-UP >OUT=MAG1:,LIST=PR:  
BACK-UP >SINCE=MAR/17/84,12:30  
BACK-UP >VERI,END
```

### 3.4.1 Multiple Disk Back-Up

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

#### NOTE

Fixed disk-to-removable disk back-up operations must be performed in a stand-alone environment with no other tasks running on the system. If other tasks are running while back-up 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 back-up 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 back-up operations. Backup will not overwrite these files, but will use the remaining free space on the disk.

### 3.5 MESSAGES

DEVICE xxxx: MUST BE MARKED ON

| indicates that an attempt was made to start the Backup  
| operation without first marking on xxxx device with the MARK  
| command.

ENTER SELECT FILE DESCRIPTORS (MAX=nnnn)

indicates a program request for filenames that are to be restored. The maximum number of entries possible is nnnn.

EQUALS SIGN EXPECTED - xxxxxx

indicates that a required equal sign has been omitted from parameter xxxxxx of the START command.

ERROR (xx) ASSIGNING fd

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. The returned SVC7 status is indicated by xx.

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 a tape-to-disk operation.

INPUT DISK MUST BE MARKED ON "PROTECT"

indicates that in an attempt to back up data from a fixed disk of a 10Mb disk to a removable disk of the same pack, the input was not marked on-line with a protected status. Backup pauses. The input disk must be marked off and then marked on protected, and the task continued.

I/O ERROR (yyyy) ON LU xx-fd

indicates that an I/O error was encountered during an SVC1 read or write operation from a device or file. The lu is xx; yyyy is the error status.

INPUT TAPE SEQ. ERROR: EXPECTING-xx MOUNTED-yy

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.

INSUFFICIENT MEMORY AVAILABLE

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

INTERNAL FAILURE: ADDRESS=aaaaaa CODE=xxxx

indicates a failure in one of Backup's internal consistency checks. Please report this to your local Perkin-Elmer analyst after taking a dump of impure task space.

| INVALID DATE SPECIFICATION

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

| INVALID PARAMETER - xxxxxxxx

| indicates an error in one of the START parameters; xxxxxxxx  
| is the parameter in error.

| INVALID PARAMETER DELIMITER

| indicates syntax error in START command.

| INVALID SELECT FILE DESCRIPTOR-fn

| indicates invalid syntax in filename for selective restore.

| INVALID SELECT KEYWORD - xxxxxx

| indicates a keyword other than !INCLUDE or !EXCLUDE was found  
| in the select file record.

INVALID TAPE VOLUME xxxx, EXPECTING xxxx

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

| LOGICAL RECORD CANNOT FIT INTO OUTPUT BLOCK SIZE FOR fn

| indicates that the output block size is too small for the  
| logical record.

| NO FILES ON DISK - dev:

| indicates that no directory was found on dev: disk.

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 Rxx-yy OR HIGHER IS REQUIRED

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

OUTPUT DEVICE IS PROTECTED

indicates that the desired operation cannot proceed because the output device is protected.

PARAMETER ALREADY SPECIFIED

indicates that a parameter of the START command was entered more than once.

PARAMETER IN CONFLICT - xxxxxxxx

indicates that two parameters are in conflict; i.e., the APPEND, POSITION or NOREWIND options were used along with the VERIFY option.

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

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

PLEASE MARK OFF THE INPUT DISC

indicates that Backup has filled an output disk during a multiple disk back-up operation. Follow the procedure for multiple disk back-up detailed above.

PLEASE MARK OFF THE OUTPUT DISC AND MOUNT NEXT DISC VOLUME

indicates that the EOVS 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.

| REQUIRED PARAMETER MISSING - xxxxxxxx

| indicates a required parameter, such as IN=, OUT= or LIST=,  
| is not present. xxxxxxxx gives the expected parameter.

SELECT FILES EXCEED MAXIMUM

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

SELECT 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.

SELECT 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.



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. The Accounting Reporting Utility provides selection of the following:

- Collected accounting data to be processed (input files)
- Account numbers on which the reports or archives are generated
- The 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
- 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.

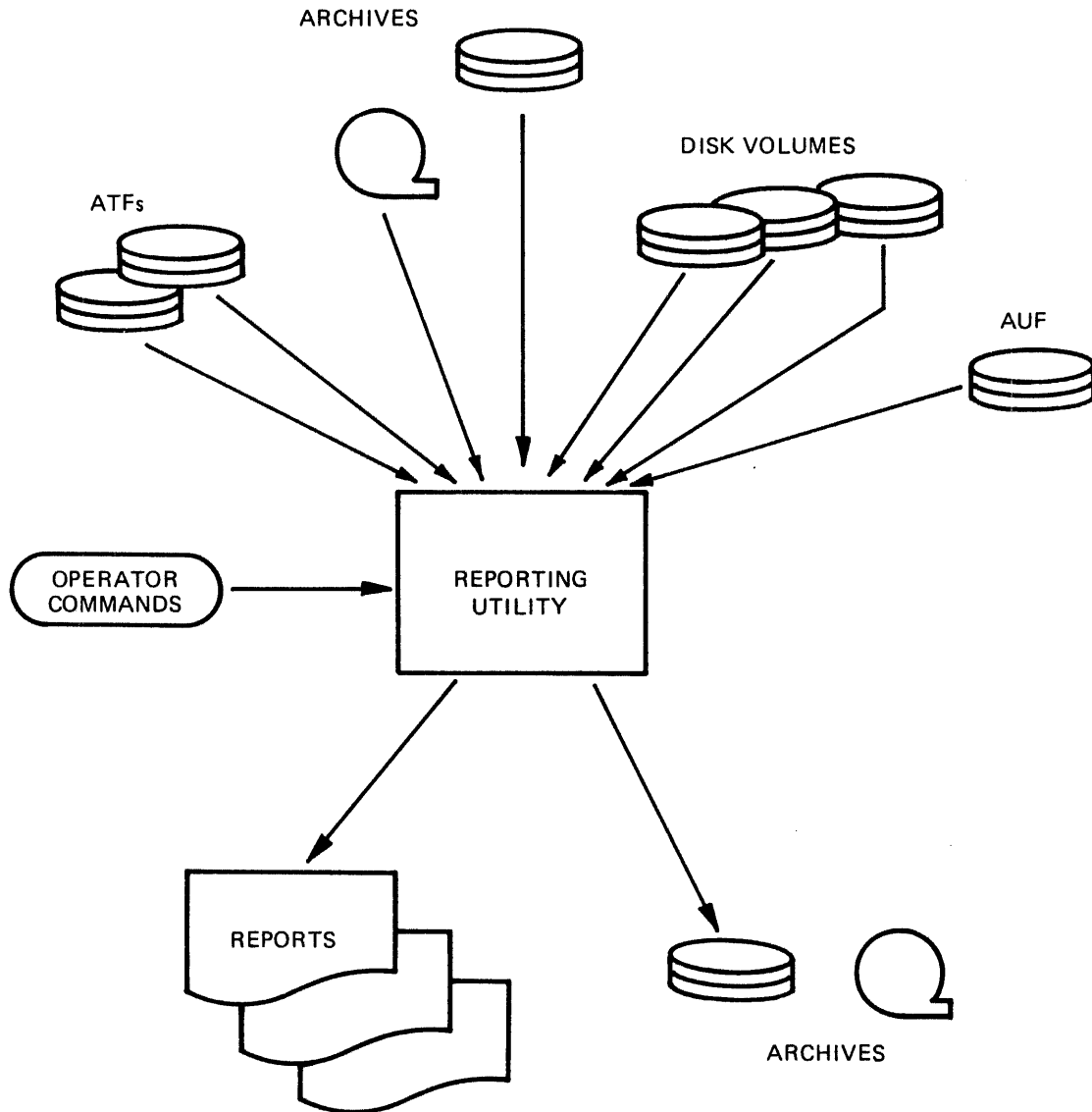


Figure 4-1 The Accounting Reporting Utility Process

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. Charge amounts specify the charge per unit of usage that 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 Operating System Without Multi-Terminal Monitor (MTM)

First, an operating system must have been generated (sysgened) 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 ATF.

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

```
.MTM QUIESCE
```

and

```
.MTM ATF fd
```

Where:

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=fd<sub>1</sub> ] [ ,LIST=fd<sub>2</sub> ] [ ,LOG=fd<sub>3</sub> ]

#### Parameters:

COMMAND= fd<sub>1</sub> specifies the input device from which commands are to be entered. If this parameter is omitted, the default is 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= fd<sub>2</sub> specifies the output device to which reports are sent. If this parameter is omitted, the default is 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= fd<sub>3</sub> 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.

UNABLE TO ASSIGN      COMMAND  
                          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 on-line 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:**

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

$$\left[ , \text{ARCHIVE} = \left\{ \begin{array}{c} \text{fd}_1 \\ ( \text{fd}_1, \text{fd}_2 [ , \dots, \text{fd}_n ] ) \end{array} \right\} \right]$$

$$\left[ , \text{DISC} = \left\{ \begin{array}{c} \text{volid}_1 \\ ( \text{volid}_1, \text{volid}_2 [ , \dots, \text{volid}_n ] ) \end{array} \right\} \right]$$

**Parameters:**

**ATF=** specifies the file descriptors of the ATFs created by the MTM START command 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= 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= specifies the disk volumes to be scanned for current file allocation data. If a specified volume is not on-line, a message requesting that volume to be marked on-line 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 DISC parameter includes all files that 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:

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

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

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 and must be enclosed within parentheses. The account numbers that can be specified by the SELECT command range from 0 to 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 that 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:

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

**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 is:

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

where 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 charge factors used to calculate charges for system usage. This command is used in conjunction with the REPORT command.

Format:

```

CHARGE [ CPU TIME = { O/dlrs.cts, U/dlrs.cts, A/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.

A/dlrs.cts specifies the charge in dollars and cents for each second of user auxiliary processing unit (APU) time used. If this parameter is omitted, no charges are calculated or included in the generated report. However, the time used is included. This applies to users of Model 3200MPS Systems.

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

dlrs.cts specifies the charge in dollars and cents (7 digits) for each second of operating system, user CPU time and, for Model 3200MPS Systems, user APU time used.

IOCOUNT=

classno/dlrs.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.

dlrs.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.

MEMORY=-

I/dlrs.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/dlrs.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 the number of .25kb segments of impure memory, pure memory or system space used are 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,A/0.60
    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:

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

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. Account numbers can be entered as a list or as a range and, in both instances, must be enclosed within parentheses. 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, an 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: OCT/ 4/83 TO OCT/ 5/83

REPORT DATE: NOV/14/83

NUMBER OF TASKS LOADED	1.	
OS CPU TIME	0: 0: 2.346	
USER CPU TIME	0: 0: 0.768	
USER APU TIME	0: 0: 3.277	
WAIT TIME	0: 3:11.056	
ROLL TIME	0: 0: 0.000	
ROLL CPU TIME	0: 0: 0.000	
ROLL COUNT	0.	
IMPURE MEMORY	6.00 KB	
PURE MEMORY	0.00 KB	
SYSTEM SPACE USED	1.25 KB	
NO. I/Os CLASS 0	2056.	
BYTES TRANSFERRED	263168.	
NO. I/Os CLASS 1	1054.	
BYTES TRANSFERRED	269824.	
NO. I/Os CLASS 2	40.	
BYTES TRANSFERRED	3219.00	
SECTOR DAYS PER VOLUME		
MTM	3036.0000	

Figure 4-2 System Summary Report

| The sample system summary report displayed in Figure 4-2  
| summarizes how the computer time was used by one user task  
| (u-task) from October 4, 1983 to October 5, 1983. The report  
| shows that:

- | ● There is one task.
- | ● The operating system occupied the CPU for 2.346 seconds.
- | ● The u-task occupied the CPU for 0.768 minutes and occupied the  
| APUs for 3.277 minutes.
- | ● The system occupied 6.00kb of impure and 0.00kb of pure memory  
| and 1.25kb of system space.
- | ● There were 2,056 index I/Os, transferring 263,168 bytes.
- | ● There were 1,054 contiguous I/Os, transferring 269,824 bytes.
- | ● There were 40 device I/Os, transferring 3,219 bytes.
- | ● The amount of sector days for volume=MTM was 3,036.



PERKIN-ELMER OS/32 ACCOUNTING REPORT

ACCOUNT SUMMARY REPORT

REPORT PERIOD: OCT/ 4/83 TO OCT 5/83

REPORT DATE: NOV/14/83

ACCOUNT NUMBER : 0-65535

PROCESSOR USAGE:

OS TIME:	0: 0: 2.346	
@ 1.5000/SEC.		3.52
USER CPU TIME:	0: 0: 0.768	
@ 0.7500/SEC.		0.58
USER APU TIME:	0: 0: 3.277	
@ 0.6000/SEC.		1.97
PROCESSOR SUBTOTAL		6.06

MEMORY USAGE:

SYSTEM SPACE:	1.25 KB	
@ 1.5000/0.25KB		7.50
IMPURE MEMORY:	6.00 KB	
@ 0.9000/0.25KB		21.60
PURE MEMORY:	0.00 KB	
@ 1.1500/0.25KB		0.00
MEMORY SUBTOTAL		29.10

I/O USAGE:

CLASS 0:		
BYTES TRANSFERRED:	0.263168D+06	
NO. OF I/Os:	2056.	
@ 1.0000/IO		2056.00
CLASS 1:		
BYTES TRANSFERRED:	0.269824D+06	
NO. OF I/Os:	1054.	
@ 1.1000/IO		1159.40
CLASS 2:		
BYTES TRANSFERRED:	0.321900D+04	
NO. OF I/Os:	40.	
@ 1.1500/IO		46.00
I/O SUBTOTAL		3261.40

DISC USAGE:

MTM : SECTOR DAYS	3036.0000	
@ 1.0500/SECTOR DAY		3187.80
DISC USAGE SUBTOTAL		3187.80

TOTAL CHARGES FOR THIS PERIOD 6484.36

J. DOE

Figure 4-3 Account Summary Report

The sample account summary report displayed in Figure 4-3 summarizes the amount of time an individual user, J. Doe, used the computer. It also shows the cost of user I/Os and prints out the total charges for the period from October 4, 1983 to October 5, 1983. This report shows that:

- The OS occupied the CPU for 2.346 seconds, the u-task occupied the CPU for 0.768 minutes and the APUs for 3.277 minutes.
- Memory was divided into 1.25kb of system space and 6.00kb of impure memory.
- There were 2,056 class 0 I/Os, transferring 0.263168D+06 bytes charged at \$1.0000 per I/O for a total of \$2,056.00.
- There were 1,054 class 1 I/Os, transferring 0.269824D+06 bytes charged at \$1.1000 per I/O transaction for a total of \$1,159.40.
- There were 40 class 2 I/Os, transferring 0.321900D+04 bytes charged at \$1.1500 per I/O transmission for a total of \$46.00.
- The charges for the disk usage were 3,036.0 sector days at \$1.0500 per sector day for a total of \$3,187.80.
- The total charges against J. Doe for the period are \$6.484.36.

PERKIN-ELMER OS/32 ACCOUNTING REPORT

ACCOUNT NUMBER : 255

REPORT PERIOD: OCT/ 4/83 TO OCT/ 5/83

REPORT DATE: NOV/14/83

ACCOUNT NUMBER : 255

PROCESSOR USAGE:

OS TIME:	0: 0: 2.346	
@ 1.5000/SEC.		3.52
USER CPU TIME:	0: 0: 0.768	
@ 0.7500/SEC.		0.58
USER APU TIME:	0: 0: 3.277	
@ 0.6000/SEC.		1.97
PROCESSOR SUBTOTAL		6.06

MEMORY USAGE:

SYSTEM SPACE:	1.25 KB	
@ 1.5000/0.25KB		7.50
IMPURE MEMORY:	6.00 KB	
@ 0.9000/0.25KB		21.60
PURE MEMORY:	0.00 KB	
@ 1.1500/0.25KB		0.00
MEMORY SUBTOTAL		29.10

I/O USAGE:

CLASS 0:		
BYTES TRANSFERRED:	0.263168D+06	
NO. OF I/Os:	2056.	
@ 1.0000/IO		2056.00
CLASS 1:		
BYTES TRANSFERRED:	0.269824D+06	
NO. OF I/Os:	1054.	
@ 1.0000/IO		1159.40
CLASS 2:		
BYTES TRANSFERRED:	0.321900D+04	
NO. OF I/Os:	40.	
@ 1.1500/IO		46.00
I/O SUBTOTAL		3261.40

DISC USAGE:

MTM : SECTOR DAYS	3036.0000	
@ 1.0500/SECTOR DAY		3187.80
DISC USAGE SUBTOTAL		3187.80

TOTAL CHARGES FOR THIS PERIOD 6484.36

J. DOE

Figure 4-4 Individual Account Summary Report

| The sample individual account summary summarizes how account  
| number 255 utilized the OS, CPU and APUs from October 4, 1983 to  
| October 5, 1983. The report shows that:

- | ● The OS occupied the CPU for 2.346 seconds.
- | ● The u-task occupied the CPU for 0.768 minutes and occupied the  
| u-task occupied the CPU for 0.768 minutes and the APUs for  
| 3.277 minutes.
- | ● System space occupied 1.25kb of memory and impure memory  
| occupied 6.00kb.
- | ● There were 2,056 class I/Os, transferring 0.263168D+06.
- | ● There were 1,054 class 1 I/Os, transferring 0.269824D+06  
| bytes.
- | ● There were 40 class 2 I/Os, transferring 0.321900D+04 bytes.
- | ● The amount of sector days used for volume=MTM was 3,036.

#### 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.

- A task termination code of 0 indicates normal termination.
- A task termination code of 1 indicates an invalid start option was used (see Section 4.2.2).
- 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 omitted.

FILE ERROR ON: fd 'ASSIGNMENT ERROR'

lu is already assigned or device is off-line.

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 SVCL 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 filename 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 off-line.

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 the '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 on-line to the system, causing the Reporting Utility to pause. Mark the specified disk on-line 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.
- Save error log information on an archival file.

### 5.2 DESCRIBING MEMORY TO INTERPRET ERRORS

To describe memory to interpret errors, users 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 the following:

- Blocks
- Banks
- Modules

Memory is also logically broken into storage units.

A block is an area in memory to which a specific number of megabytes 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 Model 3220, 3240 and 3250 Processors Shared Memory Systems Installation and Maintenance Manuals.

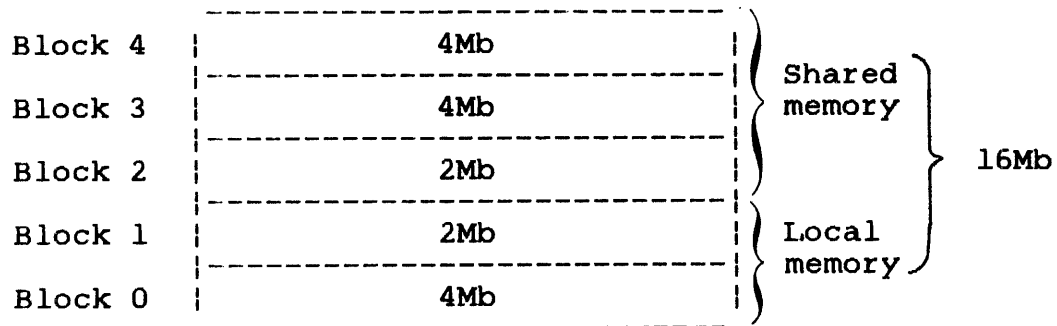


Figure 5-1 Memory Described as Blocks

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

- noninterleaved (1 bank),
- 2-way interleaved (2 banks), or
- 4-way interleaved (4 banks).

Banks within a block are equal in size and each 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	} 16Mb
Block 3	1Mb	1Mb	1Mb	1Mb	
Block 2	1Mb	1Mb			
Block 1	1Mb	1Mb			
Block 0	4Mb				
					4-way interleaved
					4-way interleaved
					2-way interleaved
					2-way interleaved
					Noninterleaved

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 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** (1,024kb or 1Mb)	DDSTM (half populated with 64K chips)
F04** (2,048kb or 2Mb)	DDSTM (fully populated with 64K chips)

\* Used on Perkin-Elmer Model 3220 and 3240 processors.

\*\* Used on Perkin-Elmer Model 3210, 3230, 3250XP and 3200MPS 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.

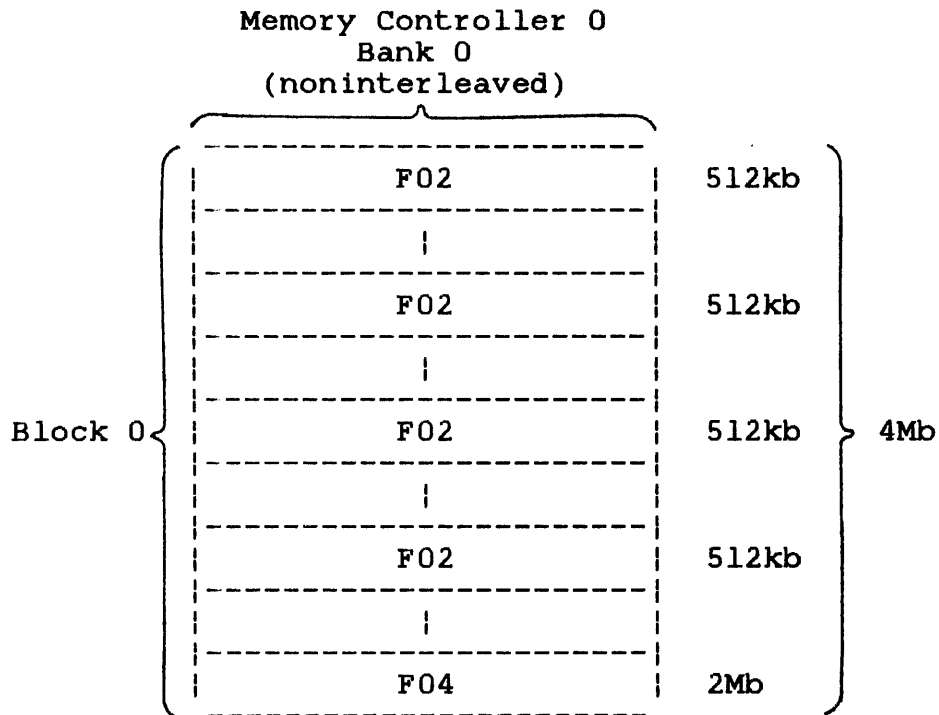


Figure 5-3 A Bank Broken into Modules

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 to 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 to 16, depending on the size of the bank. See Figure 5-4.

Memory Controller 0  
 Bank 0  
 (noninterleaved)

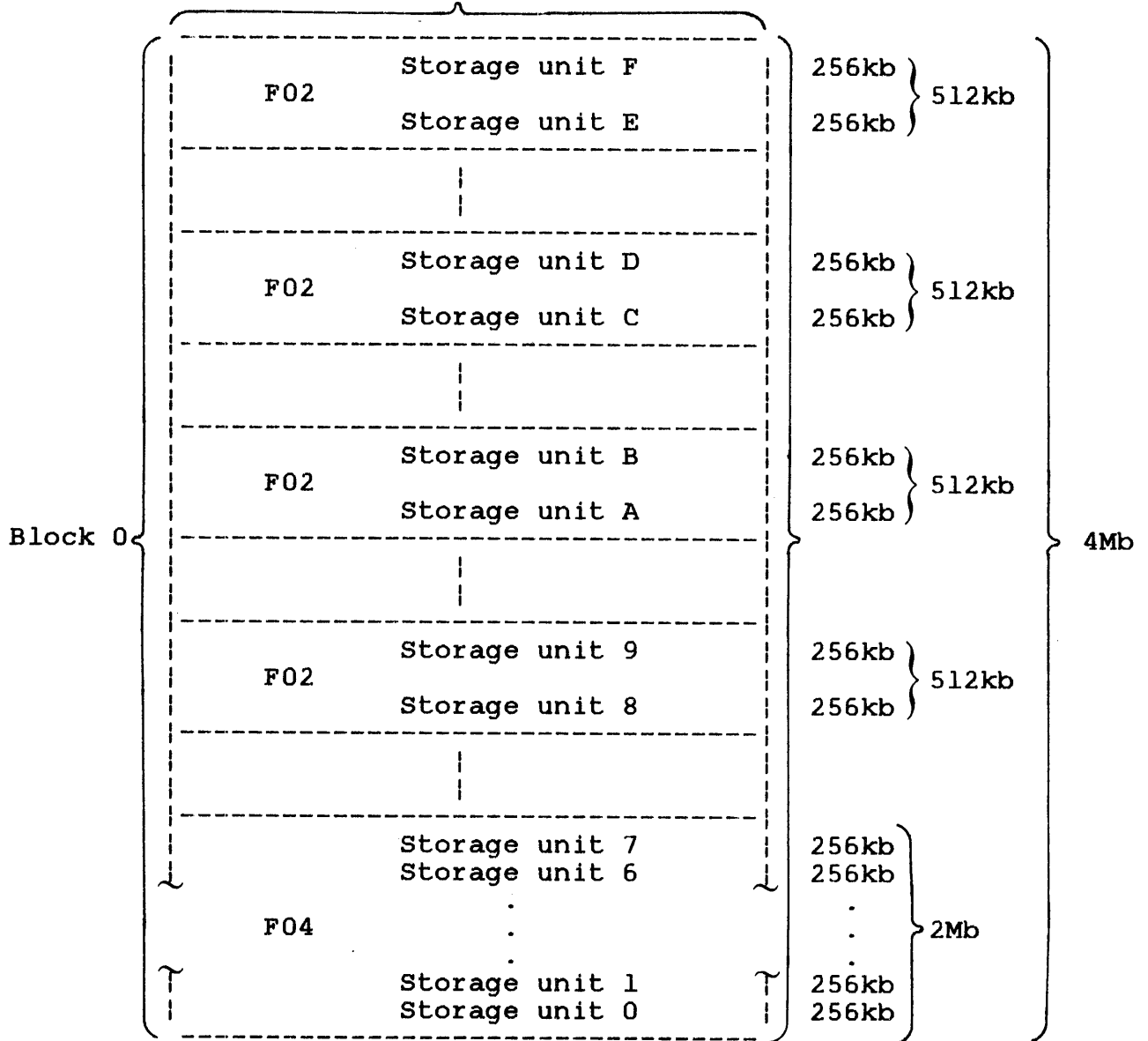


Figure 5-4 A Bank Broken into Storage Units

### 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=fd<sub>1</sub> ] [ , LOG=fd<sub>2</sub> ] ]

#### Parameters:

COMMAND= fd<sub>1</sub> specifies the file descriptor of the command input device. If this parameter is omitted, the console device (CON:) is the default. If fd is not an interactive device, a log device must also be specified.

LOG= fd<sub>2</sub> 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.
- 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 to 16. A block consists of from 1Mb 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 or 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:

- The memory configuration procedure does not allow the entry of block sizes of less than a whole megabyte.
- 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. A block has more than one bank only if it is interleaved. The valid responses have the following meanings:

- 1 = Noninterleaved
- 2 = 2-way interleaved
- 4 = 4-way interleaved

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 to 64.

Error Message: INVALID NUMBER OF MODULES

Recovery: The program reissues the prompt.

- Prompt: ENTER MODULE TYPES

Response: module type ,module type ,...,module type

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	16kb	Single
F01	256kb	16kb	Double
F02	512kb	16kb	Double
F03	1024kb or 1Mb	64kb	Double
F04	2048kb or 2Mb	64kb	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

Examples:

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 define BLOCK 1, BANK 0.

```
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.

```
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.

```
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:

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

## Parameters:

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

## Functional Details:

The first date format is valid only if the United States 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 sections show formats for list and summary reports.

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

#### Format:

031-1

```
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 = iiiiiiiii
```

#### 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 can also 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 allocated disk sector

**Format:**

031-2

```
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:**

031-3

```
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:**

031-4

```
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 or index data sector beyond disk range

**Format:**

031-5

```
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:**

031-6

```
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 report types are generated when the system detects an error.

- Report Type 1 - Task queue overflow

#### Format:

031-7

```
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; overflow possible

#### Format:

031-8

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

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

#### Format:

031-9

```
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:

031-10

```
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 that 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:

031-11

```
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:**

031-12

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

The following report types are only applicable to the Model 3200MPS System and are supported by the R07.2 software release and higher.

- Report Type 7 - During task wait removal, it is found that the APU is waiting for a control block that is inconsistent with the TCB's "APU wait" status.

**Format:**

031-13

```
mon/dd/yy hh:mm:ss REMOVE APU WAIT ERROR
APU#n APU WAIT = wwww TCB STATUS = ssssssss TCB WAIT = twtwtwtw
```

- Report Type 8 - During task cancellation, it is found that the the APU is waiting for a TCB that is inconsistent with the TCB's "APU wait" status.

**Format:**

031-14

```
mon/dd/yy hh:mm:ss APU RELEASE RESOURCE ERROR
APU#n APU WAIT = wwww TCB STATUS = ssssssss TCB WAIT = twtwtwtw
```

**Fields:**

APU#            n is a hexadecimal number from 1 to 9 that identifies the particular APU involved.

APU WAIT =    wwww is a hexadecimal number indicating the TCB ID of the task for which the APU is waiting.

TCB STATUS = ssssssss is a hexadecimal number indicating the task status of the TCB.

TCB WAIT =    twtwtwtw is a hexadecimal number indicating the task wait state of the TCB.

- Report Type 9 - An APU signal indicating task passback is detected, whereupon the CPU task receive queue is found to be empty.

Format:

031-15

```
mon/dd/yy hh:mm:ss CPU RECEIVE QUEUE EMPTY ERROR
APU SIGNAL = ssssssss APU LEAF ADDRESS = 11111111
```

Fields:

APU SIGNAL = ssssssss is a hexadecimal number representing the APU signal received.

APU LEAF ADDRESS = 11111111 is the hexadecimal address of the APU event leaf.

- Report Type 10 - An APU signal indicating task passback is detected, whereupon the CPU task receive queue is found to be locked.

Format:

031-16

```
mon/dd/yy hh:m:ss CPU RECEIVE QUEUE LOCKED ERROR
COUNT = cccc QUEUE = qqqqqqqq
```

Fields:

COUNT = cccc is a hexadecimal number indicating the number of tasks present on the CPU receive queue when it locked.

QUEUE = qqqqqqqq is a hexadecimal number indicating the receive queue address.

- Report Type 11 - APU execution queue is locked indefinitely, perhaps by an APU assigned to the queue.

Format:

031-17

```
mon/dd/yy hh:m:ss APU EXECUTION QUEUE LOCKED ERROR
BIT MAP = bbbb QUEUE = qqqqqqqq
```

**Fields:**

BIT MAP = bbbb is a hexadecimal number representing the APU-to-queue assignment bit map.

QUEUE = qqqqqqqq is a hexadecimal number indicating the APU queue address.

- Report Type 12 - the APU does not respond to status interrogation and/or control commands.

**Format:**

031-18

```
mon/dd/yy hh:mm:ss APU NO RESPONSE ERROR
APU#n FLAGS = ffffffff
```

**Fields:**

APU# n is a hexadecimal number from 1 to 9 that identifies the particular APU involved.

FLAGS = ffffffff is a hexadecimal number indicating the value of the APU interval state field (APB.FLGS).

- Report Type 13 - APU data or a signal is received through a real-time support module (RTSM) with parity error.

**Format:**

031-19

```
mon/dd/yy hh:mm:ss RTSM RECEIVE PARITY ERROR
APU#n DATA BYTE = vv FLAGS = ffffffff
```

**Fields:**

APU# n is a hexadecimal number from 1 to 9 that identifies the particular APU involved.

DATA BYTE = vv is a hexadecimal number representing the data or signal byte received.

FLAGS = ffffffff is a hexadecimal number indicating the value of the APU internal state field (APB.FLGS).

#### 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:

031-20

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

##### Fields:

DEVICE                xxxx is the mnemonic of the device being marked on-line.

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 on-line.

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

- Report Type 1B - Marking on a nonbulk device

##### Format:

031-21

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

##### Fields:

DEVICE                xxxx is the mnemonic of the device being marked on-line.

DEVICE NUMBER        yyyyyyyy is a decimal number specifying the device being marked on-line.

● Report Type 2A - Marking off a bulk device

Format:

031-24

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

Fields:

DEVICE                    xxxx is the mnemonic of the device being marked off-line.

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 off-line.

TOTAL ERRORS =            nnnnnnnnnn is a decimal number specifying the total number of errors that occurred on the device before it was marked off-line.

● Report Type 2B - Marking off a nonbulk device

Format:

031-23

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

Fields:

DEVICE                    xxxx is the mnemonic of the device being marked off-line.

DEVICE NUMBER =            YYYYYYYY is a decimal number specifying the device being marked off-line.

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

Format:

031-22

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



**Fields:**

xxxx is the volume name of the device.  
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:**

031-25

```
mon/dd/yy hh:mm:ss ERROR RECORDING RESUMED
TOTAL ERRORS LOST          bbbbbbbb
MEMORY ERRORS             dddddddd
BULK DEVICE I/O ERRORS    mmmmmmmmm
FILE MANAGER DETECTED ERRORS nnnnnnnn
SYSTEM DETECTED ERRORS   xxxxxxxxx
SYSTEM MILESTONES        yyyyyyyyy
```

**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 mmmmmmmmm 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 xxxxxxxxx is the total number of system detected errors.

SYSTEM MILESTONES yyyyyyyyy is the total number of system milestones.

- Report Type 5 - System queue reached its depth.

**Format:**

031-26

```
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 is 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:**

031-27

```
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 on all Perkin-Elmer systems except the Model 3205. The module data is omitted if a memory configuration definition file was not specified.

**Format:**

031-28

```
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 through F) where the error occurred. S is included if the block is shared memory.

**BANK=** n is the bank number (0 to 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 through 38) where the error occurred. This field can also contain asterisks (\*\*), indicating that 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 (e.g., F01, F02, F03, F04).

If the Error Reporting Utility is executing on a Model 3205 System, the report has the following format.

**Format:**

031-29

mon/dd/yy hh:mm:ss MEMORY ERROR ROW=r CHIP=c

**Fields:**

**ROW=** r is a numeric value (0 through 32) indicating the row number of the chip where the error occurred.

**CHIP=** c is an alphanumeric value indicating which chip in the row has failed. Valid values are 0 through 15 for data bits and C0, C1, C2, C4 and C8 for check bits.

## 5.7.2 Summary Reports

There are three types of summary reports:

- System errors and milestones
- Device errors
- Memory errors

### 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. Additional errors of the same type, but different tasknames, are listed under the category ALL OTHER TASKS. Error types itemized by APU number indicate the number of occurrences of the error for each APU within the Model 3200MPS System.

Format:

031-30

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

## REMOVE APU WAIT ERROR

APU#1	nnnnnn
APU#2	nnnnnn
APU#3	nnnnnn
.	
APU#9	nnnnnn

## APU RELEASE RESOURCE ERROR

APU#1	nnnnnn
APU#2	nnnnnn
APU#3	nnnnnn
.	
APU#9	nnnnnn

CPU RECEIVE QUEUE EMPTY ERROR	jjj
-------------------------------	-----

CPU RECEIVE QUEUE LOCKED ERROR	kkk
--------------------------------	-----

APU EXECUTION QUEUE LOCKED ERROR	lll
----------------------------------	-----

APU NO RESPONSE ERROR	nnnnnn
-----------------------	--------

APU#1	nnnnnn
APU#2	nnnnnn
APU#3	nnnnnn
.	
APU#9	nnnnnn

## RTSM RECEIVE PARITY ERROR

APU#1	nnnnnn
APU#2	nnnnnn
APU#3	nnnnnn
.	
APU#9	nnnnnn

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.

taskname 1... taskname 9      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.

| \*APU#1... APU#9      nnnnnn is a decimal number indicating the total errors recorded for each APU.

| \*CPU RECEIVE QUEUE EMPTY      jjj is a decimal number indicating the total number of times the CPU task receive queue was found to be empty in error.

| \*CPU RECEIVE QUEUE LOCKED      kkk is a decimal number indicating the total number of times the CPU task receive queue was locked in error.

| \*APU EXECUTION QUEUE LOCKED      lll is a decimal number indicating the total number of times the APU execution queue locked.

| \* These messages are only applicable to Model 3200MPS Systems using APUs.

### 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:**

031-32

```

DEVICE  device name1

  BIT MAP ALLOCATION ERRORS          nnnnnn
  BIT MAP DEALLOCATE ERRORS         nnnnnn
  DIRECTORY BLOCK ERRORS            nnnnnn
  DIRECTORY ENTRY ERRORS            nnnnnn
  INDEX/DATA BLOCK ERRORS           nnnnnn

DEVICE  device name9

  BIT MAP ALLOCATION ERRORS          nnnnnn
  BIT MAP DEALLOCATE ERRORS         nnnnnn
  DIRECTORY BLOCK ERRORS            nnnnnn
  DIRECTORY FILE ERRORS             nnnnnn
  INDEX/DATA BLOCK ERRORS           nnnnnn

  .
  .
  .

ALL OTHER DEVICES

  BIT MAP ALLOCATION ERRORS          mmmmmn
  BIT MAP DEALLOCATE ERRORS         mmmmmn
  DIRECTORY BLOCK ERRORS            mmmmmn
  DIRECTORY ENTRY ERRORS            mmmmmn
  INDEX/DATA BLOCK ERRORS           mmmmmn

I/O ERRORS

  DEVICE NAME      ADDRESS      I/O ERRORS
  xxxx             YYYYYYYY     bbbbbbbb
  xxxx             YYYYYYYY     bbbbbbbb
  xxxx             YYYYYYYY     bbbbbbbb
  .                .                .
  .                .                .
  .                .                .
  xxxx             YYYYYYYY     ssssssss

ALL OTHER I/O ERRORS
  
```

**Fields:**

device name1...  
device name9            are the names of up to nine devices on 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 is the name 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 the 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:**

031-33

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

**Fields:**

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



**BANK** n is the bank (0 through 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 OF FAILURES** nnnn is a decimal number indicating the total number of memory errors occurring within the blocks displayed.

The following memory errors summary report is generated for the Model 3205 System.

**Example:**

031-34

ROW	CHIP	NUMBER OF FAILURES
r	c	nnnn
.	.	.
.	.	.
.	.	.

**Fields:**

**ROW** r is the row number of the chip where the error occurred. The row number will be in the range of 0 to 32.

CHIP c is an alphanumeric value indicating what chip in what row has failed. Valid values are 0 through 15 for data bits and C0, C1, C2, C4 and C8 for check bits.

NUMBER OF FAILURES nnnn is a numeric value indicating the number of failures.

## 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 nonzero 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 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.
- 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 BACKUP OR FASTBACK 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.
- 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  $48 \times 256$  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 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 using the ASSIGN command as follows:
  - Disk device to logical unit 1 (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
```

Where  $xx$  is the current revision level of the program and  $yy$  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 7.2 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:

FNМ:	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 [ { ED=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 multisector 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:**

The following example dumps sectors 0 through 10 in hexadecimal and ASCII to lu3.

DUH SECTOR=0,10

The following example dumps sector 1 in hexadecimal and ASCII to lu3.

DUH SECTOR=1

The following example dumps to lu3, in ASCII, all sectors occupied by file RUN.CSS.

DUA F=RUN.CSS

The following example dumps all sectors in ASCII to lu3.

DUA

The following example dumps binary to lu2 sectors marked as allocated in the bit map.

DUS \*

The following example dumps all blocks of sectors to lu2.

DUL

The following example dumps to lu2 only those blocks of sectors with at least one sector allocated.

DUL \*

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 multisection 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 16Mb disk should not be dumped to a 67Mb or 256Mb disk, and vice versa.

**Examples:**

The following example reads a sector from lu2 and writes it to the disk as sector 0.

```
UDS SEC=0
```

The following example 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.

```
UDS *
```

The following example 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 is 48.

UDL \*

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
REWIND lu	Rewind
RW lu	Rewind
WFILE lu	Write filemark
FFILE lu	Forward to end of file
BFILE lu	Backspace to beginning of file
BRECORD lu	Backspace record
FRECORD lu	Forward space record

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

COMMAND	FUNCTION
WIDTH 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 for 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 3 (SVC3) 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 lul is not a disk device or is an unsupported type of disk. 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 nonzero 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 nonzero status is returned on a disk I/O operation. The SVCL device-independent status is ss; dd is the SVCL 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 SVCL status, the I/O operation is retried 10 times before the message is logged; another command is then read from lu5.

#### MEM-FULL

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

#### OS/32 Rxx-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
- Contents of memory

The Dump Print Utility requires the following system resources:

- OS/32
- 18.25kb of memory in addition to that of the operating system
- A disk device
- 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 7.2 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

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 lu1.

### NOTE

This prompt is not issued if lu1 was preassigned to a permanent disk file.

Prompt:           ENTER LIST DEVICE FD:>

Response:        fd

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 until the end of operating system code to the list device. If the user response is NO, the following 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: ENTER PRINT RANGE (LOLIM,HILIM)?>

Response:  $\left[ \left[ \left( \begin{array}{l} \text{st mem adr} \\ 0 \\ \text{OS} \\ \text{SYSTEM} \end{array} \right) \right] \left[ \left( \begin{array}{l} \text{end mem adr} \\ \text{top of memory} \end{array} \right) \right] \right]$

Where:

- st mem adr        is the starting memory address of the range whose contents are to be printed on the list device. The default starting address is X'0'.
- end mem adr       is the ending memory address of the range whose contents are to be printed on the list device. The default ending address is top of memory (MTOF).
- OS                specifies that all memory occupied by the operating system is to be printed on the list device.
- SYSTEM            specifies that all memory defined as system space is to be printed on the list device.

NOTE

The command input device defaults to the console (CON:) and is automatically assigned to lu5. The temporary file is a default and is automatically assigned to lu2. If no values are specified and a carriage return is entered, all memory is assumed.

After these user responses are entered or lu assignments are made, the contents of the memory dump on magnetic tape are copied to the temporary file or permanent file. If the contents of the specified range are corrupt and cannot be interpreted, an error message is displayed. After the Dump Print Utility dumps as much memory as it can, the following 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):>2C200,3EEFE
DUMP MORE MEMORY?>NO
.BG - END OF TASK CODE = 0 CPU TIME = n.nnn/n.nnn
*
```

Appendix A contains a sample display from the Dump Print Utility in effect for release R06.2 and higher corresponding to the dialogue above. 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 B contains fragments of a sample display from the Dump Print Utility, in effect for software release R07.2 and higher, corresponding to the dialogue above. The dump print display contained in Appendix B 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 operating system 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 (SCL) are not produced.

## 7.6 DUMP PRINT STRUCTURE DISPLAY

The following sequence of structure display is in effect for the Dump Print Utility, release R07.2 and higher, for all processors. Please note differences for the Model 3200MPS System.

- SPT
- TCB followed by its context block (CTX) for each task assigned to the central processing unit (CPU)
- For each APU queue parameter block (QPB) in a Model 3200MPS System, the TCB and its associated CTX for each task assigned to the QPB, followed by the QPB
- TCBs and their associated CTXs for all tasks in the TCB table not already output
- An auxiliary processor block (APB) for each APU in a Model 3200MPS System
- Segment descriptor entries (SDEs) for any pure segments in the SCL
- A device control block (DCB) followed by its associated CTX (if any) for each device in the system
- A volume mnemonic table (VMT)
- Coordination nodes (EVNs)

## NOTE

For systems other than the Model 3200MPS, no APBs or QPBs 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 AUXILIARY PROCESSING UNIT (APU) QUEUE PARAMETER BLOCK (QPB) SUMMARY DISPLAY

This summary page will be displayed following the display of the floating point registers. Output of the summary will be contingent upon finding a nonzero quadword aligned address in the SPT.QPB0 field of the SPT. Following are the summary contents.

TITLE	DUMP OF QPB INFORMATION
ID	This field contains the queue ID from the QPB.ID field of the current QPB.
ADDR	This field contains the hexadecimal address of the QPB.
STATUS	This field displays the queue processing status.
APUS	This field displays the APU assignment bitmap in hexadecimal form. The bit number of the bitmap corresponds to the number of the APU assigned to the queue.
QUEUED	This field displays the number of tasks on the queue as contained in the respective counter in the QPB.
MAPPING TASK	This field displays the TCB.ID of a task granted mapping rights (via SVC13) to the queue. The word NONE will be displayed if no task is assigned control rights.
EXCL.TASK	This field displays the TCB.ID of the task holding exclusive rights to the queue.

Example of the APU queue summary:

DUMP OF QPB INFORMATION

ID	ADDR	STATUS	APUS	QUEUED	MTID	XTID
0	09C0	1098	1000	1	000F	000F
1	09E0	....	....	.	....	....

7.8 AUXILIARY PROCESSING UNIT (APU) AUXILIARY PROCESSOR  
BLOCK (APB) SUMMARY DISPLAY

This summary page will be displayed following output of the floating point registers. Output of the APB summary page is contingent upon finding nonzero values at locations X'C0' and X'C4'. Following are the summary contents.

TITLE	DUMP OF APB INFORMATION
ID	This field contains the APU number from the APB.ID field of the current APB.
ADDR	This field contains the hexadecimal address of the current APB. It must be quadword aligned or the word INVALID will be output in this field and no further fields for the APB will be processed.
STATUS	This field displays the APU state as contained in the APB.STAT field of the current APB.
CTCB	This field displays the address of the current TCB contained in the APB.CTCB field of the current APB. If the address is not quadword aligned, the message INVALID is displayed and no further fields for this APB are processed.
STATE	This field will display internal state information for the APU contained in the APB.FLGS field of the current APB.
QUEUE	This field will display the address of the QPB to which the APU is assigned. The address is contained in the APB.ARQP field. If this address is not quadword aligned, the word INVALID will be displayed in this field.
CONTROL TASK	This field will display the TCB.ID of the task granted control rights (via SVC13) for this APU. The word NONE will be displayed if no task is assigned control rights.

WAIT TASK            This field will display the TCB.ID of the task on which this APU is "waiting" for completion of fault servicing. The word NONE will be displayed if there is no task wait on the APU.

Example of the APB summary report:

DUMP OF APB INFORMATION

ID	ADDR	STATUS	CTCB	STATE	QUEUE	CTID	WRID
1	07C0	FFFF	FE37C0	A0000001	E7C890	NONE	0010
2	08FF	....	.....	.....	.....	....	....

## 7.9 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 off-line.

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 or 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 logical unit.



#### NAME-ERR

indicates that the specified filename and extension do not exist.

#### POINTER TO SPT INVALID-CANNOT DUMP OS

A dump of operating system memory was requested but the memory range of the operating system 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 a device is marked off-line.

#### 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 SVCL parameter block; dd is the device-dependent status byte. See the OS/32 Application Level Programmer Reference Manual.

## CHAPTER 8 MIRROR DISK SYNCHRONIZATION UTILITY

### 8.1 MIRROR DISK FACILITY

The mirror disk facility ensures continued operation of the system, without operator intervention, following a disk failure by automatic maintenance of pairs of disks in physical synchronization. It provides for automatic continuation, on-line resynchronization and uninterrupted availability of data while being transparent to users. It also allows for orderly reconfiguration through operator requested switching of mirrored disks.

When the operating system is informed by the MARK ON command that there are to be mirror disks, it directs all writes to two disks. This not only includes writes to the files, but also file allocation, file renaming, etc. No program changes are required. Reads continue to be scheduled from one of the disks. When mirror disks are present, this read disk is called the primary disk. The other partner of the pair is called the secondary disk.

If a single disk fails for any reason, the system detects the failure, alerts the system operator and automatically switches to use the remaining disk. When the failure is corrected, the recovery process is able to proceed in parallel to normal operation.

The operation of mirror disks has impact in the following areas:

- The MARK ON command informs the operating system of the presence of mirror disks with the use of an optional parameter. When the disks are marked on, certain criteria must be met. The synchronization stamps must agree and a synchronization bit must be set on both disks. If these criteria are met, both disks will be marked on and mirroring operations will proceed. If these criteria are not met, the operator will be advised that synchronization is necessary and the Synchronization Utility (DISCSYNC) must be run.
- The DISPLAY DEVICES command output indicates the presence of mirror disks and identifies the primary and secondary disks.
- The SWOP command allows the primary and secondary disks to be exchanged.
- The DISCSYNC Utility must be run to synchronize disks whenever one of the pair has been operating in an unprotected mode.

The operational features of the MARK ON, DISPLAY DEVICES and SWOP commands, within the mirrored disk environment, are discussed in the OS/32 Operator Reference Manual. Refer to the OS/32 Application Level Reference Manual for general information on disk organization and control. The operational features of the DISCSYNC Utility are discussed in the remainder of this chapter.

## 8.2 OVERVIEW

The Mirror Disks Synchronization Utility (DISCSYNC) is a support component of the mirror disk facility. It copies one disk to another so the two disks can operate as a mirrored pair. The utility can also be used to verify that two disks are identical as a result of the copy operation. While the utility is running, all OS/32 application programs can access data on the disks as usual.

When DISCSYNC has completed successfully, the disks are said to be in synchronization. All used parts of the disks are identical bit for bit, except for some system data.

DISCSYNC runs as a stand-alone utility. It is a nonrollable, privileged user task (u-task).

## 8.3 FUNCTION

DISCSYNC copies one disk to another so that the two disks can be operated as a pair of mirrored disks. It also verifies that the two disks are identical as a result of the copy operation. In addition, if synchronization is requested for disks that are already synchronized, DISCSYNC does not repeat the synchronization. This situation could occur when the operation of DISCSYNC is requested from a command substitution system (CSS) file.

The utility can be run in three different situations, although the function and operation of the utility are the same in each case. The utility is used in the following circumstances:

- To synchronize two empty disks that are being included in a system for the first time.
- To resynchronize two disks that have been operating as a mirrored pair after some error has occurred. In this case, one of the disks may or may not have been temporarily in operation as a single disk while the error situation was resolved.

- To synchronize two disks when one of them contains valid data and the other is a fresh disk. One example of this is when an existing system is upgraded to a mirrored system. Another is when one disk of a mirrored pair becomes unusable as such and must be replaced.

The use of the utility does not affect the availability of data from the good disk in the latter two cases. The utility may optionally verify that synchronization is successful after each buffer is copied by rereading the buffers from both disks and comparing them. Thus, both the read from the source disk and the write to the destination disk are verified.

The utility runs with both disks marked on, whether they were marked on together or separately. Both disks must have been initialized by the FASTCHEK Utility and given the same volume name before being synchronized for the first time. After an error situation, the good disk must be checked by the FASTCHEK Utility as needed by the marking on operation. Resynchronization can run in parallel with OS/32 application programs accessing the disk.

DISCSYNC provides some measure of optimization of the synchronization. Large contiguous areas of unused disk space will not be copied or verified.

#### 8.4 OPERATING PROCEDURES

The following are operating procedures for the DISCSYNC Utility.

##### 8.4.1 Preliminary Steps

Before running the DISCSYNC Utility, do the following:

- Use the FASTCHEK Utility to initialize and record bad sector information for both disks. This must be done for all disks being mirrored for the first time, or disks that have just been reformatted. Otherwise, this step is not necessary. Both disks must be initialized with the same volume name.

After an error situation where both disks were affected, FASTCHEK should be run in checking mode on the primary disk as normally required by the MARK ON command for a single disk. It is not normally necessary to use FASTCHEK for the secondary disk.

- The two disks supplied must be marked on as a mirrored pair by means of one or more MARK ON commands. Mark on the disks as a mirrored pair if they are compatible. The MARK ON command of mirrored disks fails if the secondary disk has any bad sectors present in positions where the primary disk has data.

| 8.4.2 Loading DISCSYNC

| When the Mirror Disk Synchronization Utility (DISCSYNC) is to be  
| loaded, use the following procedure.

| Format:

| LOAD taskname,DISCSYNC [n]  
| TASK taskname

| Parameters:

| taskname specifies any valid OS/32 task name.  
| DISCSYNC specifies the name of the file containing the  
| task.  
| n specifies the segment size increment. The  
| default and minimum value permitted is 16kb.  
| To minimize disk seek time and to avoid  
| wasting memory, the value of n should be a  
| multiple of the disk track size and a factor  
| of the cylinder size, but not larger than the  
| cylinder size. This is applicable when the  
| VERIFY option is not used. When the VERIFY  
| option of the START command is used, n should  
| be doubled, if possible. See Section 8.4.5  
| for a discussion of the effect of the segment  
| size increment upon the operation of DISCSYNC.

| 8.4.3 Starting DISCSYNC

| Start the DISCSYNC Utility using the START command.

| Format:

| START,DRIVE=drive-name [ ,LOG={log-fd} ] [ ,VERIFY={YES} ]  
| CON: {NO} ]

| The parameters of the START command may be input in any order.

## Parameters:

DRIVE= drive-name specifies the device name of the drive holding the primary disk from which data is to be copied.

LOG= log-fd specifies the device name or file descriptor to which messages are to be written. The default is CON:, the system console.

VERIFY= is an optional parameter indicating whether or not verification is desired. Verification consists of a comparison of the two disks. The default is NO.

## Functional Details:

The utility reads sector 0 from both disks and checks for synchronization. If the disks are synchronized, the utility outputs a message and goes to end of task. If the disks are not synchronized, the utility automatically proceeds to put both disks in full mirrored operation. No further operator action is required. If the VERIFY option is in effect, verification will be performed upon completion of the operation. Upon successful completion (end of task 0), the two disks will be marked on and in synchronization.

## Examples:

In the following example, D300: is the drive containing the primary disk. Messages will be sent to the file MIRR.LOG, and the two disks will be verified as identical when the mirroring operation is completed.

```
ST,DRI=D300:,LOG=MIRR.LOG,VER=Y
```

In the following example, D67B: is the drive containing the primary disk. Messages will be sent to the printer, and, by default, verification will not be performed.

```
ST,DRI=D67B:,LOG=PR:
```

#### 8.4.4 End of Task Codes

The following is a list of possible end of task codes:

- |     |  |
|-----|--|
| 0   | Successful completion.   |
| 1   | Operational error, probably in start parameters. Rerun correctly. The primary disk remains available for applications.   |
| 2   | I/O errors or bad sectors on one of the disks are preventing synchronization. Reformat or replace the disk and then rerun. The primary disk will remain available if the error was on the secondary disk; otherwise, neither disk will be available. |
| 254 | Internal error. A memory dump should be taken to allow analysis of the problem.  |

#### 8.4.5 Efficient Operation

The amount of time required for synchronization is influenced by three factors: the size of the disks used as the mirrored pair, the amount of I/O activity on the mirrored pair during synchronization, and the segment size specified when DISCSYNC is loaded.

The first factor is obvious. A mirrored pair of 256Mb disks requires more time to synchronize than a mirrored pair of 64Mb disks.

The amount of I/O activity on the mirrored pair may or may not be controllable by the operator. The ideal situation exists when DISCSYNC is the only task accessing the mirrored pair. It is more likely that other tasks will require concurrent I/O on the mirrored pair and, therefore, affect synchronization time.

The third factor, segment size, allows some degree of optimization. When DISCSYNC is loaded, the segment size specified by *n* is used as buffer space for I/O operations. If VERIFY = NO is specified, the segment size increment is used as one I/O buffer. If VERIFY = YES, the segment size increment is divided into two buffers of equal size.

The greatest amount of time used in I/O to a disk is during the movement of heads from cylinder to cylinder. By tailoring the size of the I/O buffers (via segment size) to the size of the cylinders on the mirrored pair, the movement of heads can be minimized.

If the buffer size specified is a fractional factor of the cylinder size (i.e., 1/3), then it will take three read-write operations to copy one cylinder. In this case, the I/O will be arranged so that the first of every three copies is aligned on a cylinder boundary. Thus, there will be no head movement, or seeking, between cylinders during a copy operation. If the buffer size is not a fractional factor of the cylinder size (e.g., 2/3), there will be seeking between cylinders during some copy operations.

It is advisable to be aware of the following when determining the segment size to be used when loading DISCSYNC. See Table 8-1 for segment size guidelines.

- To avoid wasted memory, n should be an exact multiple of the disk track size and not more than the disk cylinder size. Any extra memory designated will be unused.
- To avoid seek time during I/O transfers, n should be a fractional factor (1/2, 1/3, etc.) of the cylinder size. This applies only when DISCSYNC is sharing the mirror disk pair with other tasks.
- If the first priority is to complete the operation of DISCSYNC as quickly as possible, while DISCSYNC is accessing the disk pair concurrently with other tasks, n should be as large as memory allows, subject to the two guidelines above.
- If the first priority is to maintain the performance and response time of u-tasks, while DISCSYNC is accessing the disk pair concurrently with other tasks, n should be as small as possible, subject to the first guideline above.

The following table provides recommended segment sizes for DISCSYNC.

TABLE 8-1 DISCSYNC SEGMENT SIZES

FORMATTED MSM DISK	SEGMENT SIZES (kb)				
67Mb	16	20	40	80	-
256Mb	16	38	76	152	304



## 8.5 MESSAGES

When the utility detects an error situation, it displays an explanatory message on the log device and abandons the run. If the error is of an operational nature, such as the mistyping of a device name, the utility can be rerun. If the error results from a problem with one of the disks that prevents them from being resynchronized as they are, corrective action should be taken before rerunning the utility. If the error is on the secondary disk, the primary disk will remain available. If the error is on the primary disk, neither disk will be available.

In the case of I/O errors, the operator is notified that an error occurred and that the mirrored pair is no longer synchronized. The operator may choose to resynchronize the failing disk or to synchronize a different disk. If a different disk is chosen, it must have the same name, but may be on a different device.

The following messages are produced by the DISCSYNC Utility on the log device or file.

### Messages:

ASSIGNMENT ERROR nn ON xx.xx

indicates that the program attempted to assign logical units to their respective file descriptors. The returned SVC7 status is nn. The device or filename is xx.xx.

This message is followed by a subsidiary message which can be any one of the following:

ILLEGAL FUNCTION  
ILLEGAL LU  
VOLUME NOT MOUNTED  
FILE DOES NOT EXIST  
INSUFFICIENT SPACE  
MISMATCH ON PROTECTION KEYS  
ACCESS PRIVILEGES CANNOT BE GRANTED  
INSUFFICIENT OS/32 SYSTEM SPACE  
DEVICE IS OFFLINE  
VOLUME NOT A DIRECT ACCESS DEVICE  
INCORRECT FILE DESCRIPTOR FORMAT  
TRAP GENERATING DEVICE CANNOT BE CONNECTED  
FILE SHOULD BE A PRIVATE FILE

DISKS ARE ALREADY SYNCHRONIZED

indicates both disks are already operating as a mirrored pair.

DRIVE drive-name DOES NOT HOLD A PRIMARY MIRRORED DISK

indicates that the utility has determined that the drive that is supposed to be holding the primary disk is actually holding the secondary disk. Primary and secondary disks may be changed with the SWOP command. (See the OS/32 Operator Reference Manual for instructions about the SWOP command.)

I/O ERROR ON SECONDARY DISK

indicates that an I/O error has occurred on the secondary mirrored disk.

INSUFFICIENT MEMORY ALLOCATION SUPPLIED

indicates there is insufficient memory to set up the I/O buffers. Reload with a segment size increment that is at least the size of one track on the disk.

INTERNAL ERROR DSnn  
TASK PAUSED

indicates that an internal error has been detected by the utility. DSnn identifies the crash point within the particular code module. Dump the memory areas, pure and impure. Then use CONTINUE or CANCEL. If CONTINUE is used, the utility stops with end of task code 254. Submit a software change request (SCR) to Perkin-Elmer with the code DSnn.

I/O ERROR nn ON dddd

indicates that an I/O error has occurred on device dddd with a status code of nn. Possible qualifying messages are:

ILLEGAL OR UNASSIGNED LU  
PARITY ERROR  
UNRECOVERABLE ERROR  
END OF FILE  
END OF MEDIUM  
DEVICE UNAVAILABLE  
ILLEGAL FUNCTION

| If the I/O error is on the log device, then this message  
| is followed by:

| MESSAGE DIVERTED TO SYSTEM LOG DEVICE

| and followed in turn by the original message being  
| logged.

| MANDATORY DRIVE START PARAMETER MISSING

| indicates the device name of the drive holding the  
| primary disk has not been specified.

| P-E OS/32 DISK SYNCHRONIZATION 03-978 Rxx-yy

| indicates that the DISCSYNC Utility is operational. The  
| program's revision level is xx; yy is the update level  
| within the revision. This is the identification message  
| produced at the start of the utility.

| SYNCHRONIZATION SUCCESSFULLY COMPLETED

| indicates successful synchronization without  
| verification, and that both disks are fully operational  
| as a mirrored pair.

| SYCHRONIZATION SUCCESSFULLY COMPLETED AND VERIFIED

| indicates successful synchronization and verification;  
| both disks are fully operational as a mirrored pair.

| SYNTAX ERROR IN START PARAMETERS

| indicates the parameters specified in the start command  
| do not correspond to the valid options available.

| VERIFICATION ERROR

| indicates a verification error has been detected.

APPENDIX A  
CONTENTS OF MAGNETIC TAPE PRODUCED  
BY A STAND-ALONE DUMP

The following pages are a portion of a sample of a stand-alone dump formatted by the OS/32 Dump Print Utility effective for release R06.2 and higher.

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	000005BC	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	0001D0AC	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

## 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				RF
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				RWF
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	4C0B		RWE
FD390	MTM CAL32	TSK	8B800	95400	9C00	1	0	PURE	49	5	00	FF007740	53B8		RF
FCC88	MTM TEXT3	TSK	95400	9B300	5F00	1	0	PURE	49	D	00	FF007740	4AB7		RF
CE140	LAURIE		9B300	A0500	5200	0	0	IMPURE	48		00	FF007740	4748		RWE
**FREE**			A0500	B5000	14B00										

## CONTENT OF MAGTAPE PRODUCED BY A STAND-ALONE DUMP

12:57:23 01/07/83

PAGE 4

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	00009378	0000B2E4
003	TNRDISP	000D9544	00001000	000072F0	00043E6A
003	SVC1	0005345C	0005345C	000072F0	00043E00
003	TNRDISP	000D9544	00001000	000072F0	00043E00
003	SVC9	00000000	000423A4	000072F4	00042D40
003	TNUCHN	C8015E30	00001000	000072F0	00042D40
000	TNRDISP	000D6954	00001000	000077F0	000C4040
000	SVC2	000C0650	0003AC50	000077F2	000C04EA
000	TNRSIN	000070F0	0003AC50	000077F2	000C04FA
000	TNRDISP	000D6954	00001000	000077F0	000C04FA
000	SVC2	00000534	00061034	000077F2	000C0514
000	TNRSIN	000070F0	00061034	000077F2	000C0514
000	SQS	0001C6AC	00000000	000D6788	0000B2F4
000	TNREHW	00000006	00000400	000D6788	8001CF60
000	TNCHN	00000006	00000400	000D6788	8001CF60
003	SQS	0001C6AC	00000000	0000B2E4	0000B2E4
003	TNRDISP	000D9544	00001000	000072F0	00042CE2
003	SVC9	00000000	000423A4	000072F4	00042D40
003	TNUCHN	C8015E30	00001000	000072F0	00042D40
000	TNRD.STS	000D69A4	00000000	00007260	00020788
000	TNRDISP	000D6954	00001000	000077F0	000C0514
000	SVC2	0000C50C	0106100C	000077F2	000C3526
000	TNRSIN	000G70F0	0006100C	000077F2	000C3526
000	TNRDISP	000D6954	00001000	000077F8	000C3526
000	SVC1	00000884	00061384	000077F0	000C4040
000	TNRSAIN	000FD868	000D687C	000D4D10	000FD820
000	EVQCON	80025BF2	000D687C	000D4D10	000D4DC4
000	SVC1	000D4E14	000D4E14	00007260	000263EC
000	TNUCHN	000D687C	00008000	000D4080	000D6788
009	SQS	0001C6AC	00000000	000D6788	000AF74
009	TNRDISP	000D5B6C	00001000	000077F2	0005651E
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	TNRDISP	000D5B6C	00001000	000077F0	00056182
009	SVC1	00000174	0006BC74	000077F0	00051206
009	TNRSAIN	000FCDB0	000D5A94	000D11F0	000FCD68
009	EVQCON	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353636	000D17E8	000D12A4
009	TNRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	TNREHW	000000FF	00008000	000000EC	00004080
009	TNCHN	000000FF	00008000	000000EC	00004080
000	TNRD.STS	000FD868	00000000	00007260	000263EC
000	SVC1	000D4E7C	000D4E7C	00007260	00026894
000	TNUCHN	000D687C	00008000	000D4080	00000000
009	SQS	0001C6AC	00000000	000D6788	000AF74
009	TNRDISP	000D5B6C	00001000	000077F0	00055164
009	SVC1	00000174	0006BC74	000077F0	00051206
009	TNRSAIN	000FCDB0	000D5A94	000D11F0	000FCD68
009	EVQCON	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353637	000D1838	000D12A4
009	TNRDISP	000D5B6C	00001000	000077F0	00051206
009	SVC1	00000174	0006BC74	000077F0	00051206
009	TNRSAIN	000FCDB0	000D5A94	000D11F0	000FCD68
009	EVQCON	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353638	000D1888	000D12A4
009	TNRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	0000B234
009	TNREHW	00000006	00000400	000D59A0	8001CF60
009	TNCHN	00000006	00000400	000D59A0	8001CF60
005	TNRDISP	000D77EC	00001000	000077F0	00000C46
005	SVC2	00003594	0005D194	000077F9	00000D3A
005	TNRSOUT	000D77FC	00001000	000077F9	00000D3A
005	SVC1	000044F4	0005F0B4	000077F0	00001106
005	TNRSAIN	000FD9C8	000D7714	000FCEA8	000FD8C0
005	EVQCON	80025BF2	000D7714	000FCEA8	000FC5FC
005	EVDIS	00025F7C	00000020	000FD2F3	000FC5FC
005	TNRDISP	000D77FC	00001000	000077F0	00001106
005	SVC1	000044B4	0005E0B4	000077F0	0000121A
005	SQS	0001C6AC	00000000	000D7620	0000R234
005	TNRDISP	000D77EC	00001000	000077F0	0000121A
005	SVC9	000FD6F8	0005D408	000077F1	00000C96
005	TNUCHN	DE001600	00001000	000077F0	00000C96
009	TNRDISP	000D5B6C	00001000	000077F2	0005654E
009	SVC1	00000174	0006BC74	000077F0	00051206
009	TNRSAIN	000FCDB0	000D5A94	000D11F0	000FCD68
009	EVQCON	80025BF2	000D5A94	000D11F0	000D12A4
009	SQS	0001C6AC	00000000	000D59A0	0000B234
009	TNRD.STS	000FCDB0	00000000	00007262	0002B02A
009	EVDIS	00025F7C	31353639	000D18D8	000D12A4
009	TNRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	TNRDISP	000D5B6C	00001000	000077F0	00051944
009	SVC1	00000174	0006BC74	000077F0	00051206
009	TNRSAIN	000FCDB0	000D5A94	000D11F0	000FCD68
009	EVQCON	80025BF2	000D5A94	000D11F0	000D12A4
009	EVDIS	00025F7C	31353730	000D1928	000D12A4
009	TNRDISP	000D5B6C	00001000	000077F0	00051206
009	SQS	0001C6AC	00000000	000D59A0	000AF74
009	TNREHW	000000FF	00008000	000000EC	00004080
000	TNCHN	000000FF	00008000	000000FC	00004080
000	TNRD.STS	000FD868	00000000	00007260	00026894
000	EVDIS	000D4E14	20202020	000D4FE8	000D4DC4
000	TNRDISP	000D6954	00001000	000077F0	000C4040
000	SVC7	000FD670	00061058	000077F2	000C3FF4
000	TNRSIN	000FD670	00061058	000077F2	000C3FF4
009	TNUCHN	000D69A4	00000000	000D6788	000C3FF4
009	TNREHW	000D8658	00000400	000FD330	0002B944
009	TNCHN	000D8658	00000400	000FD330	0002B944
007	IIN	430080R6	430080BE	00007002	000080B6

```

000000: FFFFFFFF 0001B186 0000FFFF 00000000 00000000 000D86BC 00000000 00000000 * .....1.....<..... *
000020: 88018801 88018801 00000002 FFFF0000 00000040 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-...^/..-..-/V... *
0000C0: 00000000 00000000 00007000 0001AA6C 3018302C 0501302C 302C302C 302C302C * .....P...*LO.0,..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 * :00,0,0,0,0,0,0,0,0,0,0,0,0,0,0, *
000120: 302C302C 302C302C 302C302C 302C302C 302C302C 302C302C 05A105BD 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 0E0,0,0,0,0,0,0,0,0,0, *
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&O.....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 * .....E.O...I@.... *
000420: 00000000 00000058 00000000 00000000 00000000 26103000 00006E50 00000000 * .....X.....E.O...NP.... *
000440: 00000000 00000000 00000000 00000000 00000000 26103000 000072E0 00000000 * .....E.O...R..... *
000460: 00000000 00000177 00000000 00000000 00000000 26103000 00007740 00000000 * .....W.....E.O...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 * .....E0..... *
0004E0: 00000000 00000000 00000000 00000000 00000000 26B00000 00008660 00000000 * .....E0..... *
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,....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..E\.... *
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 00055FF3 FFB41A36 00000000 0000C55A 124E0500 00009790 FF0C0001 * .....[S.4.6.....EZ.N..... *
000700: 0005253A 004E0000 0000999D 0000F8A4 302C0000 00009790 0000C5DC FF86233A * ..%:..N.....XS0,....E\..#:. *
000720: FF01FF89 000556F3 000020FC 00000000 0000C55A 124E0500 000099B0 FF0C0001 * .....VS.....EZ.N.....0.... *
000740: 0005253A 00490000 00009BBD 0000F8A4 302C0000 000099B0 0000C5DC FF061444 * ..%:..I.....=..XS0,....0..E\..D *
000760: FF01FF89 000558F3 FF9E1A36 0000FFFF 0000C55A 124E0500 00009BD0 FF0C0001 * .....XS...6.....EZ.N.....P.... *
000780: 0005253A 00AC0000 00009DD0 0000F8A4 302C0000 00009BD0 0000C5DC FF86233A * ..%:.....]..XS0,....P..E\..#:. *
0007A0: FF01FFB1 00054483 FFA20FC 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 0000007F 00000000 00000000 00000000 * ..... *
000820: 00000000 FF060001 0005253A 00200001 00000000 0002EBEC 394A0001 0000A010 * .....%:.....KL9J..... *
000840: 00052537 38CC0000 00000000 10040000 00040000 000C007F 00000000 0000A178 * ..%78L.....!X *
000860: 00000000 00000000 00000000 00000000 FF000001 00000000 00000001 00000000 * ..... *

```

```

000880: 0002EBEC 30BE0C00 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 *** *
0009A0: 00000000 00000000 0000A440 00000000 00000000 00000000 00000000 00000000 * .....Sa..... *
0009C0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
0009E0: 00000000 00000000 00000000 0000A440 00000000 00000000 00000000 00000000 * .....Sa..... *
000A00: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000A20: 00000000 00000000 00000000 00000000 00000000 00000000 0000A580 00000000 * .....X..... *
000A40: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000A60: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000A580 * .....X..... *
000A80: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000AA0- 000ABF ***SAME AS ABOVE *** *
000AC0: 00000000 00000000 0000A6C0 00000000 00000000 00000000 00000000 00000000 * .....&a..... *
000AE0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000B00: 00000000 00000000 00000000 0000A6C0 00000000 00000000 00000000 00000000 * .....&a..... *
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 *** *
000BE0: 00000000 00000000 0000A940 00000000 00000000 00000000 00000000 00000000 * .....)a..... *
000C00: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000C20: 00000000 00000000 00000000 0000A940 00000000 00000000 00000000 00000000 * .....)a..... *
000C40: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000C60: 00000000 00000000 00000000 00000000 00000000 00000000 0000A800 00000000 * ..... *
000C80: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000CA0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000A800 * ..... *
000CC0: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000CE0- 000CFF ***SAME AS ABOVE *** *
000D00: 00000000 00000000 0000AB00 00000000 00000000 00000000 00000000 00000000 * .....+a..... *
000D20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000D40: 00000000 00000000 00000000 0000AB00 00000000 00000000 00000000 00000000 * .....+a..... *
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...S0 *
000DE0: 00000007 00002002 00024756 000D7060 30302C30 000FD330 430080B6 430080BE * .....GV..P 00,0..S0C..6C..> *
000E00: 00007002 000080B6 00000000 00000000 00000000 00000000 00000000 00000000 * .....P...6..... *
000E20: 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 * ..... *
000E40- 000EDF ***SAME AS ABOVE *** *
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 0001D0AC 000D6788 00000000 00000000 00000000 0000B700 * .....R.../T..P,..G.....7. *
000F40: 00004080 00000000 FFFF50 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 *** *
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..#3J0.<@5., *

```

001020:	C3700020	2338C860	10604064	0014C370	00802134	24687564	00009477	9E271800	* CP.#8H . @D..CP..I4SHUD...W.'.. *
001040:	24697464	00162138	C860FF8C	40640000	C8600058	9E26C860	10604064	00141800	* SITD..I8H ..@D..H .X.EH . @D.... *
001060:	212F4210	80929D23	4270808C	9B25E354	00004220	8022C550	000D233F	18002468	* 1/B....#BP...XT..B ."EP..#?.SH *
001080:	76640000	E6608006	40640014	1800245B	76540016	42308076	24305854	0018D235	* VD..F ..@D...S[VT..BO.VSOXT..R5 *
0010A0:	00724835	002C2339	25314035	002C5855	00046450	4000B700	C85000C4	9E251800	* .RH5.,#9X1a5.,XU..DPa.7.HP.D.%.. *
0010C0:	C3300020	42308028	C5300008	2335C830	00844300	FFC45854	00182439	74340016	* CO. BO.(EO..#5H0..C..DXT..S9T4.. *
0010E0:	23384865	002E2761	208D4065	002E1800	C8300082	4300FFA2	C3300001	2135C530	* #8HE..'A .@E....HO...CO...I5FO *
001100:	00044230	FFBAC830	00A04300	FF8C2450	58640018	4556002C	4330FF9C	58560038	* ..BO.:HO. C...\$PXD..EV.,CO..XV.8 *
001120:	40540002	25524054	000A0755	4B540002	CA50003C	4056002C	C8500FF0	40540014	* @T..Xr@T...UKT..JP.<@V.,HP.PaT.. *
001140:	C850FF82	40540000	C8500064	9E251800	245B7554	00162305	C430007F	E3340000	* HP..@T..HP.D.%..S[UT..#.DO..C4.. *
001160:	25524054	000A4300	FEDE5854	00184864	00025565	00382334	27614064	00021800	* Xr@T..C..^XT..HD..UE.8#4'AaD.... *
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.,SIT5..BO.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..VFE..S6..EO..#9AP..FE *
001200:	009D2470	24334300	81089B26	E670803E	D3350070	C3300001	2333E670	80824074	* ..SPS3C...@FP..>S..CO..#3FP..@T *
001220:	00142435	74350100	42308FE4	9D23C330	00014230	8FDA2466	74650100	2335C330	* ..S5T5..BO.D.#CO..BO.ZSfTE..#5CO *
001240:	00024230	888AC330	00080337	18009B26	08334230	8852C460	007FC560	00202383	* ..BO..CO...7...E.3BO.RD .E .# . *
001260:	E7640010	417085AE	41708402	E3640000	46280558	18005854	00182432	74350100	* GD..AP..AP..CD..B .X..XT..S2T5.. *
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	* .E.3BO..D ..AP.FSVTU.V#5E..CO.L *
0012C0:	9A26E364	00004260	88221800	07335854	0018D235	00FA2461	76650100	586500EC	* .@CD..B ."...3XT..R5.ZSAVE..XE.( *
0012E0:	D3750104	D2760000	487500F4	26710B67	4300801E	58540018	24715175	00A45865	* SU..RV..HU.TEQ.GC...XT..SQQU.SYE *
001300:	00ECD375	0102D276	00002761	487500F4	0B677345	009A4034	00164074	000A5064	* .LSU..RV..'AHU.T.GSE..@4..@T..PD *
001320:	000C7345	0098E660	800C4064	00147345	009A4300	80D69B26	C3300001	42308ED0	* ..SE..F ..@D..HE..C..V.ECO..BO.P *
001340:	C3300002	42308788	18009926	0833203C	C460007F	58540018	C560007F	43308044	* CO..BO.....E.3 <D ..XT..E .CO.D *
001360:	C5600020	238D0876	11715A74	00104877	00001171	F5700000	12764230	80264875	* E .#..V.QZT..HW...QUP...VBO.EHU *
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..SfTE..#4 *
0013C0:	9D234220	870A2463	746500F6	233E9D23	232C4170	849E2302	230F0766	40650072	* .#B ..\$CTE.V#>..#.#.AP..#.#.F@E.R *
0013E0:	50650038	430087BC	07664B64	00021062	26644065	002C9B26	C460007F	C5600014	* PE.8C..<.FKD...B&D@E..@D .E.F .. *
001400:	43308194	C5600013	4330818C	2621DE20	9032E660	80184064	00140834	73450098	* CO..E ..CO..E1^ .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..SNTD.. *
001460:	23394864	00025A64	0004D336	0000E734	0010E364	00004260	80449A26	18004220	* #9HD..ZD..S6..G4..CD..B .D.E..B *
001480:	803E2335	08334230	861E1800	58540018	24617665	00AC4330	86524875	00F04075	* .>#5.3BO...XT..SAVE..CO.RHU.PaU *
0014A0:	00ACD375	00F2D275	00AE4170	83E62478	76740000	41704000	F1244300	862E9A26	* .,SU.RRU..AP.FSxVT..AP@.QSC....E *
0014C0:	48640016	42308022	58540018	D3750070	C3700001	2137247E	76740000	4874000A	* HD..BO."XT..SU.PCP..I7S.VT..HT.. *
0014E0:	23242478	76740000	1800C860	FF0C4064	00004170	4000F124	24624564	00164330	* #SSXVT...H ..@D..AP@.QSSBED..CO *
001500:	87EC5854	00182721	E6604000	302C4064	00140834	73450098	24634563	00164330	* .LXT..'IF @.0,@D...4SE..SSEC..CO *
001520:	FCE84300	FD629B26	2621C460	007F5854	0018C560	000E4330	8844C560	00144330	* .HC..B.E&EID ..XT..E .CO.DE ..CO *
001540:	80A2C560	00134330	809AC560	00184330	85680833	42308550	C560000F	21352461	* ."E ..CO..E ..CO.H.3BO.PE ..15SA *
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..@U.. *
0015A0:	73450098	41704000	F1249B26	08334230	84F6C460	007FC560	00184330	84FCC560	* SE..AP@.QS.E.3BO.VD .E .CO..E *
0015C0:	00122335	C5600011	4230849A	E660FF56	40640014	58540018	4170803E	7345009A	* ..#5E ..BO..F .V@D..XT..AP.>SE.. *
0015E0:	4300FE28	DE208E5E	C8607FFF	E67086BE	58540018	4835002C	03374065	002C0307	* C..(^ .^H ..FP.>XT..H5.,,7aE.,, *
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	88B22133	* ..FKD..B&DC..@XT..S2T5..FO.2I3 *
001640:	E630FC0A	40340014	7345009A	E6304000	302C4034	0014DE20	8DEC7345	00980307	* FO..@4..SE..FO@.0,@4..^ .LSE.... *
001660:	58540018	585500E8	D3550001	23022551	48340016	211BC560	007F2338	C5600020	* XT..XU.HSU..#.%QH4..I.E ..#8E. *
001680:	2389C560	00094330	80405854	00182531	03070855	21172531	C3507F00	21359A25	* #.E ..CO.@XT..%1...!1.%1CP..15.% *
0016A0:	23039A26	24315854	00185135	00845835	00B45535	00B02183	503500B0	453500A2	* #..ES1XT..Q5.4X5.4U5.01.P5.0E5.. *
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.FU0..J#5FO.F@4..@T..^ .H44.. *
001700:	42108076	587500E8	D3670001	11684065	00C4C330	7F004330	801F5575	00E0213D	* B..VXU.HSG...H@E.DCO..CO..U..I = *
001720:	D36740FF	FFFFC660	20004065	00C4DA27	40FFF3FD	4170808A	583500B4	27315035	* SG@...F .@E.DZ'@...AP..X5.4'1P5 *

001740:	00B4DA25	00C44535	00A22387	74350162	21344170	806C220F	D33500C5	Q8332335	* .4Z%.DEF.##.T5.R14AP.L".S5.F.3#5 *
001760:	4170805E	DA2500C5	487500C6	40740014	7374000C	24692431	03075835	00B44535	* AP.^Z%.EHU.F@T..ST..SIS1..X5.4F5 *
001780:	00A22383	75350162	C8600020	9A265835	00B42631	503500B4	553500B0	21835035	* .#.U5.BH . .EX5.4E1P5.4U5.01.P5 *
0017A0:	00B04535	00A24380	FFBE7635	01625875	00D87437	00004230	FFAE4170	80044300	* .0E5."C.>V5.BXU.XT7..BO..AP..C. *
0017C0:	FFC64074	000A7345	009A7364	00144065	00BC4170	4000F124	C5300008	22355854	* .F@T..SE..SD..@E.<AP@.QSEO..%5YT *
0017E0:	00187365	00BC4064	00147345	00982721	7374000A	03074074	000A4170	4000F124	* ..SE.<@D..SE..'IST....@T..AP@.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...7@T..RD..SE..E!^ ..SJ.E *
001840:	41704000	F124C530	00082235	DE208BF6	E6604000	302C4064	00145854	00182461	* AP@.QSEO..%5^ .VF @.0.PD..XT..FA *
001860:	516500A4	73450098	2721D364	00087374	000A0307	D3650070	C4600021	C5600021	* QE.SSE..'ISD..ST....@D..@T..IE .! *
001880:	42370002	5835006C	24617463	00144337	00020307	48640000	40640020	40740022	* B7..X5.LSATC..C7....HD..@D. @T." *
0018A0:	58540018	D3650103	D37500F8	C3700040	2133D365	01022470	24337635	00AC4230	* XT..SE..SU.XCP..@13SE..SPS3V5..BO *
0018C0:	80365875	00A4D365	01042439	76350100	42308024	D3650102	587500A4	26715575	* .6XU.SSE..S9V5..BO.SSE..XU.S@QUU *
0018E0:	0090218B	58350090	23382470	24307435	001C2133	D3650103	507500A4	587500EC	* ..I.X5..#8SPS0T5..I3SE..PU.SXU.L *
001900:	D2670000	483500F4	26314034	000A0B73	5074000C	C870FF8C	40740000	D37500F9	* RG..H5.TE1@4....SPT..HP..@T..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..AP@.QS *
001960:	24687564	0000220B	24679A26	DE208AD6	41704000	F1244210	812E5854	00182461	* SHUD..".SG.E^ .VAP@.QSB...XT..FA *
001980:	D26500FA	43008C28	24640B63	10624A65	002C4170	FC5A9D23	2185E364	00002168	* RE.ZC..(SD.C.BJE..AP.Z.#I.CD..IH *
0019A0:	9A264170	4000F124	421080FC	23029A26	58540018	D36500AE	2761D265	00AE4220	* .EAP@.QSB...#.EXT..SE..TARE..B *
0019C0:	FF12E660	FAB84064	00145865	00ECD375	0104D276	00004875	00F42671	0B675064	* ..F Z@D..XE.LSU.RV..HU.TEQ.GPD *
0019E0:	000C4074	000A4864	0020C660	00804064	00004874	00220307	E630F93A	40340014	* ..@T..HD. F ..@D..HT..".FOY:@4.. *
001A00:	DE208A40	C6200001	9D23C330	00084230	80204834	00025A34	0004D363	0000E364	* ^ .@F ...#CO..BO. H4..Z4..SC..CD *
001A20:	00004170	FC482316	586500EC	D3660001	9A264170	FD8C4834	00024534	00084230	* ..AP.H#.XE.LSF...EAP..H4..E4..BO *
001A40:	FFD42479	76750100	42308218	4300F838	9B26C330	00014230	87B6C330	00202334	* .TSYVU..BO..C.X8.&CO..BO.6CO. #4 *
001A60:	08664330	802E2531	C460007F	C560000E	4330830A	C560000F	43308376	C560001B	* .FCO.%X1D ..E ..CO..E ..CO.VE .. *
001A80:	233AC560	00112184	C5600015	21832467	9A261800	58540018	24717475	00F62036	* #:E ..I.E ..I.SG.E..XT..SQTU.V 6 *
001AA0:	247A7575	01001800	C3300001	42308760	C3300020	233E0866	213C5854	00182461	* SZUU....CO..BO. CO. #>.F1<XT..SA *
001AC0:	745500F6	4230FA20	C8608200	43008046	C8608A00	C42003FE	9D274220	8038C330	* TE.VBOZ H ..C..FH ..D ...'B .8CO *
001AE0:	00084330	FFD41800	E3640000	07665854	0018D375	0070C370	0001213E	487500AC	* ..CO.T..CD...FXT..SU.PCP..I>HU.. *
001B00:	C470C000	C570C000	23372477	75750100	2471D275	00AE5854	00184065	00720866	* DP@.EP@.#7SHUU..SQRU..XT..@E.R.F *
001B20:	233DD235	00730733	D23500FA	24307635	01002333	24377635	01000866	42308064	* #=R5.S.3R5.ZSOV5..#4S7V5...FRO.D *
001B40:	24667565	01004170	FD2A2303	43008054	4170822E	24607465	00F64330	80462561	* SFUE..AP.*#.C.TAP..S TE.VCO.FXA *
001B60:	9A264170	4000F124	58540018	24667665	01002463	75650100	24607665	01002463	* .EAP@.QSXT..SFVE..SCUE..S VE..SC *
001B80:	4170FA6C	C42003FE	E6604000	302C4062	420000D0	40624200	00D2DE20	88A5DE20	* APZLD ..F @.0.@BB..P@BB..R^ .%X^ *
001BA0:	88A31800	C5608400	4330FFC6	4835002C	213ED335	0070C330	00404330	801F5835	* .#.E ..CO.FH5.,I>S5.PCO.@CO..X5 *
001BC0:	000CF330	000C0000	213A2306	58650004	64604000	B7002531	4035002C	C42003FE	* ..SO...I:#.XE..D @.7.%1@5.,D .. *
001BE0:	E6304000	302C4032	420000D2	E630FE60	73450098	40340014	26414042	420000D0	* FO@.0.@2B..RFO. SE..@4..E@BB..P *
001C00:	DE20883E	9B262621	DE20883A	18005854	00184865	003A4874	00024565	002E2134	* ^ .>.E&E!^ ..:XT..HE..HT..EE..I4 *
001C20:	05674330	FE684065	002E4064	00024835	00F84034	00160733	503500B4	503500B0	* .GC0.H@E..@D..HT.X@4...3P5.4P5.0 *
001C40:	24327635	01002431	4300F6A8	58540018	48740002	4975002E	23162479	75750100	* S2V5..S1C.V(XT..HT..IU..#.SYUU.. *
001C60:	430083FE	246DE364	00004300	F65E2470	23022571	58540018	48640002	4565003A	* C...SHCD..C.V^SP#.XQXT..H..FE.. *
001C80:	4330FE0A	4965002E	21174065	002E6175	002E5175	00B02761	40640002	5A640004	* CO..IE..I.@E..AU..QU.0'@D..ZD.. *
001CA0:	0877D376	00002115	08674170	F9B21800	C8300020	D2360000	0867C850	7F004170	* .WSV..I..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@2C.V.XT..SE..HD..HU..V *
001D00:	4330F584	40640008	40740002	07775075	00B44300	FCE24170	FADC4170	F9124300	* COU.@D..@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.U<HD..XT..IE..C.2ZD.. *
001D60:	D3660000	4170F906	F3640000	48640002	4965002E	21142462	76650100	1800E670	* SF..APY.CD..HD..IE..I.SBVE...FP *
001D80:	FFFA5854	0018D365	00FBD265	009D2468	583500E8	D2630001	D2630003	D36500F8	* .ZXT..SE..RE..SHX5.HRC..RC..SE.X *
001DA0:	C4600004	0A365035	00E00766	D26500F9	D33500F8	C4300001	21382436	D4350105	* D ...6P5..FRE.YSS.XD0..!8S6T5.. *
001DC0:	2184D335	01052631	0B634065	00F42461	76650100	03074170	FA1C9A26	4300F4A8	* I.S5..E1.C@E.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..FE.. *
001E20:	4330FC6A	4170F9CE	406500C0	48740016	4210FC5A	58540018	48640002	4565003A	* CO.JAPYN@E.@HT..R..ZXT..HD..FE.. *
001E40:	4330FC4A	4965002E	21134065	002EE660	F4E44064	00147345	009AE660	80084064	* CO.JIE..I.@E..F TD@D..SE..F ..@D *

001E60:	0014DE20	85DE0833	4230FC3C	58540018	73450098	48640002	4565003A	4330F682	* ..^..^..3B0.<XT..SE..HD..EE.:COV. *
001E80:	27614064	00025A64	0004D366	00004170	F7CE4875	00C02114	0567233D	23047460	* *AaD..ZD..SF..APWNHU.a!..G#=#.T *
001EA0:	85902339	08332316	586500EC	D3660001	9A261800	58540018	7345009A	07664064	* ..#9.3#.XE.LSF...E..XT..SE...F0D *
001EC0:	00160833	4210F63A	4300F61E	48640016	4210FBBA	58540018	48640002	4965002E	* ...3B.V:C.V.HD..B.:XT..HD..IF.. *
001EE0:	4310FBAA	24627765	01004170	F7421800	9B260833	4230FB80	C460007F	C5600020	* C..*SBWE..APWB...E.3B0.OD ..E. *
001F00:	2383E764	00105854	00182430	50350110	40350114	50350124	243D4035	0128D265	* #.GD..XT..\$OP5..05..P5.SS=05.(RE *
001F20:	4300012A	48340002	26314034	00082431	6135002E	E630F412	40340014	7335009A	* C..*H4..E104..S1A5..FOT.04..S5.. *
001F40:	E670800A	40730014	DE2084F8	23090833	4230FB54	27215854	00187345	00984865	* FP..0S..^..X#..3B0.T'IXT..SE..HE *
001F60:	01262761	2312246D	40650126	D3654600	012A4874	00025A74	0004D337	00004875	* .E'A#.SM0E.ESEF..*HT..ZT..S7..HU *
001F80:	01282771	2312247D	40750128	D2354700	012A4170	F6D8E364	00004260	F32E4874	* .('Q#.S.0U.(R5G..*APVXCD..B S.HT *
001FA0:	00024575	002E2335	08334210	FEFA1800	7345009A	E670800A	40740014	08332218	* ..EU..#5.3B..Z..SE..FP..0T...3". *
001FC0:	23070833	4230FAE0	27215854	00187345	00984864	00024564	0008233D	27614064	* #..3B0Z 'IXT..SE..HD..ED..#='A0D *
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..05..H4..704..AU..C .. *
002040:	4875002E	27714330	FEF24835	01280A37	40350128	273E4210	FEE24035	01284300	* HU..'QCO.RH5.(.705.(>B..005. *
002060:	FEDA4874	00025854	00184865	002E0976	4310FA1A	24327635	01004834	00162319	* .ZHT..XT..HF..VC.Z.S2V5..H4..#.. *
002080:	40640002	24797675	01004230	FBD61800	40640008	4300F960	48740016	4210F9EE	* 0D..SYVU..BO.V..0D..C.Y HT..B.YN *
0020A0:	58540018	24727475	01004230	F9E04875	01104310	F9D84170	F75CE670	F2784074	* XT..SRTU..BOY HU..C.YXAPWFFPR0T *
0020C0:	0014DE20	837E2571	23034875	00C02671	407500C0	D3654700	0111C560	000D4330	* ..^..%Q#.HU.0EQ0U.0SEG...E ..CO *
0020E0:	F1A6E364	0000216D	4170F582	23165865	00ECD366	00019A26	4170F6C6	4300FFCA	* Q0CD..IMAPU.#.XE.LSF!..0APVFC..J *
002100:	25716174	00022467	9A264170	F6B44300	F1764864	00164210	F9745854	00184864	* %QAT..SG.0APV4C.QVHD..B.YXT..HD *
002120:	00024875	002E0967	4310F962	4575003A	4330F95A	27714075	002E0967	21142432	* ..HU...GC.YBEU.:COYZ'Q0U...G!..S2 *
002140:	76350100	E630F1EE	40340014	48340002	40340008	7345009A	F6308010	40340014	* V5..FOQN04..H4..04..SE..FO..04.. *
002160:	DE2082F0	24304035	01102306	08334230	F9365854	00187345	00984874	00024575	* ^..S005..#..3B0Y6XT..SE..HT..EU *
002180:	002E4330	001E2671	00740002	5A740004	D3670000	D26740FF	FFFF4170	F4D04210	* ..CO..EQ0T..ZT..SG..RG0..APT0B. *
0021A0:	F0061800	5A740004	C8600020	D2670000	25315875	00B05B75	00B44320	FD250710	* ...ZT..H ..RG..X1XU.0IU.4C.RPU *
0021C0:	00B8C860	00204170	F4A44170	F5F45875	00B45575	00B0208A	C0860020	4170F480	* .8H ..APTSAPUTXU.4UU.0.H ..APT. *
0021E0:	25715175	00B82324	4170F5D6	220A5875	00B45075	00B00733	4300FDB4	58540018	* %QQU.8#SAPUV'.XU.4PU.0.3C..4XT.. *
002200:	24727475	00F64330	F06CC830	C0002303	C8304000	73640014	4064000C	E6608040	* SRTU.VCOPLHO0.#.HO0.SD..0D..F.0 *
002220:	40640014	73700062	486700A6	4064000E	9B265854	00182460	76650100	24657565	* 0D..SP.BHG.00D..0XT..S VE..SEUE *
002240:	01004865	009CC460	3FFF0663	4065009C	DE2081EE	DE2081EE	DE25009C	4300FB1E	* ..HE..D ?..CaE..^..N^..N^..C... *
002260:	98260873	C8304000	C370000B	4230FFB4	08774230	802AC460	007FC560	000D423C	* .E.SHO0.CP..BO.4.WB0.'D ..E..RO *
002280:	801E7364	000C4064	0014DE20	81B65854	00182465	76650100	C860002A	9A261800	* ..SD..0D..^..6XT..SEVE..H ..E.. *
0022A0:	73700062	486700A6	4564000E	22374064	000E5854	00184865	009CF660	000F0000	* SP.BHG.0ED..*70D..XT..HE..V .... *
0022C0:	C8604000	4065009C	DE25009C	18005854	0018D265	01042471	74750100	233C2477	* K 0.0E..^%...XT..RE..SQTU..#<SW *
0022E0:	75750100	2471D275	00AE2471	767500AC	4300806C	24797575	01004864	00022661	* UU..SQRU..SQVU..C..LSYUU..HD..0A *
002300:	506500A8	43008028	58540018	D2650102	24707475	00AC233D	48640002	26615065	* PE.(C..(XT..RE..SPTU..#='HD..0APE *
002320:	00A8246D	24717475	01004230	FFB09A26	2471D275	00AE4170	F55AE364	00004260	* .(SMSQTU..BO.0.0SQURU..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	* .E.3ST...XT..SPTU..*?SXTU..CO.L *
002380:	48640002	26615065	00A8246D	24717475	01004230	FF489A26	2471D275	00AE4170	* HD..0APE.(SMSQTU..BO.H.0SQURU..AP *
0023A0:	F4F25865	00DC5875	00A44475	00A07476	00004230	FF84220F	58540018	24707475	* TRXE.\XU.\$JU. TV..BO..XT..SPTU *
0023C0:	00AC4330	FF904874	00025B75	00A84074	00204874	00144074	0022C860	0020E334	* ..CO..HT..IU.(0T. HT..0T."H. C4 *
0023E0:	00004260	FF7AC860	00209A26	48740020	58540018	00240575	00A24380	FF605865	* ..B.ZH ..0HT.0Q0T. EU."C.XE *
002400:	00D87476	00004230	FF542571	517500A8	24787674	00004170	4000F124	58540018	* .XTV..BO.T%QU.(SXTV..AP0.QSXT.. *
002420:	48740022	40740014	24787574	00004300	FFB40000	00000000	FFC07FFF	FFE07FFF	* HT."0T..SXUT..C..4.....0... *
002440:	FFE061A1	6383A323	012C000F	00000000	58540018	4865002C	233BD235	00732561	* .AIC.##.,.....XT..HE.,#;R5.S%A *
002460:	4065002C	58650004	64604000	B700E660	4000302C	40624200	00D01800	58540018	* 0E.,XE..D 0.7.F 0.0,0BB..P..XT.. *
002480:	4835009C	4825001A	24744075	002C2641	40434300	00D02741	E670806C	40740014	* H5..H%.ST0U.,0A0CC..P'AFP.L0T.. *
0024A0:	DE308090	DA350039	D835003A	DA350079	D835007A	4865009E	00A24380	43308034	* ^0..Z5.9X5.:Z5.YX5.ZHE..HT..CO.4 *
0024C0:	D8240006	DA640003	24717474	0010233A	9577C470	B7FF9557	DF608054	DE308051	* XS..ZD..SQTU..#:..WDP7..W^..T^0.Q *
0024E0:	18009577	C470B7FF	9557DE60	8044DE30	80411800	DA640003	D3740005	11744674	* ...WDP7..W^..D^0.A..ZD..ST..SFT *
002500:	00069867	4300FFC0	58540018	4525009C	4230FF40	DE20801C	4824000A	21321800	* ..GC.0XT..F%.#..BO.0^..H.S..!2.. *
002520:	4825009E	9D23C330	00804230	FF261800	41304210	48002000	00000000	00000000	* H%..#CO..BO.0..A0B.H. .... *
002540:	58540018	4865002C	433080F6	403503C6	C3300001	423080FC	430080D6	42000000	* XT..HE.,CO.V05.FCO..BO..C.C.VB.. *
002560:	58540018	4835009C	4865001A	487503F4	4075002C	9D670475	03C6C370	007B4230	* XT..H5..HE..HU.T0U.,,0B".FCP..BO *

002580:	80D2E674	00014073	430000D0	E6708060	40740014	DE3080C2	DA3503C9	D83503CA	* .RFT..@SC..PFP. @T..^O.BZ5.IX5.J *
0025A0:	DA3503CD	D83503CE	D37503E2	C3700040	2337D374	0016C370	000C2332	9E672470	* Z5.MX5.NSU.BCP.@#7ST..CP..#2.GSP *
0025C0:	9E27DA25	03DD375	03DF9867	117A4675	03E09827	D37503F2	10761171	48678076	* .Z%.JSU.<.G.ZFU. .SU.B.V.QHG.V *
0025E0:	9577C470	B7FF9557	9F269466	9E361800	58540018	4865002C	43308046	403503D6	* .WDP7..W.E.F.6..YT..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.FXA@E.,YE..D @ *
002640:	B700C860	00C09E26	F6604000	302C4062	420000D0	18001442	34414C00	42000000	* 7.H .@.EF @.O.@BB..P...B4AL.B... *
002660:	41308210	4835009C	DE3081D0	41704000	F12A9577	C470B7FF	9567DE25	00F6DE25	* AO..H5..^O.PAP@.Q.*WDP7..G^X.V^X *
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..I>AP@.QSAP@.Q.E%..I4CO.. *
0026C0:	21321800	D2350073	4825001A	DE2500F8	41708194	4130819C	4835009C	DE30815C	* I2..R5.SH%..^X.XAP..AO..H5..^O.\ *
0026E0:	41704000	F1529577	C470B7FF	95672469	746500E4	23367665	00E4C860	004B9E26	* AP@.QR.WDP7..GSITE.D#6VE.DH .K.C *
002700:	DE2500F6	DE2500FD	DE30812F	4160817E	41704000	F1244170	4000F118	4525009C	* ^X.V^X..^O./A ..AP@.QSAP@.Q.E%.. *
002720:	23322209	4865001A	9D63C330	0011213E	41704000	F1244170	4000F118	4525001A	* #2".HE...CCO..I>AP@.QSAP@.Q.E%.. *
002740:	2134C330	00112132	18004825	001AD235	0073DE25	00F84170	810F4170	80E04170	* I4CO..I2..H%..R5.S^X.XAP..AP. AP *
002760:	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	00F84170	80E04170	* @.QSAP@.Q.CO..I2..R5.S^X.XAP.JAP *
002780:	80D24170	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	00F84170	* .RAP@.QSAP@.Q.CO..I2..R5.S^X.XAP *
0027A0:	80C64170	80984170	4000F124	41704000	F118C330	00112132	1800D235	0073DE25	* .FAP..AP@.QSAP@.Q.CO..I2..R5.S^X *
0027C0:	00F84170	80A24170	80744170	4000F124	41704000	F118C330	00112132	1800D235	* .XAP."AP.TAP@.QSAP@.Q.CO..I2..P5 *
0027E0:	0073DE25	00F84170	807E4170	80504160	809C4170	4000F124	41704000	F118D235	* .S^X.XAP..AP.PA ..AP@.QSAP@.Q.P5 *
002800:	0073C330	00112132	1800DE25	00F84170	80564170	80284170	4000F124	41704000	* .SCO..I2..^X.XAP.VAP.(AP@.QSAP@. *
002820:	F118C330	00132132	1800D235	0073DE25	00F84170	8032E1E2	20C03414	48005854	* Q.CO..I2..R5.S^X.XAP.2AB @4.H.XT *
002840:	0018DE25	00F64865	00EE4065	002CDE25	00F40307	58540018	DE2500F6	24614065	* ..^X.VHE.N@E.,^X.T..XT..^X.VSA@E *
002860:	002CDE25	00F40307	58650004	64604000	00701800	58540018	58650875	00785B75	* .,^X.T..XE..D @.7...XT..S XU.[U *
002880:	00745D65	01BC2679	4075002C	0303D375	0019C570	00440386	9D27C370	00100336	* .T]E.<EY@U.,..SU..EP.D...^C'P...6 *
0028A0:	24777575	00E42471	4075002C	58540018	DE2500F8	24614065	002C4170	4000F124	* SHUU.DSQ@U.,XT..^X.XSA@E.,AP@.Q5 *
0028C0:	41704000	F1181800	58540018	C8607080	DE2500E2	26724075	002CC330	0010213D	* AP@.Q...XT..H P.ME.BER@U.,CO..I1 *
0028E0:	DE2500F6	41704000	F1244170	4000F118	C3300011	21321800	40350072	DE2500F8	* ^X.VAP@.QSAP@.Q.CO..I2..@5.R^X.X *
002900:	4170FF64	00000000	00000000	00000000	58540018	C860001E	4065002C	C5300004	* AP.D.....XT..H .@E.,FO.. *
002920:	43308042	C3300021	4230803A	C3300010	213FC860	00489E26	9D23C330	00102136	* CO.BCO.IBO.:CO..I7H .H.E.#CO..I6 *
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	E670802A	* ...H.HP. #.HP..XT..RU.RC..SFP.. *
002980:	40740014	24787674	00009D23	C33000C1	4230FFD8	C3300002	21349B26	9B261800	* @T..SXVT...#CO.ABO.XCO..I4.E.E.. *
0029A0:	58540018	4875002C	23392571	4070002C	58750004	64704000	B700DE20	FFA61800	* XT..HU.,#9XQ@P.,XU..DP@.7.^ .E.. *
0029C0:	C33000D1	4230FFA4	C3300008	0337C530	00084330	FF961800	4170FFE4	23030833	* CO.QBO.SCO..7EO..CO..CO..AP.D#..3 *
0029E0:	20349B25	9B260855	43308044	24770475	43308040	73774700	2A7A4330	80502774	* 4.% .E.UCO.DSW.UCO.@SWG.*ZCO.P^T *
002A00:	C360003C	42308046	C4500038	94669062	C56000FF	43808036	06560657	58740010	* C <BO.FDP.8.F.BF .C..6.V.WXT.. *
002A20:	D3754700	0000E374	00004220	FF4E1800	08662238	0876C470	003C4330	FFCA4660	* 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	* I:HT..ZT..[U.B@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	9052F354	00004220	FE9E9166	* .....3BO.4.EST...V.V.RC^..B .E.F *
002AE0:	9B250656	E3540000	4220FE90	E670FFAE	40740014	18004170	FEC64300	FFA64170	* .X.VCT..B .FP..@T...AP.FC..EAP *
002B00:	FEBE4300	FFC44170	FEB62303	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...E.CD..B .D..... *
002B40:	58540018	D2350073	C3300008	4330804C	8C700040	9E279D23	D2350073	23844170	* XT..R5.SCO..CO.LHP.@.'#R5.S.#AP *
002B60:	4000F124	41704000	F118D235	0073C330	00084330	8020C530	0008213E	7375002E	* @.QSAP@.Q.R5.SCO..CO. FO..I>SU.. *
002B80:	26714575	00D22384	4075002E	1800C830	0082D235	0073C870	00C09E27	25714075	* @QEU.R#@U...HO..R5.SHP.@.*%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.SXQ@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	430080B6	* HP..C..RH .C..FH .C..>H .C..6 *
002C40:	C860001C	430080AE	C860001B	430080A6	C860001A	4300809E	C8600019	43008096	* H .C...H .C..EH .C...H .C...V *
002C60:	C8600018	4300808E	C8600017	43008086	C8600016	4300807E	C8600015	43008076	* H .C...H .C...H .C...H .C...V *
002C80:	C8600014	4300806E	C8600013	43008066	C8600012	4300805E	C8600011	43008056	* H .C..NH .C..FH .C..^H .C...V *

CONTENT OF MAGTAPE PRODUCED BY A STAND-ALONE DUMP

12:57:23 01/07/83

PAGE 12

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	* \$DC..OSBC..*SAC..\$\$ C...SO#.SN#. *
002CEO:	2468230A	24692308	246A2306	246B2304	246C2302	246D0876	11725A74	001C5877	* SH#.SI#.SJ#.SK#.SL#.SM.V.RZT..YX *
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.@...R *
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 @.VHAO@.04C. *
002E00:	40021BEA	41C081C0	58C90068	4230802E	E6BA002F	41804002	118E42A0	4001A57A	* @..JA@.@XI.HBO..F:./A.@...B @.%Z *
002E20:	08CD08DA	41804002	0FE00066	58A04000	56C841B0	40024F34	43004002	1EA04180	* .M.ZA.@.. .FX @.VHAO@.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#6E.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	* VHAO@.04C.@.R.A.@.M4A.@.MLC...A@ *
002EE0:	80E6E6BA	00074180	4002118E	42804001	A57A08DA	41804002	0FE00069	43004002	* .FF:..A.@...R.@.%Z.ZA.@.. .IC.@. *
002F00:	19525890	400056AC	433080F8	5899026C	21334300	FDA08DF	27D641C0	80AAE6BA	* .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.*AO@.O.C. *
002F60:	40022FE0	41C08034	41804002	0FE0006E	43004002	19FA41C0	804EE6BA	000E4180	* @.. 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 B  
CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

The following pages contain fragments of a sample display from the Dump Print Utility for release R07.2 and higher. It is from a Model 3200MPS System.

DUMP OF GENERAL PURPOSE REGISTER SETS

	SET F	SET 0	SET 1	SET 2	SET 3	SET 4	SET 5	SET 6
R0	00000000	00047200	00000000	00000000	00000000	00000000	00000006	00000009
R1	00005682	00000070	00000000	00000000	00000000	00000000	FFFFFFF	000020DC
R2	00000000	00000010	00000000	00000000	00000000	00000000	00000C00	00002B60
R3	0001A0B2	00000008	00000000	00000000	00000000	00000000	00000000	0002C874
R4	00000000	00000400	00000000	00000000	00000000	00000000	0001F324	00000FFF
R5	00100000	00000000	00000000	00000000	00000000	00000000	00000400	00019F14
R6	0001BC92	00000006	00000000	00000000	00000000	00000000	0004205A	00000000
R7	00000000	001F3FE0	00000000	00000000	00000000	00000000	00047052	00000000
R8	00018E38	00027C88	00000000	00000000	00000000	00000000	0001F204	0002828C
R9	00000001	00000000	00000000	00000000	00000000	00000000	000278E8	00002B60
RA	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000
RB	0001980A	00000000	00000000	00000000	00000000	00000000	0000072B	00000000
RC	00000000	00009160	00000000	00000000	00000000	00000000	00000000	00002E40
RD	0000001F	0002859A	00000000	00000000	00000000	00000000	00008484	00000000
RE	000133E8	00002B60	00000000	00000000	00000000	00000000	0004F200	000472F0
RF	0001A3F3	00027B54	00000000	00000000	00000000	00000000	00007F08	00000010

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  
 UT REGISTERS OWNER 2B60 \*\*NONE\*\*  
 RS REGISTERS OWNER \*\*NONE\*\*

## DUMP OF APB TABLE

ID	ADDR	STATUS	CTCB	CONTROL STATE	WAIT TCB	CONTROL TCB	QUEUE	PASSBACK
1	84F0	0000	00000000	DISABLED	NONE	NCNE	1	00000000
2	8910	0000	00000000	DISABLED	NONE	NONE	2	00000000
3	8030	0000	00000000	DISABLED	NONE	NONE	3	00000000

## DUMP OF THE QPBS

ID	ADDR	NO. TASKS	READY QUEUE	STATE	EXCLUSIVE TASK	MAPPING TASK
0	9160	0	00000000	ON	NONE	NONE
1	9180	2	001F3480	ON	NONE	NONE
2	91A0	0	00000000	OFF	NONE	NONE
3	91C0	0	00000000	OFF	NONE	NONE

## DUMP OF TCB TABLE

ID	TCB NAME	ADDRESS	TASK FILE NAME	MID	GID	NLU	START	END	SIZE	SHSZ	OPTION	STATUS	WAIT
001	.CSL	2780		000	001	13	18184	18184	0	0	32108008	00000000	00000400
002	.CMDP	2860		000	001	13	18D04	18D04	0	0	32108008	00000000	00000400
003	B1	1F3480	MTM :BACKUP .TSK/00000	000	002	8	3E000	43000	5000	5100	00200C2C	08000000	00000000
004	B2	1F2F60	MTM :BACKUP .TSK/00000	000	002	8	43C00	48000	5000	5100	00200C2C	08000000	00000000

48-031 F00 R02

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 4

SEGMENT CONTROL LIST

ADDRESS	NAME	START	END	SIZE	USE	ROLL	TYPE	FLGS	SREG	KEY	ROLL DCB	SECTOR	SSTP	PRIV
1F33B0	MTM BACKUP TSK	38800	3D900	5100	2	0	PURE	48	9	00			300	RE
**FREE**		3D900	3E000	700										
1F34BC	91	3E000	43000	5000	0	0	IMPURE	48		00				RWE
1F301C	82	43000	48000	5000	0	0	IMPURE	48		00				RWE
**FREE**		48000	1C2000	17A000										

B-5



CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

SPT.SCTT 00000000  
 SPT.OSUP 20303120  
 SPT.PANC 00031504  
 SPT.PSV 00047062 000224C2  
 SPT.RSV 001F3FE0  
 SPT.TSV 00000000  
 SPT.AFSV 00000000 00000000 00000000 00000000 00000000  
 SPT.SCL 001F3380  
 SPT.FLST 00048000  
 SPT.FSYP 001F3850  
 00000000  
 SPT.RINQ 000278E4  
 SPT.RDCB 00000000  
 SPT.SPCT 0000  
 0000  
 SPT.EDMA 00000000  
 SPT.EMHD 00000000  
 SPT.EMTL 00000000  
 SPT.CTCB 00000000  
 SPT.UTOW 00002E40  
 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 00000001  
 SPT.HLDA 00000000  
 SPT.HLDB 00000000  
 SPT.PAGE 00000800  
 SPT.SCLP 00000008  
 SPT.ADDM 0FFFFFFF  
 SPT.OFFM 000007FF  
 SPT.PAGM 0000F800  
 SPT.SEGM 00FF0000  
 SPT.CPID 00000000  
 SPT.CPIE 00124096  
 SPT.MISS 00000000  
 SPT.NODE 20202020  
 20202020  
 SPT.PSDD 00000000  
 SPT.PSDT 00000000  
 SPT.QH 00000000  
 SPT.TID 00000002  
 SPT.PID 0000  
 SPT.IPID 0000  
 SPT.ERBL 00005A98  
 SPT.RDYQ 000278DC  
 00000000  
 SPT.LPMT 00009210  
 SPT.QP80 00009160  
 00000000 00000000 00002780 00002860

15:33:27 08/22/83

PAGE 6

```

*....
*-01
*....
*..pb..$.
*..?.
*....
*.....
*..3.
*....
*..8P
*....
*..x.
*....
*..
*..
*....
*....
*....
*....
*..B
*....
*....
*....
*..r`
*..p`
*....
*....
*....
*....
*....
*..B
*....
*..
*..Z.
*..x.
*....
*....
*..
*.....

```





CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 8

```

TCB.CLC 00000000 *... *
TCB.TGD 00000000 *... *
TCB.VOFF 0000 *.. *
TCB.RSV 0000 *.. *
TCB.RSAC 0000 *.. *
TCB.RCNT 0000 *.. *
TCB.PRI 01 *.. *
TCB.RPRI 01 *.. *
TCB.DPRI 01 *.. *
TCB.HPRI 01 *.. *
TCB.CTSW C80044C0 *..D. *
TCB.SLOC 00000000 *... *
TCB.CTOP 00000000 *... *
TCB.UTOP 00000000 *... *
TCB.TTOP 00000000 *... *
TCB.OBOT 00000000 *... *
TCB.OCB 00000000 *... *
TCB.TEQM 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 *.. *
00 *.. *
TCB.MSEG 0000 *.. *
TCB.SOPT 00000000 *... *
TCB.DLAY 00000000 *... *
TCB.VOL 00000000 *... *
TCB.SVAD 00018224 *...S *
TCB.TMP1 00000000 *... *
TCB.USER 00000000 00000000 *..... *
00000000 00000000 *..... *
TCB.SYS 000183D8 *... *
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 *.. *
0000 *.. *

```

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 9

```

TCB.CPLM 00000000          *....          *
TCB.CPU  000008A8 000000FB 00000000 00000000  *.....          *
TCB.ACUM          *....          *
TCB.WTIM 001438D9          *..8.          *
TCB.RTIM 00000000          *....          *
TCB.HOLD          *....          *
TCB.TMWT 001448F0          *..K.          *
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 0000293C          *..)<          *
TCB.CIOB 000058A0          *..[.          *
IOB.NXT  00000000          *....          *
IOB.RFLG 0000          *..          *
IOB.PRI  01          *..          *
IOB.TYPE 01          *..          *
IOB.DONE 00000000          *....          *
IOB.DCB  00005850          *..[P          *
IOB.TCB  00002780          *..          *
IOB.ESR  00009320          *....          *
IOB.UPBK 00018480          *....          *
IOB.PBLK 00018480          *....          *
IOB.FC   41          *A          *
IOB.LU   00          *..          *
IOB.STAT 00          *..          *
IOB.DDPS 00          *..          *
IOB.SADR 00005679          *..Vy          *
IOB.EADR 000056DC          *..V.          *
IOB.RAND 00000000          *....          *
IOB.LUE  E8005850          *..[P          *
IOB.SVIX 20000000          *..          *
IOB.WCHN 00000000          *....          *
IOB.CYL  0000          *..          *
IOB.SECT 00          *..          *
IOB.LSEC 00          *..          *

TCB.UCTX 00002A60          *..*^          *

CTX.FPTR 00000000          *....          *
CTX.BPTR 00002A60          *..*^          *
CTX.TCB  00002780          *..          *
CTX.TRCE 000183D6          *....          *
CTX.PSW  00047062 0002595E          *..pb..Y^          *
CTX.PSTD 00000000          *....          *
CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....          *
          00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *.....          *

TCB.LPU  00          *..          *

```







CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 13

```

TCB.XFRS 00000000      *....      *
TCB.IOC  0000      *..      *
TCB.IOAC 0000      *..      *
TCB.IOBL 00002D1C      *..-      *
TCB.CIOB 00007D1C      *..>      *
IOB.NXT  00000000      *....      *
IOB.RFLG 0000      *..      *
IOB.PRI  02      *..      *
IOB.TYPE 01      *..      *
IOB.DONE 00000000      *....      *
IOB.DCB  00007CCC      *..|      *
IOB.TCB  00002B60      *..+      *
IOB.ESR  0000AA04      *....      *
IOB.UPBK 00012ECO      *....      *
IOB.PBLK 00012ECO      *....      *
IOB.FC   C0      *..      *
IOB.LU   03      *..      *
IOB.STAT 00      *..      *
IOB.DOPS 00      *..      *
IOB.SADR FFFFFFFF      *....      *
IOB.EADR 0001A40F      *....      *
IOB.RAND 00000000      *....      *
IOB.LUE  5B007CCC      *|.      *
IOB.SV1X 4C030000      *L...      *
IOB.WCHN 00000000      *....      *
IOB.CYL  0002      *..      *
IOB.SECT 01      *..      *
IOB.LSEC 01      *..      *

TCB.UCTX 00002E40      *...B      *

CTX.FPTR 00000000      *....      *
CTX.BPTR 00002E40      *...B      *
CTX.TCB  00002B60      *..+      *
CTX.TRCE 00018E34      *...4      *
CTX.PSW  00047062 0002595E      *..pb..Y^      *
CTX.PSTD 00000000      *....      *
CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
          00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
          *.....*
          *.....*
          *

TCB.LPU  00      *..      *
TCB.QUE  00      *..      *
          00      *..      *
TCB.NLU  00      *..      *
TCB.UCON 00000000      *....      *
TCB.LTFL 00002E00      *....      *
TCB.FMLU 00000000      *....      *
TCB.LTAB EB011074      *..+      *
TCB.TUB  00000000      *....      *
TCB.AF   00000000      *....      *
TCB.II   00000000      *....      *
TCB.MF   00000000      *....      *
TCB.DF   00000000      *....      *
TCB.SVI  00000000      *....      *
TCB.MM   00000000      *....      *

```



CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 15

```

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 001F301C          *..0.          *
SDE.BPTR 001F3380          *..3.          *
SDE.NAME 42312020 20202020 20202020 00000000 *B1          *
SDE.SADR 0003E000          *....          *
SDE.SIZE 00005000          *..P.          *
SDE.USE  0000          *..          *
SDE.ROLL 0000          *..          *
SDE.FLGS 48          *H          *
SDE.TYPE 01          *..          *
SDE.SREG          *..          *
SDE.PREG 09          *..          *
SDE.KEY  00          *....          *
SDE.ASG  00000000          *....          *
SDE.FSEC 00000000          *..          *
SDE.SSTP 0000          *..          *
SDE.PRIV 00          *..          *
          *..          *

TCB.TID  03000001          *....          *
TCB.STAT 08000000          *....          *
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.WAIT 00000000          *....          *
TCB.OPT  00200C2C          *..          *
TCB.CLC  00000000          *....          *
TCB.TGD  00000000          *....          *
TCB.VOFF 0000          *..          *
TCB.RSV  0000          *..          *
TCB.RSAC 0000          *..          *
TCB.RCNT 0000          *..          *
TCB.PRI  96          *..          *
TCB.RPRI 96          *..          *

```





CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 17

```

TCB.STIM 02B187J3      *....      *
TCB.TSL  0000          *..        *
TCB.RLSL 0000          *..        *
TCB.TIMR 0000          *..        *
TCB.LOAD  00           *..        *
                00           *..        *
TCB.XFRS 00000001     *....      *
TCB.IOC   0000          *..        *
TCB.IOAC  02B4          *..        *
TCB.IOBL  001F363C     *..6>      *
TCB.CIOB  00000000     *....      *
IOB.NXT   001F3764     *..7d      *
IOB.RFLG  0000          *..        *
IOB.PRI   96           *..        *
IOB.TYPE  01           *..        *
IOB.DONE  00000000     *....      *
IOB.DCB   00000000     *....      *
IOB.TCB   00000000     *....      *
IOB.ESR   00000000     *....      *
IOB.UPBK  00000000     *....      *
IOB.PBLK  00000000     *....      *
IOB.FC    00           *..        *
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           *..        *

TCB.UCTX  001F37B0     *..7.      *

CTX.FPTR  00000000     *....      *
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 *.....      *
                *.....      *

TCB.LPU   01           *..        *
TCB.QUE   01           *..        *
                00           *..        *
TCB.NLU   08           *..        *
TCB.UCON  00000000     *....      *
TCB.LTFL  001F3710     *..7.      *
TCB.FMLU  00000000     *....      *
TCB.LTAB  00000000     *....      *
TCB.TUB   00000000     *....      *

```

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 18

TCB.AF	00000000									*....	*
TCB.II	00000000									*....	*
TCB.MF	00000000									*....	*
TCB.DF	00000000									*....	*
TCB.SVI	00000000									*....	*
TCB.MM	00000000									*....	*
TCB.RSCH	00000000									*....	*

CTX	1F3780										
CTX.FPTR	00000000									*....	*
CTX.BPTR	00000000									*....	*
CTX.TCB	001F3480									*..4.	*
CTX.TRCE	00000000									*....	*
CTX.PSW	000477F0	00000100								*..W.....	*
CTX.PSTD	00123E68									*..>h	*
CTX.REGS	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*	*
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00090000		*.....*	*
										*	*
CTX.SCRG	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*	*
	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*.....*	*
										*	*
CTX.SFLT	00000000	001F3450	00000790	00000000	00000640	001F3860	001F3860	00000000		*.....4P.....B..8^..8^.....*	*
										*	*
CTX.DFLT	001F3090	00000000	00000000	00000000	00000000	00000000	00000000	00000000	00000000	*..=.....*	*
	00000000	00000000	00000000	00000000	00000000	00000000	42322020	20202020		*.....B2	*
										*	*

TCB	1F2FE0										
TCB.FPTR	001F3480									*..4.	*
TCB.BPTR	001F3480									*..4.	*
TCB.QPTR	00009180									*....	*
TCB.CTX	001F3310									*..3.	*
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	001F3480									*..4.	*
SDE.NAME	42322020	20202020	20202020	20202020	00000000					*B2	*
SDE.SADR	00043000									*..0.	*

CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 19

SDE.SIZE	00005000	*..P.	*
SDE.USE	0000	*..	*
SDE.ROLL	0000	*..	*
SDE.FLGS	48	*H	*
SDE.TYPE	01	*.	*
SDE.SREG			
SDE.PREG	09	*.	*
SDE.KEY	00	*.	*
SDE.ASG	00000000	*....	*
SDE.FSEC	00000000	*....	*
SDE.SSTP	0000	*..	*
SDE.PRIV	00	*.	*
	00	*.	*
TCB.TID	04000002	*....	*
TCB.STAT	08000000	*....	*
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.WAIT	00000000	*....	*
TCB.OPT	00200C2C	*.. /	*
TCB.CLC	00000000	*....	*
TCB.TGD	00000000	*....	*
TCB.VOFF	0000	*..	*
TCB.RSV	0000	*..	*
TCB.RSAC	0000	*..	*
TCB.RCNT	0000	*..	*
TCB.PRI	96	*.	*
TCB.RPRI	96	*.	*
TCB.OPRI	96	*.	*
TCB.MPRI	08	*.	*
TCB.CTSW	00000000	*....	*
TCB.SLOC	00090000	*....	*
TCB.CTOP	00047FFE	*....	*
TCB.UTOP	00001880	*....	*
TCB.TTOP	00001880	*....	*
TCB.OBOT	00000000	*....	*
TCB.OCB	00000000	*....	*
TCB.TEQH	00000000	*....	*
TCB.SEG	001F2F80	*.. /	*
TCB.LRA			
TCB.PSTD	00123E5F	*..>_	*
TCB.SSTD	003E0006	*..>..	*





CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

15:33:27 08/22/83

PAGE 22

CTX.BPTR 00000000  
 CTX.TCB 001F2FE0  
 CTX.TRCE 00000000  
 CTX.PSW 000477F0 00000100  
 CTX.PSTD 00123E5F  
 CTX.REGS 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  
 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00090000

\*.... \*  
 \*../. \*  
 \*.... \*  
 \*..#..... \*  
 \*..>\_ \*  
 \*..... \*  
 \*..... \*  
 \*

CTX.SCRG 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  
 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

\*..... \*  
 \*..... \*  
 \*

CTX.SFLT 00000000 001F348C 00002634 40544020 4241434B 55502020 54534800 03275053

\*.....4...&4MTM BACKUP TSK..'JS\*

CTX.DFLT 00038800 00005100 00020000 48020900 00000000 00000000 03000500 00000000  
 00000000 00000000 00000000 00000000 00000060 5C1207C0 00000000 00000000

\*.....Q.....H..... \*  
 \*.....\..... \*  
 \*

QPB 009180  
 TKQ.LOKS 0000  
 TKQ.CNT 0002  
 TKQ.FRNT 001F3480

\*.. \*  
 \*.. \*  
 \*..4. \*

QPB.TPTR 00000000  
 QPB.QWTC 00000000  
 QPB.MTID 0000  
 QPB.ETID 0000  
 QPB.STAT 0010  
 QPB.BMAP 4000  
 QPB.ACNT 01  
 QPB.LCNT 01  
 QPB.ID 01  
 00000000 00

\*.... \*  
 \*.... \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*8. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*..... \*

QPB 0091A0  
 TKQ.LOKS 0000  
 TKQ.CNT 0000  
 TKQ.FRNT 00000000

\*.. \*  
 \*.. \*  
 \*..... \*

QPB.TPTR 00000000  
 QPB.QWTC 00000000  
 QPB.MTID 0000  
 QPB.ETID 0000  
 QPB.STAT 0000  
 QPB.BMAP 2000  
 QPB.ACNT 01  
 QPB.LCNT 00  
 QPB.ID 02

\*.... \*  
 \*.... \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*  
 \*.. \*













CONTENTS OF MEMORY PRODUCED BY A PANIC DUMP

```

0000C0: 00000040 00035A9C 00000000 00000000
000020: 88018801 88018801 00000000 00000000
000040: 88018801 88018801 00047000 0001027C
000060: 23022540 43004003 3E640000 00000000
000080: 00008484 000006B0 00047050 00010F0A
0000A0: 21F62234 22462266 229822DE 2180236A
0000C0: 00009150 00000003 00047000 000108B8
0000E0: 24FC24FC 24FC24FC 24FC24FC 24FC24FC
000100: 24FC24FC 24FC24FC 24FC24FC 24FC24FC
000120: 054124FC 058124FC 05C124FC 060124FC
000140: 24FC24FC 24FC24FC 24FC24FC 24FC24FC
000160-00019F ***SAME AS ABOVE ***
0001A0: 24FC24FC 24FC24FC 24F024F6 24FC24FC
0001C0: 24FC24FC 24FC24FC 24FC24FC 24FC24FC
0001E0-0002FF ***SAME AS ABOVE ***
000300: 54140710 001F33B0 00000000 00000000
000320: 00000000 00000000 00000000 00000000
000340-0003FF ***SAME AS ABOVE ***
000400: FF01FF9D 0000560C 00001952 00000000
000420: 000184EA 00280000 00005D5D 0000C7F4
000440: FF010000 00000000 00000000 00000000
000460: 00000000 00000000 00000000 0000C7F4
000480: FF010000 00000000 00000000 00000000
0004A0: 00000000 00000000 00000000 0000C7F4
0004C0: FF010000 00000000 00000000 00000000
0004E0: 00000000 00000000 00000000 0000C7F4
000500: FF010000 00000000 00000000 00000000
000520: 00000000 00000000 00000000 0000C7F4
000540: FF010000 00000000 00000000 00000000
000560: 00000000 00000000 00000000 0000C7F4
000580: FF010000 00000000 00000000 00000000
0005A0: 00000000 00000000 00000000 0000C7F4
0005C0: FF010000 00000000 00000000 00000000
0005E0: 00000000 00000000 00000000 0000C7F4
000600: FF010000 00000000 00000000 00000000
000620: 00000000 00000000 00000000 0000C7F4
000640: 00000000 00000000 00000000 00000000
000660: 00000002 00000000 00000000 00000000
000680: 00000000 00000000 0000C000 0000FF00
0006A0: 00000000 00000000 00000000 00007CCC
0006C0: 00000000 00100000 00018C92 00000000
0006E0: 00000000 0000001F 000133E8 0001A3F3
000700: 00000400 00000000 00000006 001F3FE0
000720: 00009160 0002859A 00002860 00027B54
000740: 00000000 00000000 00000000 00000000
000760-00081F ***SAME AS ABOVE ***
000820: 00000000 00000000 00000000 00000000
000840: 0001F324 00000400 0004205A 00047052
000860: 00000000 00008484 0004F200 00007F08
000880: 00000FFF 00019F14 00000000 00000000
0008A0: 00002E40 00000000 000472F0 00000010
0008C0: 00000000 00000000 00000000 00000000
0008E0-00095F ***SAME AS ABOVE ***
000960: 00000000 00000000 00000000 00000000
000980: 00000000 00000000 00000000 00000000
0009A0: 00000000 00000000 00000000 00000000
0009C0: DE2092D6 DE2092D6 58540018 483092D2

```

15:33:27 08/22/83

PAGE 61

```

00000000 00000000 00000000 00000000 * ...@..Z..... *
00000040 00035A9C 00000000 000105EC * .....@..Z.....UL *
050000CF 43000080 F81001FF 00621313 * .....P...R.U...OC...X...B.. *
430C4003 15040000 00000000 00000000 * #.X&C.>D.....C.&..... *
00047000 0001CED4 00047000 218021BA * .....0..PP.....P...NT..P..101: *
238E2180 218023EE 24142426 00000000 * !V"4"F"F".."A"10#J#.1010#N$.8.... *
24E824FC 24FC24FC 24FC24FC 24FC24FC * ...P.....P...X8$$.S$.S$.S$.S$. *
0401041D 24FC24FC 24FC24FC 24FC24FC * $.S$.S$.S$.S$.S$.S$.S$.S$.S$. *
044124FC 048124FC 04C124FC 050124FC * $.S$.S$.S$.S$.S$.S$.S$.S$.S$. *
24FC24FC 24FC24FC 24FC24FC 24FC24FC * .A$.S$.S$.S$.S$.S$.S$.S$.S$. *
24FC24FC 24FC24FC 24FC24FC 24FC24FC * $.S$.S$.S$.S$.S$.S$.S$.S$.S$. *
- SAME - *
24FC24FC 24FC24FC 24FC24FC 24FC24FC * $.S$.S$.S$.S$.S$.S$.S$.S$.S$. *
24FC24FC 24FC24FC 24FC24FC 24FC24FC * $.S$.S$.S$.S$.S$.S$.S$.S$.S$. *
- SAME - *
00000000 00000000 00000000 00000000 * T.....30..... *
- SAME - *
00009BDC 0A824500 00005B50 FF0C0001 * .....V\...R.....\...E...CP... *
24FC0000 00005850 00009C5C FF060C7E * ...J.+...]]..GT$.S$.S$.S$.S$. *
00009BDC 128E0000 00005D68 00000000 * .....GT$.....\...JH... *
24FC0000 00005D68 00009C5C 00000000 * .....GT$.....\...JH... *
00009BDC 128E0000 00005F80 00000000 * .....GT$.....\...A... *
24FC0000 00005F80 00009C5C 00000000 * .....GT$.....\...A... *
00009BDC 128E0000 00006198 00000000 * .....GT$.....\...A... *
00009BDC 128E0000 00006380 00000000 * .....GT$.....\...CO... *
24FC0000 00006380 00009C5C 00000000 * .....GT$.....\...CO... *
00009BDC 128E0000 000065C8 00000000 * .....GT$.....\...EH... *
24FC0000 000065C8 00009C5C 00000000 * .....GT$.....\...EH... *
00009BDC 128E0000 000067E0 00000000 * .....GT$.....\...G... *
24FC0000 000067E0 00009C5C 00000000 * .....GT$.....\...G... *
00009BDC 128E0000 000069F8 00000000 * .....GT$.....\...IX... *
24FC0000 000069F8 00009C5C 00000000 * .....GT$.....\...IX... *
00009BDC 128E0000 00006C10 00000000 * .....GT$.....\...L... *
24FC0000 00006C10 00009C5C 00000000 * .....GT$.....\...L... *
0000C7F4 20A00000 00006E28 00000000 * .....GT.....N... *
10983000 0F006F4C 00000000 FFFFFFFF * .....G...OL..... *
000073D0 00000000 00000000 00000000 * .....SP..... *
00000000 00005682 00000000 0001ADB2 * .....L.....V.....-2 *
00018E38 00000001 00000000 0001980A * .....<.....8..... *
00047200 00000070 00000010 00000008 * .....3H...#S...R...P..... *
00027C88 00000000 0000C000 00000000 * .....?..... *
00000000 00000000 00000000 00000000 * ... ..+ ...T..... *
- SAME - *
00000006 FFFFFFFF 00000000 00000000 * ..... *
0001F204 000278E8 00000000 00000728 * ..SS.....Z..PR...R...X.....+ *
00000009 00002DDC 00002B60 0002C874 * .....R.....-\...+ ..HT *
0002828C 00002860 00000000 00000000 * .....+..... *
00000000 00000000 00000000 00000000 * ..B.....RP..... *
- SAME - *
00000000 00000000 00008870 00000000 * .....P... *
00000000 00000000 00008C90 00000000 * ..... *
00000000 00000000 00009080 00000000 * .....O... *
^ .V^ .VXT..HO.R#5..$1T5...BO.^X5 *

```

CCNTEENTS OF MEMORY PFOUDCED BY A PANIC DUMP

```

0009EG: 0084F330 2000G000 43308050 D3350C70
000A00: 233F4170 864C7335 009A4873 000A4220
000A20: 009D0336 0000C530 00182339 41708622
000A40: 80405835 0084F330 20000000 2133E670
000A60: 9023C330 00014230 8FFC2466 74650100
000A80: 18009826 03334230 835CC460 007FC560
000AA0: E3640000 4260805E 18005854 00182432
00CAC0: 25314C35 00C01800 41708586 DA25009F
000AEO: 4170856E 24767475 00F62335 C560000D
000B00: 42608830 18000733 58540018 023500FA
000B20: 00004875 00F42671 08674300 801E5854
000B40: 02760000 27614875 00F40867 7345009A
000B60: E660800C 40640014 7345009A 43008006
000B80: 878C1800 98260833 203CC460 007F5854
000BA0: 08761171 5A740010 48770000 1171F570
000BC0: 000F238E 40750110 48750114 27712312
000BE0: FF8E4064 00141800 27210E20 90AC5654
000C00: 24637465 00F6233E 9D23232C 417084A6
000C20: 87080766 48640002 10622664 4065002C
000C40: 00134330 818E2621 DE209050 E6608018
000C60: 42308E02 08439D23 08332332 18005854
000C80: 76350100 1800E660 802E4064 00142468
000CA0: 5A640004 D3360000 E7340010 E3640000
000CC0: 42308622 18005854 00182461 7665004C
000CE0: 027500AE 417083EA 24787674 00004170
000D00: 80245854 00185875 0084F370 20000000
000D20: 76740000 1800C860 FFOC4064 00004170
000D40: 00182721 E6604000 24FC4064 00140834
000D60: F05A9826 2621C460 007F5854 0018C560
000D80: 00134330 809AC560 00184330 856A0833
000DA0: 18002470 74750100 2337C560 000D4330
000DC0: 00114330 80764855 002CC550 7FFF4330
000DE0: 41704000 C0749826 08334230 84F8C460
000E00: C5600011 4230849C E660FF56 40640014
000E20: DE208E7A C8607FFF E6708608 58540018
000E40: E67086C0 58540018 24607665 0100E660
000E60: 48640002 10622664 4300FFC0 58540018
000E80: 40340014 7345009A E6304000 24FC4034
000EA0: 585500E8 D3550001 23022551 48340016
000EC0: 00094330 80405854 00182531 03070855
000EE0: 24315854 00185135 00845835 00845535
000F00: 01620733 03074054 000E5854 00182430
000F20: F5300000 08842335 E630FC44 40340014
000F40: 587500E8 D3670001 11684065 00C4C330
000F60: FFFFC660 20004065 00C4DA27 40FFFFFD
000F80: 00C44535 00A22387 74350162 21344170
000FA0: DA2500C5 487500C6 40740014 7374000C
000FC0: 75350162 C8600020 9A265935 00842631
000FE0: 00A24380 FF8E7635 01625875 00087437
001000: 000A7345 009A7364 C0144065 008C4170
001020: 008C4064 00147345 C0982721 7374000A
001040: 423082A2 C460007F 58540018 7374000A
001060: 01000337 24357435 01000237 4074000A
001080: 9A264170 4000C074 C5300008 2235DE20
0010A0: 24615165 00A47345 00982721 03640008
0010C0: 00214237 00022461 74650084 43370002
0010E0: 0018D365 0103D375 00F8C370 00402133

```

15:33:27 08/22/83

```

C3300020 21375835 0084F330 02000000
802A4873 00164230 81424300 8E9AE665
E665009D 24702433 43008110 9826E670
80824074 00142435 74350100 42309006
2335C330 00024230 8894C330 00080337
00202383 E7640010 41708586 4170840A
74350100 42308CA4 4300FFDC 417083AC
4300FF68 98260833 4230880A C460007F
4330883C 48340016 21129A26 E3640000
24617665 01005865 00ECD375 01040276
00182471 517500A4 586500EC D3750102
40340016 4074000A 5064000C 73450098
9826C330 00014230 8EECC330 00024230
0018C560 007F4330 8044C560 0020238D
00000AAA 42308026 48750110 2671C570
24704075 01140265 47000116 1800E660
00182466 74650100 23349D23 4220870E
2302230F 07664065 00725065 00384300
9826C460 007FC560 00144330 8196C560
40640014 08347345 00982465 74650100
00182437 74350100 23374170 84542437
75640000 246E7464 00002339 48640002
42608044 9A261800 4220803E 23350833
43308660 487500F0 407500AC D37500F2
4000C074 43008674 9A264864 00164230
2337247E 76740000 4874000A 23242478
4000C074 24624564 00164330 88065854
73450098 24634563 00164330 F0CE4300
000E4330 885EC560 00144330 80A2C560
42308552 C560000F 21352461 77650100
808A4300 8516C560 00124330 807EC560
84FA1800 C8707FFF 4075002C 73450098
007FC560 00184330 84FC560 00122335
58540018 4170803E 7345009A 4300FE26
4835002C 03374065 002C0307 DE208E5C
FF104064 00145854 00187345 009A0766
24327435 0100E630 88CC2133 E630FC02
00140E20 8E087345 00980307 58540018
2118C560 007F2338 C5600020 2389C560
21172531 C3507F00 21359A25 23039A26
00802183 50350080 453500A2 23837635
743500AC 4330FFAE 48340014 403500C6
4074000C DE208D64 4834000E 42108076
7F004330 801E5575 00E02130 D36740FF
4170808A 58350084 27315035 0084DA25
806C220F D33500C5 08332335 4170805E
24692431 03075835 00844535 00A22383
50350084 55350080 21835035 00804535
00004230 FFAE4170 80044300 FFC64074
4000C074 C5300008 22355854 00187365
03074074 000A4170 4C00C074 98260833
03075854 00184834 00160217 24377635
02640008 7345009A 2621DE20 8C1E246A
8C06E660 400024FC 40640014 58540018
7374000A 0307D365 0070C460 0021C560
03074864 00004064 00204074 00225854
D3650102 24702433 763500AC 42308036

```

```

* .SO .CO.P55.PC0. 17X5..SO... *
* #?AP.L55..HS..B .#HS..BO.BC...FE *
* .S6..EO.#9AP..FE..$P33C...8FP *
* .X5..SO .13FP..BT..$5T5..BO.. *
* .#CO..BO..$FTE..#5CO..BO..CO..7 *
* .B..380.\D .E .#.GO..AP.6AP.. *
* CO..B .^..XT..$2T5..BO.$C..AP.. *
* X165..AP..ZX..C..H.8.380..D .. *
* AP.NSVTU.V#5E .CO.<H4..I..&CD.. *
* B .O...XT..R5.ZSAVE..XE.LSU..RV *
* .HU.T&Q.GC...XT..$QQU.$XE.LSU.. *
* RV..AMU.T.GSE..@4..@T..PD..SE.. *
* F ..@D..SE..C..V.&CO..BO.LCO..BO *
* .....&.3 <D ..XT..E ..CO.DE .#. *
* .V.QZT..HW...QUP..#BO.&HU..&@P *
* .#.BU..HU..Q#..S.BU..REG...F *
* ..@D...^I^ ..XT..$FTE..#4.#B .. *
* SCTE.V#>..#AP.&#..#F@E.RPE.BC. *
* .X.FKD...B&D@E..&D ..E ..CO.E *
* ..CO..&I^ .PF ..@D...4SE..$ETE.. *
* BO...C.#.3#2..XT..$7T5..#7AP.T$7 *
* V5...F ..@D..$HUD..$NTD..#9HD.. *
* ZD..S6..G4..CO..B .D..8..B .>#5.3 *
* BO..XT..SAVE..CO.HU.P@U..SU.R *
* RU..AP.J$XVT..AP@.BT..<BHD..BO *
* .XT..XU..SP ..#7$.VT..HT..#S$X *
* VT...H ..@D..AP@.BT$BED..CO..XT *
* .IF @.$@D...4SE..SCEC.CO.CT *
* .Z.&&ID ..XT..E ..CO.^E ..CO.E *
* ..CO..E ..CO.J.380.RE .I$@E.. *
* ..$PTU..#7E ..CO.CCO.E..$@E.. *
* ..CO.VHU..EP..CO.Z..HP..@U..SE.. *
* AP@.BT.&.380.XD .E ..CO.E ..#5 *
* E ..BO..F .V@D..XT..AP.>SE..C.& *
* ^ .ZH ..FP.XT..H5..7@E..S^ .\ *
* FP.@XT..$ VE..F ..@D..XT..SE..F *
* KD...B&OC..@XT..$2T5..FO.LI3FO.. *
* @4..SE..F08.$@4..^ .SE...XT.. *
* XU.HSU..#.XQH4..I.E ..#8E .#E *
* ..CO.@XT..X1...UI.X1CP..I5.X#..& *
* $1XT..Q5.4X5.4U5.OI.P5.OE5."#V5 *
* .B.3..@T..XT..$0T5..CO.H4..@5.F *
* UO...#5FO.D@4..@T..^ .DH4..B..V *
* XU.HSG..H@E.DCO..CO..UU. I$SG@ *
* .F .@E.DZ'@..AP..X5.4'1P5.4Z *
* .DES."#T5.BI4AP.L".SS.E.3#5AP.^ *
* ZX.EHU.F@T..ST..I$1..X5.4E5."# *
* U5.BH .&X5.4&1P5.4U5.OI.P5.OE5 *
* "C.>V5.BXU.XT7..BO..AP..C..F@T *
* ..SE..SD..@E.<AP@.@TEO..$5XT..SE *
* <@D..SE..!ST...@T..AP@.L@.3 *
* BO."D ..XT..ST...XT..H4...$7V5 *
* ...7$5T5...7@T..RD..SE..&I^ ..$J *
* .&AP@.@TEO..$^ .F @.S.@D..XT.. *
* $AQE.$SE..!SD..ST...SE.PD .I.E *
* .I@7..$ATE..C7...HD..@D. @T.XT *
* .SE..SU.XCP.@I3SE..$P33V5..BO.6 *

```









## Selective back-up operations

(Continued)

use of hyphen	3-16
.SPL command	
OS/32 Spooler	2-11
SPL/32 Spooler	2-1
Spool queue	2-4
START command	
Accounting Reporting	4-6
DISCSYNC	8-4
Disk Backup	3-7
Disk Dump	6-2
Dump Print	7-2
Error Reporting	5-6
OS/32 Spooler	2-5
.STM. See modules.	
Summary reports	
device errors	5-33
memory errors	5-34
system errors	5-30
system milestones	5-30
SWOP command	8-1
System Support Utilities.	
See Utilities.	

## T

TASK command	
Disk Backup	3-7
Disk Dump	6-2
Dump Print	7-2
Task termination codes. See	
end of task codes.	

## U

UNDUMP command	
Disk Dump	6-7
Utilities	
Accounting Reporting	4-1
Disk Backup	3-1
Disk Dump	6-1
Dump Print	7-1
Error Reporting	5-1
Mirror Disk	
Synchronization	8-1
OS/32 Spooler	2-1
overview	1-1

## V

VERIFY command	
Disk Dump	6-8
VOLUME command	
Disk Dump	6-3

## W, X, Y, Z

WIDTH command	
Disk Dump	6-9

PERKIN-ELMER  
Data Systems Group

D O C U M E N T A T I O N   C H A N G E   N O T I C E

The purpose of this documentation change notice (DCN) is to provide a quick and efficient way of making changes to technical manuals before they are formally updated or revised.

The manual affected by these changes is:

-----  
**48-031 F00 R02   OS/32 SYSTEM SUPPORT UTILITIES Reference Manual**  
-----

Please disregard Chapter 3 of the System Support Utilities Reference Manual, 48-031 F00 R02, and refer to the attached pages for the correct documentation of the Disk Backup Utility.

The R07.2 OS/32 software package (04-082 F00 M99 R15) contains the R03-02 version of the Disk Backup Utility (03-153). However, Chapter 3 of the System Support Utilities Reference Manual, 48-031 F00 R02, documents the R04-00 version of Backup. As a result, Chapter 3 of the System Support Utilities Reference Manual, 48-031 F00 R01, has been reprinted and included here to provide documentation of the Disk Backup Utility.

When the R04-00 version of Backup is released, Chapter 3 of the System Support Utilities Reference Manual, 48-031 F00 R02, will provide accurate documentation of the new version of Backup.

The previous Appendix B, in revision R01, of this manual documented the Disk Integrity Check Utility (03-080 F00 R08-01). This utility was scheduled to be removed from OS/32, but ultimately was not. The previous Appendix B has therefore been reprinted and included here for reference. This should not be confused with the current Appendix B, entitled "Contents of Memory Produced by a Panic Dump."

## 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

Volume header	FIB	Data	EOF	FIB	Data	EOF	EOV
---------------	-----	------	-----	-----	------	-----	-----

#### TAPE 2

Volume header	Remaining data	EOF	EOV
---------------	----------------	-----	-----

### 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
    { POSITION
    { NOREWIND } } [ ,NEWDATE ]
    { dd/mon/yy }
```

**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.



## APPENDIX B DISK INTEGRITY CHECK UTILITY

### B.1 GENERAL DESCRIPTION

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:

```
START ,dev: ,[list fd] [ { CLOSE }  
                   { NOREADCHECK }  
                   { READCHECK } ]
```

#### 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:**

<b>ST ,DSC1:,CON:</b>	Normal program with read check
<b>ST ,DSC1:</b>	Normal program with read check (list device was preassigned)
<b>ST ,DSC1:,PR:</b>	Normal program with read check
<b>ST ,DSC1:,CON: ,R</b>	Normal program with read check
<b>ST ,DSC1:,CON:</b>	Normal program
<b>ST ,DSC1:,CON: ,CLOSE</b>	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.



**PERKIN-ELMER**

**PUBLICATION COMMENT FORM**

We try to make our publications easy to understand and free of errors. Our users are an integral source of information for improving future revisions. Please use this postage paid form to send us comments, corrections, suggestions, etc.

1. Publication number \_\_\_\_\_

2. Title of publication \_\_\_\_\_

3. Describe, providing page numbers, any technical errors you found. Attach additional sheet if necessary.  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

4. Was the publication easy to understand? If no, why not?  
\_\_\_\_\_

5. Were illustrations adequate? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

6. What additions or deletions would you suggest? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

7. Other comments: \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

From \_\_\_\_\_ Date \_\_\_\_\_

Position/Title \_\_\_\_\_

Company \_\_\_\_\_

Address \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

STAPLE

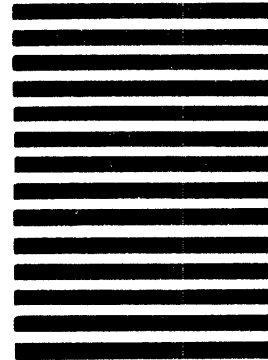
STAPLE

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**

Data Systems Group  
106 Apple Street  
Tinton Falls, NJ 07724

**ATTN:  
TECHNICAL SYSTEMS PUBLICATIONS DEPT.**

FOLD

FOLD

STAPLE

STAPLE