TOPS–10/TOPS–20 SPEAR Manual

AA–J833B–TK

September 1985

This manual describes the SPEAR product (Standard Package for Error Analysis and Reporting). SPEAR is a library of functions that reports on the errors and events that are recorded by the operating system.

This manual supersedes the *TOPS–10/TOPS–20 SPEAR Manual*, order number AA–J833A–TK.

OPERATING SYSTEM: TOPS–10 V7.02
TOPS–20 (KS/KL Model A) V4.1
TOPS–20 (KL Model B) V6.1

SOFTWARE: SPEAR V2.0

Software and manuals should be ordered by title and order number. In the United States, send orders to the nearest distribution center. Outside the United States, orders should be directed to the nearest DIGITAL Field Sales Office or representative.

Northeast/Mid–Atlantic Region

Digital Equipment Corporation
PO Box CS2008
Nashua, New Hampshire 03061
Telephone:(603)884–6660

Central Region

Digital Equipment Corporation
Accessories and Supplies Center
1050 East Remington Road
Schaumburg, Illinois 60195
Telephone:(312)640–5612

Western Region

Digital Equipment Corporation
Accessories and Supplies Center
632 Caribbean Drive
Sunnyvale, California 94086
Telephone:(408)734–4915

digital equipment corporation • marlboro, massachusetts

First Printing, April 1982
Revised, September 1985

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment 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 may only be used or copied in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by DIGITAL or its affiliated companies.

The following are trademarks of Digital Equipment Corporation:

digital™

DEC	MASSBUS	RSX
DECmate	PDP	RT
DECsystem–10	P/OS	UNIBUS
DECSYSTEM–20	Professional	VAX
DECUS	Q–BUS	VMS
DECwriter	Rainbow	VT
DIBOL	RSTS	Work Processor

The postage-prepaid READER'S COMMENTS form on the last page of this document requests the user's critical evaluation to assist us in preparing future documentation.

CONTENTS

FIGURES

TABLES

PREFACE

This manual describes Version 2.Ø of SPEAR on TOPS-1Ø and TOPS-2Ø. The primary audience for this manual is a person with experience in the following areas:

1. Fault isolation techniques
2. KL1Ø instruction set
3. All hardware connected to the various configurations of TOPS-1Ø or TOPS-2Ø

If you do not have the above experience, refer to:

TOPS-1Ø Operators Guide

TOPS-2Ø Operators Guide

DECsystem-1Ø/DECSYSTEM-2Ø Processor Reference Manual

DECsystem-1Ø Hardware Reference Manual

READING PATH

This manual has three functions: it serves as a learning aid, a user's guide, and a reference tool for those who already have learned to use the SPEAR library.

As a learning aid: Chapters 1, 2, and 3 provide an overview of the SPEAR library. They also provide background information necessary to understand and use the SPEAR library.

As a user's guide: Chapter 4 provides step-by-step procedures for using the SPEAR functions; INSTRUCT, RETRIEVE, KLERR, SUMMARIZE, and COMPUTE. This chapter explains the command syntax and the response parameters associated with each function.

As a reference tool: Chapter 5 and the appendixes provide reference material such as system event file formats, error messages, and a glossary. This material is not meant to be read from beginning to end. Use Chapter 5 and the appendixes as a reference when you need them.

CONVENTIONS USED IN THIS MANUAL

The following conventions are used throughout this manual:

Contrasting colors	Red - where examples contain both user input and computer output, the characters you type are in red; the characters SPEAR prints are in black.
Lowercase letters	Lowercase letters in a command string indicate variable information you must supply.
UPPERCASE LETTERS	Uppercase letters in a command string indicate fixed (literal) information that you must enter as shown.
[ ]	Square brackets indicate optional information that you can omit from a command string. Do not type the square brackets.
Examples	All examples were produced on either the TOPS-10 or the TOPS-20 operating system.
(ESC)	This symbol represents where you press the Escape key.
(RET)	This symbol represents where you press the RETURN key.

CHAPTER 1

SPEAR OVERVIEW

1.1 INTRODUCTION

This chapter introduces you to the SPEAR product and gives an overview of its use.

The name SPEAR is an acronym for Standard Package for Error Analysis and Reporting. The main function of SPEAR is to help isolate the cause of a failure through information contained in the system event file. Most failures are intermittent; that is, they are active at one instant causing system malfunction and inactive at another instant allowing system operation. The task at hand is to find the cause of the failure and correct the problem in the least amount of time. SPEAR helps to accomplish this task.

SPEAR is a library of functions that reports on the errors and events that are recorded by the operating system, TOPS-10 or TOPS-20. In the past, the field service engineer was forced to analyze intermittent failures by sorting through error reports generated by SYSERR, looking for common failure patterns. For example, the engineer examined several disk reports looking for common media failures, common disk head failures, or common failures of the read/write circuitry. Now, SPEAR can do the tedious work.

SPEAR uses the system event file for analysis. The system event file contains entries made by the operating system and the communications subsystems (if any). Each time certain events occur, the operating system records and stores pertinent data in the system event file. The operating system continually monitors and records information about every disk, tape, and memory parity error as they occur, along with errors from other subsystems. At your discretion, you can call on SPEAR to generate a report of selected events.

For more information on the system event file, refer to Chapter 2. For samples of events your operating system can record, refer to Chapter 5.

The SPEAR program consists of a library of five functions:

- INSTRUCT
- RETRIEVE
- KLERR
- SUMMARIZE
- COMPUTE

These function names are also the primary commands you type to run the particular function of SPEAR in which you are interested.

INSTRUCT is a computer-aided instruction program designed to ensure that you have the background knowledge and experience necessary to use the other functions in the SPEAR library. To run INSTRUCT, refer to Section 4.3.

RETRIEVE reads the binary data in the system event file and produces an ASCII report for each entry selected. RETRIEVE also allows you to save specific entries either for later analysis and translation or for record-keeping purposes. RETRIEVE is described in Section 4.4.

KLERR provides signal name translation and summaries, CRAM word translation, and other useful features to help you analyze log files resulting from a KL1Ø crash. KLERR is described in Section 4.5.

SUMMARIZE reads the binary data in the system event file and produces an ASCII report. Refer to Section 4.6 for a description of SUMMARIZE.

COMPUTE calculates and reports overall system availability, effectiveness, and reliability. COMPUTE is described in Section 4.8.

Chapter 4 describes these functions in detail, along with an additional feature available only on TOPS-2Ø, KLSTAT mode.

1.2 USER PROFILES AND INTERACTION

There are three main groups of SPEAR users:

1. Field Service and Software Support personnel who have specific maintenance responsibilities.

2. System operators who must recognize failures and initiate recovery procedures.

3. System managers who have a need to monitor overall system performance and schedule system use.

These groups each have varying degrees of expertise in software and hardware areas. SPEAR can not only handle the needs of each group but can also guide the new user as well as the experienced user.

The system operator and Field Service engineer can cooperate by using SPEAR as a tool for both preventive and corrective maintenance. SPEAR also has the COMPUTE function that allows the system manager a closer look at system performance. Refer to Chapter 4 for information on COMPUTE.

CHAPTER 2

THE SYSTEM EVENT FILE

2.1 INTRODUCTION

This chapter discusses the file that SPEAR uses for input, the system event file. Specifically, this chapter discusses what events are recorded, how they are recorded, and what form they take within their respective files.

Each operating system and communications subsystem has its own error logging facility to gather and maintain information on system errors and events as they occur. The error logging facility detects a variety of hardware and software errors, providing a detailed record of system activity. When an error occurs, the facility gathers significant data about the current state of the system; the type of data it gathers depends on the type of error detected. In addition to detecting actual errors, the facility monitors events that reflect other aspects of system performance. The recording of such events helps to define the system context in which actual errors occur.

The events are recorded in a system event file, ERROR.SYS. The logical name for the location of this file (structure and directory) depends on which operating system you are using. The following list gives you the names to use to locate your system event file:

- TOPS-10 V7.02 SYS:ERROR.SYS
- TOPS-20 V4.1 SYSTEM:ERROR.SYS
- TOPS-20 V6.1 SERR:ERROR.SYS

Events that occur during the operation of the system are logged into the system event file for use in preventive maintenance as well as corrective maintenance. These events occur within the various hardware and software components of the system, such as:

Hardware	Software
CPU	Operating system
Memory	Memory management
I/O	I/O
Console	File system

Some of the events that can occur include parity errors, address failures, operator log entries, system reloads, device mounts and dismounts. Each time one of these events occurs, an entry is appended to the system event file in binary format.

2.2 ENTRY CATEGORIES

There are two general categories of entries in the system event file, error and nonerror. Both categories can be broken down further into the following:

1. Software entries
2. Hardware entries
3. Performance entries

The following three sections describe the entry types that can be found in the system event file.

2.2.1 Software Entries

The software error entries that SPEAR is concerned with are internal software errors. On TOPS-1Ø, these errors result in a STOPCD; on TOPS-2Ø, these errors result in a BUGHLT, BUGCHK, or BUGINF.

A STOPCD is represented by a 3-letter message that is printed at the operator's terminal (CTY) when the operating system detects a serious error. Sometimes the operating system crashes immediately following this message; at other times the operating system continues to run but halts the current job. The action the operating system takes depends on the severity of the problem. There are five types of STOPCDs:

1. HALT - The system halts and you must manually dump and and reload the operating system.
2. STOP - All jobs are aborted, and the system automatically dumps and reloads itself.
3. CPU - This is the same as STOP except this message occurs on dual processors. Jobs are aborted only on the processor where the error occurs.
4. JOB - The current job is aborted and processing continues.
5. DEBUG - A message prints and processing continues.

The list of all stopcode messages is documented in the STOPCD specification in the TOPS-1Ø Software Notebooks.

The TOPS-2Ø operating system errors also range in severity. A BUGHLT is the most serious. It is a non-recoverable error detected by the operating system. A BUGCHK is a recoverable error detected by the operating system, while a BUGINF is a message informing you that a certain event related to the operating system has occurred. BUGHLTs, BUGCHKs, and BUGINFs are listed in the TOPS-2Ø Operators Guide.

2.2.2 Hardware Entries

The hardware entries come from a variety of subsystems; CPU, memory, I/O, console, and networks. The number and type of components depends on the system configuration. In general, Figure 2-1 represents the major components or subsystems that can contribute entries to the system event file.

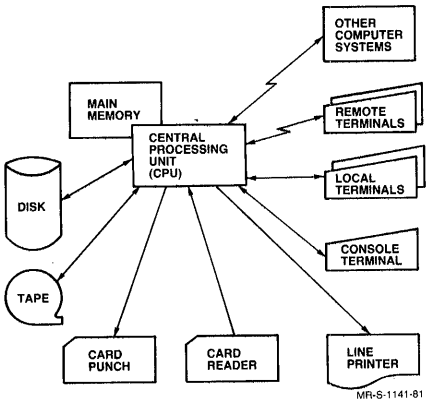


Figure 2-1: Components of a Computer System

Hardware error entries are the most frequent type of error. These errors are caused by a failure in the hardware itself. Each time an event of this type occurs, an entry is made into the system event file. Hardware error entries can be divided into three general categories:

1. CPU-instruction and CPU-addressing failures
2. Controller and channel failures
3. I/O errors

Because the system hardware cannot be expected to operate continuously without failure, the design of the hardware includes facilities to monitor the hardware operation. (One such facility is the parity check.) Once the system has detected an error, it can either signal the CPU and system software that an error has occurred or attempt to recover from the error and notify the software if it cannot recover successfully. This activity is recorded in the form of one or more entries in the system event file.

2.2.2.1 **CPU and Memory Failures** - The first category is a failure occurring in the CPU and main storage section of the system. This type of failure is perhaps the most difficult to handle correctly. These failures can easily modify either the operating system software or a user program or cause instructions to be incorrectly executed. A failure in an addressing section can cause the system to operate with incorrect data or unknowingly modify some other job's program or data. For these reasons, CPU errors ordinarily cause the crash of a job or the entire system, depending on whether a user or the operating system is in control.

2.2.2.2 **Channel and Controller Failures** - The second category of hardware error entry is a channel or controller failure. The system controllers monitor and control several I/O devices of the same type, and the channels of various types connect the CPU and/or main storage units with the I/O controllers and devices. These errors are likely to affect several jobs or users because each controller or channel can handle several I/O devices being used by many jobs or processes. Detected errors are signalled to the CPU, and the operating system may

stop the current operation if the error is serious. An example is a controller's parity check of a command issued by the CPU. If this parity check fails, the command will not be performed, and the error will be signalled to the CPU. Such an event is recorded in the system event file for subsequent retrieval by SPEAR.

2.2.2.3 I/O Device Failures - The third category of hardware error entry is a failure of an I/O device. Errors detected by a single I/O device are recovered in the same manner as channel and controller failures but usually the error affects only one job or task. Some I/O failures are caused by faulty media. The most frequently used form of error recovery in this case is to retry the failing operation. If the failure continues for a specified number of consecutive retries, the job or task is crashed. Each failure is recorded in the system event file.

2.2.3 Performance Entries

The system event file contains more than just error entries. It also contains entries concerning day-to-day events of the system. These events vary depending on the operating system. But in general, you might find entries of the following nature:

1. System reloads
2. Tape and disk mounts/dismounts
3. Operator messages

These entries add another dimension to your environment. Keeping track of system performance can be a useful tool in preventive maintenance. The COMPUTE function, described in Chapter 5, also uses this type of entry to help derive system availability and effectiveness.

2.3 RECORDING EVENTS

The operating system continually detects and records events concerning every disk, tape, and memory parity error as they occur. The operating system:

1. Detects the event
2. Identifies the type of event
3. Associates it with a device
4. Gathers information about it
5. Records the date and time
6. Stores the information as an entry by appending it to the system event file
7. In some cases, tries to recover or find a way around the error

The system event file is a sequential file, therefore, each new entry is written to the end of the file. SPEAR can format these entries into an ASCII report with its RETRIEVE facility. Refer to Section 4.4 for information on RETRIEVE. The following section describes the template that each entry fills.

2.3.1 Record Format

Each entry in a TOPS-10 and TOPS-20 system event file is composed of two sections: a header section and a body section. The top section (contained in asterisks) of each entry report is the header section. It contains the following information:

1. The entry type
2. The time the entry was recorded
3. The operating system uptime at the time of the entry
4. The serial number of the CPU where the entry occurred
5. The record sequence number

The record sequence number is a number indicating the position of the entry in the file. SPEAR assigns the record sequence number to the entry when you decide to RETRIEVE it.

For each operating system, the format of the header is the same. The following is a sample of an entry header on TOPS-20 after it has been translated by SPEAR:

```
************************************************************
MASSBUS DEVICE ERROR
 LOGGED ON FRI 13 JUN 80 03:23:15 MONITOR UPTIME WAS 2:34:08
        DETECTED ON SYSTEM #2137.
        RECORD SEQUENCE NUMBER:   344.
************************************************************
```

On TOPS-10, if the system crashed and the entry has been copied from the CRASH.EXE file, the header states this fact at the top of the section. For example:

```
************************************************************
**THIS ENTRY COPIED FROM A SAVED CRASH**
                    .
                    .
                    .
                    .
************************************************************
```

Because the information was extracted from a saved crash instead of a running operating system, the date and time of the entry and the uptime listed in the header are the last values recorded by the operating system before it crashed. (Note that multiple entries extracted from a crash will have identical DATE, TIME, and UPTIME.)

The body section of the entry contains the various data items that make up the entry. The format of the header is constant regardless of the entry type but the body varies according to the type of entry. The amount of information that is reported in the body also varies depending on the format you specify to RETRIEVE. You can receive a SHORT version of an entry with only summary information or a FULL entry with all the information that is in the system event file. Refer to Section 4.4 for more information on the RETRIEVE function.

2.3.2 Record Conventions for Numbers and Dates

In the entries on TOPS-10 and TOPS-20, most numbers output by SPEAR are either decimal or octal. If SPEAR uses another numbering system, it is so noted on any report you request. Decimal values always contain a decimal point; all other values are octal. Values printed in half-word format have leading zeroes suppressed in each half of the word, and the halves are separated with a comma.

All register values that are translated to text, such as the CONI value, have text translations only for bits or bytes of interest, and the whole value is dumped. For example, the CONI value might include a DONE bit and a PI assignment, but these bits are not translated to text.

All dates and times printed by SPEAR are from your local time zone, for example EST, unless otherwise stated.

Refer to Chapter 5 for samples of entries that can appear in the system event file of your operating system.

CHAPTER 3

ANALYZING FAILURES

3.1 INTRODUCTION

The main reason for using SPEAR is to isolate the faults that are causing intermittent failure of the system. In case you are unaware of the various problems you can run into trying to find the cause of these failures, this chapter discusses:

1. The types of failures that can occur and what causes them.
2. The various error-checking schemes built into the system.
3. Some techniques to follow in isolating these failures.

3.2 TYPES OF FAILURES

A fault is a condition that causes a system component to fail to perform as expected. For example, such a condition could be a broken wire, a power supply fluctuation, or an unexpected interaction between two or more software routines. As a matter of course, the operating system records the symptoms of these occurrences in the system event file for later reference.

A fault is not necessarily noticeable until a failure occurs. A failure occurs only when a fault causes an adverse effect on system performance. The fault probably does not become apparent until a failure occurs. This is one reason for a system manager or system operator to use the COMPUTE function (Section 4.8) of SPEAR to check system performance.

You are likely to find several faults before you find the one that is causing the failure. Therefore, always confirm that the fault you corrected is indeed the one that is causing the failure. Refer to Section 3.4.1 for verification techniques.

You should also be on the lookout for changes in performance that may indicate an impending failure. By running SPEAR daily and keeping a record of its output, you could prevent a problem with the system.

There are two general categories of failures caused by faults. They are:

- Solid failures
- Intermittent failures

3.2.1 Characteristics of Solid Failures

A fault that affects the system in a permanent manner results in a solid failure. A solid failure is easier to solve than an intermittent failure.

Because the failure is solid; that is, reproducible, you have a basis by which to research, identify, and eliminate the cause of the failure.

3.2.2 Characteristics of Intermittent Failures

A fault that affects the system in a temporary manner can result in an intermittent failure. An intermittent failure is more difficult to solve than a solid failure. Something must be causing the failure to occur and something must be making it go away. The secret behind finding the cause of an intermittent failure is knowing that somehow, somewhere, something is changing the conditions under which the system is running. The changing conditions, in turn, make the problem intermittent.

For field service engineers: the next time you are working on a really tough intermittent problem (after checking the power supplies and ground system and running the appropriate diagnostics), try stepping back and thinking about the problem. Think about what the system is doing. Watch it for a while. See if you can identify the exact conditions at the time of the failure. Use SPEAR to watch the conditions of the system and check the events before and after they occur by checking the system event file.

If you can identify the conditions, then maybe you can reproduce them. If you can reproduce the conditions, then you have changed the intermittent failure into a solid failure. Although the approach to solving a solid failure is the same as the approach to solving an intermittent failure, in many cases, you will find that solving a solid failure is easier.

3.3 ERROR DETECTING AND ERROR CHECKING

The system has several means by which to check for errors in both the hardware and software. The hardware contains error-detection circuits, and the software contains error-checking routines. Both the detection circuits and checking routines serve a dual purpose: (1) to minimize the effects of a failure on overall system performance, (2) to help isolate the cause of a failure.

3.3.1 Hardware Error Detectors

There are three basic types of hardware error detectors in common use:

1. Threshold error detectors
2. Timing error detectors
3. Parity error detectors

Threshold error detectors monitor critical analog circuits, such as power supplies, servomechanisms, write current circuits, and temperature probes.

Timing error detectors monitor asynchronous events within the system, such as data requests to main memory or cache. The memory or cache must respond to the request within a certain amount of time. If it does not, the nonexistent-memory timing-error detector sets an error condition. Other asynchronous events that must be monitored for proper timing are: index and sector pulses, disk and tape up-to-speed operations, and internal and external clocks.

Parity error detectors monitor the transfer of information. The parity generator adds one or more extra bits to the information being transferred to satisfy a particular parity algorithm. For example, in the case of the single-bit odd parity, the information is in the form of ones and zeros, the extra parity bit assures that the total number of one bits in the transfer is odd. The parity error detector monitors each transfer. Should a transfer ever contain an even number of one bits, the parity error detector raises a parity error condition. Note that in some cases, two bits can be dropped leaving odd parity. However, this is an undetectable error condition.

Once any one of these detectors detects an error condition, the operating system records the information as an entry in the system event file. These are the kinds of events you will be looking for when using the SPEAR library.

3.3.2 Software Error Checking

There are four types of software error checking routines in common use:

1. Range checking
2. Validity checking
3. Sum checking (checksum)
4. Loop checking

A range checking routine verifies that the arguments supplied to a routine fall between two known values.

A validity checking routine verifies that a routine written to accept only certain arguments indeed accepts only those arguments. Any other response causes an error condition.

A sum checking routine (checksum) checks file storage. When the monitor assembles a group of blocks to write contiguously on the disk, it checksums the first word of that group and saves that checksum in the retrieval information block (RIB). If, when read back, that checksum does not match the first word; the monitor assumes it read the wrong block. If there are no hardware errors, this is the best assumption. These errors probably indicate a disk addressing failure.

If the monitor crashes before it is able to write the new RIB of an old file, the checksum may change in core but not on disk. An obscure software problem may also be responsible. Reproducing the error is one way for you to narrow the problem down. Also check the crash log and look for other error types.

Note that a checksum error is not a substitute for parity. Its purpose is to make sure that a data set was written in the right place. If it was not, either the software failed to keep track of the data, or the hardware failed to address the correct place.

A loop checking routine keeps count of the number of times a program entered a loop and reports an error when a maximum count is reached, indicating that the loop is unable to reach a decision.

Any time one of these error conditions is set, the operating system records the event in the system event file. You can check on these events by using the SPEAR library.

3.4 ISOLATION TECHNIQUES

When you are faced with the problem of finding the cause of an intermittent failure, you should take the time to define the problem. First check the symptoms:

1. What is happening that should?
2. What is happening that should not?
3. What are the conditions and circumstances?

As you probably know, here are some possible causes of intermittent failures:

1. An environmental violation (power, grounding, temperature, humidity, contamination)
2. A damaged, defective, or worn component
3. A faulty mechanical or electrical connection
4. A mechanical misalignment
5. An electrical misadjustment
6. A software design oversight
7. A hardware design oversight

What you have to work with are the symptoms of the failure and the SPEAR library of functions. Hopefully, the system operator has been running SPEAR analysis on a daily basis so that you can get a picture of the conditions leading up to the problem. If not, you can run SPEAR and receive a report within a short period of time. With SPEAR analysis and reported symptoms, you should be able to venture a guess as to the cause of the problem. You might even be able to pinpoint the failure right away. If you are not that fortunate, your next plan of action is to do the following:

1. Devise an experiment
2. Predict the results
3. Conduct the experiment

4. Evaluate the results

5. Refine the experiment

6. Repeat the process

For example, if you suspect that a disk pack is bad, move the pack to another disk drive. If the media is bad, the error pattern will move to the other drive. Once you believe you have isolated the failure, you should confirm your findings. After moving the disk pack, run the system for a couple of days. Then run SPEAR analysis. Check to see if the same error patterns occur on the second drive.

3.4.1 Verification

There are two general methods of verifying your findings. The first method is to reinsert the problem. If the symptoms recur, you can be relatively sure that you have identified the cause of the problem, thereby verifying your findings. If the symptoms do not recur, you should proceed with the second method.

The second method is called the time window. You should use the time window for intermittent problems or when reinserting the probable cause is not feasible; that is, when reinserting would be too time consuming or potentially damaging to the system.

The time window is simply a period of time during which you closely monitor the performance of the system. If the problem does not recur during that period, then you assume the problem is solved, and your findings are verified.

The duration of the time window depends on whether the problem was solid or intermittent. If the problem was solid, then monitor the system for 24 hours. If the problem was intermittent, wait at least three times as long as the frequency of the error. Experience will dictate the method that works best for you.

Your site may have its own specific isolation and verification techniques that are tried and true. If so, stay with the most successful method.

CHAPTER 4

THE SPEAR LIBRARY

4.1 INTRODUCTION

The previous chapters introduced you to SPEAR, described where SPEAR gets its information, and listed techniques for intermittent fault isolation. This chapter explains how to use the SPEAR dialogue with its help facilities and describes the following six functions in the SPEAR library:

- INSTRUCT
- RETRIEVE
- KLERR
- SUMMARIZE
- KLSTAT (TOPS-2Ø only)
- COMPUTE

SPEAR is set up in such a way that after you use it a number of times you can run through it without any problems. The reason for its ease of use is the way you interact with SPEAR. SPEAR has a dialogue that prompts and helps you along as much as you want.

4.2 RUNNING SPEAR

To run SPEAR, first log in to your operating system, then type one of the following:

```
.R SPEAR        On TOPS-1Ø based systems

@SPEAR          On TOPS-2Ø based systems
```

SPEAR indicates that it is waiting for instructions by displaying the following prompt:

```
SPEAR>
```

After you see the SPEAR prompt, you can type any one of the function names, (you can type KLSTAT on TOPS-2Ø only) or type HELP or question mark, or EXIT back to operating system command level. If you type a function name, you need only specify enough characters to make it unique to SPEAR. In this case, you need type only the first character of the name for SPEAR to recognize it.

If you type a question mark (?) at this point, SPEAR prints a list of the features available to you in your version of the SPEAR Library.

CAUTION

> The SPEAR library is not transportable across operating systems. You cannot run SPEAR for TOPS-10 on TOPS-20 and so on. Consequently, you cannot use the system event file from one operating system with a SPEAR library from another system.

SPEAR has several features to guide you in its use. The following subsections describe these features.

4.2.1 Prompts, Responses, and Arguments

Each function of SPEAR has several levels of questions for you to answer. SPEAR prompts you and gives you a selection of acceptable responses. The default is listed in parentheses with each prompt.

If you have been through this before, you can speed up the process by responding to all the prompts on the first line, using legal separators, or by specifying an indirect file containing your responses.

SPEAR can process commands from a disk file as well as from your terminal. This disk file, known as an indirect file, is useful if you have a set of responses you often use. To use this function, create a disk file while at operating system command level with a text editor. The file should contain responses that you would normally type to SPEAR on the terminal.

NOTE

> Be sure to delete any line-sequence numbers from your indirect file. SPEAR will not accept them.

Once you have created the file and saved it in your disk area, all you need to do is to run SPEAR and type the file name preceded by an at sign (@). The at sign (@) signifies an indirect file. The default file name for an indirect file is SPRCMD.CMD. Note that you can specify an indirect file at any prompt level of SPEAR, as long as the file contains only the remaining information necessary to complete the SPEAR requests.

You can choose to be prompted at every step or decide to supply all required information without prompting. In fact, at SPEAR command level, you can input an entire SPEAR session on one line, separating each field with a space. For example:

```
SPEAR>RETRIEVE A0916.PAK 5,6,10 ASCII FULL /G (RET)
```

By using special characters as separators, you can also speed up the process within the SPEAR dialogue. Section 4.2.2 describes these characters.

4.2.2 Separators and Terminators

The following characters and terminal keys have special meaning to SPEAR:

1. The RETURN key (RET) - indicates that you have completed input to a SPEAR prompt in one way or another. You have either input your own arguments or taken the default.

2. A comma (,) - indicates that you are inputting a list of items within one request for input, for example a list of sequence numbers or packet identifiers.

3. A colon (:) - indicates that you have either input a device name within a file specification or you have specified devices within an error type specification.

4. A plus sign (+) - separates more than one major error type on one line.

5. A semicolon (;) - indicates that the next argument is a version number in a file specification.

6. An exclamation point (!) - allows you to insert comments. SPEAR ignores anything it sees on the current line after an exclamation point.

4.2.3 Help Features

There are five major help features in SPEAR, the question mark (?), the HELP command, the @HELP command, the question mark switch (/?), and the /HELP switch.

1. The question mark (?) provides enough information to refresh your memory about the acceptable responses.

2. The HELP command provides detailed information on both the prompt and on acceptable commands.

3. The @HELP command displays information concerning indirect files.

4. The question mark switch (/?) provides a list of switches you can type as response to a particular prompt.

5. The /HELP switch provides an explanation of the acceptable switches that you can type as response to a particular prompt.

You can type any of these help features after any prompt in the SPEAR dialogue and also after you have typed a response to the prompt. For example, if you type a question mark in response to a prompt, SPEAR does the following:

1. Lists all acceptable responses.

2. Gives a brief description of the desired response if it is general (for example, file specification).

If you type a question mark after supplying characters to a prompt, SPEAR lists all acceptable responses matching the characters typed.

You can also type the HELP command after any prompt. SPEAR prints up to 22 lines of information about the use of the prompt.

The Escape key is another help feature in the SPEAR library. The Escape key fills in a response if you type enough characters for SPEAR to know what you want. For example:

Output mode (ASCII):B (ESC) INARY

If you do not supply enough information before typing (ESC), SPEAR prompts you for more input by sending a bell to the terminal. If you press <ESC> without typing any characters in response to a prompt, SPEAR fills in the default response. For example:

Event file (SERR:ERROR.SYS): (ESC) SERR:ERROR.SYS

The following keys can also help you through the SPEAR dialogue:

1. CTRL/U - deletes the current input line
2. CTRL/W - deletes back to the last punctuation character
3. CTRL/F - completes the next field of a file specification with the default

4.2.4 File Specifications

The following are the formats of the file specifications that can be given in a SPEAR command string. These formats are listed according to operating system:

TOPS-10	dev:filename.file extension[directory]
TOPS-20	dev:<directory>filename.file type.file version

4.2.5 SPEAR Switches

The following is a list of the switches available in SPEAR. Note that the square brackets indicate optional information that you can omit. You do not type the square brackets.

/?	lists the available switches.
/B[REAK]	returns you to the SPEAR> prompt.
/G[O]	executes the current SPEAR command with the parameters you have given so far. It takes the defaults for the rest of the parameters. This is the default switch.
/H[ELP]	lists the available switches and gives a brief explanation of their uses.
/R[EVERSE]	returns you one level back to the previous prompt, where you can change any parameters.
/S[HOW]	shows all the parameters you have specified so far and fills in the defaults for the ones you have not specified.

The following is an example (from TOPS-1Ø) using the /SHOW switch with the RETRIEVE and SUMMARIZE commands. Note that all the defaults are shown because no other parameters have been specified.

```
SPEAR> SUMMARIZE/SHOW


Event file: SYS:ERROR.SYS
Report to: DSK:SUMMAR.RPT
Time from: 8-Mar-85
Time to: LATEST
Show Error Distribution: YES


SPEAR> RETRIEVE/SHOW


Event or packet file: SYS:ERROR.SYS
Output to: DSK:RETRIE.RPT
Merge with: NONE
Time from: EARLIEST
Time to: LATEST
Selection to be: INCLUDED
Output mode: ASCII
Report format: SHORT
Selection type: ALL

SPEAR> RETRIEVE/REVERSE

SPEAR> EXIT
.
```

4.2.6 Exiting from SPEAR

To exit from SPEAR, first return to the SPEAR> prompt by typing /BREAK. Then type the EXIT command. You can also exit from SPEAR by typing CONTROL/C at any prompt.

4.3 INSTRUCT

INSTRUCT is a computer-aided instruction program that explains how to use the SPEAR library. You can use INSTRUCT as a course on how to use SPEAR, or as a reference to a particular piece of information on the SPEAR library.

The SPEAR (CAI) course consists of four main modules:

1. Fault Isolation Techniques - This module describes the nature of intermittent faults and discusses some of the most common methods used to isolate intermittent system and subsystem failures.

2. System Event File Organization and Content - This module describes the overall organization and content of TOPS-1Ø and TOPS-2Ø system event files.

3. SPEAR Library Functions - This module explains how to use each of the SPEAR maintenance functions: RETRIEVE, KLERR, COMPUTE, SUMMARIZE, and KLSTAT.

4. Guaranteed Uptime Program - This module explains how to use the NOTIFY program to measure system uptime.

Each module consists of an introduction and a menu of subordinate topics. When appropriate, the subordinate topics are also broken down into introduction and menus. Thus, you can use INSTRUCT as either a tutorial or a reference.

INSTRUCT is frame-oriented, that is, it displays one frame of information at a time. Thus, you can study each frame for as long as you like. Then, when you are ready, you can proceed to the next frame by pressing the RETURN key.

To use INSTRUCT as a tutorial, refer to Section 4.3.1. To use INSTRUCT as a reference, refer to Section 4.3.2.

4.3.1 Setting Up a Student ID

To access INSTRUCT right now, do the following:

Log in to your operating system.

Run SPEAR - .R SPEAR (TOPS-10)
@SPEAR (TOPS-20)

To begin the teaching session, type:

@SPEAR>INSTRUCT (RET)

This response places you at the beginning of the course. First INSTRUCT displays an overview of the SPEAR library. You must press the RETURN key to see the next frame of information. INSTRUCT then gives you an introduction to the course. If there is no instruction or question to answer at the bottom of the screen, press the RETURN key to see the next frame of information. After the explanation of common responses, you will be asked if you want to establish a student identification number:

Badge number (REFERENCE):

If you want to establish an ID, enter an alphanumeric string; something you are not likely to forget. Then press the RETURN key. From this point on, INSTRUCT will keep track of where you are in the course.

After you have established your Student ID, you can leave INSTRUCT any time you want by typing /B. When you return, type your ID in response to the SPEAR prompt:

@SPEAR>INSTRUCT ID n (RET)

where

n is your Student ID.

INSTRUCT will return you to the exact location where you typed the break switch, /B.

4.3.2 Using INSTRUCT as a Reference Tool

The quickest way to access the INSTRUCT menus is by typing the following:

```
SPEAR> i i r/g (RET)
```

where

The first i represents INSTRUCT.

The second i represents ID.

The r is for REFERENCE.

The /g is for /GO.

INSTRUCT responds with the following menu:

```
                Spear Course Menu

 1. Course Administrator/Student Guide

 2. Troubleshooting

 3. System Event Files

 4. Using The Spear Library

 5. Guaranteed Uptime Program

 6.  Feedback

 7.  Random Questions

 8.  Dialog Changes

Your selection please (#)>
```

At this point enter one of the numbers or letters in the menu and press the RETURN key.

The Course Administrator's Guide gives a brief description of how to administer the course along with a sample answer sheet. The Troubleshooting section gives some tips on how to approach the problem of isolating intermittent system faults. The System Event File section is a question and answer session concerning that topic. Using the SPEAR Library is a combination of information and questions and answers. The Guaranteed Uptime Program explains how to use the NOTIFY program with the COMPUTE function of SPEAR to measure system uptime.

The Feedback section is a request for your opinion of the SPEAR Library. The Random Questions section gives you another opportunity to test your knowledge of SPEAR. The section, Using the SPEAR Manual, describes the use of the SPEAR manual with the SPEAR program.

Remember to press the RETURN key after each frame of information, unless instructed otherwise.

4.4 RETRIEVE

RETRIEVE provides a means by which to convert the entries in the system event file from internal binary format to a readable ASCII format. It also allows you to select specific entries from the system event file and save them in a separate file.

4.4.1 RETRIEVE Input

RETRIEVE accepts the following types of input:

1. The system event file
2. A file created by the RETRIEVE process
3. Any file containing entries from the system event file

With RETRIEVE, you have the option of translating the entire system event file or specific entries in the file by sequence number. In order to have more control over the selection of specific types of entries, you can use RETRIEVE to extract the entry types in which you are interested and then translate them.

You can select entries on the basis of the following:

1. Date/time limits
2. Sequence numbers
3. Event codes
4. Error
5. Statistics
6. Configuration
7. Diagnostics

Error, Statistics, Configuration and Diagnostics can be further subdivided into the following categories:

1. Mainframe (CPU, memory, front-end)
2. Disk
3. Tape
4. CI
5. NI
6. Unit record
7. Network
8. Operating system
9. Disk pack identifier
10. Tape reel identifier

Once you have defined a category, you can specify physical names or device types within a class, such as LPT for unit record device. Table 4-1 lists the available device types that you can specify.

Table 4-1: Device Types

Category	Device Types
Mainframe	ALL, MEM, FE, CPU
Disk	ALL, RM03, RM05, RP04, RP05, RP06, RP07, RS04, RP20, RA60, RA80, RA81
Tape	ALL, TU16, TU45, TU70, TU71, TU72, TU73, TU77, TU78, TA78
CI	CI20, HSC50
NI	NIA20, ALL
Unit Record	ALL, LPT, CDR
Network	ALL, Decimal number in range 0-511 (see Table 4-2)

Table 4-2 lists the classes available for selection of DECnet events.

Table 4-2: Network Event Classes

Class	Description
0	Management layer
1	Application layer
2	Session Control layer
3	Network services layer
4	Transport layer
5	Data link layer
6	Physical link layer
007-031	Reserved for other common event classes
032-063	Reserved for RSTS specific event classes
064-095	Reserved for RSX specific event classes
096-127	Reserved for TOPS-20 specific event classes
128-159	Reserved for VMS specific event classes
160-191	Reserved for RT specific event classes
192-479	Reserved for future use
480-511	Reserved for Customer specific event classes

For more information concerning network entries from DECnet, refer to the DECnet documentation for system managers and operators.

If you specify Error as an entry selection, you can also specify an error type. See Table 4-4 for a list of error types.

4.4.2 RETRIEVE Output

RETRIEVE output can be in the following forms:

1. One or two lines containing the most pertinent data in ASCII format.

2. All data about each event, in ASCII format.

3. All data about each event in octal dump format. This format is useful only for debugging the error-reporting system.

4. Specific events saved in binary format, for future reference.

Your default output can be an ASCII file, RETRIE.RPT, or a binary file, RETRIE.SYS.

You should be aware that user-defined entries that are unknown to SPEAR cannot be translated into ASCII. You can, however, get an octal dump of these entries by specifying OCTAL to the Output Mode prompt when running RETRIEVE.

An unusual event you may find in the system event file is a KLERR entry. The KLERR entries are different from most entries in that it takes several event file records to make up one complete entry. This is because the front-end must send information in pieces through the DTE interface along with all communications, console, and hard-copy data. Because of this, there is a chance that not all records will actually get through to the event file. When SPEAR sees that a KLERR entry is incomplete, it will type an error message (non-fatal) and will translate all available data anyway.

Each KLERR entry uses one sequence number. When looking at a RETRIEVE report, you may notice gaps between sequence numbers even if you have selected ALL entries. A KLERR entry is listed using the sequence number of the first record in the entry, but it is not listed until all records of the entry have been received. Because other entries may enter the event file before the front-end has sent all records of one KLERR entry, the KLERR entry will appear to be out of sequence. For example, you may find entries with the following sequence numbers:

1. Configuration status change

3. Disk error

6. Tape error

2. KLERR

8. Reload

You can translate the KLERR entry into its components by using the KLERR function. See Section 4.5 for details.

For step-by-step procedures for using RETRIEVE, refer to Section 4.4.3.

4.4.3 RETRIEVE Procedure

RETRIEVE allows you the option of converting events in the system event file into an ASCII format for listing on the terminal or lineprinter. To begin with, RETRIEVE prompts with one or more of the following guidewords:

```
RETRIEVE Mode
-------------

        Event or packet file(SERR:ERROR.SYS):

        Packet numbers:

        Selection to be (INCLUDED):

        Selection type (ALL):

        Sequence numbers:

        Event codes:

        Category (ALL):

        Next category (FINISHED):

        Mainframe devices (ALL):

        Disk drives (ALL):

        Tape drives (ALL):

        CI controller (ALL):

        Unit record devices (ALL):

        Disk (structure IDs):

        Tape (reel IDs):

        Time from (EARLIEST):

        Time to (LATEST):

        Output mode (ASCII):

        Merge with (NONE):

        Report format (SHORT):

        Output to (DSK:RETRIE.RPT):
```

4.4.3.1 Retrieving Selected Events - If you want to take all the defaults, type R/G to the SPEAR> prompt; otherwise, read the following procedure.

➡ STEP 1

After typing RETRIEVE to the SPEAR> prompt, you are asked for the name of the input file:

```
Event or packet file (SERR:ERROR.SYS):        TOPS-20

                    or

Event or packet file (SYS:ERROR.SYS):         TOPS-10
```

Type one of the following:

1. The RETURN key - to select the default, the system event file.
2. Any file name, in the proper format, containing events stored in binary.
3. The name of a previous file that you RETRIEVEd in BINARY mode.

➡ STEP 2

RETRIEVE then prompts for the method of selection:

```
Selection to be (INCLUDED):
```

Type one of the following:

1. The RETURN key - to select the default I[NCLUDED]. INCLUDED moves a few selected entries of various types into a separate file.
2. E[XCLUDED] - to select all but a few entry types.

➡ STEP 3

After selecting INCLUDED or EXCLUDED, you receive the following prompt:

```
Selection type (ALL):
```

At this prompt, you have two separate lists from which to choose. Type one or more of the following from the first group:

1. E[RROR] - to select entries that contain actual failure data.
2. ST[ATISTICS] - to select statistic entries.
3. D[IAGNOSTICS] - to select entries created by a diagnostic.
4. CON[FIGURATION] - to select configuration entries.
5. O[THER] - to select entries that do not fit into the other types.

If you choose more than one of these types, separate each with a comma.

Or type one of the following from the second group:

1. The RETURN key or A[LL] - to select the default that extracts all entries. You will be asked for date and time limits next.

2. SE[QUENCE] - to select entries by sequence number.

   If you choose SEQUENCE, RETRIEVE prompts further with:

   Sequence numbers:

   Here you can specify one number, several numbers separated by commas, or a range of numbers separated by a hyphen.

3. COD[E] - to select entries on the basis of their octal code number. These numbers are listed in Table D-1 and in the SPEAR Reference card.

   If you choose CODE, RETRIEVE prompts you further with:

   Event codes:

   Here you can specify one number, several numbers separated by commas, or a range of numbers separated by a hyphen.

If you chose ERROR, STATISTICS, CONFIGURATION, OTHER, or a combination of these, proceed with Step 3A. If you chose ALL or CODE, proceed to Step 4. If you chose SEQUENCE proceed to Step 6.

➡STEP 3A

If you choose ERROR, STATISTICS, CONFIGURATION, OTHER, or a combination of these types, you receive the following prompt:

Category (ALL):

Type one of the following:

1. The RETURN key or A[LL] - to select all the categories. This is the default.

2. M[AINFRAME] - to select errors occurring in specific mainframe components.

3. D[ISK] - to select entries occurring on disk subsystems or individual drives.

4. T[APE] - to select entries occurring on tape subsystems or individual drives.

5. CI - to select entries occurring on the CI interconnect or the HSC5Ø disk controller.

6. NI - to select entries occurring on the NI.

7. U[NITRECORD] - to select entries occurring on unit-record devices such as card readers and line printers.

8. NE[TWORK] - to select entries occurring on the network nodes.

9. O[PERATING-SYSTEM] - to select entries that are software related.

10. CO[MM] - to select entries occurring on communications devices.

11. P[ACKID] - to select entries occurring on specific disk packs.

12. R[EELID] - to select entries occurring on specific tape reels.

All categories except COMM and NI, prompt further for specific device types. Table 4-3 lists the subprompts you can expect.

Table 4-3: Subprompts for Device Types

Device Type	Subprompt
MAINFRAME	Mainframe devices (ALL):
DISK	Disk drives (ALL):
TAPE	Tape drives (ALL):
CI	CI controllers (ALL):
UNITRECORD	Unit record devices (ALL):
NETWORK	Event class and type (ALL):
OPERATING-SYSTEM	Operating System codes (ALL):
PACKID	Disk (structure IDs):
REELID	Tape (reel IDs):

Type ? at the subprompt level to get a list of acceptable responses, or refer to Table 4-1 in this manual.

If you chose ERROR as one of the selection types in STEP 3, you can also specify the particular error types for which you are looking in relation to the specific device. Table 4-4 lists the error types for the devices.

Table 4-4: Error Types

Prompts	Error Types
Disk error type (ALL):	OFFLINE WRITE-LOCK UNSAFE MICROPROCESSOR SOFTWARE BUS CHANNEL-CONTROLLER READ-WRITE SEEK-SEARCH TIMING OTHER
Tape error type (ALL):	READ WRITE DEVICE-FORMATTER BUS CHANNEL-CONTROLLER SOFTWARE OFFLINE OPERATOR OTHER
CI error type (ALL): for CI2Ø	EBUS MBUS CRAM-PARITY CHANNEL-ERROR SERDES-OVERRUN EDS INCONSISTENT-DATA
CI error type (ALL): for HSC5Ø	SERDES-OVERRUN EDC INCONSISTENT-DATA
NI error type (ALL):	EBUS MBUS CRAM-PARITY CHANNEL-ERROR

➡STEP 3B

RETRIEVE keeps prompting you for categories until you either type FINISHED or press the RETURN key:

```
Next category (FINISHED):
```

Type one of the following:

1. The RETURN key or F[INISHED] to take the default.
2. Another category.

Note that you can select disk entries by either DISK or PACKID and tape entries by either TAPE or REELID. If you are interested in media, use PACKID or REELID; otherwise, use DISK or TAPE. If you specify both DISK and PACKID (or TAPE and REELID), you select all disk entries (or tape entries), not just those that match the selected media. If you want to select entries with a specific device and media, you must run RETRIEVE twice.

You can specify more than one device name by separating them with commas. For example:

```
Disk drives (ALL):DISK:RP06,RM03,RP05
```

You can always come back to error category selection (by using /REVERSE) to add parameters. Everything typed here remains until you type CTRL/U or CTRL/W.

Note that supplying a device type (RP06, RM03) causes SPEAR to search a different field than if you had supplied a physical name (DP130, MTA1, and so forth). If the name you supply does not match one of the known device types, SPEAR assumes that it is a physical name.

➡ STEP 4

RETRIEVE then prompts you for the date and time limits of the entries you want to select:

```
Time from (EARLIEST):
```

Type one of the following:

1. The RETURN key or E[ARLIEST] - to select the beginning of the file. This is the default.

2. A date and time in the format dd-mmm-yy hh:mm:ss - to signify where to begin extracting entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -7 causes RETRIEVE to begin extracting entries from seven days prior to the current day.

➡ STEP 5

RETRIEVE then prompts for the end of the time period:

```
Time to (LATEST):
```

Type one of the following:

1. The RETURN key or L[ATEST] - to select the end of the file. This is the default.

2. A date and time in the format dd-mmm-yy hh:mm:ss - to indicate the last date for extracted entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -13 causes RETRIEVE to stop extracting entries recorded thirteen days before the current date.

➡STEP 6

RETRIEVE next prompts for style of output:

Output mode (ASCII):

Type one of the following:

1. The RETURN key or A[SCII] - to convert entries into ASCII format. This is the default.

2. B[INARY] - to retain the entries in their internal format.

If you choose ASCII, proceed to STEP 7. If you choose BINARY, skip to STEP 8.

➡STEP 7

After choosing ASCII, RETRIEVE prompts you for the form of your output:

Report format (SHORT):

Type one of the following:

1. The RETURN key or S[HORT] - to select the default. This selection produces a report with only the most essential information. No entry will be longer than three lines of 72 columns.

2. F[ULL] - to display all the information that the operating system recorded for that entry.

3. O[CTAL] - to produce a ones and zeros ASCII report. The ones and zeros represent the actual binary contents of the entry. Unless you are familiar with the internal format of the individual entries, this format has very little value. Its primary purpose is to aid in debugging the SPEAR program library.

➡STEP 8

If you specified BINARY as output style, RETRIEVE then prompts for another file name to give you an opportunity to combine two files into one for record-keeping purposes. The merged output file will be in the proper chronological order. Both files must be in binary format. The prompt is:

Merge with (NONE):

Type one of the following:

1. The RETURN key - to select the default of NONE.

2. A file name of another file containing entries from the system event file.

➡ STEP 9

The last thing RETRIEVE asks for is the destination of the output. If you chose ASCII, the prompt is:

```
Output to (DSK:RETRIE.RPT):
```

If you chose BINARY, the prompt is:

```
Output to (DSK:RETRIE.SYS):
```

Type one of the following:

1. The RETURN key - to select the default RETRIE.RPT or RETRIE.SYS.

2. TTY: - to direct ASCII formatted output to the terminal. You should not request BINARY formatted output to be printed on the terminal.

3. Any file name in the proper format for your system.

After you select the output destination and press RETURN, SPEAR asks you to confirm your decision:

```
Type <cr> to confirm (/GO):
```

At this point, you can:

1. Press RETURN or type /GO to execute the RETRIEVE process.

2. Type /SHOW to list the parameters you have chosen.

3. Type /REVERSE to return to the previous prompt.

4. Type /BREAK to return to SPEAR> level.

5. Type question mark (?), HELP, the question mark switch (/?), or /HELP to find out what your options are.

If your output is formatted in ASCII and you decide to output the file to your disk area, you can list the file on the lineprinter by doing the following:

Return to operating system command level by typing EXIT to the SPEAR> prompt.

Use the PRINT command with any options available on your operating system.

4.4.3.2 Sample RETRIEVE Session - The following is a sample RETRIEVE session using the TOPS-20 system event file for input:

```
@spear

Welcome to SPEAR for TOPS-20. Version 2(605)
Type "?" for help.


SPEAR> retrieve

RETRIEVE mode
-------------
    Event or packet file (SERR:ERROR.SYS):

    Selection to be (INCLUDED):

    Selection type (ALL): error,diagnostic

        Category (ALL): disk

            Disk drives (ALL): RP07

                Disk error type (ALL): ?

                  One or more of the following:
                  ALL
                  OFFLINE
                  WRITE-LOCK
                  UNSAFE
                  MICROPROCESSOR
                  SOFTWARE
                  BUS
                  CHANNEL-CONTROLLER
                  READ-WRITE
                  SEEK-SEARCH
                  TIMING
                  OTHER
                  HELP

                Disk error type (ALL): read-write

        Next Category (FINISHED):

    Time from (EARLIEST):

    Time to (LATEST):

    Output mode (ASCII):

        Report format (SHORT): full

    Output to (DSK:RETRIE.RPT):

Type <cr> to confirm (/GO):
```

4.4.3.3 **Short Format** - The following is a sample of a RETRIEVE report in short format:

```
@ty retrie.RPT

SPEAR Version 2(565). Retrieval from SERR:ERROR.SYS
  Report generated  6-Mar-84 15:57:46-EST
  As directed by user
  Selected window: 23-Feb-84 00:00:01-EST to 26-Feb-84 00:00:01-EST.
  Selected records are included
  Selection type is ERRORS,
  Report sent to DSK:RETRIE.RPT


SEQ    TIME    Thu 23 Feb 84

1249. 03:12:43 DP100 WORK: RP07 SERIAL #2861. CONI RH= 0,222715
                     CHN STS= 540100,174632 SR= 0,51700 ER= 0,100000
                     CYL/SURF/SEC= 212./27./3.
1713. 08:15:49 DP040 RP06 SERIAL #0125. CONI RH= 0,202615
                     CHN STS= 500000,305600 SR= 0,51700 ER= 0,100000
                     CYL/SURF/SEC= 0./0./1.
1875. 11:26:39 DP000 SERR: RP06 SERIAL #0941. CONI RH= 0,222615
                     CHN STS= 540100,174024 SR= 0,51700 ER= 0,100000
                     CYL/SURF/SEC= 603./10./16.

SEQ    TIME    Fri 24 Feb 84

328. 13:14:20 DP010 PUBLIC: RP06 SERIAL #0484. CONI RH= 0,222615
                     CHN STS= 540100,174066 SR= 0,51700 ER= 0,100000
                     CYL/SURF/SEC= 93./12./0.
372. 17:04:09 DP000 SERR: RP06 SERIAL #0941. CONI RH= 0,222615
                     CHN STS= 540100,174024 SR= 0,51700 ER= 0,100000
                     CYL/SURF/SEC= 361./15./16.

SEQ    TIME    Sat 25 Feb 84

85. 10:43:36 DP110 GALAXY: RP07 SERIAL #251D. CONI RH= 0,322615
                     CHN STS= 540100,174632 SR= 0,51700 ER= 0,400
                     CYL/SURF/SEC= 623./15./35.
```

4.4.3.4 Octal Format - The following is a sample of a RETRIEVE report in octal format.

```
SPEAR Version 2(565). Retrieval from SERR:ERROR.SYS
  Report generated  6-Mar-84 16:08:12-EST
  As directed by user
  Selected window: 23-Feb-84 00:00:01-EST to 26-Feb-84 00:00:01-EST.
  Selected records are included
  Selection type is ERRORS,
  Report sent to DSK:RETRIE.OCTAL


Sequence # 1249 -- Record HEADER:
0/      111001,,125124
1/      131271,,257140
2/      0,,116617
3/      0,,5467
4/      0,,2341

Record BODY:
0/      0,,0
1/      675762,,530000
2/      1242,,440147
3/      1,,74014
4/      100000,,1
5/      0,,222715
6/      0,,2415
7/      0,,35624
10/     1,,234156
11/     0,,172464
12/     0,,0
13/     0,,0
14/     0,,0
15/     732200,,177471
16/     732200,,177471
17/     720000,,15403
20/     720000,,15403
21/     0,,715652
22/     600001,,0
23/     0,,1
24/     0,,0
25/     0,,0
26/     0,,0
27/     0,,324
30/     0,,2214

                .
                .
                .
```

```
Sequence # 1713 -- Record HEADER:
0/      111001,,125124
1/      131271,,432751
2/      0,,272430
3/      0,,5467
4/      0,,3261

Record BODY:
0/      0,,0
1/      0,,0
2/      1242,,440146
3/      0,,1
4/      100000,,1
5/      0,,202615
6/      0,,2415
7/      0,,0
10/     0,,466
11/     0,,0
12/     0,,0
13/     0,,0
14/     0,,0
15/     732204,,177771
16/     732204,,177771
17/     720004,,1
20/     720004,,1
21/     0,,715436
22/     200001,,0
23/     0,,1
24/     0,,0
25/     0,,0
26/     0,,0
27/     0,,0
30/     0,,1

                .
                .
                .
```

4.4.3.5 Full Format - The following is an example of a full format:

RETRIEVE SESSION

```
SPEAR Version 2(565). Retrieval from SERR:ERROR.SYS
  Report generated  6-Mar-84 16:02:31-EST
  As directed by user
  Selected window: 23-Feb-84 00:00:01-EST to 26-Feb-84 00:00:01-EST.
  Selected records are included
  Selection type is ERRORS,
  Report sent to DSK:RETRIE.FULL



**********************************************
MASSBUS DEVICE ERROR
 LOGGED ON Thu 23 Feb 84 03:12:43      MONITOR UPTIME WAS  3:41:34
        DETECTED ON SYSTEM # 2871.
        RECORD SEQUENCE NUMBER: 1249.
**********************************************
        UNIT NAME:       DP100
        UNIT TYPE:       RP07
        UNIT SERIAL #:   2861.
        VOLUME ID:       WORK
        LBN AT START OF XFER:            1074014  =
        CYL:   212.      SURF:   27.     SECT:   3.
        OPERATION AT ERROR:       DEV.AVAIL., GO +  READ DATA(70)
        FINAL ERROR STATUS:      100000,1
        RETRIES PERFORMED:       2.
        ERROR:  RECOVERABLE
DRIVE EXCEPTION,CHN ERROR, IN CONTROLLER CONI
                DCK, IN DEVICE ERROR REGISTER

CONTROLLER INFORMATION:
CONTROLLER:      RH20 # 1
CONI AT ERROR:   0,222715 =
        DRIVE EXCEPTION,CHN ERROR,
CONI AT END:     0,2415 =
         NO ERROR BITS DETECTED
        DATAI PTCR AT ERROR:     732200,177471
        DATAI PTCR AT END:       732200,177471
        DATAI PBAR AT ERROR:     720000,15403
        DATAI PBAR AT END:       720000,15403

CHANNEL INFORMATION:
CHAN STATUS WD 0:        200000,174567
        CW1:  0,0  CW2:  0,0
CHN STATUS WD 1:         540100,174632 =
       NOT SBUS ERR,NOT WC = 0,LONG WC ERR,
CHN STATUS WD 2:         614005,377200
```

```
DEVICE REGISTER INFORMATION:
        AT ERROR          AT END          DIFF.
CR(ØØ): 4Ø7Ø              4Ø7Ø               Ø
         DEV.AVAIL., READ DATA(7Ø)
SR(Ø1): 51ØØ              11ØØ            4ØØØØ
        ERR,MOL,PGM,DPR,DRY,VV,
ER(Ø2): 1ØØØØØ               Ø            1ØØØØØ
        DCK,
MR(Ø3):    Ø                 Ø               Ø
AS(Ø4):    Ø                 Ø               Ø
DA(Ø5): 15404             15407            3
        D. TRK = 33, D.SECT. = 4
DT(Ø6): 24Ø42             24Ø42              Ø
LA(Ø7): 17ØØ              7ØØ              1ØØØ
SN(1Ø): 24141             24141              Ø
OF(11):    Ø                 Ø               Ø
DC(12): 324               324                Ø
        212.
CC(13): 324               324                Ø
        212.
E2(14):    Ø                 Ø               Ø
        NO ERROR BITS DETECTED
E3(15):    Ø                 Ø               Ø
        NO ERROR BITS DETECTED
EP(16): 1454                 Ø            1454
PL(17): 24ØØ                 Ø            24ØØ

DEVICE STATISTICS AT TIME OF ERROR:
# OF READS:     342126. # OF WRITES:    62772.  # OF SEEKS:     15252.
# SOFT READ ERRORS:     1.       # SOFT WRITE ERRORS:    Ø.
# HARD READ ERRORS:     Ø.       # HARD WRITE ERRORS:    Ø.
# SOFT POSITIONING ERRORS:       Ø.
# HARD POSITIONING ERRORS:       Ø.
# OF MPE:  Ø.   # OF NXM:  Ø.    # OF OVERRUNS:  Ø.
```

4.5 KLERR

The KLERR function translates the front-end log. This log is summarized in the system event file as the FRONT END DEVICE REPORT "KLERR" entry. This entry is written into the system event file when the KL clock stops for any of several errors (FAST MEMORY, PARITY ERRORS, CRAM PARITY, DRAM PARITY ERROR, or FIELD SERVICE STOP). Any significant error signal will be listed just after the header.

You can use KLERR to generate a detailed report of and/or summaries of KLERR data blocks. You always get a summary but you must select one of three formats if you want a detailed report of each event.

KLERR helps KL1Ø maintainers by automating some of the time-consuming tasks associated with interpreting front-end snapshots logged in the TOPS-1Ø and TOPS-2Ø system event files. RSX-2ØF stores a list of function reads (FREADs) and their results in octal. To determine the cause of a crash by reading these octal function-read words is difficult because:

- The KL1Ø registers are split between function-read words and must be reconstructed manually.
- It takes time to find the signal names associated with each bit.

- Some registers are difficult to reconstruct.
- It is difficult to see patterns across multiple events.

To use KLERR effectively, check the daily ANALYZE report. If KLERR records are being written, the ANALYZE report will include a message to that effect. The report will also show whether any error bits were set. You can use the ANALYZE packet number as input to RETRIEVE short format to find what error bits were set or use full format to get all the function reads in octal. If this does not successfully localize the fault, use the KLERR function.

4.5.1 KLERR Input

KLERR accepts the following types of input:

- The system event file
- A binary file created by the RETRIEVE process
- Any binary file containing entries from the system event file

4.5.2 KLERR Procedure

KLERR prompts you with one or more of the following guidewords:

KLERR mode

```
Event file (SERR:ERROR.SYS):

Selection (ALL):

Sequence numbers:

Time from (EARLIEST):

Time to (LATEST):

Report style (SUMMARY-ONLY):

Summary type (ERRORS-ONLY):

Output to (DSK:KLERR.RPT):
```

If you want to take all the defaults, type KLE/G to the SPEAR> prompt. Otherwise, read the following procedure:

➡STEP 1

After you type KLERR to the SPEAR> prompt, KLERR requests the name of the input file:

```
Event file (SERR:ERROR.SYS):        TOPS-20

                  or

Event file (SYS:ERROR.SYS):         TOPS-10
```

Type one of the following:

1. The RETURN key - to take the default, the system event file.

2. Any file in binary format containing KLERR events.

➡STEP 2

Next KLERR prompts you to select all KLERR events or specific ones by sequence number:

Selection (ALL):

Type one of the following:

1. The RETURN key or A[LL] - to take the default of all KLERR events in the file. You will be prompted for date and time constraints.

2. S[EQUENCE] - to select specific KLERR events by sequence number.

   If you choose SEQUENCE, KLERR prompts you further with:

   Sequence numbers:

   Here you can specify one number, several numbers separated by commas, or a range of numbers separated by hyphens.

If you chose ALL, continue with STEP 3. If you chose SEQUENCE, continue with STEP 5.

➡STEP 3

KLERR then prompts you for the date and time limits of the entries you want to select:

Time from (EARLIEST):

Type one of the following:

1. The RETURN key or E[ARLIEST] - to select the beginning of the file. This is the default.

2. A date and time in the format dd-mmm-yy hh:mm:ss - to signify where to begin extracting entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -7 causes KLERR to begin extracting entries seven days prior to the current day.

➡STEP 4

KLERR then prompts for the end of the time period:

```
Time to (LATEST):
```

Type one of the following:

1. The RETURN key or L[ATEST] - to select the end of the file. This is the default.

2. A date and time in the format dd-mmm-yy hh:mm:ss: - to indicate the last date for extracting entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -13 causes KLERR to stop extracting entries recorded thirteen days before the current date.

➡STEP 5

KLERR then prompts for the type of report in which you are interested:

```
Report type (SUMMARY-ONLY):
```

Type one of the following:

1. The RETURN key or S[UMMARY-ONLY] - to take the default. This report will contain only the final summary of signals. It will not have the entry-by-entry output.

2. F[ULL] - to select a set of detailed reports that list all the registers and signals (true or false) as well as their fields.

3. T[RUE] - to select a set of detailed reports that list all of the registers, but only the true signals and not the fields.

4. C[RAM-BAD-WORD] - to select a set of reports consisting of one line for each record that includes a CRAM parity error. This line contains the CRAM location and contents.

If you chose CRAM-BAD-WORD, continue with STEP 5A, otherwise continue with STEP 6.

➡STEP 5A

If you choose CRAM-BAD-WORD, you are then prompted with a choice of formats:

```
Cram word formats (MICROCODE):
```

Type one of the following:

1. The RETURN key or M[ICROCODE] - to select the default. This format is a comparison of the bad cram word with the microcode listing.

2. O[CTAL] - to select a format that matches the one shown in the KL1Ø Maintenance Handbook and can help isolate the failing cram module.

3. T[RACON] - to select a format that compares TRACON snapshots.

➡STEP 6

The next information KLERR prompts for is the type of summary in which you are interested:

Summary type (ERRORS-ONLY):

Type one of the following:

1. The RETURN key or E[RRORS-ONLY] - to select the default. This summary is in the form of a single page list containing the number of times an error signal was true and the number of times it was false.

2. A[LL] - to select a summary with a complete listing of the number of times each signal was true or false.

3. N[ONE] - to select the option of receiving no summary.

➡STEP 7

The last thing KLERR asks for is the destination of the output file:

Output to (DSK:KLERR.RPT):

Type one of the following:

1. The RETURN key - to select the default of KLERR.RPT.

2. TTY: - to direct the ASCII formatted output to your terminal.

3. Any file name in the proper format for your system.

After you select the output destination and press RETURN, SPEAR asks you to confirm your decision:

Type [cr] to confirm (/GO):

At this point, you can:

1. Press the RETURN key or type /GO to execute the KLERR process.

2. Type /SHOW to list the parameters you have chosen.

3. Type /REVERSE to return to the previous prompt.

4. Type /BREAK to return to the SPEAR prompt.

5. Type question mark (?), HELP, the question mark switch (/?), or /HELP to find out what your options are.

4.5.3 Sample KLERR Session

The following is a sample session of the KLERR dialogue:

```
@spear

Welcome to SPEAR for TOPS-20. Version 2(605)
Type "?" for help.

SPEAR> klerr

KLERR mode
----------
    Event file (SERR:ERROR.SYS):

    Selection (ALL): sequence

        Sequence numbers: 846

    Report style (SUMMARY-ONLY): ?

      One of the following:
      SUMMARY-ONLY
      TRUE-SIGNALS
      FULL
      CRAM-BAD-WORD
      HELP

    Report style (SUMMARY-ONLY): cram

        Cram word format (MICROCODE): ?

          One of the following:
          MICROCODE
          OCTAL
          TRACON
          ALL
          HELP

        Cram word format (MICROCODE): tracon

    Summary type (ERRORS-ONLY):

    Output to (DSK:KLERR.RPT):

Type <cr> to confirm (/GO):
```

4.5.4 KLERR Output

The following is a sample of KLERR output:

```
**********************************************
FRONT END DEVICE REPORT "KLERR" TYPE 205
 LOGGED ON 15-Nov-83 04:52:57          MONITOR UPTIME WAS 0 DAY(S) 0:0:14
        DETECTED ON SYSTEM # 2241
        RECORD SEQUENCE NUMBER: 316
**********************************************


Registers:

    AR: 000000,,000000  ARX: 000000,,000000  FM: 000000,,273041
    BR: 000000,,000000  BRX: 002000,,020000  AD: 000000,,000000
    MQ: 001100,,002000  ADX: 000000,,000000

    PC:        00,,005636   PI ON:    177  SC: 0000  FM BLOCK: 00
    VMA:       00,,005636   PI HOLD: 000  FE: 0000  FM ADDR:  04
    VMA HELD: 00,,005636   PI GEN:  000

CRAM word in octal:

   LOC    0-15    16-31   32-47   48-63   64-79   80-85
   1044/ 001044 070000 104041 000020 000002 10

CRAM word by field (microcode listing format):

   LOC    A     B     C     D     E     F     G
   1044, 1044,0001,0400,0020,1020,7110,0000

CRAM word by field (TRACON format):

   LOC /  J   T AR  AD BR MQ FM SCAD SC FE SH  # VMA MEM COND SPEC M
   1044/1044 1 40 1000 0  0  0  200  0  0  1 000  0  00   71   10 0

DRAM word by field:

    ADR: A B P J
    254/ 2 0 0 144
```

Signal name breakdown follows (Error signals first) :

- Signals in alphabetical order -

```
STATE   NAME
-----   ----

  F     APR2-M8539-APR C DIR P ERR IN H
  F     APR1-M8539-APR I/O PF ERR IN H
  F     APR1-M8539-APR MB PAR ERR IN H
  F     APR1-M8539-APR NXM ERR IN H
  F     APR2-M8539-APR S ADR P ERR IN H
  F     APR1-M8539-APR SBUS ERR IN H
  F     APR2-M8539-APR ANY EBOX ERR FLG H
  F     APR2-M8539-APR PWR FAIL IN H
  F     CHC1-M8533-CBUS ERROR E H
                .
                .
                .
```

- Fields from function reads -

```
VALUE    FIELD
-----    -----

     Ø   CCW2-M8534-CCW CHA 18-23 H
     Ø   CCW2-M8534-CCW CHA 14-17 H
     Ø   CCW2-M8534-CCW CHA 24-29 H
     Ø   CCW2-M8534-CCW CHA 3Ø-35 H
     Ø   PIC4-M8532-EBUS CSØØ-Ø3 E H
     Ø   MBZ1-M8537-EBUS REG ØØ-Ø8 H
     Ø   MBZ1-M8537-EBUS REG 14-26 H
    33   MBC1-M8531-EBUS REG 27-33 H
     2   MBZ1-M8537-EBUS REG 34,35 H
    1Ø   IRD1-M8522-IR AC Ø9-12 H
 77400   MTR1-M8538-MTR CACHE COUNT Ø2-17 H
     Ø   MTR1-M8538-MTR EBOX COUNT Ø2-17 H
   6ØØ   MTR1-M8538-MTR INTERVAL Ø6-17 H
 11Ø7Ø   MTR1-M8538-MTR PERF COUNT Ø2-17 H
     2   MTR3-M8538-MTR PERIOD Ø6-17 H
 11ØØØ   MTR1-M8538-MTR TIME Ø2-17 H

                .
                .
                .
```

** End of KLERR report. 1. entries were processed.

4.6 SUMMARIZE

SUMMARIZE reads the system event file and summarizes its contents according to the following categories:

1. Event code
2. STOPCODE (TOPS-1Ø)
3. BUGCHK, BUGHLT, BUGINF (TOPS-2Ø)
4. Front-end reloads
5. Channel errors
6. Disk errors
7. Magnetic tape errors

The SUMMARIZE report also contains Error Distribution tables. These tables show a 24 hour distribution of events listed according to subsystem. With these tables, you can determine when the large number of events is occurring. Once you know the subsystem (Mainframe, Disk, Tape, and so forth) and the timeframe, you can use RETRIEVE or ANALYZE to pinpoint the specific device that is causing the problem.

After reading the file, SUMMARIZE produces an ASCII report file containing the summaries and Error Distribution tables and stores it in your disk area (or wherever you specify). You can then print the report on the lineprinter for inspection. You can also print the report on the terminal by specifying TTY: to SPEAR's request for the output destination.

SUMMARIZE allows you to pinpoint the timeframe of the summaries by requesting a beginning date and an ending date to search for in the system event file. In addition, you can also specify a binary file created with the RETRIEVE process (RETRIE.SYS) for input. See Section 4.4 for information on RETRIEVE.

4.6.1 The SUMMARIZE Report

The following example is representative of a SUMMARIZE report in that it contains:

- File environment information
- Entry occurrence counts
- System event codes, shown in parentheses under entry occurrence counts
- Summaries of bugchecks and subsystems
- Error distribution tables

Note that if the media name cannot be identified in reports that include media identification, SUMMARIZE uses three specific formats:

1. <unknown> - if SUMMARIZE does not find a mount record in the error file prior to the time of the error.

2. <none> - if a series of mount and dismount records indicate no medium was mounted at the time of the error, such as an error occurring during the mount process.

3. <blank> - if SUMMARIZE finds a mount record but the medium-name field of the mount record is empty.

Note the error register codes listed in the report are described in Section 4.6.2.

File Environment

```
SPEAR Version 2(613)
Input file:   SERR:ERROR.SYS Created: 12-Mar-84 Ø8:49:ØØ-EST
Output file:  DSK:SUMMAR.RPT

Selection Criteria: ALL

Date of first entry processed: 14-Mar Ø1:22:13
Date of last entry processed:  14-Mar 23:53:38

Number of entries processed: 1128.
Number of inconsistencies detected in error file: Ø.
```

Entry Occurrence Counts:

```
  9. SYSTEM RELOAD ...(1Ø1)
496. MONITOR BUG ...(1Ø2)
 36. MASSBUS ERROR ...(111)
12Ø. STATISTICS ...(114)
  8. CONFIGURATION CHANGE ...(115)
1Ø2. FRONT END DEVICE ERROR ...(13Ø)
  1. CPU PARITY INTERRUPT ...(162)
294. PHASE III DECNET ENTRY ...(24Ø)
 62. HSC5Ø ERROR LOG ...(243)
```

Monitor Detected Errors and Reloads:

```
 43. BUGCHK
  4. BUGHLT
449. BUGINF
```

Monitor Error and Reload Breakdown:

```
BUGCHK Breakdown
  8. FLKTIM
  2. KLPERR
 17. MSCORO
  3. NODDMP
  5. PI2ERR
  4. SCACVC
  4. SCATMO

BUGHLT Breakdown
  1. ILPSEC
  1. NOTOFN
  1. SKDPF1
  1. UNPGF2
```

```
BUGINF Breakdown
     8. CFCONN
     4. KLPCVC
    29. KLPNUP
     1. KLPRRQ
     1. KLPSTR
    28. MSCAVA
     2. MSCDSR
     7. MSCPTG
   324. NSPBAD
    29. NSPLAT
     2. NTOHNG
     1. SPRZRO
     1. TM8AEI
    12. TTYSTP

Front-end Summary:

    10. CD20
    10. DH11
    10. DL11C
    10. DM11
     1. DM11-3
     6. KLCPU
    45. KLERR records forming 5. full entries
    10. LP20

DECnet Summary:

  Class.Type     Count    Description

      0.0          10.    Event records lost
      0.3           8.    Automatic line service
      2.0           2.    Local node state change
      4.0          29.    Aged packet loss
      4.1         233.    Node unreachable packet loss
      4.4           1.    Packet format error
      4.7           6.    Circuit down, circuit fault
      4.10          5.    Circuit up


RH20 Channel/Controller Summary:

                Hard    Soft
     # 1           0.      1.
     # 2           5.     30.


RP07 Summary:

                Hard    Soft
  S/N 2861
      DP100       0.      1.
```

TM78 Summary:

	Hard	Soft
S/N 4404		
MT200	2.	4.
S/N 5242		
MT210	3.	26.

RH20 Breakdown (CONI)

	PAR ERR	EXC	LWC ERR	SWC ERR	CHN ERR	RES ERR	RAE	OVR RUN
DP100								
SOFT		1.			1.			
MT200								
HARD		2.						
MT200								
SOFT		4.						
MT210								
HARD		3.						
MT210								
SOFT		26.						

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*                                                         *
*       Disk Subsystem Error Summary                      *
*                                                         *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Disk Subsystem Error Entries Summarized by Device, then Error Type.
Where the Error Types are the following:

OTHER	=	OTHER
TIMIN	=	TIMING
SK-SR	=	SEEK-SEARCH
READ	=	READ-WRITE
CH-CO	=	CHANNEL-CONTROLLER
BUS	=	BUS
SOFT	=	HARDWARE DETECTED SOFTWARE ERROR
MICRO	=	MICROPROCESSOR DETECTED ERROR
UNSAF	=	UNSAFE
WRTLK	=	WRITE LOCK
OFFLI	=	OFFLINE

	OTHER	TIMIN	SK-SR	READ	CH-CO	BUS	SOFT	MICRO	UNSAF	WRTLK	OFFLI
DP100											
				1.							
DU-7-14-17											
	36.					3.					
DU-7-3-17											
	19.					3.			1.		

Read Data Errors further summarized by Drive and Media ID.

Drive	Media	Error Totals
DP100	WORK	1.

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*                                                                               *
* This report summarizes all Read Data Errors by Drive and Media ID             *
*                                                                               *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

DRIVE	MEDIA	CYL	TRK	SECT	HARD	SOFT	RETRIES	LBN
DP1ØØ	WORK	565.	5.	15.	Ø.	1.	2.	2,,756704

RPØ7 BREAKDOWN:

Error Register 1

D C K	U N S	O P I	D T E	W L E	I A E	A O E	H C R C	H C E	E C H	W C F	F E R	P A R	R M R	I L R	I L F

S/N 2861
DP1ØØ S 1.

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*                                                                               *
*       Tape Subsystem Error Summary                                            *
*                                                                               *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Tape Subsystem Error Entries Summarized by Device, then Error Type.
Where the Error Types are the following:

OTHER	=	OTHER
READ	=	READ
WRITE	=	WRITE
FORMT	=	DEVICE FORMAT
CH-CO	=	CHANNEL-CONTROLLER
BUS	=	BUS
SOFT	=	HARDWARE DETECTED SOFTWARE ERROR
OPER	=	OPERATOR
OFFLI	=	OFFLINE

	OTHER	READ	WRITE	FORMT	CH-CO	BUS	SOFT	OPER	OFFLI
MT2ØØ			6.						
MT21Ø			29.						

```
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
*                                                                           *
* SUMMARY of all Errors sorted by Media and Drive by                        *
* Operation.                                                                *
*                                                                           *
*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*-*
```

Operation : WRITE Related

MEDIA ID	UNIT ID		
	MT200	MT210	TOTAL
unknown	6.	29.	35.
TOTAL	6.	29.	35.

TM78 Breakdown:
(Interrupt and Failure Codes are OCTAL)

	Interrupt Code		Failure Code	Hard	Soft
S/N 4404					
MT200	22	(WRITE)	7	0.	3.
MT200	22	(WRITE)	10	0.	1.
MT200	22	(WRITE)	14	2.	0.
S/N 5242					
MT210	22	(WRITE)	1	0.	7.
MT210	22	(WRITE)	4	0.	10.
MT210	22	(WRITE)	7	0.	1.
MT210	22	(WRITE)	10	0.	8.
MT210	22	(WRITE)	14	3.	0.

Error distribution

14-Mar-84	Main-frame	Disk	Tape	Unit rec	Comm	Net-work	Soft-ware	Crash	Totals
1:00 - 2:00						6.		5.	11.
6:00 - 7:00		7.				6.		12.	25.
8:00 - 9:00	19.	35.				13.		64.	133.
9:00 - 10:00		20.				5.		31.	56.
10:00 - 11:00	9.					10.		7.	28.
11:00 - 12:00	9.					6.		6.	22.
12:00 - 13:00								3.	3.
13:00 - 14:00						1.		9.	10.
14:00 - 15:00						3.		7.	10.
15:00 - 16:00	9.		4.			27.		45.	86.
16:00 - 17:00						91.		76.	167.
17:00 - 18:00						19.		6.	25.
18:00 - 19:00			2.			22.		38.	62.
19:00 - 20:00			11.		1.	17.		43.	72.
20:00 - 21:00		1.	8.			21.		39.	69.
21:00 - 22:00			4.			19.		38.	61.
22:00 - 23:00			2.			12.		38.	52.
23:00 - 0:00			4.			16.		38.	58.
Totals	46.	63.	35.		1.	294.		505.	950.

Due to the addition of the CI and HSC5Ø, you will find another format for listing the names of disks in the SUMMARIZE report. In the previous report, you will find the following:

```
DU-7-14-17
DU-7-3-17
```

Starting from left to right, these four fields represent the following:

Field one	Device type DU = RA8Ø, RA81 DJ = RA6Ø ?? = unknown
Field two	RH slot number for the CI2Ø. This is always number 7.
Field three	HSC5Ø node number on the CI.
Field four	Drive number on the push button. If the HSC5Ø cannot get this number, the number 4Ø95 appears in this field.

Note you will find a description of the Disk Subsystem Error Bits in Appendix D.

4.6.2 Error Register Codes

The following tables contain brief explanations of the abbreviations of the error register codes (MASSBUS disk registers for RPØ4s and RPØ6s and tape registers for TU45s, TU77s, and TE16s).

Table 4-5: MASSBUS Disk Registers

	Error Register 1
Code	Meaning
DCK	Data Check
UNS	Unsafe
OPI	Operation Incomplete
DTE	Drive Timing Error
WLE	Write Lock Error
IAE	Invalid Address Error
AOE	Address Overflow Error
HCRC	Header CRC Error
HCE	Header Compare Error
ECH	ECC Hard Error
WCF	Write Clock Fail
FER	Format Error
PAR	Parity Error
RMR	Register Modification Refused
ILR	Illegal Register
ILF	Illegal Function

Table 4-5: MASSBUS Disk Registers (Cont.)

	Error Register 2
Code	Meaning
ACU	RPØ4 - AC Unsafe RPØ6 - Unused
PLU	Phase Locked Oscillator Unsafe
3ØVU	RPØ4 - 3Ø Volts Unsafe RPØ6 - Unused
IXE	Index Error
NHS	No Head Select
MHS	Multiple Head Select
WRU	Write Ready Unsafe
FEN	RPØ4 - Failsafe Enabled
ABS	RPØ6 - Abnormal Stop
TUF	Transition Unsafe
TDF	Transition Detector Failure
MSE	RPØ4 - Motor Sequence Error
R&W	RPØ6 - Read and Write
CSU	Current Switch Unsafe
WSU	Write Select Unsafe
CSF	Current Sink Failure
WCU	Write Current Unsafe

	Error Register 3
Code	Meaning
OCYL	Off Cylinder
SKI	Seek Incomplete
OPE	RPØ4 - Unused RPØ6 - Operator Plug Error
ACL	AC Voltage Unsafe
DCL	DC Voltage Unsafe
DIS	RPØ4 - Unused
35V	35 Volts Unsafe
UWR	RPØ4 - Any Unsafe Except Read/Write RPØ6 - Unused
VUF	RPØ4 - Velocity Unsafe
WOF	RPØ6 - Write and Unsafe
PSU	RPØ4 - Pack Speed Unsafe
DCU	RPØ6 - DC Voltage Unsafe

Table 4-6: Tape Registers

Code	Meaning
COR/CRC	PE - Correctable Data Error NRZI - CRC Does Not Match Computed CRCC
UNS	Unsafe
OPI	Operation Incomplete
DTE	Drive Timing Error
NEF	Nonexecutable Function
CS/ITM	PE - Correctable Skew NRZI - Illegal Tape Mark
FCE	Frame Count Error
NSG	Nonstandard Gap Tape Character
PEF/LRC	PE - Format Error NRZI - Longitudinal Redundancy Check
INC/VPE	PE - Noncorrectable Data Error NRZI - Vertical Parity Error
DPA	Data Bus Parity Error
FMT	Format Error
PAR	Control Bus Parity
RMR	Register Modification Refused
ILR	Illegal Register
ILF	Illegal Function

4.6.3 SUMMARIZE Procedure

SUMMARIZE prompts with one or more of the following guidewords:

SUMMARIZE Mode

Event file (SERR:ERROR.SYS):

Category (ALL):

Time from (EARLIEST):

Time to (LATEST):

Show Error Distribution (YES):

Report to (DSK:SUMMAR.RPT):

If you want to take all the defaults, type S/G to the SPEAR> prompt; otherwise, read the following procedure:

➡STEP 1

After you type SUMMARIZE to the SPEAR> prompt, SUMMARIZE requests the name of the input file:

Event file (SERR:ERROR.SYS): TOPS-2Ø

or

Event file (SYS:ERROR.SYS): TOPS-1Ø

Type one of the following:

1. The RETURN key - to take the default, the system event file.
2. The name of a file you have previously RETRIEVEd, in binary format, for example RETRIE.SYS.
3. Any file in binary format containing events from the system event file.

➡STEP 2

SUMMARIZE asks for the category of the summary in which you are interested:

Category (ALL):

Type one of the following:

1. The RETURN key or A[LL] - to take the default of all categories.
2. M[AINFRAME] - to select a summary for mainframe events.
3. D[ISK] - to select a summary for disk devices.
4. T[APE] - to select a summary of tape devices.
5. CI - to select a summary of CI-related events.
6. NI - to select a summary of NI-related events.
7. U[NITRECORD] - to select a summary of hard-copy devices.
8. NE[TWORK] - to select a summary of network-related events.
9. O[PERATING-SYSTEM] - to select a summary of software-related events.
10. CO[MM] - to select a summary of communication devices.
11. P[ACKID] - to select a summary of specific disk packs.
12. R[EELID] - to select a summary of specific tape reels.

All categories except for COMM and NI prompt for specific device types. Table 4-7 lists the subprompts you can expect.

Table 4-7: Subprompts for Device Types

Device Type	Subprompt
MAINFRAME	Mainframe devices (ALL):
DISK	Disk drives (ALL):
TAPE	Tape drives (ALL):
CI	CI controllers (ALL):
UNITRECORD	Unit record devices (ALL):
NETWORK	Event class and type (ALL):
OPERATING-SYSTEM	Operating System codes (ALL):
PACKID	Disk (structure IDs):
REELID	Tape (reel IDs):

➡STEP 3

SUMMARIZE keeps prompting you for categories until you either type FINISHED or press the RETURN key:

```
Next Category (FINISHED):
```

Type one of the following:

1. The RETURN key or F[INISHED] - to take the default.

2. Another category.

➡STEP 4

After you have specified the source of input, SUMMARIZE prompts you for the date and time at which you want the summary to begin:

```
Time from (EARLIEST):
```

Type one of the following:

1. The RETURN key - to take the default EARLIEST, the first event in the file.

2. A date and time in the format dd-mmm-yy hh:mm:ss - to signify where to begin extracting entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -7 causes SUMMARIZE to begin extracting entries seven days prior to the current day.

➡STEP 5

SUMMARIZE then prompts for the end of the time period:

```
Time to (LATEST):
```

Type one of the following:

1. The RETURN key - to take the default LATEST, the last entry in the system event file.

2. A date and time in the format dd-mmm-yy hh:mm:ss - to indicate the last date for extracted entries. A date by itself defaults to one second after midnight.

3. A date and time in the format -nn to indicate a reference point prior to the current date. For example, -13 causes SUMMARIZE to stop extracting entries recorded thirteen days before the current date.

➡STEP 6

After specifying a timeframe, you can choose whether or not to receive the error distribution tables:

Show Error Distribution (YES):

Type one of the following:

1. The RETURN key or Y[ES] - to take the default. This will give you all the error distribution charts relevant to the time constraints you specify.

2. N[O] - to suppress the error distribution charts from the report.

➡STEP 7

The last thing SUMMARIZE asks for is the destination of the output:

Report to (DSK:SUMMAR.RPT):

Type one of the following:

1. The RETURN key - to take the default DSK:SUMMAR.RPT.

2. Any file name in the proper format.

3. TTY: - to have the report printed on your terminal. Note that if you specify TTY:, SUMMARIZE does not save the file in your disk area.

After you select the output destination and press RETURN, SPEAR asks you to confirm your decision.

Type <cr> to confirm (/GO):

At this point you can:

1. Press RETURN or type /GO to execute the SUMMARIZE process.

2. Type /SHOW to list the parameters you have chosen.

3. Type /REVERSE to return to the previous prompt.

4. Type /BREAK to return to SPEAR level.

5. Type question mark (?), HELP, the question mark switch (/?), or /HELP to find out what your options are.

To read the SUMMARIZE report, you can list the file on the lineprinter by doing the following:

Return to operating system command level by typing EXIT to the SPEAR> prompt.

Use the PRINT command with any options available on your operating system.

Note that if you specified TTY: to the Report to: prompt, you will not have a file saved in your area to print.

4.6.4 Sample SUMMARIZE Session

The following is a sample of a SUMMARIZE session using the system event file for input:

```
@spear

Welcome to SPEAR for TOPS-20. Version 2(605)
Type "?" for help.

SPEAR> summarize

SUMMARIZE mode
--------------
    Event file (SERR:ERROR.SYS):

        Category (ALL): main

            Mainframe devices (ALL): cpu

        Next Category (FINISHED): disk

            Disk drives (ALL): rpo7

        Next Category (FINISHED):

    Time from (EARLIEST):

    Time to (LATEST):

    Show Error Distribution (YES): no

    Report to (DSK:SUMMAR.RPT):

Type <cr> to confirm (/GO):
INFO - Summarizing ST:GIDNEY.02-27
INFO - Now sending summary to DSK:SUMMAR.RPT
INFO - Summary output finished


SPEAR> ex
```

Table 4-8 lists the supported devices, according to subsystem, from which you can expect summaries.

Table 4-8: Supported Devices

SUBSYSTEM	DEVICE	DETAILED SUMMARIES?
MAINFRAME	KL10	YES
	KS10	NO
	FRONT-END	YES
CI	CI20	YES
	HSC	YES
DISK	RP03	YES
	RM03	YES
	RP04	YES
	RP05	YES
	RP06	YES
	RP07	YES
	RP20	YES (DX20)
	RS04	YES
	RA60	YES
	RA80	YES
	RA81	YES
TAPE	TU16	YES
	TU45	YES
	TU70	YES
	TU71	YES
	TU72	YES
	TU73	YES
	TU77	YES
	TU78	YES
UNIT RECORD	LPT	YES
	CDR	YES
COMM	DH11	YES
	DQ11	YES
NET	DECNET	
	PHASE 2, 3, 4	YES
	ANF 10	YES
	SNA 20	YES
	NIA20	YES

4.7 TOPS-20 KLSTAT MODE

On TOPS-20, there is an additional troubleshouting aid that can be helpful if severe intermittent faults do not leave enough information in the system event file. This feature is the KLSTAT mode. When you turn KLSTAT on, you are actually turning on a monitor flag that tells the monitor to record additional information into the system event file when any CPU, memory, or MASSBUS errors occur.

Note that turning on this flag causes severe system degradation (the system goes down while KLSTAT is collecting data) you should turn it on only when absolutely necessary. In fact, you must have special privileges to turn it on or off.

When the KLSTAT mode is in operation, the system event file will contain KL CPU STATUS BLOCK entries. For a sample of such an entry, turn to Section 5.3.12. For the KLSTAT procedure, read the following section, Section 4.7.1.

4.7.1 KLSTAT Procedure

The KLSTAT mode has three functions: ON, OFF, and CHECK. The following procedure describes their use:

➡STEP 1

First, enable your special privileges at monitor level, either OPERATOR or WHEEL privileges. Then access SPEAR. (Note, you do not need privileges to CHECK the status of KLSTAT.)

➡STEP 2

Once at the SPEAR prompt, type K[LSTAT]:

```
SPEAR>KLSTAT
```

SPEAR responds with:

```
SPEAR>KLSTAT

KLSTAT mode
----------

Extra reporting (CHECK):
```

➡STEP 3

At this point, type one of the three options. Pressing the Escape key gets you the default, CHECK. If you type ON, you will get this message:

```
The following should be noted before proceeding!
This function can cause SEVERE system degradation!
```

If you decide not to risk it, type /R to return to the SPEAR prompt.

➡STEP 4

If you respond with one of the three choices, SPEAR prompts with:

```
Type <cr> to confirm (/GO):
```

If you chose ON or OFF, SPEAR returns you to the SPEAR prompt. If you chose CHECK, the default, SPEAR prints one of the following:

```
(KLSTAT) Extra error reporting is currently enabled.
```

or

```
(KLSTAT) Extra error reporting is currently disabled.
```

You can check the information gathered by turning on the KLSTAT mode by looking for the KL CPU STATUS BLOCK entry in the system event file. See Section 5.3.12.

4.8 COMPUTE

COMPUTE allows you to generate an ASCII report on the availability of system resources. When compiling its report, COMPUTE considers system statistics and monitor failures in its calculations. The data base that COMPUTE uses differs slightly between the operating systems.

On TOPS-10, the report data base is a file written by the monitor in the same format as the system event file. This TOPS-10 file contains reload information, device status-change data, date and time changes, and other pertinent information. The entries are written into this file when they occur, in the same manner as the entries are written into the system event file.

COMPUTE files on TOPS-10 are grouped starting with the first monitor load and ending with the last reload in the selected directory. The files are named AVAIL.Ann beginning with AVAIL.A01 for the first week, (the oldest file in the group) AVAIL.A02 for the second week, and so forth up to the current (incomplete) file AVAIL.SYS. To find out the file names of the available weeks, do a directory search of SYS:AVAIL.*, by typing DIR SYS:AVAIL.* at operating system command level.

On TOPS-20, the report data base is the system event file, ERROR.SYS. For COMPUTE purposes, TOPS-20 also has a buffer file called COMPUTE.STATISTICS. Approximately every 20 seconds, any available runtime information is written into this buffer file. Then every hour the information in this buffer file is dumped into the system event file as a special entry called LOGGER ENTRY (octal code 500). Also, during a system reload, the last entry in COMPUTE.STATISTICS is written into the system event file. When you run COMPUTE on TOPS-20, it looks for these LOGGER entries to compile its report.

Although TOPS-20 does not have separate weekly files, COMPUTE can break down the system event file into a calendar week from Sunday at 00:00:01 hours to Saturday at 23:59:59 (approximately) to come up with single weekly reports. COMPUTE uses hourly dumps from COMPUTE.STATISTICS on TOPS-20 to approximate the calendar week. In this way, you can specify date and time limits when running COMPUTE.

4.8.1 COMPUTE Reports

With COMPUTE, you can output your report in one of three ways:

1. A single report containing statistics from a single week.

2. A single report containing statistics from several weeks, merged into one report.

3. Several reports containing statistics from individual weeks.

In addition to the COMPUTE report, you also receive a report containing information concerning reloads. This report is called RELOAD.RPT. You will receive the same number of reload reports as you do COMPUTE reports.

If you decide you want individual weekly reports, COMPUTE prompts you for the beginning and ending dates of the weeks of interest. The default is the first week's file, the oldest file, to the file containing the last full week. If you use the default, you will receive one report for every week from the last monitor load to the last reload of the COMPUTE file in your selected directory.

4.8.2 COMPUTE Formulas

The following formulas are used by COMPUTE to derive the values reported in the full report:

FORMULA 1 System Availability (SA)

SA = (1.Ø) - Chargeable Downtime/Usage Cycle

where

Chargeable Downtime — is any nonscheduled period of time that the system is not running as determined by the answer the operator gives to the WHY RELOAD? question. The answers that constitute a charge to downtime are:

1. STOPCD or BUGHALT
2. Halt
3. Parity
4. Hardware
5. NXM (nonexistent memory)
6. Hung
7. Loop
8. CM (corrective maintenance)

Time is not charged when the answer to WHY RELOAD? is:

1. Power
2. Static
3. OPR (operator)
4. PM (preventive maintenance)
5. New
6. Sched (scheduled)
7. SA (standalone)
8. Other

Total Downtime is the sum of Chargeable Downtime and Nonchargeable Downtime.

Usage Cycle — is Total Downtime plus Total Run time.

Total Run Time is the sum of all monitor Run Times within the period you specify for the report.

FORMULA 2 User Availability (UA)

UA = (1.Ø) - Chargeable Downtime/(Chargeable Downtime + Total Run Time)

FORMULA 3 System Effectiveness (SE)

SE = System Availability * (e**(-t/MTBF))

where

e — is the natural base of logarithms, (2.71828+), also known as the Napierian logarithm.

** — represents the words "raised to the power of".

t can be one of four different values: Ø.1 hrs., Ø.5 hrs., 1.Ø hrs., or 4.Ø hrs.

MTBF is the abbreviation for Mean Time Between Failures. This is the usage cycle divided by the number of crashes. Usage cycle is Total Run Time plus Total Down Time.

System Effectiveness considers both the probability of the system being up at time zero (System Availability) and the probability of the system staying up (System Reliability) for some time period "t".

You should be aware of the following facts about the COMPUTE function:

1. The accuracy of this function depends heavily on correct operator response to the WHY RELOAD question and accurate insertion of the time of day. If "Other" is selected for reason for reloading, the preceding Downtime is not counted against availability.

2. An incorrect reload time should be corrected by the operator before another reload occurs to avoid negative Downtimes or Runtimes. Because date/time changes are logged in the COMPUTE files, COMPUTE can adjust times as necessary.

3. Total Runtime and Downtime figures are not precise. On TOPS-1Ø, the monitor keeps track of time by updating the availability file every six minutes. On TOPS-2Ø, the buffer file COMPUTE.STATISTICS is updated every 2Ø seconds, and the system event file is updated every hour. If one crash/reload sequence is immediately followed by another, these times may not be correctly updated. COMPUTE compensates for this by assuming the system did not resume service after the previous reload.

4.8.3 COMPUTE Procedures

If you want to take all the defaults, type C/G to the SPEAR> prompt; otherwise, read the following procedures:

COMPUTE uses the following guideword prompts:

COMPUTE Mode

Event file (SERR:ERROR.SYS):

Report period (LAST-WEEK):

Time from (EARLIEST):

Time to (LATEST):

Report type (SINGLE-REPORT):

Availability report to (DSK:COMPUT.RPT):

Reload report to (DSK:RELOAD.RPT):

➡STEP 1

COMPUTE begins by asking for the file containing the records you want to use in the COMPUTE calculations:

```
Event file (SERR:ERROR.SYS):        TOPS-20

            or

Event file (SYS:AVAIL.LWK):         TOPS-10
```

Type one of the following:

1. The RETURN key - to take the default COMPUTE file for your system; SERR:ERROR.SYS on TOPS-20, SYS:AVAIL.LWK on TOPS-10.

2. If you are on TOPS-20, and you know of another file containing COMPUTE statistics, specify that file name here.

   If you are on TOPS-10, and you know of a specific AVAIL file (for example, AVAIL.A14) specify the file name here.

➡STEP 2

The next prompt asks for the period of time for which you want system performance calculated:

```
Report period (LAST-WEEK):
```

Type one of the following:

1. The RETURN key or L[AST-WEEK] - to take the default. This report covers the last 7 days (168 hours) prior to last Sunday at 00:00:01.

2. T[HIS-WEEK] - if you want the report to cover the current week. This report will begin with last Sunday at 00:00:01 and continue through the present. This will be an incomplete week.

3. O[THER] - if you want the report to cover a period of time other than last week or this week. If you choose OTHER, you will be prompted for the date and time parameters.

If you specify OTHER, continue with STEP 3. If you specified THIS-WEEK or LAST-WEEK, skip to STEP 6.

➡STEP 3

After you type OTHER, COMPUTE prompts you for the beginning date of the time period in which you are interested:

```
Time from (EARLIEST):
```

Type one of the following:

1. The RETURN key or E[ARLIEST] - to take the default. This is the first entry in the file.

2. The date and time (real time) in the form dd-mmm-yy hh:mm:ss where dd is the numerical day, mmm is the first three letters of the month, yy is the year, hh is the hour, mm is the minute, and ss is the second. If you specify only the date, the default time is one second after midnight.

3. The date (relative time) in the form -nn where -nn indicates a date prior to the current date. For example, -6 causes COMPUTE to begin processing from 6 days prior to the current day.

➡STEP 4

COMPUTE prompts next for the time at which you want to end the calculations:

Time to (LATEST):

Type one of the following:

1. The RETURN key or L[ATEST] - to take the default. This is the last entry in the file.

2. The date and time (real time) in the form dd-mmm-yy hh:mm:ss where dd is the numerical day, mmm is the first three letters of the month, yy is the year, hh is the hour, mm is the minute, and ss is the second. If you do not specify the date, the default time is one second after midnight.

3. The date (relative time) in the form -nn where -nn indicates a date prior to the current day. For example, -2 causes COMPUTE to end the calculations 2 days prior to the current day.

➡STEP 5

COMPUTE asks for the type of report you want:

Report type (SINGLE-REPORT):

Type one of the following:

1. The RETURN key or S[INGLE-REPORT] - to take the default. This choice will give you one report containing the information for as many weeks as you specified.

2. M[ULTIPLE-REPORTS] - to receive a report for each week within the timeframe you specified. Each report will reflect system performance for a 7 day period beginning on Sunday at 00:00:01 and ending on the following Sunday 00:00:00.

➡STEP 6

COMPUTE prompts for the destination of the availability report:

Availability report to (DSK:COMPUT.RPT):

Type one of the following:

1. The RETURN key - to take the default file specification DSK:COMPUT.RPT.

2. A file specification in the proper format for your system.

➡STEP 7

The last thing COMPUTE asks for is the destination of the reload report:

Reload report to (DSK:RELOAD.RPT):

Type one of the following:

1. The RETURN key - to take the default file specification DSK:RELOAD.RPT.

2. A file specification in the proper format for your system.

After you select the output destination and press RETURN, SPEAR asks you to confirm your decision:

Type <cr> to confirm (/GO):

At this point, you can:

1. Press RETURN or type /GO to execute the COMPUTE process.

2. Type /SHOW to list the parameters you have chosen.

3. Type /REVERSE to return to the previous prompt.

4. Type /BREAK to return to SPEAR> level.

5. Type question mark (?), HELP, the question mark switch (/?), or /HELP to find out what your options are.

After you execute COMPUTE, if you specified MULTIPLE-REPORTS, you will receive several individual reports with the file names Cmmdd.RPT and RLmmdd.RPT,

where

mm is the month of the start of the usage cycle.

dd is the day of the week of the usage cycle.

You will also receive a COMPUT.RPT and a RELOAD.RPT, combining all the information in the individual reports.

When COMPUTE has finished its calculations, it prints a summary report on your terminal and outputs the full report(s) to your disk area (or wherever you specify). The COMPUTE Summary report is a condensed version of the information you will find in the full report.

4.8.4 COMPUTE Summary Report

The following is a sample COMPUTE Summary report:

```
COMPUTE Summary Report From: 28-Sep-81 03:44 To:  1-Oct-81 14:05
        period length (HRS):      82.351, usage cycle =   82.350

     SYSTEM Availability % :     100.000
     USER Availability %   :     100.000

Total Reloads           4.      MTB Reloads       20.587
Total Crashes           0.      MTB Crashes       82.350

Effectiveness   Six minutes     Thirty minutes  One Hour        Four Hours
  factor          98.559          93.002          86.495          55.972

                Run times       Down times      Crash times
Totals            27.571          54.779          0.000
Means              6.892          18.259          0.000
Maxima            12.270          52.153          0.000
Minima             1.401           0.375          0.000
Std. Dev.          4.234          23.978          0.000

Bug/Stopcode    count
   DIRPG1       7.   DN20ST     7.
   DTEIPR       17.
   DX2HLT       1.
   ITRLGO       5.
   NSPLAT       5.
   OVRDTA       1.
Report file name: DSK:COMPUT.RPT
```

4.8.5 COMPUTE Full Report

The following is a sample of a COMPUTE Full report. This type of report is saved on your disk area for printing on a line printer. You can print it on your terminal but it will be unreadable because of the 132 column width of the report.

SYSTEM AVAILABILITY REPORT FOR THE PERIOD: 28-Sep-81 03:44 TO 1-Oct-81 14:05

CUSTOMER SATISFIED(Y OR N)?____ CUSTOMER SIGNATURE_____

***** SYSTEM STATISTICS *****(ALL TIMES IN HOURS)

AVAILABILITY		SYSTEM EFFECTIVENESS		RUNTIME		DOWNTIME	
OPERATIONAL CYCLE :	82.351	T= 0.1HRS:	98.559	TOTAL RUN TIME :	27.571	SYSTEM NOT RUNNING:	54.779
SYSTEM AVAILABILITY:	100.000	T= 0.5HRS:	93.002	MAXIMUM RUN TIME :	12.270	MAXIMUM DOWNTIME :	52.153
USER AVAILABILITY :	100.000	T= 1.0HRS:	86.495	MINIMUM RUN TIME :	1.401	MINIMUM DOWNTIME :	0.375
NUMBER OF RELOADS :	4.	T= 4.0HRS:	55.972	MEAN RUN TIME :	6.892	MEAN DOWN TIME :	18.259

***** RELOADS NOT AFFECTING MEASURED AVAILABILITY *****

MONITOR NAME & VERSION

	POWER FAIL	STATIC	OPERATOR	PM	NEW	SCHEDULED	STANDALONE	OTHER /UNK	TOTALS	
SYSTEM 2116 THE BIG ORANGE, TOPS-20 MONITOR 4(3556) 500,,4363										
	0.	0.	1.	0.	0.	1.	0.	2.	4.	Count
	0.000	0.000	2.251	0.000	0.000	52.153	0.000	0.375	54.779	Time (HRS)

Bug/Stopcode	Count
DN20ST	7.
DTEIPR	7.
DX2HLT	17.
ITRLGO	1.
NSPLAT	5.
OVRDTA	5.
OVRDTA	1.

CHAPTER 5

ENTRY DESCRIPTIONS

5.1 INTRODUCTION

This chapter provides a sample of most of the events that can be recorded in the system event file. These samples appear just as you see them when you use RETRIEVE to translate entries from binary to ASCII. Although the entries may differ in format, they each have sections in common, some more than others depending on the operating system involved. Each entry may contain from one to six sections of information:

Section 1 Entry Description
Section 2 Unit Identification
Section 3 Software Status
Section 4 Controller Status
Section 5 Device or Unit Status
Section 6 Statistical Information

Every entry has at least a Section 1, Entry Description. This section contains:

1. Type of entry and/or type of error
2. Error-entry date and time that it was logged
3. Monitor uptime
4. System serial number

Entries may contain Sections 2 through Section 6. Section 2 contains the following information:

1. Unit logical name
2. Unit physical name
3. Unit type
4. Media identification

Section 3 contains the following:

1. Highest process requesting service (user)
2. Lowest process requesting service (author)

3. User/process identification (user identification, program name, file name, program location in memory, and so forth)

4. Pertinent system registers (processor flags, program counter, and so forth) before and/or after error as applicable

5. Disposition of event (retry count, recovered or not, the point in the retry algorithm where recovery was affected, and so forth)

6. Other I/O activity at error time

Section 4 contains the following:

1. Controller name and/or address

2. Controller type

3. Name and value of all information available from the controller

Section 5 contains the following:

1. Name and value of all status information available from the unit

2. Function that was active at error time

3. Logical and physical address of the unit before error

4. Logical and physical address of the unit at error

5. Transfer size and starting memory location of I/O if applicable

Section 6 contains unit activity since start-up.

The default radix in these entries is decimal; however, some entries may have numbers displayed in octal or binary.

5.2 TOPS-10 ENTRIES

The following sections list both the FULL and SHORT versions of the entries that TOPS-10 can record in its system event file.

5.2.1 System Reload

The monitor generates a System Reload entry into the system event file whenever it is loaded. Note that HALT, STOP, and CPU stopcode information is also recorded in this entry, if applicable.

FULL

```
**********************************************
SYSTEM RELOAD
 LOGGED ON  5-Aug-80 AT  0:16:39      MONITOR UPTIME WAS  0:00:38
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 190.
**********************************************
CONFIGURATION INFORMATION
        SYSTEM  NAME:                RZ064A KL #1026/1042
        MONITOR BUILT ON:        07-23-80
        CPU SERIAL #:            1026.
        STATES WORD:             771165,0
        MONITOR VERSION  %701(0)

RELOAD BREAKDOWN
        CAUSE:                   SCHED
        COMMENTS                 ;PUT 1
MEMORY ON-LINE AT RELOAD:
FROM:  0 P  TO:  2048 P
```

SHORT

```
SEQ    TIME     5-Aug-80

190.  0:16:39 RELOAD OF     RZ064A KL #1026/1042 VERSION (70100)
                    BUILT ON 07-23-80 REASON SCHED
```

5.2.2 Non-Reload Monitor Error

Each time a JOB or DEBUG stopcode occurs, the monitor records the information as a Non-Reload Monitor Error in the system event file. The JOB stopcode endangers the integrity of the job currently running; therefore, the monitor aborts the current job, then continues. A DEBUG stopcode is not immediately harmful to any job or the system; therefore, the monitor prints the stopcode message on the operator's terminal (CTY) and then continues processing.

FULL

```
**********************************************
NON-RELOAD MONITOR ERROR
 LOGGED ON  5-Aug-80 AT 10:51:49      MONITOR UPTIME WAS  2:26:26
        DETECTED ON SYSTEM # 1042.
        RECORD SEQUENCE NUMBER: 863.
**********************************************
        SYSTEM NAME:         RZ64C  KL #1026/1042
        SYSTEM SERIAL #:        1026.
        MONITOR DATE:   07-23-80
        MONITOR VERSION  %701(0)
        STOPCD NAME:    BAZ
        RESULT:
                JOB #:  6.
                USER'S ID:      [1,2]
                TTY NAME:          470
                PROGRAM NAME:   ACTDAE

        CONTENTS OF AC'S AT STOPCD:

    0:  20,0
    1:  777642,377507
    2:  0,100
    3:  5777,371000
    4:  526200,340000
    5:  664145,663167
    6:  440004,0
    7:  0,50
   10:  0,0
   11:  0,505273
   12:  0,250255
   13:  47040,1
   14:  0,1
   15:  0,1
   16:  0,4
   17:  0,146

        PI STATUS:      440004,0
```

SHORT

```
SEQ    TIME     5-Aug-80

863. 10:51:49 STOPCD    BAZ ON CPU SERIAL # 1026 FOR JOB # 6 ON    470
                    USER WAS [1,2] RUNNING ACTDAE
```

5.2.3 Crash Extract

A Crash Extract becomes a part of the system event file whenever the program DAEMON starts. When DAEMON starts, it checks the system search list for a CRASH.EXE file. If it finds one, it extracts the information and appends it to the system event file.

NOTE

It is strongly recommended that, each time the monitor is started, you save a dump as a CRASH.EXE file so that DAEMON/SPEAR can provide a complete picture of system activity. You can do this by saving each monitor core image (dumping the crash) after each run; that is, before PM or CM periods, before scheduled reloads, after stand-alone periods, and so forth. To save core-image, use the /D command to MONBTS.

Because DAEMON extracted the information from a saved crash, the date and time and the monitor uptime in the header are the last values recorded by the monitor before the crash.

FULL

```
*********************************************
  ** THIS ENTRY COPIED FROM A SAVED CRASH **
CRASH EXTRACT
 LOGGED ON  5-Aug-80 AT  0:11:25      MONITOR UPTIME WAS 11:50:09
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 187.
*********************************************
CRASH.EXE READ FROM:    DSKB
SYSTEM WIDE ERROR COUNT:        162.
CONTENTS OF GETTAB'D ITEMS:
                  TIME OF DAY:  0:11:24          SYSTEM MEMORY SIZE: 336000          LAST ADDR POKED: 13415
   # JOBS LOGGED
IN: 26.
             DEBUG STATUS WORD: 0,0          START OF MONITOR HIGH SEG: 2501000          #OF WORDS OF CORE: 4000000
               UPTIME IN TICKS: 2556574.                # UNREC EXEC PDL OV: 0.      # RECOVERED EXEC PDL OV: 0.
              SWAP ERROR COUNT: 0.       DISABLED HARDWARE ERROR COUNT: 20.              LAST STOPCD:
               # DEBUG STOPCDS: 0.                    # JOB STOPCDS: 0.          LAST STOPCD-JOB NUMBER: 0
      LAST STOPCD-PROGRAM NAME:                      LAST STOPCD-UUO: 0,0          LAST STOPCD-P,PN: [0,0]


PARITY ERROR INFORMATION:
          TOTAL MEM PAR ERRORS: 0.       TOTAL SPURIOUS PARITY ERRORS: 0.          MULTIPLE PARITY ERRORS: 0.
             LAST PARITY ADDR: 0                  LAST PARITY WORD: 0,0                LAST PARITY PC: 0
  HIGHEST ADDR OF PARITY ERROR: 0       ADDRESS IN SEGMENT OF PAR ERR: 0          # PAR ERRORS THIS SWEEP: 0.
                  # SWEEPS: 0.              USER ENABLED ERRORS: 0.             LOGICAL AND  OF ADDR: 0,0
          LOGICAL AND OF DATA: 0,0              LOGICAL OR OF ADDR: 0,0             LOGICAL OR OF DATA: 0,0
  COUNT OF SPUR CHANNEL ERRORS: 0.

SYSTEM RESPONSE INFORMATION:
                    MEAN//ST.DEV. RESP/MIN   # of RESP
   'til TTY output:   2.1//0.3       15.7      11135.
   'til TTY input:    2.0//0.4       15.4      10944.
   'til requeued:     11.3//1.2      4.0       2853.
   'til lst of above: 1.0//0.3       17.2      12210.
   'TIL JOB STARTED:  0.6//0.3       18.1      12817.

      TOTAL UUO COUNT:              5474654.       AVG. = 7710.8 PER MIN.
TOTAL JOB CONTEXT SWITCH COUNT:     1736842.       AVG. = 2446.3 PER MIN.
          SUM TTY OUT UUO RES: 1400074.                    NUM TTY OUT UUO: 11135.
        HI-SUM SQ TTY OUT UUO: 0.            LO-SUM SQ TTY OUT UUO: 10710992548.          SUM TTY INP UUO: 1283336.
           NUMBER TTY INP UUO: 10944.        HI-SUM SQ TTY INP UUO: 9.          LO-SUM SQ TTY INP UUO: 22975521908.
          SUM QUANTUM REQ RES: 1936068.           NUMBER QUANTUM REQ RES: 2853.       HI-SUM SQ QUANTUM RES: 6.
    LO-SUM SQ QUANTUM REQ RES: 22256557290.             SUM ONE OF ABOVE: 696571.        NUMBER ONE OF ABOVE: 12210.
       HI-SUM SQ ONE OF ABOVE: 0.          LO-SUM SQ ONE OF ABOVE: 3363170363.             SUM CPU RES: 473960.
              NUMBER CPU RES: 12817.              HI-SUM CPU RES: 0.             LO-SUM CPU RES: 3255471104.
```

```
                        UPTIME: 11:50:09                    LOST TIME:  0:11:48                 NULL TIME:  5:12:47
          OVERHEAD
TIME:  2:12:05                TOTAL UUO COUNT: 5474654.  TOTAL JOB CONTEXT SWITCH COUNT: 1736842.                  TOTAL NXM: 0
•
                    TOTAL SPUR NXM: 0.          # JOBS AFFECTED LAST NXM: 0.              FIRST ADDR LAST NXM: 0
```

SHORT

```
SEQ     TIME      5-Aug-80

187.  0:11:25 CRASH EXTRACT-STOPCD WAS FOR JOB 0 UUO WAS 0,0
                    SYSTEM WIDE ERROR COUNT WAS 162
```

5.2.4 Data Channel Error

When a channel detects an error or a device connected to a channel detects an error during a data transfer, the monitor logs a Data Channel Error into the system event file. The entry is made at the time of first error; thus, the entry can be a soft or a hard error. Because the monitor programs the channel to stop when it encounters an error (except on the last retry), this entry gives valuable information about the word in error and its address, whether or not the error was detected by the channel.

The Data Channel Error is generated only for DF10 data channels and is not generated for devices using the KL10 internal channels (RH20).

FULL

```
**********************************************
DATA CHANNEL ERROR
 LOGGED ON  1-Oct-80 AT  9:03:12       MONITOR UPTIME WAS  1:02:10
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 3122.
**********************************************
DATA CHANNEL ERROR TOTALS
        NXM'S AND OVERRUNS:      1.
        MEM PE SEEN BY CHANNEL: 0.
        CONTROLLER DATA PE
        OR CCW TERM CHK FAILS:   0.

CHANNEL COMMAND LIST BREAKDOWN
        DEVICE  USING CHANNEL:   RPA5
        INITIAL CONTROL WORD:    0,454
        TERMINATION WD WRITTEN: 11323,313216
        EXPECTED TERM. WORD:    11323,313413
        CHANNEL COMMAND LIST:    0,454
                                 774003,313213
                                 0,0
        3RD FROM LAST DATA WORD:0,0
        2ND FROM LAST DATA WORD:0,0
        LAST DATA WORD XFERRED: 0,0
```

SHORT

```
SEQ     TIME      1-Oct-80

3122.   9:03:12 RPA5 CHANNEL ERROR COUNTS: NXM/MPE/DPE 1/0/0
                    WRITTEN TERM WD = 11323,313216
                    EXPECTED TERM WD = 11323,313413
```

5.2.5 DAEMON Started

The monitor logs this entry into the system event file each time DAEMON is started, either after a system reload or a restart of DAEMON. If DAEMON is modified at the site, the customer version number should be edited to track the modifications.

FULL

```
**********************************************
DAEMON STARTED
 LOGGED ON  5-Aug-80 AT  0:16:30      MONITOR UPTIME WAS  0:00:28
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 184.
**********************************************
        DAEMON VERSION  20(757)
```

SHORT

```
SEQ    TIME     5-Aug-80

184.  0:16:30 DAEMON STARTED--VERSION 20(757)
```

5.2.6 MASSBUS Disk Error

Any time the monitor detects an error in any portion of the MASSBUS system (either hardware or software), DAEMON is called to collect and record all pertinent hardware and software information in the error file.

In this entry, the MEDIA ID is the value given to the disk when structured with ONCE or TWICE. The STR ID is the logical name of the media such as DSKB0. Both are recorded in the HOME block. The LBN (logical block number) is the location of the first block in the transfer. If LBN n, n+1, n+2, and n+3 were transferred, it is possible that LBN n, n+1, and n+2 are alright, but LBN n+3 is bad. This value is broken into either the cylinder #, surface #, and sector # (for disks) or the track # and sector # (for RS04s) to determine the physical location of the failure.

The OPERATION AT ERROR is the text translation of the last command issued to the device before the error was detected (presumably the command that caused the error). The text translation should match the translation of the bits in DATAI RHCR AT ERROR for the RH10 and DATAI PTCR AT ERROR for an RH20. If the information does not match, look for an error in the control bus.

NOTE

> Because of dual-port capabilities for disk drives, the physical device number can change according to the port assignment. For example, on dual-ported drives, one drive may be RPA3 on PORT A and RPC3 on PORT B.

MASSBUS devices store and make available significant amounts of device-dependent information. The contents of all registers are listed in the entry both at error time and after the last retry, along with the difference between the two values. Text translations are always from the AT ERROR value with the exception of the OFFSET Register; offsets are not normally used.

Note that software errors are checked only after the hardware has completed the transfer without a detected error.

```
                                        FULL

*********************************************
MASSBUS DISK ERROR
 LOGGED ON  4-Aug-80 AT 13:36:27      MONITOR UPTIME WAS  1:15:13
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 2.
*********************************************
        UNIT ID:          RPB5
        UNIT TYPE:        RP06
        UNIT SERIAL #:    0058.
        MEDIA ID:               !
        STR ID:
        USER'S ID:        [1,2]
        USER'S PGM:       PULSAR
        USER'S  FILE:     .
        LBN AT START OF XFER:    1.  =
        CYL:   0.         SURF:   0.      SECT:   1.
        OPERATION AT ERROR:       DEV.AVAIL., GO +  READ DATA(70)
        ERROR:  RECOVERABLE DRIVE EXCEPTION, IN CONTROLLER CONI
                DCK, IN DEVICE ERROR REGISTER
        REMAINING ENTRIES IN
        UNIT'S BAT BLOCK:         UNKNOWN
        RETRY COUNT:      16.
CONTROLLER INFORMATION:
        CONTROLLER:        RH20  #540
        CONI AT ERROR:   0,202415 = DRIVE EXCEPTION,
        CONI AT END:     0,2415 =  NO ERROR BITS DETECTED
          CHN STATUS AT ERROR:  500000,0 = NOT SBUS ERR,
          CHN STATUS AT END:    400000,0 =  NO ERROR BITS DETECTED
        DATAI PTCR AT ERROR:    732605,177771
        DATAI PTCR AT END:      732605,177771
        DATAI PBAR AT ERROR:    723617,605735
        DATAI PBAR AT END:      723617,605735
DEVICE REGISTER INFORMATION:
                AT ERROR        AT END          DIFF.   TEXT
        CR(00): 4070            4070              0     DEV.AVAIL., READ DATA(70)
        SR(01): 51700           11700           40000   ERR,MOL,PGM,DPR,DRY,VV,
        ER(02): 100000             0            100000  DCK,
        MR(03): 400             400                0    ZERO DET,
        AS(04):    0               0               0
        DA(05): 2               2                  0    D. TRK = 0, D.SECT. = 2
        DT(06): 24022           24022              0
        LA(07): 240             240                0
        SN(10): 130             130                0
        OF(11): 116000          100000          16000   AT END:SIGN CHANGE, OFFSET = NONE
        DC(12):    0               0               0    0.
        CC(13):    0               0               0    0.
        E2(14):    0               0               0    NO ERROR BITS DETECTED
        E3(15):    0               0               0    NO ERROR BITS DETECTED
        EP(16):    0               0               0
        PL(17):    0            177771          177771

                                SHORT

SEQ    TIME     4-Aug-80

2. 13:36:27 RPB5 RP06 SERIAL # 0058. CONI RH = 0,202415
                  CHNSTS1 = 500000,0 SR = 51700 ER = 100000
                  CYL/SURF/SEC= 0./0./1. RETRIES: 16
```

5.2.7 DX20 Device Error

The monitor records a DX20 Device Error in the system event file when it detects an error in any portion of the MASSBUS system connected to the DX20 channel interface.

In this entry, the MASSBUS REGISTER INFORMATION contains the nonzero contents of all registers both at error time and after the last retry. Also the SB (sense bytes) describe the device type and status of the device (in octal) attached to the DX20.

FULL

```
**********************************************
DX20 ERROR
 LOGGED ON  8-Sep-80 AT 22:41:10      MONITOR UPTIME WAS  3:23:01
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 1471.
**********************************************
        UNIT NAME:      RNB0
        UNIT TYPE:      RP20
        VOLUME ID:      SCR00
        LOCATION:  LBN = 463454.
        OPERATION AT ERROR:     GO+ READ DATA(70)
        USER'S P,PN     [10,664]
        USER'S PGM:             FILCHK
        USER'S FILE:
        RETRIES PERFORMED:      1.
        ERROR:  RECOVERABLE DRIVE EXCEPTION,CHN ERROR, IN CONTROLLER CONI
                MPER, IN DEVICE ERROR REGISTER
CONTROLLER INFORMATION:
        CONTROLLER:     RH20 # 554  DX20 #:0
        DX20 U-CODE VERSION:    0(4)
        CONI AT ERROR:  540100,222615 = DRIVE EXCEPTION,CHN ERROR,
        CONI AT END:    540100,222615 = DRIVE EXCEPTION,CHN ERROR,
        DATAI PTCR AT ERROR:    732600,171771
        DATAI PTCR AT END:      732600,171771
        DATAI PBAR AT ERROR:    723617,777417
        DATAI PBAR AT END:      723617,777417
CHANNEL INFORMATION:
        CHAN STATUS WD 0:       200000,464      CW1:  414721,475143  CW2:  420000,721000
        CHN STATUS WD 1:        540100,466 = NOT SBUS ERR,NOT WC = 0,LONG WC ERR,
        CHN STATUS WD 2:        414720,721143
MASSBUS REGISTER INFORMATION:
                AT ERROR        AT END          DIFF.   TEXT
        CR  00: 70              70                0     READ DATA(70)
        SR  01: 170000          170000            0     ATA,ERR,LINK PRESENT,MP RUN,
        ER  02: 10600           10600             0     ERROR CLASS = 1,SUBCLASS = 1 ;MPER,
                                                        = UNUSUAL STATUS FROM INITIAL SELECTION SEQUENCE
        MR  03: 4               4                 0     MICRO P START,
        AS  04: 1               1                 0
        HR  05: 16005           16005             0     HEAD#: 28. RECORD#:5.
        DT  06: 10061           10061             0
        ESSI20: 1               1                 0     STATUS INDEX FOR ESR0&1=1
                                                        DEV STATUS:  NO ERROR BITS DETECTED
        ASYN21:    0               0              0     CTRL: 0 DRIVE: 0
                                                        DEV STATUS:  NO ERROR BITS DETECTED
        FA  22:    0               0              0     ARGUMENT:0 FLAGS:  NO ERROR BITS DETECTED
        DN  23: 30              30                0     CTRL: 1 DRIVE: 10
        CL  24: 1151            1151              0     CYL: 617.
        HR  25: 16005           16005             0     HEAD#: 28. RECORD#:5.
        ESR026: 100151          100151            0
        ESR127: 56123           56123             0
```

```
DIAG30: 161231          161231              0
DIAG31: 133025          133025              0

RP20 SENSE BYTES LISTED IN HEXIDECIMAL
BYTE 00:  08     = DATA CHK,
BYTE 01:  00     =  NO ERROR BITS DETECTED
BYTE 02:  40     = CORRECTABLE,
BYTE 03:  06     ; RESTART COMMAND
BYTE 04:  80     ; PHYSICAL DRIVE ID
BYTE 05:  69
BYTE 06:  5C    LOGICAL CYL. ADDR. = 617.
                LOGICAL HEAD = 28.
BYTE 07:  53    = FORMAT 5 , MESSAGE 3
        DATA FIELD CORRECTABLE DATA AREA
CYL OF LAST SEEK ADDRESS:       617.
SURF. OF LAST SEEK ADDRESS:     28.
RECORD # IN ERROR:      4.
SECTOR # IN ERROR:      20.
# OF BYTES XFERRED:     576. BYTES
ERROR DISPLACEMENT:     553. BYTES
ERROR PATTERN:  100000
```

SHORT

```
SEQ    TIME     8-Sep-80

1471. 22:41:10 RNB0 SCR00: RP20  CONI=540100,222615 CHNSTS1=540100,466
                    SR=0,170000 ER=0,10600 SENSE BYTE 7:  53
                    LBN: 463454. RETRIES: 1
```

5.2.8 Software Event

This entry is logged into the system event file when a user with special privileges, for example the system operator, issues one of the following monitor calls: POKE, RTTRP, SNOOP, or TRPSET. These monitor calls have the following effect:

1. POKE changes the value of a word in monitor core.

2. RTTRP connects a device to or releases it from the realtime interrupt facility.

3. SNOOP allows privileged programs to insert breakpoints in the monitor that trap to a user program. The user program must be locked in core when the trap occurs. This feature is used for fault insertion, performance analysis, and trace functions.

4. TRPSET prevents jobs other than the calling job from running. You can use this call to guarantee fast response to realtime interrupts.

For more information on monitor calls, refer to the TOPS-10 Monitor Calls Manual.

FULL

```
**********************************************
SOFTWARE EVENT
 LOGGED ON 14-Jul-80 AT  8:56:45      MONITOR UPTIME WAS  0:42:42
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
        EVENT TYPE: POKE
        JOB #: 46.
        USER PPN: [10,5324]
        LOCATION OF USER:
                NODE:26
                LINE:154
                TTY154
        PROGRAM: SPICE
STORED DATA VALUES:
        0,34030
```

SHORT

```
SEQ     TIME     14-Jul-80

1.  8:56:45 SOFTWARE EVENT  TYPE:   POKE  BY JOB 46 USER WAS [10,5324]
                    RUNNING SPICE AT NODE: 26 LINE: 154 TTY154
```

5.2.9 Configuration Status Change

The monitor records a Configuration Status Change whenever the system operator marks disk units and sections of core memory on-line or off-line. The system operator uses the either the CONFIG program or the SET command to change the system configuration. These tools are useful because they can prevent further errors to users until a unit can be repaired, or they can be used to split and later join dual CPU systems. For more information on the CONFIG program, refer to the file CONFIG.DOC.

With the SET command, the system operator can also give a 2-character reason for the change in configuration. Any two characters can be used, but the following codes are suggested:

PM - preventive maintenance

CM - corrective maintenance

DN - unit is down

OT - other

CAUTION

When the system operator adds memory to the system, the monitor checks to verify the availability of the specified addresses. Mistakes are reported at the operator's terminal (CTY), but the error logging system treats these as valid NXMs and generates the appropriate NXM reports. You can identify a NXM report of this type because no physical memory is placed off-line and the user's directory is [1,2].

FULL

```
**********************************************
CONFIGURATION STATUS CHANGE
 LOGGED ON  4-Aug-80 AT 14:06:05      MONITOR UPTIME WAS  1:44:50
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 15.
**********************************************
COMMAND:DETACH
        DEVICE:RNA0
```

SHORT

```
SEQ    TIME     4-Aug-80

15. 14:06:05 CONFIGURATION CHANGE  DETACHED RNA0
```

5.2.10 System Log Entry

The monitor records a System Log Entry when the system operator enters a log entry into the system event file with the OPR program.

A system operator, or anyone with operator privileges, can make an entry into the system event file by doing the following:

1. Run the OPR program

   ```
   .OPR (RET)
   OPR>
   ```

2. When you see the prompt, specify the REPORT command:

   ```
   OPR>REPORT
   ```

3. Use the following syntax:

   ```
   OPR>REPORT user text (RET)
   ```

   where user can be directory name and/or device name and text can be a single-line or multiple-line response.

For more information on OPR, refer to the TOPS-10 Operator's Command Language Reference Manual.

FULL

```
**********************************************
SYSTEM LOG ENTRY
 LOGGED ON 15-Sep-80 AT 10:40:12      MONITOR UPTIME WAS  5:30:10
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 37.
**********************************************
ENTRY CREATED BY:
        JOB #, TTY #:    77,502
        P,PN:            [27,2617]
        WHO:             MASELL
        DEV:             TTY
        MESSAGE:         : THIS IS A TEST.
```

SHORT

```
SEQ    TIME    15-Sep-80

37. 10:40:12 SYSTEM LOG ENTRY BY MASELL FOR DEVICE TTY ON TTY # 502
                    MESSAGE: : THIS IS A TEST.
```

5.2.11 Software Requested Data

At certain times during system operation, some problems can arise that are not easily understood. Most frequently, the source of the failure is a hardware failure but the failure is detected by the software. In order to troubleshoot this type of failure, you may require additional data from the monitor. You can obtain this information by patching the monitor to collect the information at the proper point and passing it to the system event file for listing.

CAUTION

> Patching a monitor can easily produce drastic, undesired results such as loss of customer data, system crashes, and so forth. Be EXTREMELY CAREFUL and enlist the help of someone who is familiar with the monitor structure and internal workings.

SPEAR lists the information in this entry in octal and sixbit.

```
**********************************************
SOFTWARE REQUESTED DATA
 LOGGED ON  4-Jan-81 AT  6:50:34      MONITOR UPTIME WAS  3:13:34
        DETECTED ON SYSTEM # 2263.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
        OCTAL VALUE     SIXBIT VALUE
        504554,545700   HELLO
        675762,544400   WORLD
        123456,654321   *<NUC1
        654321,123456   UC1*<N
        555762,450063   MORE S
        517042,516400   IXBIT
```

5.2.12 Magtape System Error

The monitor records any magtape errors it detects as a Magtape System Error. Errors that are non-recoverable are classified as HARD, recoverable errors are classified as SOFT.

If the monitor detects a data channel error, it records the appropriate information under error code 6 or Data Channel Error. After a user issues an UNLOAD command or UUO, the monitor records the performance statistics for the tape, including the total number of characters transferred and the number of errors (soft read, soft write, hard read, hard write) encountered.

Note that if someone mounts unlabelled tapes without specifying any kind of ID, there will be no MEDIA identified in the error file.

FULL

```
**********************************************
MAGTAPE SYSTEM ERROR
 LOGGED ON  8-Sep-80 AT  9:05:11      MONITOR UPTIME WAS  0:57:06
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 11.
**********************************************

        UNIT NAME:      MTB261
        UNIT TYPE:      TU70
        USER'S ID:      [1,2]
        USER'S PROGRAM: BACKUP
        MEDIA   ID:
        LOCATION
         OF FAILURE:    RECORD:  0.  OF FILE:  5.
        POSITION
        BEFORE ERROR:   RECORD:  262143.  OF FILE:  5.
        CHAR. INTO RECORD:  5458276711.
        OPERATION:      S.I., IMM, BYTE,  DEV.CMD.:   READ
        STATUS:  CU IS:  TX01,7 & 9 TRK NRZI DEVICE IS:   WRITE ENB
        THIS ENTRY CREATED AS A RESULT OF A 'HUNG DEVICE'

        ERROR:  NON-RECOVERABLE RUNNING,CSR, IN DX10 CONI, UNIT EXC, IN ICPC+1
        ***AS OF DX10 MICROCODE VERSION 4(0), RECOVERABLE ERRORS
        ARE NOT REPORTED TO MONITOR IF DX10 MICROCODE ERROR
        RETRY IS ENABLED.***

        RETRY COUNT:    0.  ----
 CONTROLLER INFORMATION:
        CONTROLLER:     DX10 #0
        CONI AT ERROR:  1,422034 = RUNNING,CSR,
        CONI AT END:    1,422034 = RUNNING,CSR,
        ICPC+1 AT ERROR:        32201,1 = UNIT EXC,
        ICPC+1 AT END:  32201,1 = UNIT EXC,
        ICPC+2 AT ERROR:        710040,457
        ICPC+2 AT END:          710040,457
       REGISTER AT ERROR        AT END          DIFF            TEXT
         B CNT: 0,0             0,0             0,0
        TAGBUS: 10,2            10,2            0,0     OPL OUT,
           DAC: 1,226233        1,226233        0,0
           REV: 150000,2660             150000,2660             0,0
        CPMA&MD:        0,0
        DR:     0,0
DEVICE INFORMATION:  *IN OCTAL BYTES*
     SENSE BYTE AT ERROR        AT END          DIFF            TEXT
           0-3: 0 102 3 0       0 102 3 0       0 0 0 0 FILE PROT, TIE = 00000011
           4-7: 0 100 5 0       0 100 5 0       0 0 0 0  NO ERROR BITS DETECTED
          8-11: 0 0 0 0         0 0 0 0         0 0 0 0 NO ERROR BITS DETECTED
         12-15: 0 305 213 0     0 305 213 0     0 0 0 0
         16-19: 0 232 0 35      0 232 0 35      0 0 0 0
         20-23: 0 0 0 0         0 0 0 0         0 0 0 0

CHAN CMD LIST:
        CPC:    0,0
        CMDS:   262020,20001
                140000,454
```

SHORT

```
SEQ    TIME     8-Sep-80

11.  9:05:11 MTB261 TU7x DX10 CONI = 1,422034 ICPC+1 = 32201,1
                  SB(0-3) = 0/102/3/0 FILE/REC = 4/0 RETRIES: 0 HARD
```

5.2.13 Front End Device Report

You will find a Front End Device Report in the system event file when the front end passes a packet of error information to the monitor. This information contains errors detected by the front end and KLCPU hardware and software. If the device being reported on is unknown to SPEAR, the entry is reported in octal.

FULL

```
**********************************************
FRONT END DEVICE REPORT
 LOGGED ON  3-Nov-80 AT  9:44:10      MONITOR UPTIME WAS 2 DAYS 14:37:29
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 67.
**********************************************
        CPU #,DTE #:              0,0
        FE SOFTWARE VER:          0.
        DEVICE: KLCPU
        STD. STATUS:    100 = ERROR LOG REQUEST,
        KL RELOAD STATUS FROM FRONT END:  0      =  NO ERROR BITS DETECTED
```

SHORT

```
SEQ    TIME     3-Nov-80

67.  9:44:10 KLCPU STD STAT=100 RELOAD STAT=0
```

5.2.14 Front End Reload

The monitor logs a Front End Reload entry into the system event file when it determines that one of its front ends (attached to a DTE on a KL10 only) has crashed and has attempted to reload. Before rebooting the front end, the monitor dumps the crashed front end's core image to a disk file for later analysis.

FULL

```
**********************************************
FRONT END RELOAD
 LOGGED ON  9-Sep-80 AT  0:01:05      MONITOR UPTIME WAS  0:01:57
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 1494.
**********************************************
        CPU # :,,FRONT END #:   1,1
        STATUS AT RELOAD:       DUMP FAILED,RELOAD FAILED,ROM DIDN'T ACK THE -10,
        RETRIES:        0
```

SHORT

```
SEQ    TIME     9-Sep-80

1494.  0:01:05 FRONT END RELOAD ON PDP11 #1 RELOAD STATUS: 104400
                  RETRIES: 0
```

5.2.15 KS10 Halt Status Block

The monitor records a KS10 Halt Status Block entry into the system event file when the KS10 microcode executes a HALT stopcode. A snapshot of the condition of the system is taken just prior to the HALT, and this information is written as the entry.

FULL

```
**********************************************
KS10 HALT STATUS BLOCK
 LOGGED ON  9-Feb-81 AT 14:21:55      MONITOR UPTIME WAS  0:01:12

        DETECTED ON SYSTEM # 4145.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
HALT STATUS CODE:       2
PROGRAM  COUNTER:       1000
HALT STATUS BLOCK
        MAG:    0,2
        PC:     0,1000
        HR:     777756,4
        AR:     0,0
        ARX:    377777,777777
        BR:     0,1000
        BRX:    254000,1000
        ONE:    241200,200000
        EBR:    0,1
        UBR:    0,31463
        MASK:   774777,470177
        FLAGS,,PAGE FAIL WORD:  0,1
        PI STATUS:      400060,120000
        XWD1:   500101,553000
        T0:     777777,777777
        T1:     4000,0
        VMA:    0,177
```

SHORT

```
SEQ    TIME     9-Feb-81

1. 14:21:55 HALT STATUS CODE =  PC = 0,1000 HR = 254000,1000
                    PAGE FAIL = 4000,0 PI = 0,177 FLAGS,,VMA = 0,0
```

5.2.16 Magtape Statistics

Each time an UNLOAD UUO or monitor command is given to a tape drive the monitor creates a Magtape Statistics entry. The same information is printed in summary form on both the user's terminal and the operator's terminal (CTY).

In this entry, the REEL IDENTIFICATION is the name supplied to the monitor at the time the tape was mounted. It has nothing to do with any label information found on the tape. The CHARS READ is the number of characters or frames of tape read on this unit since the last UNLOAD command was issued to this unit. The CHARS WRITTEN is the number of characters or frames of tape written on this unit since the last UNLOAD command was issued.

FULL

```
**********************************************
MAGTAPE STATISTICS
 LOGGED ON  4-Aug-80 AT 13:40:05      MONITOR UPTIME WAS  1:18:50
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 5.
**********************************************
MAGTAPE STATISTICS
        UNIT NAME:                MTB261
        REEL IDENTIFICATION:
        USER'S P,PN:              1,2
        CHARS READ:               2720.
        CHARS WRITTEN:            0.
        SOFT READ ERRORS:         0.
        HARD READ ERRORS:         1.
        SOFT WRITE ERRORS:        0.
        HARD WRITE ERRORS:        0.
```

SHORT

```
SEQ    TIME     4-Aug-80

5. 13:40:05 MTB261 STATISTICS  READ CH/H/S: 2720/1/0 WRITE CH/H/S: 0/0/0
```

5.2.17 Disk Statistics

This entry reports the performance of each disk unit since the monitor was loaded. It is useful for computing the disk error rate and disk throughput. This information is usually not recorded by DAEMON in the system event file because it takes up a great deal of space. Installations that want this entry should reassemble DAEMON with the conditional assembly switch FTUSN set.

The monitor records this entry type for each disk unit on the system each hour. You can find the same type of information for each monitor run in the Crash Extract entry (Section 5.2.3).

FULL

```
**********************************************
  ** THIS ENTRY COPIED FROM A SAVED CRASH **
DISK STATISTICS
 LOGGED ON  5-Aug-80 AT  0:11:25      MONITOR UPTIME WAS 11:50:09
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 188.
**********************************************
```

					*** ERROR COUNTS						***
UNIT	PACK	SEEKS	BLOCKS READ	BLOCKS WRITTEN	! DATA	DEVICE	SEEK	HUNG	SAT	RIB	! CHECKSUM
RPA0		0	0	0	0	0	0	0	0	0	0
RPA1		0	0	0	0	0	0	0	0	0	0
RPA2		2145	26986	2	17	0	0	0	0	0	0
RPA3		3758	9800	2	5	0	0	0	0	0	0
RPA4		0	0	0	0	0	0	0	0	0	0
RPA5		4614	13740	1355	1	0	0	0	0	0	0
RPA6		0	0	0	0	0	0	0	0	0	0
RPA7		0	0	0	0	0	0	0	0	0	0
RPB0	BLKX0	549	4037	44	0	0	0	0	0	0	0
RPB1		2166	35998	26	0	0	0	0	0	0	0
RPB2	DSKC0	92665	2149512	1306027	2	0	0	0	0	0	0
RPB3		0	0	0	0	0	0	0	0	0	0
RPB4	FTN0	3577	39762	58	1	0	4	0	0	0	0
RPB5		2032	14114	4940	1	0	0	0	0	0	0
RPB6	GAL00	592	4550	61	0	0	0	0	0	0	0
RPB7		0	0	0	0	0	0	0	0	0	0
RPD0	DSKC1	131943	1576709	1162512	0	0	0	0	0	0	0
RPD1		0	0	0	0	0	0	0	0	0	0
RPD2		0	0	0	0	0	0	0	0	0	0
RPD3		12525	331776	56	1	0	0	0	0	0	0
RPD4	DSKB0	56962	1102911	863534	5	0	0	0	0	0	0
RPD5		0	0	0	0	0	0	0	0	0	0
RPD6		0	0	0	0	0	0	0	0	0	0
RPD7	DSKB1	25648	100568	18486	1	0	0	0	0	0	0
RNA0	DSKR0	17196	28596	11667	0	0	0	1	0	0	0
RNA1	DSKP0	3029	3884	422	0	0	0	0	0	0	0
RNA2	BLKK0	550	843	4	0	0	0	0	0	0	0
RNA3	!	2	2	0	0	0	0	0	0	0	0

SHORT

```
SEQ    TIME     5-Aug-80

188.  0:11:25 DISK STATISTICS
```

5.2.18 DL1Ø Communications Error

The monitor records a DL1Ø Communications Error into the system event file when the DL1Ø detects an error on the communications link.

FULL

```
**********************************************
DL1Ø COMMUNICATIONS ERROR
 LOGGED ON  4-Aug-8Ø AT 16:45:Ø9      MONITOR UPTIME WAS  4:23:54
        DETECTED ON SYSTEM # 1Ø26.
        RECORD SEQUENCE NUMBER: 86.
**********************************************
        UNIT:           DC76
        DL1Ø PORT:      Ø
        ERROR:   NO ERROR BITS DETECTED

        11 PROGRAM NAME:        DC76
CONTROLLER INFORMATION:
        CONI DLC:       6Ø,2ØØ2Ø4 = P1 ENB,
        DATAI DLC:      Ø,75Ø =  NO ERROR BITS DETECTED
        CONI DLB (R=Ø): Ø,5Ø37
        CONI DLB (R=1): 4ØØØØ,6ØØ5
        CONI DLB (R=2): 2ØØØ,46Ø1
        CONI DLB (R=3): 577777,464ØØ
        DATAI DLB (R=1)(MB):    Ø,Ø
```

SHORT

```
SEQ    TIME     4-Aug-8Ø

86. 16:45:Ø9 DL1Ø ERROR ON PDP11 # Ø CONI DLC = 6Ø,2ØØ2Ø4
                    DATAI DLC = Ø,75Ø
```

5.2.19 KL1Ø Parity or NXM Interrupt

The monitor records a KL1Ø Parity or NXM Interrupt in the system event file when the KL1Ø detects a parity error or an attempt to access a nonexistent memory location.

The PC AT INTERRUPT is the status of the program counter at the time of the parity or nonexistent memory interrupt. The CONI PI AT INTERRUPT is the status of the Priority Interrupt system at the time of the parity or nonexistent memory interrupt.

```
                                    FULL

*********************************************
  ** THIS ENTRY COPIED FROM A SAVED CRASH **
KL10 PARITY OR NXM INTERRUPT
 LOGGED ON  2-Dec-80 AT  0:05:28      MONITOR UPTIME WAS 16:20:11
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 584.
*********************************************
ERROR DETECTED ON CPL0
PC AT INTERRUPT:        4000,566602
CONI PI AT INTERRUPT:   0,10377
        CONI APR AT INTERRUPT:  7760,2030 = NXM,SWEEP DONE,
        ERA:            200003,554255 = WD # 1  MEMORY READ
        BASE PHY. MEM ADDR.
         AT FAILURE:    3554255
SYSTEM MEMORY CONFIGURATION:

CONTROLLER:  #4  DMA20
        INTERLEAVE MODE:        4 WAY
        DMA:
        LAST ADDR HELD: 45220
        ERRORS DETECTED:        NONE

                                SHORT

SEQ    TIME     2-Dec-80

584.  0:05:28 PARITY OR NXM INTERRUPT ON CPL0 CONI APR = 7760,2030
                    CONI PI = 0,10377 RDERA = 200003,554255
                    PC AT INTERRUPT = 4000,566602DUMPING UNKNOWN ERROR IN OCTAL

ERROR CODE =    0
```

5.2.20 KS10 NXM Trap

When the KS10 detects a read on a nonexistent memory location, the monitor records a KS10 NXM Trap into the system event file. A trap stops execution during the current instruction.

```
                                    FULL

*********************************************
KS10 NXM TRAP
 LOGGED ON 22-Mar-81 AT  0:11:50      MONITOR UPTIME WAS  0:23:18
        DETECTED ON SYSTEM # 4608.
        RECORD SEQUENCE NUMBER: 1.
*********************************************
ERROR DETECTED ON CPS0
PC AT TRAP:     1,145267
CONI PI AT TRAP:        0,2377
        PAGE FAIL WORD: 200013,770000
        PAGE FAIL CODE: 20 = I-O NXM
PHYSICAL MEMORY ADDRESS AT TRAP:        0,0
USER'S ID AT TRAP:      [307,5515]
USER'S PROGRAM:         TSTUBA
# OF RECOVERABLE TRAPS: 0.
# OF NON-RECOVERABLE TRAPS:     0.

                                SHORT

SEQ    TIME     22-Mar-81

1.  0:11:50 NXM TRAP  PFW = 200013,770000 PMA = 0,0 NON
                    RECOVERABLE FAILURE  RETRYS: 31
                    USER AT TRAP [307,5515] RUNNING TSTUBA
```

5.2.21 KL10 or KS10 Parity Trap

The monitor records a KL10 or KS10 Parity Trap when either the KL10 or KS10 detects an internal parity error, not necessarily in memory.

In this entry, the PHYSICAL MEMORY ADDRESS AT TRAP gives the location of the parity error where the trap occurred.

FULL

```
**********************************************
KL10 OR KS10 PARITY TRAP
 LOGGED ON  4-Feb-81 AT 17:37:14      MONITOR UPTIME WAS  0:03:13
        DETECTED ON SYSTEM # 2136.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
ERROR DETECTED ON CPL0
PC AT TRAP:     316000,230
CONI PI AT TRAP:        0,377
PHYSICAL MEMORY ADDRESS AT TRAP:        547001,436241
USER'S ID AT TRAP:      [1,2]
USER'S PROGRAM:         KLPAR4
        PAGE FAIL WORD: 767000,241
        PAGE FAIL CODE: 36 = AR
        BAD DATA WORD:  252525,252525
        GOOD DATA WORD: 0,0
        DIFFERENCE:     252525,252525
        RECOVERY:       CRASH USER
        RETRY COUNT:
                W CACHE:        4.
                W-O CACHE:      0.ERROR DURING CACHE SWEEP TO CORE
# OF RECOVERABLE TRAPS: 0.
# OF NON-RECOVERABLE TRAPS:     3.
```

SHORT

```
SEQ    TIME     4-Feb-81

1. 17:37:14 PARITY TRAP  PFW = 767000,241 PMA = 547001,436241
                    NON RECOVERABLE FAILURE  USER AT TRAP [1,2]
                    RUNNING KLPAR4 RETRIES: 4
```

5.2.22 Memory Sweep for NXM

When the monitor detects an attempt to access a nonexistent memory location in user core, it scans core by doing a memory sweep, looking for more NXMs. The monitor then records the results of this scan as a Memory Sweep for NXM in the system event file.

The ADDRESSES DETECTED BY SWEEP gives you the locations, if any, of more attempts to access nonexistent memory locations.

FULL

```
**********************************************
MEMORY SWEEP FOR NXM
 LOGGED ON  1-Oct-80 AT  9:03:14      MONITOR UPTIME WAS  1:02:21
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 3124.
**********************************************
NXM CORE SWEEP TOTALS FOR CPL0
        REPRODUCIBLE:   0.
        NON-REPRODUCIBLE:       0.
        DETECTED BY DATA
         CHANNEL BUT NOT
         BY CPU:                20.
SWEEP INFORMATION:
        ERRORS DETECTED:        0.
        LOGICAL "AND" OF BAD
         PHYSICAL ADDRESSES:    777777,777777
        LOGICAL "OR" OF BAD
         PHYSICAL ADDRESSES:    0,0
MEMORY PLACED OFF-LINE:
```

SHORT

```
SEQ    TIME     1-Oct-80

3124.  9:03:14 NXM SWEEP ON CPL0 # OF ERRORS SEEN = 0
```

5.2.23 Memory Sweep for Parity

When the monitor detects a parity error on a read attempt, it sweeps memory looking for more of the same. The results of the sweep are recorded in the system event file as a Memory Sweep for Parity.

The SWEEP INFORMATION contains the number of words found with bad parity. It also contains the logical AND and logical OR of the bad addresses and bad contents.

FULL

```
**********************************************
MEMORY SWEEP FOR PARITY
 LOGGED ON  4-Nov-80 AT  8:39:53      MONITOR UPTIME WAS  0:35:34
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 2026.
**********************************************
DATA PARITY CORE SWEEP TOTALS FOR CPL0
        REPRODUCIBLE:   0.
        NON-REPRODUCIBLE:       0.
        USER ENABLED:   0.
        CORE SWEEPS:    1.
        DETECTED BY DATA
         CHANNEL BUT NOT
         BY CPU:                1.
SWEEP INFORMATION:
        ERRORS DETECTED:        0.
        LOGICAL "AND" OF BAD
         PHYSICAL ADDRESSES:    777777,777777
        LOGICAL "OR" OF BAD
         PHYSICAL ADDRESSES:    0,0
        LOGICAL "AND" OF BAD DATA:      777777,777777
        LOGICAL "OR" OF BAD DATA:       0,0
```

SHORT

```
SEQ    TIME     4-Nov-80

2026.  8:39:53 DATA PARITY CORE SWEEP FOR CPL0 # OF ERRORS SEEN = 0
```

5.2.24 CPU Status Block

The monitor records this entry into the system event file after recovering from a system crash. At the time of the crash, a snapshot is taken of the condition of all the components of the CPU (such as controllers, channels, RH20s, the pager, and so forth). When the system recovers, the monitor extracts this information from the CRASH.EXE file and places it in the system event file as a CPU Status Block.

This entry contains the condition of the registers and channels just prior to the crash. Also, the SBDIAG FUNCTIONS column contains the SBUS diagnostic functions.

FULL

```
**********************************************
  ** THIS ENTRY COPIED FROM A SAVED CRASH **
CPU STATUS BLOCK
 LOGGED ON  5-Aug-80 AT  0:11:25      MONITOR UPTIME WAS 11:50:09
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 185.
**********************************************
APRID = 231,342002
CONI APR = 7760,3
RDERA = 604000,7427
CONI PI = 0,10377
DATAI PAG = 701100,3
CONI PAG = 0,620001
CONI RH0 THRU RH7
        000000,,002445  000000,,006400  000000,,002445  000000,,002445
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
CONI DTE0 THRU DTE3
        000000,,020014  000000,,100000  000000,,100014  000000,,100014
EPT LOCATIONS 0 THRU 37 (CHANNEL LOGOUT AREA)
        200000,,000454  500000,,000456  600000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        200000,,000454  500000,,000455  600001,,457000  000000,,000000
        200000,,000454  500000,,000455  600001,,014660  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
EPT LOCATIONS 140 THRU 177 (DTE CONTROL BLOCKS)
        141000,,413160  241000,,223676  264000,,057516  000000,,000000
        000000,,000442  000000,,057054  000000,,000030  000000,,057136
        000000,,000000  000000,,000000  264000,,057556  000000,,000000
        000000,,000443  000000,,057053  000000,,000030  000000,,057166
        241000,,224302  341000,,224563  264000,,057616  000000,,000000
        000000,,000444  000000,,057052  000000,,000030  000000,,057216
        341000,,232743  141000,,224000  264000,,057656  000000,,000000
        000000,,000445  000000,,057051  000000,,000030  000000,,057246
UPT LOCATIONS 424 THRU 427 (UUO AREA)
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
UPT LOCATIONS 500 THRU 503 (PAGE FAIL AREA)
        000000,,000000  304000,,112667  004000,,566102  000000,,000000
AC BLOCK 6 LOCATIONS 0 THRU 3 AND 12
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000
AC BLOCK 7 LOCATIONS 0 THRU 2
        255000,,000000  000000,,640010  000000,,000000

SBDIAG FUNCTIONS
        CTRLR   FUNCTION 0      FUNCTION 1
        4       005740,,041736  000200,,000000
```

SHORT

```
SEQ     TIME      5-Aug-80

185.  0:11:25 CPU STATUS BLOCK  APRID = 231,342002 CONI APR = 7760,3
                    CONI PI = 0,10377 CONI PAG = 0,620001
                    DATAI PAG = 701100,3
```

5.2.25 Device Status Block

The monitor records this entry into the system event file after recovering from a system crash. At the time of the crash, a snapshot is taken of the condition of all the I/O devices (such as lineprinters, cardreaders, disk drives, and so forth). When the system recovers, the monitor extracts this information from the CRASH.EXE file and places it in the system event file as a Device Status Block.

FULL

```
**********************************************
  ** THIS ENTRY COPIED FROM A SAVED CRASH **
DEVICE STATUS BLOCK
 LOGGED ON  5-Aug-80 AT  0:11:25      MONITOR UPTIME WAS 11:50:09
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 186.
**********************************************
CONI 20 : 117,63202
CONI 24 : 0,32003
CONI 120 : 0,0
CONI 104 : 0,0
CONI 100 : 0,0
CONI 240 : 0,0
CONI 320 : 0,410000
CONI 324 : 770010,4100
CONI 150 : 3,0
CONI 124 : 0,2400
CONI 140 : 0,40
CONI 344 : 0,0
CONI 340 : 0,0
CONI 220 : 1,420004
CONI 170 : 0,0
CONI 174 : 0,0
CONI 270 : 0,0
CONI 274 : 4000,5
CONI 360 : 0,0
CONI 250 : 0,0
CONI 254 : 0,0
CONI 260 : 0,0
CONI 264 : 0,0
CONI 334 : 0,0
CONI 330 : 0,0
CONI 64 : 60,200224
CONI 60 : 0,5037
CONI 164 : 0,0
CONI 160 : 0,0
CONI 110 : 0,400000
CONI 154 : 2,0
CONI 234 : 0,0
CONI 230 : 307620,32400
CONI 144 : 0,0
DATAI 0 : 0,0
DATAI 170 : 0,0
DATAI 174 : 0,0
DATAI 270 : 0,0
DATAI 274 : 4003,3
DATAI 360 : 0,0
DATAI 250 : 0,0
DATAI 254 : 0,0
DATAI 260 : 0,0
```

```
DATAI 264 : Ø,Ø
DATAI 64 : Ø,77Ø
DATAI 6Ø : Ø,162
DATAI 164 : Ø,Ø
DATAI 16Ø : Ø,Ø
```

SHORT

```
SEQ    TIME     5-Aug-8Ø

186.  Ø:11:25 DEVICE STATUS BLOCK
```

5.2.26 Line Printer Error

The monitor records any errors detected by the LP1ØØ controller as a Line Printer Error in the system event file. Note that if the line printer is taken off-line to add paper or change forms, the monitor does not record this event.

The LAST DATA WORD SENT can help to determine the location of a data parity error, if one exists. Also, the CONI AT ERROR text translation contains significant error bits to describe the mode of operation when the failure occurred.

FULL

```
**********************************************
LINE PRINTER ERROR
 LOGGED ON 22-Mar-81 AT  Ø:11:5Ø      MONITOR UPTIME WAS  Ø:23:18
        DETECTED ON SYSTEM # 1536.
        RECORD SEQUENCE NUMBER: 1.
**********************************************

        UNIT NAME:      LPTØ
        CONTROLLER TYPE:        LP1ØØ
        LAST DATA WORD SENT:    Ø,123
        CONI AT ERROR:  2ØØØ45,226465 = NOT READY,VFU ERROR,OFF LINE,
        VFU TYPE:       DIRECT ACCESS
        CHARACTER SET:  VARIABLE
        PAGE COUNTER:   37.
```

SHORT

```
SEQ    TIME    22-Mar-81

1.  Ø:11:5Ø LPTØ LP1ØØ ERROR  CONI LP = 2ØØØ45,226465
```

5.2.27 Unit Record Error

The monitor logs a Unit Record Error into the system event file when it detects an error on a unit-record device such as a line printer, a card reader, a card punch, or a plotter.

FULL

```
**********************************************
UNIT RECORD ERROR
 LOGGED ON  8-Sep-80 AT 12:06:44      MONITOR UPTIME WAS  3:58:38
        DETECTED ON SYSTEM # 1026.
        RECORD SEQUENCE NUMBER: 314.
**********************************************
UNIT NAME:              LPT262
CONTROLLER TYPE:        LP100
DEVICE TYPE:            LPT
USER ID:                [1,2]
PROGRAM NAME:           LPTSPL
VFU TYPE:               DAVFU
CHARACTER SET:          96 CHARACTER
CONI AT ERROR:          307216,632444   NOT READY,VFU ERROR,OFF LINE,
LAST DATA WD:           0,0
```

SHORT

```
SEQ    TIME     8-Sep-80

314. 12:06:44 LPT262 ERROR FOR USER [1,2] RUNNING LPTSPL
                    CONI LP100 = 307216,632444
```

5.3 TOPS-20 ENTRIES

The following sections list both the FULL and SHORT versions of the entries that TOPS-20 can record in its system event file. Note that the network entries for DECnet-20 version 2.1 are listed separately in Section 5.4. Network entries for DECnet-20 versions 3.0, and 4.0 are listed in Section 5.5

5.3.1 TOPS-20 System Reloaded

Every time the monitor is loaded a TOPS-20 System Reloaded entry is written into the system event file, explaining why the system was reloaded. If the system is on auto-reload and a BUGHLT occurs, the BUGHLT address is listed and the TOPS-20 BUGHLT-BUGCHK entry, Section 5.3.2, is also written into the system event file.

FULL

```
**********************************************
TOPS-20 SYSTEM RELOADED
 LOGGED ON Mon 23 Jun 80 08:46:31       MONITOR UPTIME WAS  0:00:22
        DETECTED ON SYSTEM # 2116.
        RECORD SEQUENCE NUMBER: 22.
**********************************************
CONFIGURATION INFORMATION
        SYSTEM NAME:    System 2116 TOPS-20 Monitor 4(3230)
        MONITOR BUILT ON:       Wed 28 Nov 79 11:00:01
        CPU SERIAL #:           2116.
        MONITOR VERSION:        4(3230)
        U-CODE VERSION:         0
RELOAD BREAKDOWN:
```

SHORT

```
SEQ     TIME    Mon 23 Jun 80

22. 08:46:31 RELOAD OF System 2116 The Big Orange Welcomes You, TOPS-20
                    Monitor 4(3230) VERSION 4(3230)
                    BUILT ON Wed 28 Nov 79 11:00:01 REASON
```

5.3.2 TOPS-20 BUGCHKs and BUGHLTs

When the monitor detects a BUGHLT, BUGCHK, or BUGINF, monitor software error, it records a TOPS-20 BUGHLT-BUGCHK entry into the system event file. The most serious of the three errors is a BUGHLT, which crashes the system. At this point, something is seriously wrong, and the monitor does not have enough integrity to attempt any further error recovery. The monitor does, however, collect pertinent information for error recording. When the system is reloaded, the information is extracted from a crash dump and recorded in the system event file.

BUGCHK and BUGINF are less serious, perhaps correctable, monitor-detected errors that can affect only particular users instead of the entire system. These errors may or may not crash the system depending on the error that occurs.

The number of errors since reload is included in this entry because only five occurrences of this entry type are allowed in the monitor's error recording buffer at any one time. In the case of an error occurring in a tight loop, more than five entries could overflow the buffer, and the information for the first occurrence might be lost. These numbers should increment by one for each entry; however, if the sequence is broken, it indicates that more than five entries occurred before the error-logger module of the monitor could empty the buffer.

The FORK # and JOB # in the entry are the numbers associated with the current user at the time of the error. A value of -1 or 777777 indicates that the monitor was performing an overhead function (such as scheduling) and that there was no current user. Note that the FORK # and JOB # indicate the current user, and not necessarily the user being serviced by the monitor interrupt-level routines.

All BUGHLTs now reside in a monitor module, BUGS.MAC. This module includes a description of what might have caused the BUGHLT and also some corrective action that you can take. For complete listing and explanation of BUGINFs, BUGCHKs, and BUGHLTs, refer to the TOPS-20 BUGINF, BUGCHK, BUGHCT Document.

FULL

```
**********************************************
TOPS-20 BUGHLT-BUGCHK
 LOGGED ON Mon 16 Jun 80 11:10:19      MONITOR UPTIME WAS  3:10:48
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 25.
**********************************************

ERROR INFORMATION:
        DATE-TIME OF ERROR:     Mon 16 Jun 80 11:10:09
        # OF ERRORS SINCE RELOAD:       1.
        FORK # & JOB #:         72,0
        USER'S LOGGED IN DIR:   OPERATOR
        PROGRAM NAME:           SYSJOB
        ERROR:          BUGINF
        ADDRESS OF ERROR:       644111
        NAME:                   DN20ST
        DESCRIPTION:            DTESRV- DN20 STOPPED
        CONI APR:       7740,3 =  NO ERROR BITS DETECTED
        CONI PAG:       0,660132
        DATAI PAG:      700100,1246
        CONTENTS OF AC'S:
    0:  0,0
    1:  777775,1
    2:  0,1
    3:  0,0
    4:  0,0
    5:  0,0
    6:  0,0
    7:  0,0
   10:  0,0
   11:  0,0
   12:  0,0
   13:  0,0
   14:  0,0
   15:  0,0
   16:  60000,0
   17:  777505,335504
        PI STATUS:      0,177
        ADDITIONAL DATA ITEMS:  1
        0,1

        ERA:            602000,5504 = WD #3 MEMORY READ
        BASE PHY. MEM ADDR.
         AT FAILURE:    5504
```

SHORT

```
SEQ    TIME    Mon 16 Jun 80

25. 11:10:19 BUGINF DN20ST AT Mon 16 Jun 80 11:10:09 USER OPERATOR
                    RUNNING SYSJOB CONI APR= 7740,3 CONI PAG= 0,660132
                    ERA= 602000,5504
```

5.3.3 MASSBUS Device Error

Every time the monitor detects an error in the MASSBUS system a MASSBUS Device Error is recorded in the system event file. The MASSBUS system includes the MASSBUS devices RP04, RP05, RP06, TU45, and RM03; the RH20 controller (RH11 and UBA for 2020); and certain errors occurring in the channel logic.

The unit name in this entry refers to the physical MASSBUS unit active at the time of the error. This is a 5-character name in the format:

xxabc

where

- xx is the device type DP (disk pack) or MT (magtape) For example, DP220 refers to disk pack 220.
- a is the logical address of the RH20 controller for this device (0-7) - RH11 and UBA in a 2020 configuration.
- b is the logical MASSBUS address for this device (0-7) For magtape units, this is the TM02 address on the MASSBUS.
- c is the slave number of a magnetic tape unit. For RP04s, RP05s, and RP06s, this number is always 0.

The following is a MASSBUS Device Error from an RPØ7 disk drive:

```
                                FULL

**********************************************
MASSBUS DEVICE ERROR
 LOGGED ON Mon 31 Aug 81 15:28:29      MONITOR UPTIME WAS  0:36:03
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 131.
**********************************************
        UNIT NAME:       DP50C
        UNIT TYPE:       RP07
        UNIT SERIAL #:   0395.
        VOLUME ID:       PS
        LBN AT START OF XFER:          1636360  =
        CYL:   344.      SURF:   23.     SECT:   19.
        OPERATION AT ERROR:      DEV.AVAIL., GO +  WRITE DATA(60)
        FINAL ERROR STATUS:      20000,3
        RETRIES PERFORMED:       2.
        ERROR:  RECOVERABLE
DATA BUS PAR ERR,DRIVE EXCEPTION,LONG WD CNT ERR,CHN ERROR, IN CONTROLLER CONI
               PAR, IN DEVICE ERROR REGISTER

CONTROLLER INFORMATION:
CONTROLLER:      RH20 # 5
CONI AT ERROR:  0,722615 =
        DATA BUS PAR ERR,DRIVE EXCEPTION,LONG WD CNT ERR,CHN ERROR,
CONI AT END:     0,2415 =
         NO ERROR BITS DETECTED
        DATAI PTCR AT ERROR:     732203,177461
        DATAI PTCR AT END:       732203,177461
        DATAI PBAR AT ERROR:     720003,13423
        DATAI PBAR AT END:       720003,13423

CHANNEL INFORMATION:
CHAN STATUS WD 0:        200000,133237
        CW1:  0,0  CW2:  0,0
CHN STATUS WD 1:         540100,133240 =
        NOT SBUS ERR,NOT WC = 0,LONG WC ERR,
CHN STATUS WD 2:         603403,510620

DEVICE REGISTER INFORMATION:
        AT ERROR         AT END           DIFF.
CR(00): 4060             4060                0
         DEV.AVAIL., WRITE DATA(60)
SR(01): 50700            10700           40000
        ERR,MOL,DPR,DRY,VV,
ER(02): 10                  0            10
        PAR,
MR(03):    0                0               0
AS(04):    0                0               0
DA(05): 13426            13427           1
        D. TRK = 27, D.SECT. = 26
DT(06): 20042            20042               0
LA(07): 2700             3000            1700
```

```
SN(10): 1625            1625              0
OF(11):    0               0              0
DC(12): 530             530               0
        344.
CC(13): 530             530               0
        344.
E2(14):    0               0              0
        NO ERROR BITS DETECTED
E3(15): 210                 0            210
        DVC,DPE,
EP(16):    0               0              0
PL(17):    0               0              0

DEVICE STATISTICS AT TIME OF ERROR:
# OF READS:     79686.  # OF WRITES:     59808.  # OF SEEKS:     14597.
# SOFT READ ERRORS:     0.      # SOFT WRITE ERRORS:    2.
# HARD READ ERRORS:     0.      # HARD WRITE ERRORS:    0.
# SOFT POSITIONING ERRORS:      0.
# HARD POSITIONING ERRORS:      0.
# OF MPE:  0.   # OF NXM:  0.   # OF OVERRUNS:  0.

                                SHORT

SEQ    TIME    Mon 31 Aug 81

131. 15:28:29 DP50C PS: RP07 SERIAL #0395. CONI RH= 0,722615
                   CHN STS= 540100,133240 SR= 0,50700 ER= 0,10
                   CYL/SURF/SEC= 344./23./19.
```

The following MASSBUS Device Error is from a TU78 magnetic tape drive:

FULL

```
**********************************************
MASSBUS DEVICE ERROR
 LOGGED ON Mon 31 Aug 81 15:42:02      MONITOR UPTIME WAS  0:08:46
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 161.
**********************************************
        UNIT NAME:       MT000
        UNIT TYPE:       TU78
        UNIT SERIAL #:   0175.
        VOLUME ID:
        LOCATION:    RECORD  #  1.    OF FILE #  0.
        USER'S LOGGED IN DIR NUMBER:     5
        USER'S PGM:      SYSJOB
        OPERATION AT ERROR:      DEV.AVAIL.  GO +  READ FWD(70)
        FINAL ERROR STATUS:      0,0
        RETRIES PERFORMED:       0.
        ERROR:  NON-RECOVERABLE
DRIVE EXCEPTION,CHN ERROR, IN CONTROLLER CONI
```

```
M8960 u-CODE REVISION LEVELS:
        0  (    0- 3777)        005
        1  ( 4000- 7777)        005
        2  (10000-13777)        005
        3  (14000-17777)        003
        4  (20000-23777)        002
        5  (24000-27777)        003
        6  (30000-33777)        007
        7  (34000-37777)        003

CONTROLLER INFORMATION:
CONTROLLER:     RH20 # 0
CONI AT ERROR:  0,222415 =
        DRIVE EXCEPTION,CHN ERROR,
CONI AT END:    0,222415 =
        DRIVE EXCEPTION,CHN ERROR,
        DATAI PTCR AT ERROR:    732200,177771
        DATAI PTCR AT END:      732200,177771
        DATAI PBAR AT ERROR:    720000,113000
        DATAI PBAR AT END:      720000,113000

CHANNEL INFORMATION:
CHAN STATUS WD 0:       200000,272774
        CW1:  0,0  CW2:  0,0
CHN STATUS WD 1:        540100,272775 =
        NOT SBUS ERR,NOT WC = 0,LONG WC ERR,
CHN STATUS WD 2:        420003,170000

DEVICE REGISTER INFORMATION:
        AT ERROR        AT END          DIFF.
CMD 00: 4070            4070              0
        DEV.AVAIL.  READ FWD(70)
DST 01: 4415            4415              0
     Interrupt code: TM
     UNEXPECTED combination -- interrupt code: 15
                               failure   code: 2

CNT 02: 30004           30004             0
        SKIP COUNT = 0.  RECORD COUNT = 1.  DRIVE # 0
DG1 03:    0               0              0
ATN 04:    0               0              0
BCT 05: 113000          113000            0
        38400. BYTES
DTR 06: 142101          142101            0
STA 07: 166200          166200            0
        RDY, PRES, ONL, PE, BOT, AVAIL,
SER 10: 565             565               0
DG2 11:    0               0              0
DG3 12:    0               0              0
NST 13: 1               1                 0
     Interrupt code: DONE
     Extended sense data not updated


NC1 14: 406             406               0
        CMD COUNT = 1. Rewind(06)
NC2 15: 10              10                0
        CMD COUNT = 0. Sense(10)
NC3 16: 10              10                0
        CMD COUNT = 0. Sense(10)
NC4 17: 10              10                0
        CMD COUNT = 0. Sense(10)
MPA 20: 2034            2034              0
MPD 21: 100000          100000            0
```

```
EXTENDED SENSE BYTE DATA NOT SUPPLIED FOR THIS ENTRY

DEVICE STATISTICS AT TIME OF ERROR:
# OF READS:      Ø.       # OF WRITES:     Ø.       # OF SEEKS:      Ø.
# SOFT READ ERRORS:       Ø.       # SOFT WRITE ERRORS:     Ø.
# HARD READ ERRORS:       1.       # HARD WRITE ERRORS:     Ø.
# SOFT POSITIONING ERRORS:         Ø.
# HARD POSITIONING ERRORS:         Ø.
# OF MPE:  Ø.    # OF NXM:  Ø.     # OF OVERRUNS:  Ø.

                                   SHORT

161. 15:42:02 MTØØØ TU78 SERIAL #Ø175. OPERATOR RUNNING SYSJOB
                    CONI RH= Ø,222415 CHN STS= 54Ø1ØØ,272775 SR= Ø,4415
                    ER= Ø,3ØØØ4 FILE/RECORD Ø./1.
```

5.3.4 DX2Ø Device Error

When the monitor detects an error in any portion of the MASSBUS system connected to the DX2Ø tape controller, the DX2Ø Device Error is recorded in the system event file.

This entry contains the octal values of the CONI and DATAI from the controller both when the error was first detected and after the last retry.

```
                                   FULL

**********************************************
DX20 DEVICE ERROR
 LOGGED ON Mon 9 Feb 81 10:33:16      MONITOR UPTIME WAS 4 DAYS 14:31:48
        DETECTED ON SYSTEM # 2116.
        RECORD SEQUENCE NUMBER: 4.
**********************************************
        UNIT NAME:       MT301
        UNIT TYPE:       TU70
        VOLUME ID:       6631
        LOCATION:     RECORD  #  1282.    OF FILE #      0.
        OPERATION AT ERROR:       GO +  WRITE DATA(60)
        FINAL ERROR STATUS:      0,3
        RETRIES PERFORMED:       0.
        ERROR:  RECOVERABLE
DRIVE EXCEPTION, IN CONTROLLER CONI
                 MPERR, IN DEVICE ERROR REGISTER

CONTROLLER INFORMATION:
CONTROLLER:      RH20 # 3  DX20 #:0  TX02  #:    0
DX20 U-CODE VERSION:     1(13)
CONI AT ERROR:  0,202615 =
        DRIVE EXCEPTION,
CONI AT END:    0,202615 =
        DRIVE EXCEPTION,
DATAI PTCR AT ERROR:     732200,177761
DATAI PTCR AT END:       732200,177761
DATAI PBAR AT ERROR:     720000,172742
DATAI PBAR AT END:       720000,172742
```

```
CHANNEL INFORMATION:
CHAN STATUS WD 0:        200000,260532
        CW1:  0,0  CW2:  0,0
CHN STATUS WD 1:         500000,260534 =
        NOT SBUS ERR,
CHN STATUS WD 2:         600001,200006

MASSBUS REGISTER INFORMATION:
        AT ERROR         AT END           DIFF.
CR(00): 60               60                  0
         WRITE DATA(60)
SR(01): 70000            70000               0
        CERR,LNKPRS,MPRUN,
ER(02): 600              600                 0

MPERR,MPERR CLASS: 1 ,SUB-CLASS: 0
         =  UNUSUAL DEVICE STATUS FROM FINAL STATUS SEQUENCE
MR(03): 4                4                   0
        MPSTR,
AS(04):    0                0                 0
SB(05): 172742           172742              0
DT(06): 50060            50060               0
         DRIVE TYPE: 60, HDWR VER: 50
SI(20): 7000             7000                0
DN(21): 10001            10001               0
ES(22): 120              120                 0
TE(23): 100              100                 0
AY(24):    0                0                 0
E0(26): 4304             4304                0
E1(27): 4214             4214                0
IR(30): 114751           114751              0
PC(31): 133662           133662              0
AL(32): 15466            15466               0
SD(33): 104030           104030              0
FP(34): 117360           117360              0
BW(35): 122377           122377              0
IB(36): 160000           160000              0
MA(37):    0                0                 0

DEVICE INFORMATION RECORDED AT TIME OF ERROR
REGISTER  CONTENTS       TEXT
SB 0-3: 10 304 10 214

DATA CHK,NOISE,SEL WR STATUS,R/W VRC,ENV CHK/SKEW REG VRC,1600 BPI, TIE = 00001000
   4-7: 0 100 5 0

 NO ERROR BITS DETECTED
  8-11: 0 10 0 0
        NO ERROR BITS DETECTED
 12-15: 0 16 374 0
 16-19: 0 2 0 74
 20-23: 0 0 201 200
  MCV:  10 0 320 33
  MRA:  343 30 0 60
  MRB:  120 0 0 4
  MRC:  200 14 1 20
  MRD:  120 0 100 0
  MRE:  0 0 342 365
  MRF:  102 0 4 0
  CB0:  0 3 152 200
  CB1:  205 17 0 16
  DP0:  14 2 6 0
  DP1:  0 0 0 0
  DP2:  30 14 0 0
  DP3:  0 14 111 70
  LAS:  1 1 4 1
```

```
DEVICE STATISTICS AT TIME OF ERROR:
# OF READS:     674226290.      # OF WRITES:    881585460.      # OF SEEKS:     0.
# SOFT READ ERRORS:     0.      # SOFT WRITE ERRORS:    39.
# HARD READ ERRORS:     0.      # HARD WRITE ERRORS:    0.
# SOFT POSITIONING ERRORS:      0.
# HARD POSITIONING ERRORS:      0.
# OF MPE:  0.   # OF NXM:  0.   # OF OVERRUNS:  0.
```

SHORT

```
SEQ     TIME     MON 9 Feb 81

4. 10:33:16 MT301 6631: TU70 OPERATOR RUNNING TAPE CONI=0,202615
                    CHN STS 1= 500000,260534 CR=0,60 SR=0,70000 ER=0,600
                    SENSE BYTES 0-3: 10 304 10 214 FILE/RECORD 0./1282.
```

5.3.5 Drive Statistics Entries

Drive Statistics Entries are written into the system event file to record the activity on the drive. For example, mounts and dismounts, reloads, and drive shutdowns are information that is recorded as a drive statistic.

FULL

```
**********************************************
DRIVE STATISTICS ENTRIES
 LOGGED ON  5-Oct 10:52:28      MONITOR UPTIME WAS 367.
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 361.
**********************************************

        Volume ID: SPARE  Reason recorded: Disk pack mount

        Channel info(CDB): RH20 # 4 on PI level 5
        Device info(UDB):  RP20,  DP401 PIA: 0

                    READS    WRITES     SEEKS

        TOTAL :        8.                  1.

**********************************************
DRIVE STATISTICS ENTRIES
 LOGGED ON  5-Oct 11:20:24      MONITOR UPTIME WAS 5454.
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 374.
**********************************************

        Volume ID: CDM    Reason recorded: Magtape unload

        Channel info(CDB): RH20 # 3 on PI level 5
        Device info(UDB):  TU70, MTA1, MT301 PIA: 0

                    READS    WRITES

        TOTAL :   353600.  7610560.

        NRZI  :
        PE    :   353600.  7610560.
        GCR   :
```

SHORT

```
361. 10:52:28 STATS DRIVE: DP401 VOLID: SPARE  REASON: Disk pack mount.
374. 11:20:24 STATS DRIVE: MT301 VOLID: CDM    REASON: Magtape unload.
```

5.3.6 Configuration Status Change

The monitor records a Configuration Status Change when the system operator takes disk units and/or sections of core memory on-line or off-line, thus changing the configuration of the system. The system operator can give a 2-character reason for the change in configuration. The following codes are suggested:

PM - preventive maintenance

CM - corrective maintenance

DN - unit is down

OT - other

This entry lists what device was affected, what action was taken, and where the action was performed (channel number, controller number, unit number).

CAUTION

When the system operator adds memory to the system, the monitor checks to verify the availability of the specified addresses. Mistakes are reported to the operator at the operator's terminal, CTY; however, the error-logging system treats these as valid NXMs and records them as NXM entries. You can identify a NXM entry of this type by the fact that no physical memory is off-line and the user's directory is [1,2].

FULL

```
**********************************************
CONFIGURATION STATUS CHANGE
 LOGGED ON Mon 23 Jun 80 08:50:21       MONITOR UPTIME WAS 2 DAYS  8:34:54
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
 DETACH TU72 S/N:28410
 AS MTA2 AT CHANNEL #0 CONTROLLER #0 UNIT #2
 REASON:
```

SHORT

```
SEQ     TIME     Mon 23 Jun 80

1. 08:50:21 DETACH TU72 S/N:28410 AS MTA2 AT CHANNEL #0 CONTROLLER #0
                    UNIT #2 REASON:
```

5.3.7 System Log Entry

The monitor records a System Log Entry when the system operator enters a log entry into the system event file with the OPR program.

A system operator, or anyone with operator privileges, can make an entry into the system event file by doing the following:

1. Run the OPR program

   ```
   @OPR (RET)
   OPR>
   ```

2. When you see the prompt, specify the REPORT command:

   ```
   OPR>REPORT (RET)
   ```

3. Use the following syntax:

   ```
   OPR>REPORT user text (RET)
   ```

   where user can be directory name and/or device name, and text can be a single-line or multiple-line response.

For more information on OPR, refer to the TOPS-20 Operator's Command Language Reference Manual.

FULL

```
**********************************************
SYSTEM LOG ENTRY
 LOGGED ON Tue 1 Jul 80 11:37:37      MONITOR UPTIME WAS  0:09:48
        DETECTED ON SYSTEM # 2116.
        RECORD SEQUENCE NUMBER: 32.
**********************************************
ENTRY CREATED BY:
        JOB #, TTY #:    11,17
        DIRECTORY:       SCHMITT
        WHO:             SCHMIT
        DEV:             NUL
        MESSAGE:         : testing
```

SHORT

```
SEQ    TIME    Tue 1 Jul 80

32. 11:37:37 SYSTEM LOG ENTRY BY SCHMIT FOR DEVICE NUL ON TTY # 17
                    MESSAGE: : testing
```

5.3.8 Front-End Device Report

You find a Front-End Device Report in the system event file when the front end passes a packet of error information to the monitor across the DTE-20. This information contains errors detected by the front end and KLCPU hardware and software. Currently, entries are created for the following devices: LP20, CD20, DH11, KLCPU, KLERROR, and KLINIK.

If the FORK # and JOB # associated with the error are 777777,777777, this indicates that the TOPS-20 monitor knows of this device but it is not currently assigned to any fork or job. If the FORK # and JOB # are 777776,777776, this indicates that the monitor does not know anything about this device.

The front end generates a standard-status word for each transfer across the DTE-20. The ERROR LOG REQUEST bit in this word causes the packet to be recorded into the system event file.

The information in the entry varies depending on the type of device being reported on. If SPEAR does not know how to list a device, this fact is stated in the entry, listed in octal.

```
                                FULL

*********************************************
FRONT END DEVICE REPORT
 LOGGED ON Mon 16 Jun 80 11:48:30      MONITOR UPTIME WAS  3:48:59
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 35.
*********************************************
        DTE20 #:                0.
        FE SOFTWARE VER:        0.
        DEVICE: DH11

        STD. STATUS:    300 = NON RECOVERABLE ERROR,ERROR LOG REQUEST,
        DH11 UNIBUS ADDRESS:    160060 = DH11 #2
        SYSTEM CONTROL REG:     30106 = TRANS & NXM INT ENA,STORAGE INT ENA,LINE #6
        RECEIVED CHAR REG:      123000 = VALID DATA PRESENT,FRAMING ERROR,LINE #6,CHAR=0

                                SHORT

SEQ    TIME    Mon 16 Jun 80

35. 11:48:30 DH11 STD STAT=300 UNIBUS ADDR=160060 SYS CONTROL=30106
                   REC CHAR=123000
```

5.3.9 Front End Reloaded

Each time the KLCPU detects that the front end has halted or is in a loop a Front End Reloaded entry is recorded in the system event file. The KL attempts to copy a crash dump file onto disk from the front end's memory and then reboots the front end.

The front-end number is the logical address of the front end and indicates whether this front end is privileged. The status at reload describes, in text, any errors that occurred during the reboot process. The file name of the core dump is listed if the crash dump was successful.

FULL

```
**********************************************
FRONT END RELOADED
 LOGGED ON Tue 1 Jul 80 00:18:51       MONITOR UPTIME WAS  0:02:24
        DETECTED ON SYSTEM # 2102.
        RECORD SEQUENCE NUMBER: 126.
**********************************************
        FRONT END #:     0
        STATUS AT RELOAD:         NO ERROR BITS DETECTED
        RETRIES:         0
        REASON FOR RELOAD:       B03
        FILENAME FOR DUMP:       <SYSTEM>0DUMP11.BIN.17, 1-Jul-80 00:18:45
```

SHORT

```
SEQ    TIME    Tue 1 Jul 80

126. 00:18:51 FRONT END RELOAD ON PDP11 #0 RELOAD STATUS,,RETRIES 0,0
                    PDP11 HALT CODE B03
```

5.3.10 Processor Parity Trap

The monitor records a Processor Parity Trap each time a page-fail trap occurs in the CPU as a result of an AR, ARX, or PAGE TABLE parity error.

The information contained in the GOOD DATA WORD is valid only if the error is recoverable; otherwise, the data is 0,0 and the DIFFERENCE DATA is a copy of the BAD DATA WORD. The DIFFERENCE is the result of an XOR between the bad data and the good data words. Note that if the user is unknown, 777777,777777 will be the FORK and JOB numbers.

FULL

```
**********************************************
PROCESSOR PARITY TRAP
 LOGGED ON Tue 8 Jul 80 11:14:04       MONITOR UPTIME WAS  8:51:58
        DETECTED ON SYSTEM # 2102.
        RECORD SEQUENCE NUMBER: 320.
**********************************************
STATUS AT ERROR:
        BAD DATA DETECTED BY:    AR
        PAGE FAIL WD AT TRAP:    763000,313
        BAD DATA WORD:  252525,252525
        GOOD DATA WORD: 525252,525252
        DIFFERENCE:     777777,777777
        PHYSICAL MEM ADDR.
         AT FAILURE:     563003,277313
        RECOVERY:        CONT. USER
        RETRY COUNT:     1.
        CACHE IN USE

        FORK #  & JOB #:                  53,17
        USER'S LOGGED IN DIR:    EIBEN
        PROGRAM NAME:            KLPAR1
```

SHORT

```
SEQ    TIME    Tue 8 Jul 80

320. 11:14:04 PARITY TRAP  PAGE FAIL WORD;763000,313
                    PHYSICAL MEMORY ADDRESS;563003,277313
                    FAILURE TYPE,,RETRIES;40000,1
```

5.3.11 Processor Parity Interrupt

When the monitor detects an APR interrupt because of a parity error, it records a Processor Parity Interrupt in the system event file. It records the entry after it has scanned all physical memory looking for more errors. If the original error also generates a page-fail trap, the monitor also creates a Processor Parity Trap entry.

The CONI APR and ERA values are the contents of these registers at the time of the first error. The PC AT INTERRUPT value includes the flags in the left half. The BASE PHYsical MEMory ADDRess AT FAILURE is from the right half of the contents of the ERA.

The # OF ERRORS on this sweep refers to the number of parity errors during this sweep of physical memory. If the value is zero, the monitor did not detect any errors, and 777777,777777 is the logical AND function for both bad addresses and bad data. The logical OR function, in this case, is 0,0.

The SYSTEM MEMORY CONFIGURATION lists the physical memory configuration and any detected errors at the time of the first error. These are the results of S-BUS DIAGNOSTIC FUNCTIONS for all memory controllers on this CPU.

FULL

```
**********************************************
PROCESSOR PARITY INTERRUPT
 LOGGED ON Tue 8 Jul 80 11:21:35      MONITOR UPTIME WAS  8:59:29
        DETECTED ON SYSTEM # 2102.
        RECORD SEQUENCE NUMBER: 323.
**********************************************
        CONI APR:       7740,413 = MB PAR ERR,
        ERA:            36001,520314 = WD #0 CACHE WRITE
        BASE PHY. MEM ADDR.
         AT FAILURE:    1520314

        PC FLAGS AT INTERRUPT:  300000,0
        PC AT INTERRUPT:        67320
        # ERRORS ON THIS SWEEP  2.
        LOGICAL AND OF
        BAD ADDRESSES:  1,520304
        LOGICAL OR OF
        BAD ADDRESSES:  1,520314
        LOGICAL AND OF
        BAD DATA:       252525,252525
        LOGICAL OR OF
        BAD DATA:       252525,252525
SYSTEM MEMORY CONFIGURATION:

CONTROLLER:  #0  MB20  128 K
F0:     6000,0  F1:     36300,36012
        INTERLEAVE MODE:        4-WAY
        REQ ENABLED:    0 2
        LOWER ADDRESS BOUNDARY: 0
        UPPER ADDRESS BOUNDARY: 777777
        ERRORS DETECTED:        NONE
```

```
CONTROLLER:  #1  MB20  128 K
F0:      6000,0  F1:      36300,36005
         INTERLEAVE MODE:          4-WAY
         REQ ENABLED:     1 3
         LOWER ADDRESS BOUNDARY: 0
         UPPER ADDRESS BOUNDARY: 777777
         ERRORS DETECTED:          NONE
CONTROLLER:  #2  MB20  128 K
F0:      6000,0  F1:      36301,36012
         INTERLEAVE MODE:          4-WAY
         REQ ENABLED:     0 2
         LOWER ADDRESS BOUNDARY: 1000000
         UPPER ADDRESS BOUNDARY: 1777777
         ERRORS DETECTED:          NONE
CONTROLLER:  #3  MB20  128 K
F0:      6000,0  F1:      36301,36005
         INTERLEAVE MODE:          4-WAY
         REQ ENABLED:     1 3
         LOWER ADDRESS BOUNDARY: 1000000
         UPPER ADDRESS BOUNDARY: 1777777
         ERRORS DETECTED:          NONE
CONTROLLER:  #10  MF20
F0:      26123,277313     F1:      500,1000
LAST WORD REQUEST:        RQ3 WRITE
LAST ADDRESS HELD:        3277313
CONTROLLER STATUS:         SF2 & SF1= 2
         ERRORS DETECTED:          WRITE PARITY
CONTROLLER:  #11  MF20
F0:      7747,631734      F1:      500,1000
LAST WORD REQUEST:        RQ0RQ1RQ2RQ3- READ
LAST ADDRESS HELD:        7631734
CONTROLLER STATUS:         SF2 & SF1= 2
         ERRORS DETECTED:          NONE
ERRORS DETECTED DURING SWEEP:
ADDRESS BAD DATA          GOOD DATA         DIFFERENCE
1520304 252525,252525               GOOD DATA NOT FOUND
1520314 252525,252525               GOOD DATA NOT FOUND

                                    SHORT

SEQ     TIME     Tue 8 Jul 80

323. 11:21:35 PARITY INTERRUPT-CONI APR;7740,413 ERA;36001,520314
                      PC AT INTERRUPT;0,67320 # OF ERRORS;2.
```

5.3.12 KL CPU Status Block

This entry is written into ERROR.SYS on TOPS-20, if KLSTAT is turned on at the time of a system crash. (See Section 4.5.1 for this procedure.)

At the time of a crash, a snapshot of the condition of all the components of the CPU (such as controllers, channels, RH20s, the pager, and so forth) is taken. When the system recovers, this information is extracted from the CRASH.EXE file and written as an entry in ERROR.SYS. This entry displays the condition of the registers and channels at the time of the crash.

FULL

```
**********************************************
KL CPU STATUS BLOCK
 LOGGED ON Mon 15 Sep 80 15:03:19       MONITOR UPTIME WAS 17:49:02
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 26.
**********************************************
APRID = 600236,364131
CONI APR = 7740,3
RDERA = 202000,132276
CONI PI = 0,2377
DATAI PAG = 701000,3201
CONI PAG = 0,660124
CONI RH0 THRU RH7
        000000,,002445  000000,,002445  000000,,002445  000000,,002445
        000000,,002000  000000,,002000  000000,,002000  000000,,002000
CONI DTE0 THRU DTE3
        000000,,001016  000000,,101016  000000,,002000  000000,,002000
EPT LOCATIONS 0 THRU 37 (CHANNEL LOGOUT AREA)
        200000,,225566  540100,,225567  620003,,477000  254340,,726001
        200000,,074442  500000,,074443  600000,,460000  254340,,726421
        200000,,075064  500000,,075065  600001,,053000  254340,,727011
        200000,,075522  500000,,075523  600001,,573000  254340,,727501
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
EPT LOCATIONS 140 THRU 177 (DTE CONTROL BLOCKS)
        241000,,223711  241000,,730250  254340,,002135  000000,,000000
        000000,,000000  000000,,223434  000000,,000030  000000,,223516
        000000,,000000  041000,,731556  254340,,002147  000000,,000000
        000000,,000226  000000,,223433  000000,,000030  000000,,223546
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
        000000,,000000  000000,,000000  000000,,000000  000000,,000000
UPT LOCATIONS 424 THRU 427 (UUO AREA)
        310100,,057200  000000,,700000  000000,,000000  601000,,003201
UPT LOCATIONS 500 THRU 503 (PAGE FAIL AREA)
        411000,,742000  000000,,000162  000006,,611327  000000,,027543
AC BLOCK 6 LOCATIONS 0 THRU 3 AND 12
        000770,,000007  301000,,002520  000000,,127000  000000,,153764
        011003,,276223
AC BLOCK 7 LOCATIONS 0 THRU 2
        000000,,000000  000000,,000000  000000,,000000

SBDIAG FUNCTIONS
        CTRLR   FUNCTION 0      FUNCTION 1
        0       006000,,000000  036300,,036012
        1       006000,,000000  036300,,036005
        10      007743,,201500  000500,,001000
```

SHORT

```
SEQ    TIME    Mon 15 Sep 80

26. 15:03:19 KL CPU STATUS BLOCK  APRID = 600236,364131
                  CONI APR = 7740,3 RDERA = 202000,132276
                  CONI PAG = 0,660124 DATAI PAG = 701000,3201
```

5.3.13 MF20 Device Report

This entry is written to ERROR.SYS when a MOS memory error occurs. A program called TGHA is called by the monitor every time a MOS memory error occurs. TGHA is responsible for recovering from the error. If TGHA places memory off-line or substitutes a spare bit, these events are recorded as an entry in ERROR.SYS. The TGHA entry is actually an ASCII text report describing the attempt to recover from an error in MOS memory.

FULL

```
**********************************************
MF20 DEVICE REPORT
 LOGGED ON Mon 30 Jun 80 10:02:41       MONITOR UPTIME WAS 1 DAY 11:39:06
        DETECTED ON SYSTEM # 2102.
        RECORD SEQUENCE NUMBER: 21.
**********************************************
        TEXT FROM TGHA:

A NEW MF20 KNOWN ERROR HAS BEEN DECLARED. DATA:
STORAGE MODULE SERIAL NUMBER: 8320021
BLOCK: 3, SUBBLOCK: 1, BIT IN FIELD (10): 5,
ROW: 174, COLUMN: 52, E NUMBER: 109, ERROR TYPE: CELL
```

SHORT

```
SEQ     TIME     Mon 30 Jun 80

21. 10:02:41 MF20 REPORT
```

5.3.14 KLERR Front End Device Report

The following entry is written into the system event file when the KL clock stops for any of several errors (FAST MEMORY, PARITY ERRORS, CRAM PARITY ERROR, DRAM PARITY ERROR, or FIELD SERVICE STOP). Any significant error signal will be listed just after the header.

FULL

```
**********************************************
FRONT END DEVICE REPORT "KLERR" TYPE 205
 LOGGED ON 23-Mar-81 09:14:54          MONITOR UPTIME WAS 0 DAYS  19:00:43
        DETECTED ON SYSTEM # 2102
        RECORD SEQUENCE NUMBER: 7
**********************************************

    No error bits are active

******* LOGGING STARTED  23-MARCH-81 09:12 ,RSX-20F YB14-41A

 OUTPUT DEVICES: TTY,LOG
KLE>EXAMINE KL
PC/ 5337
VMA/ 5337
PI ACTIVE: OFF, PI ON: 177, PI HOLD: 000, PI GEN: 000
 OVF  CY0  CY1  FOV  BIS  USR  UIO  LIP  AFI  AT1  AT0  FUF  NDV
       X    X
KLE>CLEAR OUTPUT TTY
 OUTPUT DEVICES: LOG
KLE>SET CONSOLE MAINTENANCE
 CONSOLE MODE: MAINTENANCE
KLE>SHOW HARDWARE
 KL10 S/N: 2102., MODEL B, 60. HERTZ
        MOS MASTER OSCILLATOR
        EXTENDED ADDRESSING
        INTERNAL CHANNELS
        CACHE

KLE>EXAMINE DTE
 DLYCNT: 000000
 DEXWD3: 160000
 DEXWD2: 060323
 DEXWD1: 000000
   KL10 DATA=014064 760000
 TENAD1: 000000  TENAD2: 000024
   ADDRESS SPACE=EPT
   OPERATION=EXAMINE
   PROTECTION-RELOCATION IS ON
   KL10 ADDRESS=24
 TO10BC: 010000  TO11BC: 130000
 TO10AD: 067540  TO11AD: 070572
 TO10DT: 000036  TO11DT: 142400
 DIAG1 : 001100
   KL IN HALT LOOP
   MAJOR STATE IS TO-11 TRANSFER
 DIAG2 : 040000
 STATUS: 012504
   RAM IS ZEROS
   DEX WORD 1
   11 REQUESTED 10 INTERRUPT
   E BUFFER SELECT
   DEPOSIT-EXAMINE DONE
 DIAG3 : 026000
```

```
KLE>FREAD 100:177
FR 100/ 000177 602664
FR 101/ 000000 002600
FR 102/ 000013 410202
FR 103/ 000020 212024
FR 104/ 000000 032434
FR 105/ 000000 003421
FR 106/ 000000 642000
FR 107/ 000000 715642
FR 110/ 000003 015225
FR 111/ 000104 000000
FR 112/ 007740 037411
FR 113/ 000000 044524
FR 114/ 000101 000012
FR 115/ 001107 060144
FR 116/ 001400 012003
FR 117/ 001100 002000
FR 120/ 000000 000000
FR 121/ 000000 000000
FR 122/ 001100 002000
FR 123/ 000000 270173
FR 124/ 002000 020000
FR 125/ 000000 000000
FR 126/ 000000 000001
FR 127/ 000000 000001
FR 130/ 000072 000000
FR 131/ 070054 060000
FR 132/ 014064 760000
FR 133/ 000020 414000
FR 134/ 130066 404003
FR 135/ 120024 224003
FR 136/ 104052 604003
FR 137/ 002004 244003
FR 140/ 760505 050707
FR 141/ 100201 000001
FR 142/ 110000 001010
FR 143/ 600202 061407
FR 144/ 540001 050707
FR 145/ 510000 000001
FR 146/ 650000 001010
FR 147/ 111212 071407
FR 150/ 000000 000104
FR 151/ 000000 002004
FR 152/ 000000 000104
FR 153/ 000024 002104
FR 154/ 000000 000125
FR 155/ 000000 002405
FR 156/ 000000 000125
FR 157/ 000024 002525
FR 160/ 001003 017027
FR 161/ 001006 276703
FR 162/ 001006 206017
FR 163/ 001000 000523
FR 164/ 001003 017323
FR 165/ 001006 276767
FR 166/ 001006 206017
FR 167/ 001000 000103
FR 170/ 360040 126722
FR 171/ 000000 735722
FR 172/ 011600 137230
FR 173/ 200102 377322
FR 174/ 176010 177664
FR 175/ 163000 127375
FR 176/ 000200 337375
FR 177/ 760000 533305
KLE>WHAT AC
 AC-BLOCK: 0
KLE>SWEEP
KLE>XCT CONI 0,15!      CONI APR,15
KLE>EXAMINE TEN 15
```

```
15/ 007740 000003
KLE>XCT BLKI 4,15!       RDERA
KLE>EXAMINE TEN 15
15/ 602000 005337
KLE>XCT CONI 4,15!       CONI PI,15
KLE>EXAMINE TEN 15
15/ 000000 000177
KLE>XCT DATAI 10,15!     DATAI PAG,15
KLE>EXAMINE TEN 15
15/ 700100 001270
KLE>XCT CONI 10,15!      CONI PAG,15
KLE>EXAMINE TEN 15
15/ 000000 060137
KLE>SET OUTPUT TTY
 OUTPUT DEVICES: TTY,LOG
KLE>CLEAR OUTPUT LOG

******* LOGGING FINISHED  23-MARCH-81 09:13
```

SHORT

```
SEQ.     TIME     23-Mar-81

7. 09:14:54 KLERR FRONT END DEVICE TYPE 205
   No error bits are active
   ========
```

5.3.14.1 The HSC50 Error Log - When a CPU initiates a request for data transfer from the HSC50, the HSC50 Error Log entry is written into that particular CPU's system event file. The following are examples of the full and short versions of the HSC50 Error Log event.

FULL

```
**********************************************
HSC50 ERROR LOG
 LOGGED ON 14-Jul-85 16:50:06-EDT MONITOR UPTIME WAS 0 DAY(S) 14:20:42
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 7503.
**********************************************

   COMMON DATA

COMMAND REF #:          00000000    HOST COMMAND #: 0
CI20 PORT #:                   7.
NODE #:                       15.
SEQUENCE #:                    1.
FORMAT:                       03    SDI Error
FLAGS:                        41    Operation Continuing, Sequence Number Reset
EVENT:                      002B    Drive Error, SDI command timed out
CNTLR DEVICE #:     00000230F00F
CNTLR CLASS:                  01    Mass Storage Controller
CNTLR MODEL:                  01    HSC50
CNTLR SOFTWARE VER:           02
CNTLR HARDWARE VER:           00

   UNIT IDENTIFICATION DATA

UNIT NUMBER:                  11.
MULTI-UNIT CODE:            0020
UNIT DEVICE #:      000000000FA5
UNIT CLASS:                   02    DEC Std 166 Disk
UNIT MODEL:                   05    RA81
UNIT SOFTWARE VER:            06
UNIT HARDWARE VER:            01
VOLUME S/N:             00000FA5

   SDI DATA

HEADER:                 00000000    Logical Block
                                    BLOCK AT ERROR WAS 0.

   CONTROLLER DATA

REQUEST BYTE:                 13    Drive-online or available,
MODE BYTE:                    00    Port switch in, Run/Stop switch in,
ERROR BYTE:                   00    Formatting disabled,
CONTROLLER BYTE:              00    Diagl Cyl access disabled, 512 byte
RETRY COUNT / FAILURE CODE:   00

   RA80/81 DEVICE DATA

LAST POSITION COMMAND:        87
SDI ERROR STATUS:             00
LAST SEEK CYLINDER:         0000
HEAD NUMBER:                   0.
MICROPROCESSOR LEDS:          00
FRONT PANEL FAULT CODE:       00

   EXTRANEOUS DATA IN 8 BIT OCTAL BYTES
   (UNUSED RIGHT 4 BITS IN 36-BIT WORD)

BYTES 63.-60.   002 000 000 000   (00)
BYTES 67.-64.   000 000 000 000   (00)
BYTES 71.-68.   000 000 000 000   (00)
BYTES 75.-72.   000 000 000 000   (00)
BYTES 79.-76.   000 000 000 000   (00)
```

SHORT

```
SEQ   TIME   14-Jul-85

7503. 16:50:06 HSC50 Error Message  Node 15.  Drive Error, SDI command timed out
                 on RA81 #11.  S/N FA5  SDI Error -  -  -  -
```

5.4 DECNET ENTRIES (V2.1)

The following sections list both the FULL and SHORT versions of network entries (Version 2.1) TOPS-10 or TOPS-20 can record in the system event file.

5.4.1 Network Control Started

Whenever NETCON is loaded and started, the monitor records a Network Control Started entry into the system event file. This entry includes the version number and the node on which NETCON is running.

FULL

```
**********************************************
NETWORK CONTROL STARTED
 LOGGED ON Mon 23 Jun 80 11:37:08       MONITOR UPTIME WAS 2 DAYS 11:21:41
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 15.
**********************************************
        PROGRAM NAME:            NETCON
        PROGRAM VERSION:  4(22)
        NODE NAME:               KL2137
```

SHORT

```
SEQ    TIME    Mon 23 Jun 80

15. 11:37:08 NCU STARTED PROGRAM: NETCON VER:4(22)
                    STARTED ON NODE KL2137
```

5.4.2 Network Up-Line Dump

Whenever NETCON dumps a node, the monitor records the name of the node involved, the line used, the dump-file specification, and any return code as a Network Up-Line Dump entry in the system event file.

FULL

```
**********************************************
NETWORK UP-LINE DUMP
 LOGGED ON Mon 23 Jun 80 11:07:53       MONITOR UPTIME WAS 2 DAYS 10:52:26
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 11.
**********************************************
        TARGET NODE NAME:        DN20L
        SERVER NODE NAME:        KL2137
        SERVER LINE DESIG.:      DTE20_1_0
        FILE NAME DUMPED:        PS:<SROBINSON>DN20L-R4-26.DMP
```

SHORT

```
SEQ    TIME    Mon 23 Jun 80

11. 11:07:53 UP-LINE DUMP OF NODE DN20L BY NODE KL2137
                    LINE DESIGNATION DTE20_1_0
                    FILE DUMPED TO PS:<SROBINSON>DN20L-R4-26.DMP
```

5.4.3 Network Down-Line Load

Whenever NETCON loads a node, the monitor records the name of the node involved, the line used, the load-file specification, and any return code as a Network Down-Line Load entry in the system event file.

FULL

```
**********************************************
NETWORK DOWN-LINE LOAD
 LOGGED ON Mon 23 Jun 8Ø 11:1Ø:33      MONITOR UPTIME WAS 2 DAYS 1Ø:55:Ø6
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 13.
**********************************************
        TARGET NODE NAME:        DN2ØL
        SERVER NODE NAME:        KL2137
        SERVER LINE DESIG.:      DTE2Ø_1_Ø
        FILE NAME LOADED:        PS:<NEXT-RELEASE>DN2ØL-R4-26.SYS.1
```

SHORT

```
SEQ    TIME     Mon 23 Jun 8Ø

13. 11:1Ø:33 DOWN-LINE LOAD OF NODE DN2ØL BY NODE KL2137
                    LINE DESIGNATION DTE2Ø_1_Ø
                    FILE LOADED PS:<NEXT-RELEASE>DN2ØL-R4-26.SYS.1
```

5.4.4 Network Hardware Error

Whenever NETCON detects an error in any hardware device connected to a node, the monitor records this information as a Network Hardware Error in the system event file.

FULL

```
**********************************************
NETWORK HARDWARE ERROR
 LOGGED ON Mon 23 Jun 80 08:52:48      MONITOR UPTIME WAS 2 DAYS  8:37:21
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 3.
**********************************************
        MSG SENT FROM:  DN20L
        MSG REC'D AT:  KL2137
        HDWR TYPE:  KMC-DUP11  SOFTWARE TYPE:  ILLEGAL
         PARENT SYSTEM TYPE:  UNKN
         HARDWARE ERROR MSG SEQUENCE # FROM XMIT NODE: 14.
        LINE ID:  KDP_0_1_0

        REASON FOR ENTRY:       DDCMP START REC'D DURING NORMAL OPERATION
        RECOVERY STATE: NOT SUPPLIED,
        ERROR:   NO ERROR BITS DETECTED IN RxDBUF,  NO ERROR BITS DETECTED IN TxCSR
HARDWARE REGISTER INFORMATION:
        MICROCODE:      NOT SUPPLIED
        CONTROLLER REGISTERS:
        SEL 0:  100220
        SEL 2:  0
        SEL 4:  177777
        SEL 6:  177777

        DEVICE REGISTERS:
        RXCSR:  0
        RXDBUF: 0        =  NO ERROR BITS DETECTED
        TXCSR:  0        =  NO ERROR BITS DETECTED
        TXDBUF: 0
```

SHORT

```
SEQ    TIME    Mon 23 Jun 80

3. 08:52:48 NETWORK HARDWARE ERROR  FROM DN20L FOR LINE KDP_0_1_0
                  ERROR IS DDCMP START REC'D DURING NORMAL OPERATION
```

5.4.5 Network CHECK11 Report

Whenever the DN20 or DN200 is loaded, CHECK11 (a hardware test module) is started. All messages from CHECK11, at that time, become one entry in the system event file.

Note that the log data in this entry is an ASCIZ CHECK11 message of arbitrary length.

FULL

```
**********************************************
NETWORK CHECK11 REPORT
 LOGGED ON Mon 23 Jun 80 11:09:56       MONITOR UPTIME WAS 2 DAYS 10:54:28
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 12.
**********************************************
        MSG SENT FROM:  KL2137
        MSG REC'D AT:  KL2137
        HDWR TYPE:  UNKN   SOFTWARE TYPE:  UNKN
         PARENT SYSTEM TYPE:  UNKN
         MSG SEQUENCE # FROM XMIT NODE: 2.
TEXT FROM CHK11 REPORT:

CHK11 HARDWARE TEST
version 2A(21) of 10-AUG-79 by LDW
Testing begins...

THE PROCESSOR SEEMS TO BE A KD11-E (11/34)
   CHK11 EXPECTED AN 11/34

KT11 memory management test

PHYSICAL MEMORY HAS ABSOLUTE LIMITS OF
   0 - 757777
   FOR A TOTAL OF 124KW (DECIMAL)

MAPPED PHYSICAL MEMORY TEST...
      ...COMPLETE

KW11-L checked

device scan report assumes
       DN20
       DN21
       DN25 fixed assignments (no floating)
1 Fixed DTE20 at 174440, vector at 774
1 Fixed KMC11 at 160540, vector at 540
2 Fixed DUP11s from 160300, vector at 570
2 Fixed DMC11s from 160740, vector at 670

   CHK11 complete
```

SHORT

```
SEQ    TIME    Mon 23 Jun 80

12. 11:09:56 NETWORK CHECK11 REPORT
```

5.4.6 Network Line Statistics

Periodically, NETCON records the status of each communications line, and this information becomes an entry in the system event file.

FULL

```
**********************************************
NETWORK LINE STATISTICS
 LOGGED ON Mon 16 Jun 8Ø Ø8:34:19       MONITOR UPTIME WAS  Ø:34:48
        DETECTED ON SYSTEM # 2137.
        RECORD SEQUENCE NUMBER: 1.
**********************************************
        MSG SENT FROM:  DN2ØL
        MSG REC'D AT:  KL2137
        HDWR TYPE:  DTE-2Ø  SOFTWARE TYPE:  UNKN
         PARENT SYSTEM TYPE:  UNKN
        LINE ID:  DTE_1_Ø_Ø

        REASON FOR ENTRY:  PERIODIC ENTRY
1802.   SECONDS SINCE LAST ZEROED
808.    BLOCKS RECEIVED
814.    BLOCKS SENT
Ø.      NON - LINE ERROR RETRANSMISSIONS
```

SHORT

```
SEQ    TIME    Mon 16 Jun 8Ø

1. Ø8:34:19 NETWORK LINE COUNTERS  FROM NODE DN2ØL FOR LINE DTE_1_Ø_Ø
                    LINE ERROR RETRANS  RECV LINE ERRORS
```

5.5 DECNET ENTRIES (V3.Ø AND V4.Ø)

The DECnet V3.Ø and V4.Ø module Event Logger records any significant network events into the system event file. The headers for DECnet V3.Ø and V4.Ø entries have the title:

```
DECNET ENTRY
```

The body of each entry contains numbers that correspond to specific event classes and event types. Tables 5-1 and 5-2 list the meaning of the numbers in the entry. Refer to Section 4.4.3 for information on how to RETRIEVE network entries by event class.

Table 5-1: Network Event Classes

Event Class	Description
Ø	Network Management Layer
1	Applications Layer
2	Session Control Layer
3	Network Services Layer
4	Transport Layer
5	Data Link Layer
6	Physical Link Layer
7-31	Reserved for other common event classes
32-63	Reserved for RSTS specific event classes
64-95	Reserved for RSX specific event classes
96-127	Reserved for TOPS-2Ø specific event classes
128-159	Reserved for VMS specific event classes
16Ø-191	Reserved for RT specific event classes
192-479	Reserved for future use
48Ø-511	Reserved for Customer specific event classes

Table 5-2: Network Events

Class	Type	Entity	Event Text
Ø	Ø	none	Event records lost
Ø	1	node	Automatic node counters
Ø	2	line,circuit	Automatic data link counters
Ø	3	line,circuit	Automatic data link service
Ø	4	line,circuit	Data link counters zeroed
Ø	5	node	Node counters zeroed
Ø	6	line,circuit	Passive loopback
Ø	7	line,circuit	Aborted service request
2	Ø	none	Local node state change
2	1	none	Access control reject
3	Ø	none	Invalid message
3	1	none	Invalid flow control
3	2	node	Data base reused
4	Ø	none	Aged packet loss
4	1	circuit	Node unreachable packet loss
4	2	circuit	Node out-of-range packet loss
4	3	circuit	Oversized packet loss
4	4	circuit	Packet format error
4	5	circuit	Partial routing update loss
4	6	circuit	Verification reject
4	7	circuit	Circuit down, circuit fault
4	8	circuit	Circuit down, software fault
4	9	circuit	Circuit down, operator fault
4	10	circuit	Circuit up
4	11	circuit	Initialization failure, circuit fault
4	12	circuit	Initialization failure, software fault
4	13	circuit	Initialization failure, operator fault
4	14	node	Node reachability change
5	Ø	line,circuit	Locally initiated state change
5	1	line,circuit	Remotely initiated state change
5	2	line,circuit	Protocol restart received in maintenance mode
5	3	line,circuit	Send error threshold
5	4	line,circuit	Receive error threshold
5	5	line,circuit	Select error threshold
5	6	line,circuit	Block header format error
5	7	line,circuit	Selection address error
5	8	line,circuit	Streaming tributary
5	9	line,circuit	Local buffer too small
6	Ø	line	Data set ready transition
6	1	line	Ring indicator transition
6	2	line	Unexpected carrier transition
6	3	line	Memory access error
6	4	line	Communications interface error
6	5	line	Performance error

The following are examples of three DECnet entries
in FULL format:

```
**********************************************
DECNET ENTRY
 LOGGED ON 7-Dec Ø3:Ø1:49      MONITOR UPTIME WAS Ø DAY(S) 9:9:33
        DETECTED ON SYSTEM # 21Ø2.
        RECORD SEQUENCE NUMBER: 19.
**********************************************

Event type 4.1Ø  Line up
From node 118. (MCB), occurred 7-DEC-1981 Ø:ØØ:ØØ.4ØØ
CIRCUIT = DMC-Ø

  NODE = 121


**********************************************
DECNET ENTRY
 LOGGED ON 7-Dec Ø3:Ø1:5Ø      MONITOR UPTIME WAS Ø DAY(S) 9:9:35
        DETECTED ON SYSTEM # 21Ø2.
        RECORD SEQUENCE NUMBER: 2Ø.
**********************************************

Event type 4.14  Node reachability change
From node 118. (MCB), occurred 7-DEC-1981 Ø:ØØ:ØØ.466
REMOTE NODE = 1Ø3 ()

  STATUS = REACHABLE


**********************************************
DECNET ENTRY
 LOGGED ON 7-Dec Ø3:Ø2:Ø2      MONITOR UPTIME WAS Ø DAY(S) 9:9:47
        DETECTED ON SYSTEM # 21Ø2.
        RECORD SEQUENCE NUMBER: 21.
**********************************************

Event type 5.3  Send error threshold
From node 118. (MCB), occurred 7-DEC-1981 Ø:ØØ:18.ØØØ
CIRCUIT = KDP-Ø-Ø
```

The following are examples of the same three DECnet entries above but
these are listed in SHORT format:

```
19. Ø3:Ø1:49 DECNET Event type 4.1Ø  Line up
                    From node 118. (MCB)
                    occurred 7-DEC-1981 Ø:ØØ:ØØ.4ØØ

2Ø. Ø3:Ø1:5Ø DECNET Event type 4.14  Node reachability change
                    From node 118. (MCB)
                    occurred 7-DEC-1981 Ø:ØØ:ØØ.466

21. Ø3:Ø2:Ø2 DECNET Event type 5.3  Send error threshold
                    From node 118. (MCB)
                    occurred 7-DEC-1981 Ø:ØØ:18.ØØØ
```

The following DECnet entry lists packet header information:

```
**********************************************
 DECNET ENTRY
 LOGGED ON 27-Feb-84 Ø7:23:29-EST      MONITOR UPTIME WAS 1 DAY(S) Ø:2:17
        DETECTED ON SYSTEM # 2871.
        RECORD SEQUENCE NUMBER: 12Ø.
**********************************************

Event type 4.1  Node unreachable packet loss
From node 143. (GIDDN), uptime was 1 day(s) 16:56:39

  Packet Header = 2 / 142 / 143 / 6
```

From left to right, the four fields listed with the packet header have the following meanings:

Field one (2)	- is a hexadecimal value one byte long representing the message flags.
Field two (142)	- is a decimal (unsigned) value two bytes long representing the destination node address.
Field three (143)	- is a decimal (unsigned) value two bytes long representing the source node address.
Field four (6)	- is a hexadecimal value one byte long representing the forwarding data.

Note if the packet is a control packet, the packet header will contain only two fields, the message flags (Field one) and the source node address (Field three).

For more information on network event parameters, see Appendix F.

For more information concerning DECnet Versions 3.Ø and 4.Ø entries, refer to the DECnet documentation for system managers and operators.

APPENDIX A

SPEAR MESSAGES

There are four general categories of SPEAR messages; User Validation Messages, Dialogue Usage Messages, Warning Messages, and Event File Messages. The following tables list these messages and suggested actions.

Table A-1: User Validation Messages

The following messages can occur because of an error on the user's part. Each message is preceded by the header:

?USER Validation failed

CODE or SEQUENCE not allowed in list of responses

You have selected CODE or SEQUENCE as a response and have attempted to add another selection type.

Does not match any valid response

Typed a response that did not match one of the list of valid responses.

End time must be later than begin time

Typed an ending date/time that is prior to or the same as the beginning date/time in RETRIEVE or COMPUTE.

Invalid date format

Typed date incorrectly. The correct format is dd-mmm-yy or -dd.

Invalid time format

Typed time incorrectly. The correct format is hh:mm:ss.

Matches more than one valid response

Typed a response that was not unique. Need to type more characters before pressing the RETURN key or ESCAPE key.

Table A-1: User Validation Messages (Cont.)

May not select all at this prompt You tried to select ALL when you must respond with specific names or numbers. No recognition for this prompt Typed ESCAPE key where it is impossible to fill in the blanks. Not a valid name or number If a name, typed a special character or more than the maximum number of characters. If a number, typed a special character or alphabetic character or more than the maximum number of digits. That function is not available You typed a function name in the SPEAR library that does not exist in the same directory as SPEAR. This could happen if you do not have ANALYZE or if some of the programs are kept on tape.

Table A-2: Dialogue Usage Messages

The following messages can occur when you are responding to the dialogue incorrectly. They are meant to give you some insight as to what the correct response is to the current prompt. Not one of the recognized types At RETRIEVE level, when specifying a device, you typed a ? after typing a few characters. SPEAR did not recognize the device as one of its physical devices. Please select function first Typed a switch that requires some function to have been selected first (for example, /GO or /SHOW) at the SPEAR> prompt. Unable to complete this response You typed an ESCAPE to a prompt that SPEAR does not know how to complete. This is true whenever the response is not one of a fixed list of possible responses, for example, time of day or file specification. No default response for this prompt Typed the ESCAPE key or another delimiter where there is no default (at SPEAR> prompt, for example).

Table A-3: Warning Messages

The following is a list of warning messages you may receive during a SPEAR operation. Each message is introduced with the following sentence:

-- The following should be noted before proceeding --

Impossible to input event records from the terminal!

You specified TTY: in response to a request for a file specification.

The input file will be superseded!

In RETRIEVE, you named the output file the same name as the input file. This means you will overwrite your input file if you proceed.

Will overwrite input file with ASCII output!

In RETRIEVE, you specified the same name for both input and output files and also specified ASCII as the output format. If you proceed, the input file (which is binary) will be overwritten with ASCII output.

Binary output to terminal is unreadable!

In RETRIEVE, you requested the BINARY report format and then specified TTY: in response to Output to:

Merging with self causes duplicate records!

In RETRIEVE, you specified the same name for both the input file and the merge file. If you proceed, you will end up with a file containing duplicate records.

Will create an exact copy of the input file!

In RETRIEVE, you selected all the events in the system event file and then requested them in BINARY format. This is a waste of effort because all you will have succeeded in doing is duplicating the system event file.

Will create an empty output file!

In RETRIEVE, you have excluded everything during the selection process.

This function can cause SEVERE system degradation!

You have turned on the KLSTAT switch which slows down system operation to gather extra data into the system event file.

Table A-4: Event File Messages

The following messages can occur as the result of an error in the system event file. The message indicates a recoverable error. Each message is preceded with the following header:

%SPEAR Event file error detected in module ____routine ____

Bad header found - RESYNCHing

Lost synchronization in file, resynchronizing in next file block. Some data has been lost.

EOF encountered while skipping an entry

Error file is truncated for some reason. Some data has been lost.

Internal EOF found - RESYNCHing

Internal end-of-file mark detected but still has data. (This can happen if files are appended to each other.) No data is lost.

Premature EOF detected in error file!

Encountered an EOF in the middle of a header or entry. File is truncated. Some data is lost.

You can also receive fatal error messages in the form:

?SPEAR Program error in module ____routine ____

where the blanks are filled in with the module and routine names.

These are SPEAR program errors over which you have no control. If you receive such an error, fill out a Software Performance Report describing the error and the situation leading up to the error.

Another error over which you have no control is an error from an internal program called XPORT. XPORT does not identify itself in the message. However, the message is preceded by a question mark, indicating, in this case, that this is a fatal error. If you receive an XPORT error message, you should also fill out a Software Performance Report.

Other possible messages you can receive originate from the operating system. For example:

?SPEAR Monitor call failed	TOPS-20
?SCNxxx message	TOPS-10

On TOPS-20, you should refer to the Monitor Calls Manual for a list of these messages. On TOPS-10, you should refer to the SCAN documentation for a list of SCAN messages.

APPENDIX B

COMMAND AND CONTROL FILES

Because of dialogue changes in RETRIEVE and SUMMARIZE, if you have existing SPEAR V1.Ø command or control files, you must change them accordingly or they will not run.

For RETRIEVE, the changes from V1.Ø to V2.Ø are in the Selection type, Error and Nonerror fields. There are no changes necessary if your command or control file specified a Selection type of Error, All. See Section 4.4.3 for the RETRIEVE dialogue changes.

You can maintain the same functionality for an error selection by changing the V1.Ø dialogue to the following V2.Ø dialogue:

```
SPEAR V1.Ø              SPEAR V2.Ø
@SPEAR                  @SPEAR
*RETRIEVE               *RETRIEVE
*SERR:ERROR.SYS         *SERR:ERROR.SYS
*INCLUDED               *INCLUDED
*ERROR                  *ERROR
*DISK                   *DISK
*RPØ6                   *RPØ6
*FINISHED               *ALL (Here's the difference.)
*EARLIEST               *FINISHED
*LATEST                 EARLIEST
*DSK:RETRIE.RPT         *LATEST
*/GO                    DSK:RETRIE.RPT
                        */GO
```

To RETRIEVE the events for a specific device error type, replace the ALL in the previous V2.Ø control file with one or more device error types, for example, Software, Bus, Channel-controller.

For Nonerror selection, you can now select specific devices. Instead of Nonerror, specify Statistics, Configuration, Diagnostics, Other, or a combination of these separated by commas.

```
SPEAR V1.Ø              SPEAR V2.Ø

@SPEAR                  @SPEAR
*RETRIEVE               RETRIEVE
*SERR:ERROR.SYS         *SERR:ERROR.SYS
*INCLUDED               *INCLUDED
*NONERROR               *STATISTICS,DIAGNOSTICS (Change)
*EARLIEST               *DISK (Change)
*LATEST                 *RA6Ø,RA8Ø,RA81 (Change)
*DSK:RETRIE.RPT         *FINISHED (Change)
*/GO                    *EARLIEST
                        *LATEST
                        *DSK:RETRIE.RPT
                        */GO
```

For SUMMARIZE, two new prompts have been added to the dialogue, Category and Show Error Distribution. You can maintain the same functionality by changing the V1.Ø dialogue to the following V2.Ø dialogue:

```
SPEAR V1.Ø              SPEAR V2.Ø

@SPEAR                  @SPEAR
*SUMMARIZE              *SUMMARIZE
*SERR:ERROR.SYS         *SERR:ERROR.SYS
*EARLIEST               *ALL (Change)
*LATEST                 *EARLIEST
*DSK:SUMMAR.RPT         *LATEST
*/GO                    *YES (Change)
                        *DSK:SUMMAR.RPT
                        */GO
```

To get summaries for a specific device or class of devices, replace ALL in the previous V2.Ø dialogue with device selection. For example:

```
SPEAR V2.Ø

@SPEAR
*SUMMARIZE
*SERR:ERROR.SYS
*DISK
*RA6Ø,RA8Ø
*FINISHED
*EARLIEST
*LATEST
*YES
*DSK:SUMMAR.RPT
*/GO
```

To suppress the error distribution charts, change the YES to NO in the dialogue.

Because there are no changes in the dialogue for COMPUTE or KLSTAT, you need not change your previous control or command files for these functions.

APPENDIX C

EVENT CODES

The following table contains the current list of TOPS-10 and TOPS-20 event codes along with their internal class. The dashes (---) indicate that the event code does not exist under the specified operating system.

Table C-1: TOPS-10 and TOPS-20 Event Codes

-10 Code	Name	-20 Code	Internal Class	Subsystem
001	SYSTEMRELOAD	101	ERROR	MONITOR
002	MONITORBUGDATA	102	ERROR	MONITOR
005	EXTRACTEDCRASHINFO	---	ERROR	MONITOR
006	CHANNELERRORREPORT	---	ERROR	MAINFRAME
007	DAEMONSTARTED	---	CONFIG	SOFTWARE
010	OLD DISK ERROR	---	ERROR	DISK
011	MASSBUSERR	111	ERROR	DISK/TAPE
012	DX20ERR	---	ERROR	DISK/TAPE
014	SOFTWAREEVENT	---	ERROR	SOFTWARE
---	STATISTICS	114	STATISTICS	DISK/TAPE
015	CONFIGCHANGE	115	CONFIG	(ALL)
016	SYSERRORLOG	116	ERROR	SOFTWARE
017	SOFTWAREREQDATA	---	ERROR	SOFTWARE
021	TAPEERR	---	ERROR	TAPE
030	FEDEVICE-ERR	130	ERROR/CONFIG	MAIN/UNIT/COMM
031	FERELOAD	131	CONFIG	MAINFRAME
033	KSHALTSTATUS	133	ERROR	MAINFRAME
040	OLDDISKSTATS	---	STATISTICS	DISK
042	TAPESTATS	---	STATISTICS	TAPE
045	DISKSTATS	---	STATISTICS	DISK
050	DLHARDWAREERROR	---	ERROR	COMM
052	KLPARNXMINT	---	ERROR	MAINFRAME
054	KSNXMTRAP	---	ERROR	MAINFRAME
055	KLORKSPARTRAP	---	ERROR	MAINFRAME
056	NXMMEMORYSWEEP	---	ERROR	MAINFRAME
057	PARMEMORYSWEEP	---	ERROR	MAINFRAME
061	CPUPARTRAP	160	ERROR	MAINFRAME
062	CPUPARINT	162	ERROR	MAINFRAME
063	KLCPUSTATUS	163	ERROR	CRASH
064	DEVICESTATUS	---	ERROR	CRASH
---	MF20ERR	164	ERROR	MAINFRAME
066	OLDKLADDRESSFAIL	---	ERROR	MAINFRAME
067	KLADDRESSFAIL	---	ERROR	MAINFRAME
071	LP100ERR	---	ERROR	UNITRECORD
072	HARDCOPYERR	---	ERROR	UNITRECORD
201	NETCONSTARTED	201	CONFIG	NETWORK

Table C-1: TOPS-10 and TOPS-20 Event Codes (Cont.)

-10 Code	Name	-20 Code	Internal Class	Subsystem
202	NODEDOWNLINELOAD	202	CONFIG	NETWORK
203	NODEDOWNLINEDUMP	203	CONFIG	NETWORK
210	NETHARDWAREERR	210	ERROR	NETWORK
211	NETSOFTWAREERR	211	ERROR	NETWORK
220	NETOPRLOGENTRY	220	ERROR	NETWORK
221	NNETTOPOLOGYCHANGE	221	CONFIG	NETWORK
222	NETCHECK11REPORT	222	CONFIG	NETWORK
230	NETLINESTATS	230	STATISTICS	NETWORK
231	NETNODESTATS	231	STATISTICS	NETWORK
232	OLDDN64STATS	232	STATISTICS	NETWORK
233	DN6XSTATS	233	STATISTICS	NETWORK
234	DN6XENABLEDISABLE	234	CONFIG	NETWORK
240	DECnet Entry	240	ERROR	NETWORK
242	HSC50 END PACKET	242	ERROR	DISK/TAPE
243	HSC50 ERROR LOG	243	ERROR	DISK/TAPE
244	KLIPA EVENT	244	ERROR	CI
245	MSCP ERROR	245	ERROR	CI
250	DIAGNOSTIC EVENT	250	DIAGNOSTIC	(ALL)

APPENDIX D

DISK SUBSYSTEM ERROR BITS

The following charts list the categories into which the error bits fall in the SUMMARIZE report for Disk Subsystems.

For example, if the SUMMARIZE report states that your RPØ6 has six SK-SR (SEEK-SEARCH) errors, you may want to know what specific RPØ6 error bits are considered to be in this category. If you go to the SK-SR chart and look under device for RPØ4,5,6 (which means either RPØ4, RPØ5, or RPØ6), you will see that this chart shows that any one of the three error bits listed is considered as a SEEK-SEARCH error.

The headings have the following meanings:

ERROR NAME	The name listed in the KL1Ø Maintenance Guide.
DEVICE	The device type.
REG	The register containing the error bit.
BIT	The position of the error bit.
COMMENTS	Any qualifiers if applicable

The following is a list of the charts that will follow:

TIMIN	=	TIMING
SK-SR	=	SEEK-SEARCH
READ	=	READ-WRITE
CH-CO	=	CHANNEL-CONTROLLER
BUS	=	BUS
SOFT	=	HARDWARE DETECTED SOFTWARE ERROR
MICRO	=	MICROPROCESSOR DETECTED ERROR
UNSAF	=	UNSAFE
WRTLK	=	WRITE LOCK
OFFLI	=	OFFLINE

DISK SUBSYSTEM ERROR BITS

TIMIN

ERROR NAME	DEVICE	REG	BIT	Comments
OP INC	RP04,5,6	ERR 1	13	
DRIVE TIMING ERR	RP04,5,6	ERR 1	12	
INDEX ERROR	RP04,5,6	ERR 2	11	
INDEX UNSAFE	RP07	ERR 3	06	
DRIVE TIMING ERR	RP07	ERR 1	12	
OP INC	RP07	ERR 1	13	
OP INC	RM03,5	ERR 1	13	
OP INC	RK07	RKER	13	
DRIVE TIMING ERR	RK07	RKER	12	
E0	RL02	RLCS		See note after last chart
E3	RL02	RLCS		See note after last chart

SK-SR

ERROR NAME	DEVICE	REG	BIT	Comments
SEEK INC	RP04,5,6	ERR 3	14	
OFF CYL	RP04,5,6	ERR 3	15	
HEADER COMP ERR	RP04,5,6	ERR 1	07	
SEEK INC	RP07	ERR 3	14	
LOSS CYL ERROR	RP07	ERR 3	09	
HEADER COMP ERR	RP07	ERR 1	07	
HEADER COMP ERR	RM03,5	ERR 1	07	
SEEK INC	RM03,5	ERR 2	14	
SEEK INCOMPLETE	RK07	RKER	01	
DRIVE OFF TRACK	RK07	RKDS	05	
HEADER VERTICALRC	RK07	RKER	08	
SEEK TIME OUT	RL02	RLMP	12	
E1	RL02	RLCS		See note after last chart

DISK SUBSYSTEM ERROR BITS

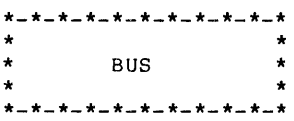
READ

ERROR NAME	DEVICE	REG	BIT	Comments
DATA CHECK	RPØ4,5,6	ERR 1	15	
HEADER CRC ERR	RPØ4,5,6	ERR 1	Ø8	
FORMAT ERR	RPØ4,5,6	ERR 1	Ø4	
BAD SECTOR ERR	RPØ7	ERR 3	15	
DATA CHECK	RPØ7	ERR 1	15	
HEADER CRC ERR	RPØ7	ERR 1	Ø8	
FORMAT ERR	RPØ7	ERR 1	Ø4	
SYNC BYTE ERROR	RPØ7	ERR 3	Ø2	
BAD SECTOR ERR	RMØ3,5	ERR 2	15	
DATA CHECK	RMØ3,5	ERR 1	15	
HEADER CRC ERR	RMØ3,5	ERR 1	Ø8	
FORMAT ERR	RMØ3,5	ERR 1	Ø4	
BAD SECTOR ERR	RKØ7	RKER	Ø7	
DATA CHECK	RKØ7	RKER	15	
ECC HARD ERR	RKØ7	RKER	Ø6	
FORMAT ERR	RKØ7	RKER	Ø4	
E2	RLØ2	RLCS		See note after last chart

CH-CO

ERROR NAME	DEVICE	REG	BIT	Comments
CHAN ERR	RH1Ø	CONI	2Ø	
OVER RUN	RH1Ø	CONI	22	and no drive errors
CHAN ERR	RH2Ø	CONI	22	
OVER RUN	RH2Ø	CONI	26	and no drive errors
IS TIMEOUT	RH78Ø	MBA SR	Ø1	
RD SUB	RH78Ø	MBA SR	Ø2	
INV MAP	RH78Ø	MBA SR	Ø4	
MAP PE	RH78Ø	MBA SR	Ø5	
DATA LATE	RH78Ø	MBA SR	11	and no drive errors
NOM EX MEM	RH75Ø	MBA SR	Ø1	
SPE	RH75Ø	MBA SR	14	
INV MAP	RH75Ø	MBA SR	Ø4	
MAP PE	RH75Ø	MBA SR	Ø5	
DATA LATE	RH75Ø	MBA SR	11	and no drive errors
NON EX MEM	RKØ7	RKCS2	11	
DATA LATE	RKØ7	RKCS2	15	
WRITECHECK	RKØ7	RKCS2	14	and Not Data Check
E4	RLØ2	RLCS		See note after last chart

DISK SUBSYSTEM ERROR BITS



ERROR NAME	DEVICE	REG	BIT	Comments
RAE	RH1Ø	CONI	29	
MDPE	RH1Ø	CONI	18	
PARITY ERR	RH1Ø	ER 1	Ø3	
RAE	RH2Ø	CONI	24	
MDPE	RH2Ø	CONI	18	and no Class B device errors
PARITY ERR	RH2Ø	ERR 1	Ø3	
MCPE	RH78Ø	MBA SR	17	
NON EX DRIVE	RH78Ø	MBA SR	18	
MDPE	RH78Ø	MBA SR	Ø6	
PARITY ERR	RH78Ø	ERR 1	Ø3	
MCPE	RH75Ø	MBA SR	17	
NON EX DRIVE	RH75Ø	MBA SR	18	
MDPE	RH75Ø	MBA SR	Ø6	
PARITY ERR	RH75Ø	ERR 1	Ø3	
PARITY ERR	RPØ7	ERR 1	Ø3	
DATA PARITY ERROR	RPØ7	ERR 3	Ø3	
NON EX DRIVE	RKØ7	RKCS2	12	
DR TO CNTRL PE	RKØ7	RKCS1	13	
CNTRL TO DR PE	RKØ7	RKER	Ø3	
CONTROLLER TIMEOUT	RKØ7	RKCS1	11	
MULTIPLE DRIVE SEL	RKØ7	RKCS2	Ø9	
UNIT FIELD ERR	RKØ7	RKCS2	Ø8	
DRIVE SEL ERR	RLØ2	RLMP	Ø8	

DISK SUBSYSTEM ERROR BITS

SOFT

ERROR NAME	DEVICE	REG	BIT	Comments
INVALID ADDR ERR	RP04,5,6	ERR 1	10	
ADDR OVERFLOW ERR	RP04,5,6	ERR 1	09	
REG MOD RFSD	RP04,5,6	ERR 1	02	
ILL REG	RP04,5,6	ERR 1	01	
ILL FUNCTION	RP04,5,6	ERR 1	00	
INVALID ADDR ERR	RP07	ERR 1	10	
ADDR OVERFLOW ERR	RP07	ERR 1	09	
REG MOD RFSD	RP07	ERR 1	02	
ILL REG	RP07	ERR 1	01	
ILL FUNCTION	RP07	ERR 1	00	
PROG ERR	RP07	ERR 2	15	
INVALID ADDR ERR	RK07	RKER	10	
PROGRAM ERROR	RK07	RKCS2	10	
ADR OVERFLOW ERR	RK07	RKER	09	
DRIVE TYPE ERR	RK07	RKER	05	
NONEXECUTIBLE FNC	RK07	RKER	02	
ILL FUNCTION	RK07	RKER	00	

MICRO

ERROR NAME	DEVICE	REG	BIT	Comments
CROM PARITY ERR	RP07	ERR 2	14	
MP UNSAFE	RP07	ERR 2	13	
DEFECT SKIP ERR	RP07	ERR 3	13	
CONTROL LGIC FAIL	RP07	ERR 3	11	
LOSS OF BIT CLOCK	RP07	ERR 3	10	
MP HANDSHAKE	RP07	ERR 3	08	
SERDES DATA FAIL	RP07	ERR 3	04	
SYNC CLOCK FAIL	RP07	ERR 3	01	
RUNTIME OUT	RP07	ERR 3	00	
FAULT CODE	RPO7	ERR 2	00-07	Any nonzero value

DISK SUBSYSTEM ERROR BITS

ERROR NAME	DEVICE	REG	BIT	Comments
AC LOW	RPØ4,5,6	ERR 3	Ø6	
DC LOW	RPØ4,5,6	ERR 3	Ø5	
WR OS	RPØ5,6	ERR 3	Ø1	
DC UN	RPØ5,6	ERR 3	ØØ	
NO H SEL	RPØ4,5,6	ERR 2	1Ø	
MULTI H SEL	RPØ4,5,6	ERR 2	Ø9	
TRAN UNSF	RPØ4,5,6	ERR 2	Ø6	
TRAN DET F	RPØ4,5,6	ERR 2	Ø5	
C SW UNSF	RPØ4,5,6	ERR 2	Ø3	
W SEL UNSF	RPØ4,5,6	ERR 2	Ø2	
C SK UNSF	RPØ4,5,6	ERR 2	Ø1	
ACUN	RPØ4	ERR 2	15	
PLO UNS	RPØ4,5,6	ERR 2	13	
3ØVU	RPØ4	ERR 2	12	
WRITE UNSF	RPØ4,5,6	ERR 2	Ø8	
WR C UNSF	RPØ4,5,6	ERR 2	ØØ	
UNSAFE	RPØ7	ERR 1	14	REG 2<11-13>RD/WRT1-3,REG3<5>DC UNS
R/W 3 UNSAFE	RPØ7	ERR 2	12	
R/W 2 UNSAFE	RPØ7	ERR 2	11	
R/W 1 UNSAFE	RPØ7	ERR 2	1Ø	
WRITE OVERRUN	RPØ7	ERR 2	Ø9	
WRITE READY UNSAF	RPØ7	ERR 2	Ø8	
WRITE CURENT FAIL	RPØ7	ERR 3	12	
DC UNSAFE	RPØ7	ERR 3	Ø5	
UNSAFE	RMØ3,5	ERR 1	14	
DEVICE CHK	RMØ3,5	ERR 2	Ø7	
UNSAFE	RKØ6,7	RKER	14	
SPEED LOSS	RKØ6,7	RKDS	Ø4	
ACLO	RKØ6,7	RKDS	Ø3	
WRITE DATA ERR	RLØ1,2	RLMP	15	
CURRENT HEAD ERR	RLØ1,2	RLMP	14	
SPEN ERR	RLØ1,2	RLMP	11	
WRITE GATE ERR	RLØ1,2	RLMP	1Ø	and Not Write Locked

```
*-*-*-*-*-*-*-*-*-*-*-*
*                     *
*        WRTLK        *
*                     *
*-*-*-*-*-*-*-*-*-*-*-*
```

ERROR NAME	DEVICE	REG	BIT	Comments
WRITE LOCK ERR	RPØ4,5,6	ERR 1	11	
WRITE LOCK ERR	RPØ7	ERR 1	11	
WRITE LOCK ERR	RMØ3,5	ERR 1	11	
WRITE LOCK ERR	RKØ7	RKER	11	
WRITE LOCK	RLØ2	RLMP	13	and Write Gate Error

```
*-*-*-*-*-*-*-*-*-*-*-*
*                     *
*        OFFLI        *
*                     *
*-*-*-*-*-*-*-*-*-*-*-*
```

ERROR NAME	DEVICE	REG	BIT	Comments
MEDIUM ON LINE	RP04,5,6	DS	12	OFFLINE when not true
MEDIUM ON LINE	RPØ7	DS	12	OFFLINE when not true
MEDIUM ON LINE	RMØ3,5	DS	12	OFFLINE when not true

```
!*****  RLØ2 NOTE ****
!
! NOTE THAT THESE 3 BITS (1Ø,11,& 12) OF THE CS REG ARE GROUPED
! TO DETERMINE THE ERROR AS FOLLOWS (x means we don't care the state of the bit)
!  12   11      1Ø      RESULT
!  DLT  CRC     OPI
!   Ø    Ø       1 = OPI                                  EØ
!   x    1       1 = HEADER CHECK                         E1
!   x    1       Ø = DATA CRC IF READ OPERATION           E2
!                    WRITE CHECK IS WRITE OPERATION
!   1    x       1 = HEADER NOT FOUND                     E3
!   1    x       Ø = DATA LATE                            E4
!
!*****
```

APPENDIX E

NETWORK EVENT PARAMETERS

Network Management Layer Event Parameters - Class Ø

Type	Keywords
Ø	SERVICE Ø = LOAD 1 = DUMP
1	STATUS Return code Ø = REQUESTED >Ø = SUCCESSFUL <Ø = FAILED Error detail (if error) Error message (optional)
2	OPERATION Ø = INITIATED 1 = TERMINATED
3	REASON Ø = Receive timeout 1 = Receive error 2 = Line state change by higher level 3 = Unrecognized request 4 = Line open error

Session Control Layer Event Parameters - Class 2

Type	Keywords
Ø	REASON Ø = Operator command 1 = Normal operation
1	OLD STATE Ø = ON 2 = SHUT 1 = OFF 3 = RESTRICTED
2	NEW STATE Ø = ON 2 = SHUT 1 = OFF 3 = RESTRICTED
3	SOURCE NODE
4	SOURCE PROCESS
5	DESTINATION PROCESS
6	USER
7	PASSWORD (Ø means password set; n parameter means not set)
8	ACCOUNT

Network Services Layer Event Parameters - Class 3

Type	Keywords
Ø	MESSAGE Message flags Destination link address Source link address Data
1	CURRENT FLOW CONTROL Ø = No flow control 1 = Segment flow control 2 = Message flow control

Routing Layer Event Parameters - Class 4

Type	Keywords
Ø	PACKET HEADER Message flags Destination node address (not for control packet) Source node address Forwarding data (not for control packet)
1	PACKET BEGINNING
2	HIGHEST ADDRESS
3	NODE
4	EXPECTED NODE
5	REASON Ø = Line synchronization lost 1 = Data errors 2 = Unexpected packet type 3 = Routing update checksum error 4 = Adjacent node address change 5 = Verification receive timeout 6 = Version skew 7 = Adjacent node address out of range 8 = Adjacent node block size too small 9 = Invalid verification seed value 1Ø = Adjacent node listener received timeout 11 = Adjacent node listener received invalid data
6	RECEIVED VERSION
7	STATUS Ø = REACHABLE 1 = UNREACHABLE

Data Link Layer Event Parameters - Class 5

Type	Keywords
Ø	OLD STATE Ø = HALTED 3 = RUNNING 1 = ISTRT 4 = MAINTENANCE 2 = ASTRT
1	NEW STATE Ø = HALTED 3 = RUNNING 1 = ISTRT 4 = MAINTENANCE 2 = ASTRT
2	HEADER
3	SELECTED TRIBUTARY
4	PREVIOUS TRIBUTARY
5	TRIBUTARY STATUS Ø = Streaming 1 = Continued send after timeout 2 = Continued send after deselect 3 = End streaming
6	RECEIVED TRIBUTARY
7	BLOCK LENGTH
8	BUFFER LENGTH
9	DTE
1Ø	REASON
11	(Reserved)
12	(Reserved)
13	PARAMETER TYPE
14	CAUSE
15	DIAGNOSTIC

Physical Line Layer Event Parameters - Class 6

Type	Keywords
Ø	DEVICE REGISTER
1	NEW STATE Ø = OFF 1 = ON

APPENDIX F

GLOSSARY

The following is a list of terms explained within the context of this document.

Term	Explanation
Body section	The data portion of an entry in the system event file.
BUGCHK	A recoverable error detected by the TOPS-20 operating system.
BUGHLT	A non-recoverable error detected by the TOPS-20 operating system.
BUGINF	A message informing you that a certain event relating to the TOPS-20 operating system has occurred.
CTY	The system operator's terminal.
Dump format	One of the three output forms of the RETRIEVE procedure.
Entry type	The type of entry within a system event file, for example, a MASSBUS Device Error, or a Crash Restart Error.
ERROR.SYS	The name of the system event file in both the TOPS-10 and TOPS-20 operating systems.
Event code	The octal code designated to a particular event in the system event file.
FRU	An acronym for Field Replaceable Unit. This is a piece of hardware that the Field Service engineer can replace on the spot.
Full format	A complete and detailed listing of an event, in ASCII as translated with RETRIEVE.
Hard error	A non-recoverable error.
Header section	The top portion of an entry in the system event file, after SPEAR formats it.

Term	Explanation
MTTR	An acronym for Mean Time To Repair. The average time it takes a Field Service engineer to isolate and repair a system malfunction.
NXM error	An attempt to address a nonexistent memory location.
Parity error	Indicates that one or more bits have been picked up or dropped to cause a nonparity condition.
RETRIE.RPT	A file containing entries converted from binary to ASCII.
RETRIE.SYS	A file in binary format containing entries extracted from the system event file.
Retry count	The number of times an operation is tried, in addition to the first time.
Sequence number	The number given to an entry in the system event file.
Short format	A brief version of an entry in the system event file, after SPEAR has translated it.
Snapshot	The information gathered by the operating system immediately after recovering from a crash.
Soft error	A recoverable error.
Stopcode	A message containing a 3-letter code printed at the CTY indicating that a serious error has occurred in the operating system's data base.
System event file	The file where the operating system records hardware and software events.
Sweep	After certain events occur, the operating system checks core looking for more of the same.

INDEX

TOPS–10/TOPS–20
SPEAR Manual
AA–J833B–TK

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify) _____

Name _____ Date _____

Organization _____ Telephone _____

Street _____

City _____ State _____ Zip Code _____
or Country

-------Do Not Tear - Fold Here and Tape-------

digital

No Postage Necessary if Mailed in the United States

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

SOFTWARE PUBLICATIONS
200 FOREST STREET MRO1–2/L12
MARLBOROUGH, MA 01752

------- Do Not Tear - Fold Here and Tape -------

Cut Along Dotted Line