RSX-11M
Error Logging
Reference Manual

Order No. AA-2542C-TC

digital

RSX-11M
Error Logging
Reference Manual

Order No. AA-2542C-TC

RSX-11M Version 3.1

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation · maynard, massachusetts

First Printing, April 1976
Revised: November 1976
December 1977

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 be used or copied only in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

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.

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECtape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-10
DECCOMM	DECSYSTEM-20	TYPESET-11

CONTENTS

FIGURES

TABLES

PREFACE

0.1 MANUAL OBJECTIVES AND READER ASSUMPTIONS

This manual contains information needed to operate the error logging subsystem. It enables the user to produce error logging reports. These reports allow the user to monitor the hardware reliability of an RSX-11M system. They also provide DIGITAL field service engineers with a convenient history of system performance.

This manual assumes a familiarity with the following documents:

The RSX-11M Operator's Procedures Manual

The RSX-11M Utilities Procedures Manual

The RSX-11M System Generation Manual

0.2 STRUCTURE OF THE DOCUMENT

Chapter 1 describes the purposes and functions of error logging.

Chapter 2 describes how the Executive logging features and the error logging tasks interface to produce the final, readable reports.

Chapter 3 describes the procedures for operating the error logging tasks and lists associated messages.

Chapter 4 describes the error log reports generated by the subsystem.

Appendix A explains how to modify the analyzer task SYE at task build.

Appendix B describes memory-resident error data formats.

Appendix C describes the record formats of temporary files produced by the error logging task ERRLOG.

Appendix D describes record formats contained in the file ERROR.SYS.

Appendix E discusses modifications made to mass storage device drivers to enable error logging. It also discusses several Executive routines supplied for the error logging subsystem.

0.3 ASSOCIATED DOCUMENTS

The RSX-11M/RSX-11S Documentation Directory defines the intended readership of each manual in the RSX-11M set and provides a brief synopsis of each manual's contents. References are made in the error logging manual to several other RSX-11M documents; readers should consult the directory for information about these documents.

CHAPTER 1

INTRODUCTION

1.1 THE PURPOSES OF ERROR LOGGING

The RSX-11M error logging subsystem monitors the hardware reliability of an RSX-11M system; it continually detects and records information about every disk, DECtape, magnetic tape, and memory parity error as it occurs, regardless of whether or not it is recoverable. Then at user-determined intervals, a report-generating task can be run to produce individual error and/or summary reports on some or all of these errors. Even without the error logging facility, the Executive automatically retries recoverable errors; but after a successful recovery, the user might be unaware that the error ever occurred.

Error logging may be implemented on any RSX-11M system that has 24K words or more of memory.

Error logging reports are useful for efficient maintenance of the hardware on which the RSX-11M system runs. Problems such as line noise, static discharges, or inherently error-prone media, for instance, can cause recoverable errors on systems that are otherwise functioning normally. By studying error logging reports, the user can learn to distinguish these kinds of errors from those that might be symptoms of an impending device failure. On the other hand, some recoverable errors that are insignificant in themselves might be related to other, more serious errors, the effects of which are not immediately apparent. Information contained in the reports about each error and about the status of the system when the error occurred may alert the system manager to a previously unforeseen hardware problem.

If the error reports seem to indicate an impending failure, the system manager should contact a DIGITAL field service engineer who will use the reports to help in diagnosing the problem. Sometimes a device fails so quickly that error logging is unable to predict the failure in time to prevent it. In this case, the field service engineer can determine the cause more quickly if a report is available which describes the errors that may have occurred immediately prior to the failure.

1.2 THE FUNCTIONS OF ERROR LOGGING

Without the error logging facility installed in the system, the RSX-11M Executive detects each hardware error, then either ignores it if it causes no immediate problem or, when appropriate, retries the function that caused the error. The user normally has no means of knowing that such an error occurred. However, when the error logging facility has been generated, the system detects the errors and then performs three further operations in order to produce an error report. In summary, error logging:

1. Detects a hardware error as it occurs,
2. Gathers information about the error,
3. Stores the information in a file, and
4. Formats the information to produce an error report.

Control of the facility is shared between routines in the Executive and specific error logging tasks. These routines and tasks interface with each other to carry out the four operations described above. The error logging tasks are ERRLOG, PSE, SYE and ERF. Chapter 2 describes how the Executive routines and error logging tasks interact to produce the final reports. Chapter 3 tells the user how to run these tasks and lists the error messages associated with each of them. The task SYE, called the Report Generator, produces the error logging reports according to user-specified switches. These reports are described in detail in Chapter 4.

1.2.1 Error Detection

Error logging routines in the Executive respond to four types of errors:

1. Unexpected traps or interrupts,
2. Device errors,
3. Interrupt timeouts, and
4. Memory parity errors.

If the error logging facility has been generated into the Executive and the error logging tasks have been installed, all unused vectors are filled with pointers to error logging routines. Therefore, when an unusually noisy electrical environment or a static discharge causes an unexpected trap or interrupt to occur to an unused vector, or a valid interrupt to be corrupted to the wrong address, Executive routines can determine and record the incorrectly used vector.

Device errors are conditions that cause a device controller to interrupt with its error bit set. When a device error occurs and error logging is active, Executive routines record the contents of the device registers, which indicate the state of the device and its controller. In addition, these routines record information about the actual I/O request to aid in the interpretation of the device error.

An interrupt timeout occurs when a device on which a transfer was initiated fails to interrupt within a specified amount of time. Interrupt timeouts are detected by software timers started when the transfer is initiated. The system records the same information for interrupt timeouts as it does for device errors, except no I/O activity is reported.

Memory uses byte parity to ensure the integrity of information. The system generates parity for both data and addresses on transfers to memory; this parity is then checked on all transfers from memory. A parity error occurs when this check fails.

1.2.2 Information Gathered

The error information gathered by the Executive routines (the state of the registers when a device error occurs, for example) provides a "snapshot" of the relevant parts of the system at the time of the error. In addition to all the system information, error routines identify the type of error and the associated device (for device errors and interrupt timeouts), record the time the error occurred and assign a sequence number to the error.

1.2.3 The Error Log File

The first two operations, error detection and the collection of error information, are continual processes. When a certain amount of error information has been gathered, the system calls the error logging task ERRLOG to copy the information to a permanent error log file. The data may be copied to a file on the system disk or to a file on a removable disk or DECtape.

1.2.4 Formatting and Report Generation

When it is convenient or necessary to generate an error log report, the user first runs the pre-formatting task PSE to make a preliminary consolidation of the raw data contained in one or more error log files. The PSE output file becomes input to SYE, the task that actually generates the finished error logging report.

SYE is capable of generating individual error reports and/or summary reports. Among many other options, the user may specify a report that covers a certain time period, a certain device or group of devices, or perhaps a certain type of error. See Chapter 4 for a detailed description of the error reports that SYE can generate.

Because the error log files may be written to a removable volume, the user can generate the reports either on site or at any other RSX-11M installation that supports the error logging facility.

CHAPTER 2

HOW ERROR LOGGING WORKS

2.1 EXECUTIVE FEATURES

The RSX-11M Executive is responsible for

- Detecting errors,
- Gathering volatile data that reflects, as closely as possible, the state of the system at the time of each error, and
- Controlling the amount of system dynamic memory used for error logging.

A set of common error routines performs these functions for all four types of detected errors (unexpected traps or interrupts, device errors, interrupt timeouts, and memory parity errors).

The routines use system information made available by the Executive. For example, the Executive maintains an I/O Active Bitmap that indicates which devices are active at any given moment. When an error occurs, an Executive routine notes the contents of the bitmap for subsequent analysis. The bitmap could show that the error was related to activity on some other device. Also, if error logging has been generated into the system, all unused vectors are filled with pointers to the error routines. Therefore, if a device improperly interrupts through an unused vector, an error routine is able to record the address of the vector.

The common error routines are called every time an error occurs. They then perform the following three error logging functions:

1. One routine allocates memory for error logging activity.
2. Another gathers relevant error data and formats the raw error messages.
3. The last routine queues the formatted error messages in preparation for logging by ERRLOG.

Figure 2-1 is a flow chart diagram of the Executive routine activity described in this section.

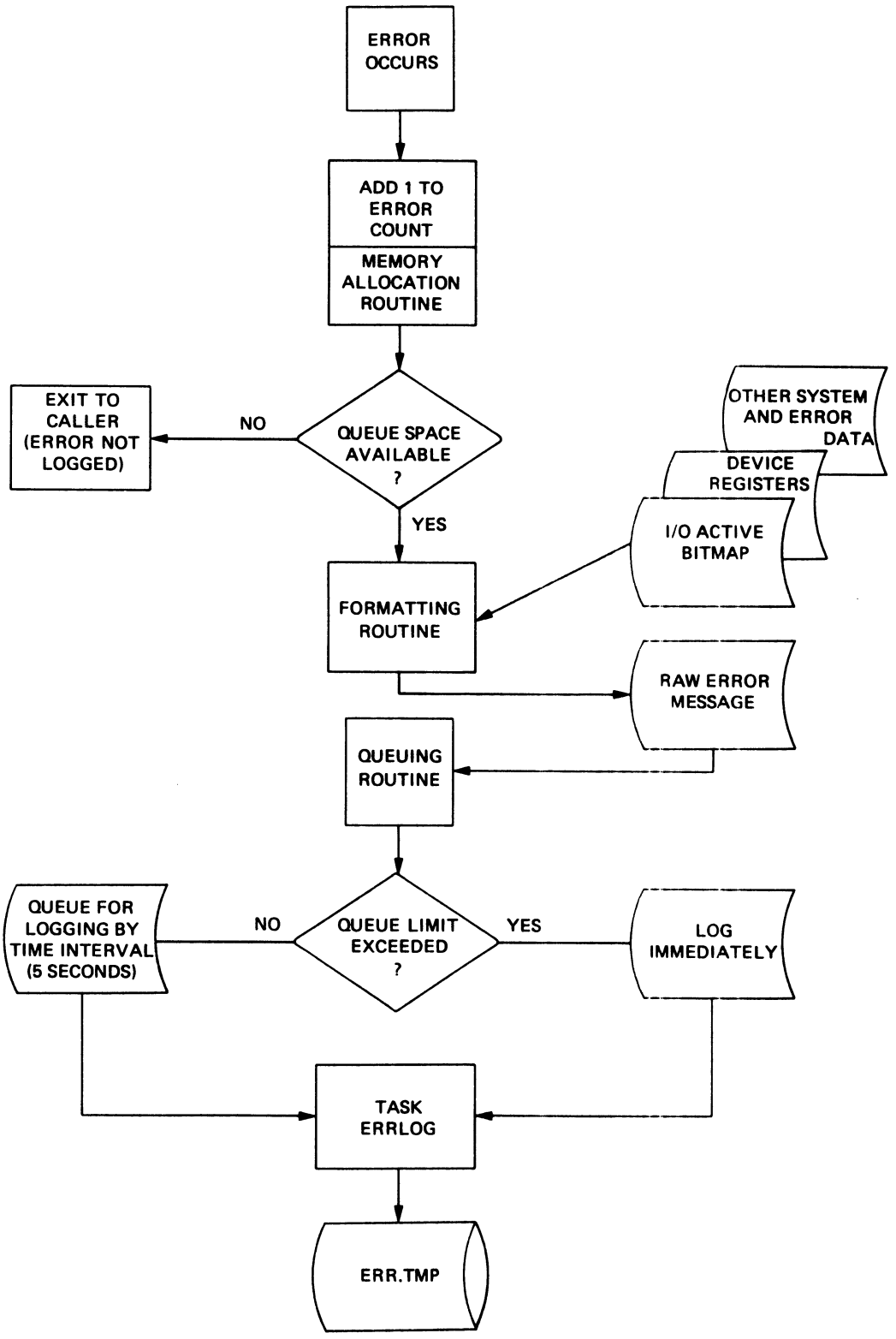

Figure 2-1 Executive Error Routines

2.1.1 Memory Allocation Routines

The memory allocation routine makes memory space available for logging error information and places a limit on the amount of space available. Normally each error results in a record being placed in this memory space, and a counter (the error sequence number) is incremented by one. The task ERRLOG then writes to disk all the error records in the memory space and frees the space for future error records. However, when a burst of errors occurs that causes the available memory space to be exceeded, the error record that caused the overflow and all subsequent error records are not placed in the memory space. The error sequence number continues to be incremented by one for each error even if it is not logged. When ERRLOG writes the error records to disk, the memory space is freed and normal recording of error records resumes.

2.1.2 Error Message Formatting Routines

When an error occurs, the formatting routine for that type of error creates a raw error message containing the following information:

- The error's sequence number,
- The date and time it occurred,
- A code that classifies the error, and
- Any additional information needed to describe the error.

Appendixes B and C contain further details about the error message formats and formatting routines.

2.1.3 Error Message Queuing Routine

The queuing routine provides the link between Executive error logging activity and the task ERRLOG. After a raw message has been formatted, the routine enters it into a queue, which ERRLOG periodically writes to a file (ERR.TMP). ERRLOG scans the queue and logs the contents at most 5 seconds after the occurrence of an error, and sooner if a burst of errors causes the queue to approach its limit. (The memory allocation routine determines the amount of space that the queue may occupy; see section 2.1.1 above.) The error data queue is cleared each time ERRLOG logs its contents.

2.2 TASK INTERACTION

The primary function of the error logging tasks is to maintain the error information gathered by the Executive in order to produce a report. ERRLOG preserves the Executive's raw information and passes it on to PSE when that task is activated. PSE prepares the data for input to SYE, which selectively generates the final, readable report. The following sections describe the file activity involved in this task interaction. Figure 2-2 is a flow chart that demonstrates the file activity.

CHAPTER 3

OPERATING PROCEDURES

This chapter describes the operating procedures for the three error logging tasks (ERRLOG, PSE and SYE) and for the task that terminates error logging (ERF). Error logging must be properly generated into the system before the user can invoke these tasks (see the RSX-11M System Generation Manual). Error logging requires very little operator intervention. After it has been activated either explicitly by the operator from a privileged terminal or automatically by a command in the system startup command file, its operation is transparent to the user until error logging reports need to be generated.

Each error logging task issues a message whenever it encounters a condition (not related to errors it is logging) that halts or interferes with the task's operation. The tasks also issue informative messages (when a new file is opened in response to a request from PSE, for example). Both types of message are listed, for each of the error logging tasks, in Section 3.5.

3.1 RUNNING ERRLOG (THE ERROR LOGGER)

ERRLOG is the only task that needs to be installed permanently for error logging to function.

To install ERRLOG, type the following at a privileged terminal:

```
>INS $ERL
```

The file ERL.TSK contains the ERRLOG task image. To ensure that error logging runs efficiently, ERL.TSK must be installed in a partition that meets the following criteria:

- All other tasks in the partition must be checkpointable and must have a lower priority than ERRLOG.
- ERRLOG should not run in the same partition as PSE and ERF unless the partition is large and system-controlled.
- ERRLOG must not run in the same partition as F11ACP since it uses its services.

To activate error logging, initiate ERRLOG from a privileged terminal by typing:

```
>RUN ERRLOG
```

The system startup command file (LB:[1,2]STARTUP.CMD) may include the above command. If so, error logging is activated automatically at system startup.

Note that ERRLOG requires that the directory [1,6] exist on the system disk. If it does not exist, ERRLOG returns the following message:

```
ERL -- GET DIRECTORY ID FAILED, CODE -26
```

Then issue the following command to create the necessary directory:

```
>UFD SY:[1,6]
```

3.2 RUNNING PSE (THE PRE-FORMATTER)

PSE must be run from a privileged terminal. To install and run PSE, type the following:

```
>INS $PSE

>PSE
```

As distributed, PSE builds to run in an 8K system-controlled partition (normally the GEN partition), providing enough space for 102. devices on a mapped system or 103. devices on an unmapped system. To increase the amount of space available to PSE, append the /INC=xxx keyword to the Install command. (Refer to the RSX-11M Operator's Procedures Manual for further information on the /INC switch to the MCR INStall command.) PSE needs additional space at the rate of 10(decimal) words per unit, where a unit is each device that could cause a loggable error. When running in a user-controlled (task) partition, PSE itself determines the amount of space available and then uses the amount of space it needs.

The pre-formatter task responds with the prompt PSE> and waits for the user to type a command line. The format of the PSE command line is:

```
outdev:[ufd]file.typ=indev:
```

where the output file is described by a standard RSX-11M file specifier. The input file specifier (indev:) consists only of the input device since the filename (assigned by the task ERRLOG) is always ERROR.TMP and the UFD is always [1,6].

Both the call to PSE and the file specifiers may be included on the same line; that is:

```
>INS $PSE

>PSE outdev:[ufd]file.typ=indev:
```

The defaults for omitted fields are:

Field	Default
outdev:	SY0:
ufd	[1,6]
file	ERROR
.typ	.SYS
indev:	SY0:

To use all the defaults for both the input and output file specifiers, press the carriage return key in response to the PSE> prompt. When PSE prompts again, type CTRL/Z to return to MCR.

Example:

```
>INS $PSE              ;INSTALL PSE
>PSE                   ;RUN PSE
PSE> <CR>              ;GENERATE PRE-FORMATTED FILE
PSE> ^Z                ;EXIT FROM PSE
>REM PSE               ;REMOVE PSE
```

3.3 USING SYE (THE REPORT GENERATOR)

Use of the report generator task SYE involves invoking the task through one of three ways and specifying a command line that will generate the desired reports. The following subsections discuss invoking SYE and the SYE command line.

3.3.1 Invoking SYE

There are two methods for invoking SYE:

1. To install and run SYE, type the following commands at a privileged terminal:

   ```
   >INS $SYE

   >SYE
   ```

2. To install and run SYE, if the conditions described in Section 3.1 have been met, type the following command at any terminal:

   ```
   >RUN $SYE
   ```

The report generator task responds with a message and the prompt SYE>, and waits for the user to type a command line.

For example:

```
>RUN $SYE
SYE>
```

3.3.2 The SYE Command Line

The format of the SYE command line is

```
outdev:[ufd]file.typ=indev:[ufd]file.typ/switch1.../switchn
```

If SYE has been explicitly installed (items 1 and 2 in Section 3.3.1 above), both the call to SYE and the file specifiers may be included on the same line; that is:

```
>SYE outdev:[ufd]file.typ=indev:[ufd]file.typ/switch1.../switchn
```

The default values for the output file specifier are:

Field	Default
outdev:	SY0:
ufd	user uic
file	ERRLOG
typ	.LST

The input file is described by a standard RSX-11M file specifier; it must be a file previously created as an output file by PSE (see Section 3.2).

The default values for the input file specifier are:

Field	Default
indev:	SY0:
ufd	[1,6]
file	ERROR
.typ	.SYS
switches	/-RP

To use all the defaults for both the input and the output file specifiers, press the carriage return key in response to the SYE prompt. When SYE prompts again, type CTRL/Z to return to MCR. For example:

```
SYE> <CR>
SYE> ^Z
>
```

Every invocation of SYE produces summary error reports. These summary error reports are described in detail in Section 4.3 of Chapter 4. Through the use of either of two command line switches, /RP and /DV, the user can cause additional error reports to be generated along with the summary reports. As shown below, the /RP switch allows the creation of general classes of error reports while the /DV switch generates reports specific to a particular device type or device unit.

Switch	Description
/RP[:class]	The /RP switch requests individual error reports; the class parameter is used to specify a general classification of errors to be reported. If no class parameter is specified with the /RP switch, SYE reports on all classes. Class has one of the following values: null — All classes. SYS — All system errors (error logging startup, shutdown, power fails, and PSE entries) which are significant to DIGITAL service groups such as field service and software support.

HDW — All hardware errors. When class is HDW, an additional parameter can be specified further defining the errors to be reported. This parameter takes the form:

```
/RP:HDW[:type]
```

where

Type is one of the following:

null	All types of hardware errors.
DSK	All disk errors.
MAG	All magnetic tape and DECtape errors.
MEM	All cache and main memory parity errors.

/-RP or /NORP — Do not include individual error reports. SYE produces a summary report only. /-RP is the default.

/DV:dev[n] — Include in the report only those errors that occurred on a specified device type or on a specified device unit.

Examples:

/DV:DK requests that error statistics on all RK03 or RK05 units be provided

/DV:DK1 requests that only the error statistics for RK03 or RK05 unit 1 be provided.

The remaining switches that can be specified on the SYE command line are:

/BEG:time-and-date — Include in the report only those errors logged after the specified time and date. The format of the time-and-date parameter is:

```
hh:mm:ss:mm:dd:yy
```

where all the numbers are decimal and leading zeros are not required. Null equals zero. The field terminator (:) must be entered for a null field. The hours (hh) field is expressed on the basis of a 24-hour clock.

Example:

/BEG:20:30:0:11:15:76 includes those errors that occurred after 8:30 PM on November 15, 1976.

The default is to include all errors of the specified type, no matter when they occurred.

/END:time-and-date

Include in the report only those errors logged prior to the specified time and date. The format of the time-and-date parameter is:

hh:mm:ss:mm:dd:yy

where all the numbers are decimal and leading zeros are not required. Null equals zero. The field terminator (:) must be entered for a null field. The hours (hh) field is expressed on the basis of a 24-hour clock.

Example:

/END:18:0:0:10:18:76 includes those errors that occurred before 6:00 PM on October 18, 1976.

The default is to include all errors of the specified type, no matter when they occurred.

/HEL[P]

The /HELP switch causes SYE to display SYE operating instructions.

To obtain this information, the user specifies the /HELP switch on its own in response to the SYE prompt as follows:

SYE>/HELP <CR>

/SP

Spool output file. This is the default. No file is printed if spooling has not been generated into the system.

SYE command line examples:

- >SYE <CR>

  Produce file ERRLOG.LST containing a summary report of all errors in file SY0:[1,6] ERROR.SYS. Spool ERRLOG.LST to the line printer.

- >SYE /RP

  Produce file ERRLOG.LST containing individual reports and a summary of all errors in file SY0:[1,6]ERROR.SYS. Spool ERRLOG.LST to the line printer.

- >SYE /DV:DB0

  Produce file ERRLOG.LST containing individual reports and a summary of only DB0 (RP04, RP05, RP06) errors in file SY0:[1,6] ERROR.SYS. Spool ERRLOG.LST to the line printer.

3.4 RUNNING ERF (THE SHUTDOWN TASK)

To terminate error logging, run the task ERF by typing the following pair of commands from a privileged terminal:

```
>INS $ERF

>ERF
```

Note that the command RUN $ERF is not acceptable.

3.5 ERROR LOGGING TASK MESSAGES

3.5.1 ERRLOG Messages

3.5.1.1 **Informational Messages** - The following two messages are informational messages issued during ERRLOG's normal operation.

ERL -- ERROR LOG INITIALIZED

The file ERR.TMP has been opened and initialized successfully. The message occurs either when error logging is first initialized or after ERRLOG has renamed the current ERR.TMP file to ERROR.TMP and opened a new ERR.TMP.

ERL -- LOGGING ENDED AFTER ddd ERRORS

ERRLOG issues this message when error logging is shut down, where "ddd" is the number of errors that were logged to the current file. This count will have been zeroed if PSE had received a file from ERRLOG while logging was active; therefore it does not necessarily indicate the total number of errors logged since ERRLOG was initialized. Note that if ERRLOG is shut down and an error occurs, the error count is updated but the error is not logged.

3.5.1.2 **Error Messages** - The error logging subsystem terminates itself in response to an error condition unless the description of the associated error message states otherwise. In messages which contain one or more error codes, the codes are File Control Service (FCS) or driver/File Control Primitive (FCP) error return codes obtained from the File Descriptor Block (FDB) at the time of the error. If the code is displayed as negative, it was generated by the driver or FCP. If displayed as positive, the code represents an FCS error. All error codes are decimal numbers. See the IAS/RSX-11 I/O Operations Reference Manual, Appendix I, for a description of the error codes.

ERL -- ASSIGN LUN FAILURE

An error occurred when ERRLOG tried to assign LUN 4 while creating file ERR.TMP. Logging is not initialized.

ERL -- DEVICE FULL. MOUNT NEW VOLUME OR
"REA ERRLOG 4 DDU:",THEN "RUN ERRLOG"

While trying to write an error to the file, ERRLOG found the device full. The buffer containing the most recent error logged is requeued in memory, and error logging is shut down. In order to reinitialize logging, either a different device with available space must be used, or the full volume must be DMOunted, a new one MOUnted and the task ERRLOG restarted; the system will then reinitialize error logging. (There may be errors queued in the dynamic pool ready to be logged.)

ERL -- DEVICE RECORD PUT ERROR, CODE ddd

During ERR.TMP initialization, a file system error was detected in writing a device configuration record.

ERL -- ERROR ON REOPEN, CODE ddd

ERRLOG was unable to open an existing, initialized ERR.TMP file in order to write an error message to it.

ERL -- ERR.TMP NOT RENAMED (ERR.TMP: ddd, ERROR.TMP: ddd)

ERRLOG failed to rename the ERR.TMP file either when passing the file contents to PSE or when error logging was being shutdown. The codes are from the respective files' FDBs; the code may be 0 for ERROR.TMP, indicating the problem was only with ERR.TMP. In this case, error logging is not shutdown since a new file will be created. Once the reason for the rename failure is determined, the file may be renamed to ERROR.TMP using PIP, and subsequently formatted (see the RSX-11M Utilities Procedures Manual).

ERL -- ERR.TMP OPEN FAILURE, CODE ddd

The file system reported a failure during initialization when ERRLOG tried to create the file ERR.TMP.

ERL -- FILE CLOSE ERROR, CODE ddd

After ERRLOG had written initialization information or error records, FCS returned an error when it tried to close ERR.TMP. The file will be locked and truncated; error records may therefore be missing from the end of the file. Use PIP to unlock the file for formatting or for further logging if the error is not persistent (see the RSX-11M Utilities Procedures Manual).

ERL -- GET DIRECTORY ID FAILED, CODE ddd

Failure during creation of a new ERR.TMP file. If the code is -26, the directory [1,6] does not exist on the system disk. Refer to Section 3.1.

ERL -- INITIALIZATION PUT ERROR, CODE ddd

During file initialization, an error occurred on the first attempt to write to the ERR.TMP file.

ERL -- MARKED FOR REMOVE, LOGGING NOT STARTED

ERRLOG must be permanently installed; therefore it may not be executed by the default "install/request/remove" method of MCR. This message will appear if the command RUN $ERL has been issued to start up error logging.

ERL -- NO ERROR FILE

This error is unlikely to occur. It indicates that some task has zeroed an internal file ID pointer. ERRLOG exits when it encounters this situation.

Rerun ERRLOG to activate error logging.

ERL -- OUTPUT ERROR, CODE ddd

This error was caused by an attempt to write data to the file ERR.TMP after the occurrence of a loggable error.

ERL -- TASK NAME NOT "ERRLOG"

This message appears when the user tries to initialize ERRLOG after the task has been installed incorrectly. The error logging task must be installed under the name "ERRLOG".

ERL -- TERMINAL NOT PRIVILEGED

The terminal from which ERRLOG is initialized must be privileged.

3.5.2 PSE Messages

PSE -- COMMAND STRING PARSE ERROR

PSE encountered a syntax or semantic error when examining an input command line.

PSE prompts for further input. Retype the corrected command line.

PSE -- DELETE ERROR ON INPUT FILE

When PSE finished processing the input file ERROR.TMP, the task was unable to delete it.

PSE -- ERROR LOGGER DID NOT PROVIDE A FILE

PSE was able to communicate with ERRLOG, but ERRLOG did not make a file available within 30 seconds.

Run PSE again.

PSE -- 1ST RECORD WAS NOT AN INITIALIZATION RECORD FILE CANNOT BE PRE-ANALYZED

The first record of a file must be an initialization record. Both the input and the output files are closed. The input file is not deleted.

Rerun PSE. If the error occurs again, delete the oldest version of ERROR.TMP.

PSE -- INPUT FILE ERROR

An error was encountered when PSE tried to open or obtain data from the input file ERROR.TMP. The input and output files are closed. The input file ERROR.TMP is not deleted.

Rerun PSE.

PSE -- OUTPUT DEVICE IS FULL

The output device became full while PSE was writing to the output file. Both the input and output files were closed. ERROR.TMP is not deleted. Select another volume for the output file.

PSE -- OUTPUT FILE ERROR

An error was encountered while PSE was accessing the output file. Both the input and output files are closed. ERROR.TMP is not deleted.

PSE -- RECORD SIZE WAS INCORRECT

When PSE reads a record from the input file, it verifies the size of the record. If the size is wrong, it ignores the record and continues to process the rest of the input file.

PSE -- TERMINAL MUST BE PRIVILEGED

An attempt was made to run PSE from a non-privileged terminal.

Run PSE from a privileged terminal.

PSE -- TOO MANY DEVICE DESCRIPTOR ENTRIES. RECOMMEND THAT PSE RUN IN A LARGER PARTITION
DEVICE = device
IDENTIFIER = id

PSE has run out of storage space in which to hold descriptor entries for the device named in the message. PSE continues to process the input file, but it is unable to process any more errors for the named device.

In the future, run PSE for this configuration in a larger partition, or specify the /INC=xxx switch when PSE is installed in a system controlled partition. PSE requires 10 (decimal) words of space for each device for which errors are to be logged.

PSE -- UNABLE TO CLOSE THE INPUT FILE

PSE is unable to close the input file ERROR.TMP. The file is not deleted.

PSE -- UNABLE TO CLOSE THE OUTPUT FILE

PSE is unable to close the output file. The message generally indicates that the output device is full and that the last block of the file will be missing.

PSE -- UNABLE TO FIND ANY ERROR.TMP FILES

PSE cannot locate any files named ERROR.TMP. Verify that the correct input device was specified or that ERRLOG has been initialized.

PSE -- UNABLE TO RECOGNIZE DEVICE IDENTIFIER
IDENTIFIER = id

PSE has received a device descriptor error entry that does not match any device described in the device tables. PSE ignores that entry record and continues to process the rest of the file.

PSE -- UNABLE TO RECOGNIZE ERROR ENTRY CODE
ENTRY CODE = code

PSE must recognize the entry code contained within each record in order to know how to process the record. If PSE does not recognize the entry code, it ignores the record and continues to process the rest of the input file.

PSE -- UNABLE TO REQUEST A FILE FROM ERRLOG

PSE tried to request ERRLOG to make a file available for formatting. This error indicates that ERRLOG has not been initialized.

3.5.3 SYE Messages

3.5.3.1 **Command Line Errors** - The messages described in this section occur when SYE encounters an error in the command line.

SYE -- COMMAND STRING ERROR
ERROR NUMBER x

SYE has encountered an error (other than end of file) when trying to obtain a command line. Refer to the IAS/RSX-11 I/O Operations Reference Manual, Appendix I, for an explanation of the error code "x", a decimal number.

SYE -- COMMAND STRING ERROR
text

SYE encountered an error when it tried to parse the command line. "text" is the command line that contains the error.

SYE -- COMMAND STRING SYNTAX ERROR
text

SYE encountered an error in the syntax of the command line. "text" is the erring portion of the command line.

SYE -- ILLEGAL REPORT SWITCH
/REPORT:text

Illegal values ("text") have been supplied with the report (/RP) switch. See Section 3.3.

SYE -- ILLEGAL SWITCH COMBINATION
NO OUTPUT CAN BE GENERATED

The SYE command line specified an illegal switch combination.

SYE -- NO ARGUMENT FOR DEVICE SWITCH

The SYE command line did not specify a group of devices or a specific device in the device switch (that is, /DV).

3.5.3.2 **File Service Errors** - The following messages are caused by file failures. Each message contains the line "FATAL ERROR - x" where x is a decimal number that represents an error code returned by File Control Services (FCS). See the IAS/RSX-11 I/O Operations Reference Manual, Appendix I.

SYE -- DEVICE ERROR INPUT FILE
FATAL ERROR - x.
filename

An error (other than end of file) occurred when SYE attempted to read data from the input file. SYE closes all files and prompts for the next command line.

SYE -- DEVICE ERROR INPUT FILE
NO SUCH FILE
filename

The input file specified in the SYE command line does not exist. SYE closes all files and prompts for the next command line.

SYE -- DEVICE ERROR OUTPUT FILE
FATAL ERROR - x.
filename

SYE encountered an error when attempting to write data to the output file. SYE closes all files and prompts for the next command line.

SYE -- OPEN FAILURE ON INPUT DEVICE
FATAL ERROR - x.
filename

SYE encountered an error when it tried to open the input file. SYE closes all files and prompts for the next command line.

SYE -- OPEN FAILURE ON OUTPUT DEVICE
FATAL ERROR - x.
filename

SYE encountered an error when it tried to open the output file. SYE closes all files and prompts for the next command line.

SYE -- SUMMARY TEMP FILE I/O ERROR REPORT CONTINUES WITHOUT SUMMARIES

SYE encountered an error while reading or writing to file SYESUM.TMP. This file is a temporary file created when SYE runs out of available work space in memory.

3.5.3.3 **Additional Messages** - The first message below is an error messages; the second message indicates that SYE has completed its formatting of the reports.

SYE -- TEMPORARY FILE ERROR
REPORT CONTINUES WITHOUT SUMMARIES

SYE was unable to produce a complete summary report because it encountered an error in a temporary file containing data for the device summary reports. SYE produces a report without the device summaries.

SYE -- x. PAGES IN REPORT

SYE displays this message when it has completed its formatting of the complete report. "x." is the decimal number of pages in the report.

3.5.4 ERF Messages

The shutdown task ERF always displays a message in response to a request to run; the message is either an error message or a confirmation of shutdown.

ERF -- "ERRLOG" NOT INSTALLED

The task ERRLOG has not been installed.

ERF -- ERROR LOGGING NOT STARTED

Error logging was not initialized, and therefore does not need to be shutdown.

ERF -- REQUESTED "ERRLOG" TO STOP LOGGING

The normal confirmation message. ERF has exited and ERRLOG should display its own termination message shortly.

ERF -- TERMINAL NOT PRIVILEGED

ERF must be requested from a privileged terminal.

CHAPTER 4

THE ERROR LOG REPORTS

4.1 GENERATING ERROR LOG REPORTS

To generate error log reports, the user must first run the task PSE from a privileged terminal (refer to Section 3.2). PSE performs a preliminary formatting of the raw data contained in files created by the error logging task ERRLOG. The PSE output file (ERROR.SYS), which contains the pre-formatted error data, becomes the input file to SYE.

SYE, the Report Generator, can be run from any terminal. When the PSE output file is available, the user must decide what kind of reports are needed and then select the appropriate switches for the SYE command line. For example, the user can select a time frame that the report is to encompass and can specify that the report include only those errors associated with a certain device type or unit. Refer to Section 3.3 for a description of SYE operating procedures and all the available command line switch options.

SYE produces an error report in the form of a printed listing or a listing file. By default, SYE creates a file called ERRLOG.LST on the system disk; this file contains summary information only on all of the errors logged in the ERROR.SYS file.

4.2 INDIVIDUAL ERROR REPORTS

Error logging detects and records four kinds of errors: unexpected traps or interrupts, device errors, interrupt timeouts, and memory parity errors (refer to Section 1.2.1). This section illustrates sample reports on each kind of error.

4.2.1 Reports For Unexpected Traps or Interrupts

Figure 4-1 is a report of an interrupt through an unused vector. Such interrupts are also called traps. The following description of the report refers to each line of text by number, where the first line described is line 1.

```
 1  ************************* ENTRY 12. ******************************
 2  UNEXPECTED TRAP OR INTERRUPT
 3  LOGGED  28-JAN-77 16:24:00
    ********************************************************************

 4  SOFTWARE STATUS
 5          VECTOR WHERE TRAP OR INTERRUPT OCCURRED  200
 6          PC      PRIOR  THIS ERROR               177000
 7          PSW     PRIOR  THIS ERROR               000004
 8          ERRORS  MISSED WHILE THIS ERROR LOGGED  0

 9  VECTORS WITH ACTIVE IO
10          204
```

Figure 4-1 Report of an Unexpected Trap or Interrupt

- Line 1 gives the record's entry number. The first error to occur after error logging has been initialized is assigned an entry number of 1. The entry number is then incremented by 1 every time a loggable error occurs.
- Line 2 describes the type of error.
- Line 3 gives the date and time the error occurred.

The next block of text (lines 4 through 8) contains information about the software status of the trap or interrupt:

- Line 5 provides the address of the vector through which the interrupt occurred.
- Line 6 is the value of the program counter immediately prior to the error.
- Line 7 is the processor status word immediately prior to the unexpected trap or interrupt.
- Line 8 notes the number of errors that occurred while the system was logging this error. (This count permits the detection of bursts of illegal interrupts possibly caused by UNIBUS noise corruption.)

The last block of data, headed "VECTORS WITH ACTIVE IO", lists all the vectors with I/O activity on the UNIBUS at the time of the error. The list includes the interrupt vector address of the device that caused the error. Lines 9 and 10 do not appear in the report if there is no I/O activity.

4.2.2 Device Error Reports

Figure 4-2 illustrates a typical report for a device error. (The figure also serves as a sample interrupt timeout report, since the report format is identical for both types of error.) The following description of the disk error report refers to each line of text by number, where the first line described is line 1.

The first two lines of text, offset by two star lines, give header information about the error:

- Line 1 gives the record's entry number.
- Line 2 describes the type of error (DEVICE HARDWARE ERROR).
- Line 3 gives the date and time the error occurred.

```
 1  ************************** ENTRY 36. ****************************
 2  DEVICE HARDWARE ERROR
 3  LOGGED   17-FEB-77 13:39:59
    ******************************************************************

 4  UNIT IDENTIFICATION
 5          UNIT LOGICAL NAME       DK0
 6          UNIT PHYSICAL NAME      DK0  (CONTROLLER-0    UNIT-0)
 7          DEVICE TYPE             RK05/RK05F/RK03 (DISK)

 8  SOFTWARE STATUS
 9          TASK NAME               ...VFY
10          TASK UIC                1,1
11          TASK START ADDRESS      373400
12          TASK BUFFER ADDRESS     431210
13          TRANSFER SIZE (BYTES)   1000    512.
14          IO FUNCTION ISSUED      READ

15  DEVICE REGISTERS
16          RKDS    004723
17          RKER    000002  CHECKSUM ERROR
18          RKCS    100744  READ
19          RKWC    000000
20          RKBA    032210
21          RKDA    000023

22  MEDIA ADDRESS           AT IO START     AT ERROR
23          CYLINDER                0       0
24          TRACK                   1       1
25          SECTOR                  2       2
26          LOGICAL BLOCK           16      16

27  ERROR DIAGNOSIS
28          NOT RECOVERED AFTER 8. RETRIES

29  VECTORS WITH ACTIVE IO
30          220
```

Figure 4-2 Report of a Device Hardware Error

The next block of data (UNIT IDENTIFICATION) describes the erring device:

- Line 5 (UNIT LOGICAL NAME) shows the device mnemonic (DK) and unit logical number (0). The unit logical number ranges from 0 to n, where n is an octal number representing the total number, minus one, of such devices supported by all controllers at a particular installation.
- Line 6 (UNIT PHYSICAL NAME) shows the device mnemonic, the physical unit number, and the controller number. The physical unit number can range from 0 to 7 for each controller.

```
 1  ********************************* ENTRY 9. ***********************
 2  MEMORY PARITY ERROR
 3  LOGGED   15-JUL-76 09:37:43
    ********************************************************************

 4  SOFTWARE STATUS
 5          TASK NAME                TT0
 6          TASK START ADDRESS       21000
 7          TASK SIZE (WORDS)        6300
 8          PC  PRIOR TO THIS ERROR  000452
 9          PSW PRIOR TO THIS ERROR  170000
10          PARTITION NAME           GEN
11          PARTITION START ADDRESS  210000

12  MEMORY REGISTERS  (11/70 TYPE)
13          LOW ERROR ADDRESS        000040
14          HIGH ERROR ADDRESS       000000
15          MEMORY SYSTEM ERROR      025167  MAIN MEMORY TIMEOUT
                                             MAIN MEMORY ADDRESS PARITY
                                             ERROR
                                             MAIN MEMORY EVEN WORD
                                             ADDRESS MEMORY GROUP 0
                                             ADDRESS MEMORY GROUP 1
                                             DATA MEMORY GROUP 0
                                             CPU UNIBUS ABORT
                                             CPU ERROR
                                             UNIBUS PARITY ERROR
24          MEMORY CONTROL           003310  FORCE MISS GROUP 1
25          MEMORY MAINTENANCE       000401
26          MEMORY HIT/MISS          001300

27  ERROR DIAGNOSIS
28  PARITY ERROR IN MEMORY LOCATION 40

29  WITH MEMORY MANAGEMENT REGISTER 5 ACTIVE IN USER MODE
```

Figure 4-3 Report of a Memory Parity Error

Lines 27 through 29 provide an error diagnosis:

- Line 28 gives the physical location in memory in which the parity error occurred.
  PDP-11/40 and /45 Processors Only - When the parity error occurred on a PDP-11/40 or /45 processor, this line gives the range of addresses in which the error occurred, rather than a precise location. For example, the line might read:

  PARITY ERROR IN LOCATIONS 210000 TO 213777

Line 29 appears for mapped systems only:

- This line gives the number of the memory management register group active at the time of the error and the operating mode of the task that was active when the error occurred.

4.3 SUMMARY REPORTS

Figure 4-4 is an example of the summary section of a report. This part of the report contains summary information about all the errors described in the report as a whole (but not necessarily about all the errors in the input file to SYE).

```
**************************************************************************
                    SYSTEM ERROR REPORT SUMMARY TOTALS
**************************************************************************

 2  DEVICE ERRORS                      80.
 3  MEMORY PARITY ERRORS               0.
 4  CACHE PARITY ERRORS                0.
 5  W.C.S. PARITY ERRORS               0.
 6  INTERRUPT TIMEOUT ERRORS           0.
 7  UNEXPECTED TRAPS OR INTERRUPTS     0.
 8  POWER FAILS                        0.
 9  ENTRIES MISSING (SATURATION)       0.


10          COMMAND LINE USED          ERK05.LST/-SP=[304,324]DKDBDS.SYS
11          INPUT FILE                 SY0:[304,324]DKDBDS.SYS;1
12          OUTPUT FILE                SY0:[304,324]ERK05.LST;2
13          DATE OF FIRST ENTRY        17-FEB-77 05:58:16
14          DATE OF LAST ENTRY         17-FEB-77 15:39:08
15          ENTRIES PROCESSED          94.
16          UNRECOGNIZABLE ENTRIES     0.




**************************************************************************
17              DETAILED DEVICE ERROR REPORT SUMMARIES
**************************************************************************




18  UNIT LOGICAL NAME       DS2
19  UNIT PHYSICAL NAME      DS2       CONTROLLER-0   UNIT-2
20  DEVICE TYPE             RS04
21                                                   [HARD]   [SOFT]
22                          DATA LATE                0.       35.


23  UNIT LOGICAL NAME       DK0
24  UNIT PHYSICAL NAME      DK0       CONTROLLER_0   UNIT_0
25  DEVICE TYPE             RK05/RK05F/RK03
26                                                   [HARD]   [SOFT]
27                          MOTION/SEEK              0.       1.
28                          CHECKSUM OR CRC          9.       0.
29                          DATA LATE                9.       0.
```

Figure 4-4 Summary Error Report

Lines 2 through 9 of the summary report provide various error statistics:

- Line 2 is the number of device errors described in this report.
- Line 3 records the total number of memory parity errors described in the report.
- Line 4 shows the total number of cache parity errors.
- Line 5 reports the number of Writeable Control Store parity errors.
- Line 6 is the number of interrupt timeouts described in this report.
- Line 7 is the number of unexpected traps or interrupts.
- Line 8 gives the number of power fails.
- Line 9 is the number of errors that were not logged because another error was already being recorded. Lines 10 through 16 of the report summarize details of the execution of SYE.
- Line 10 is the command line issued to SYE.
- Line 11 describes the complete file specifier of the input file and line 12 describes the complete file specifier of the output file.
- Lines 13 and 14 give the dates and times of the first and last entries in the input file.
- The number of entries formatted in the report appears in line 15.
- Line 16 is the number of unrecognizable errors encountered by SYE. An unrecognizable error is any entry that the current version of SYE cannot format. This situation can occur for one of the following reasons:

  1. An old version of SYE has been run.
  2. The entry refers to a device not supported by SYE. Such an entry may be encountered if the site has implemented error logging on a device SYE does not recognize.
  3. SYE has encountered a data structure (field format) error. Such encounters indicate that the wrong version of the pre-formatter PSE was used.

Line 17 marks the beginning of a series of detailed device error report summaries. Each such summary provides error statistics relating to an individual device. Figure 4-4 shows two such summaries, the first on lines 18 through 22, and the second on lines 23 through 29.

- Lines 18 and 23 show the unit logical name.
- Lines 19 and 24 give the physical unit numbers of the respective devices and their controller numbers.
- Lines 20 and 25 show the DIGITAL name for the devices.

The remaining sections in each detailed device error summary (lines 21 and 22 in the first instance and lines 26 through 29 in the second) show the number of hard (non-recovered) and soft (recovered) errors described in this report. The types of errors which can appear in these sections, examples of which appear on lines 22 and 27 through 29, are described below.

Heading	Description
CHECKSUM OR CRC	The controller detected either a data checksum error or a hard error-check correction error.
CONTROLLER PARITY	The hardware controller detected a parity error within itself.
DATA LATE	The controller detected a data error caused by one of two possible situations: • Data was ready for memory when memory had not emptied its hardware buffer (silo), or • The controller was ready for data, but memory's silo had no data to be written.
DATA PARITY	The controller detected a data parity error.
DEV-TO-MEM PARITY	The controller detected a parity error while the controller was performing a Write or Write-Check command.
DRIVE	A drive specific error, for example a drive timing error, occurred.
DRIVE SELECT	A drive selection eror, such as the selection of multiple heads, has occurred.
HEADER	The controller detected a unit error that prevents a successful data transfer. Specific errors include search errors, header compare errors, and format errors.
INTERRUPT TIMEOUT	An initiated operation did not cause an interrupt within a specified time interval.
MOTION/SEEK	The controller detected mechanical problems on a device unit, which cause the unit to produce errors such as seek incomplete, seek error, or operation incomplete.
NONEXISTENT MEMORY	The controller detected a bus timeout error.
POWER ERROR	The controller detected an error related to power difficulties.
SOFTWARE ERROR	Corrupted service routines caused a controller error.
UNIT UNSAFE	The controller detected an error that caused a selected device to be unusable.
WRITE CHECK	A write check error occurred.
WRITE LOCKED	A write operation was attempted on a write-locked unit.

APPENDIX A

MODIFYING SYE AT TASK BUILD

The command file SYEBLD.CMD contains instructions to the task builder to create the ...SYE task. There are several option statements in the command file that may be changed before task build to modify the standard analyzer task SYE:

1. The page width and buffer size for output from the terminal message processor to the terminals. These values are defined by the following TKB option statements:

   GBLDEF=PT$WTH:nnn (page width)

   and

   EXTSCT=TBUF:nnn (buffer size)

   where nnn is octal and must be the same in both directives. The default value is 110 (octal).

   The value assigned to nnn does not affect task size.

2. The page width and buffer size for the output from the error processors. These values are defined by the following TKB option statements:

   GBLDEF=PL$WTH:nnn (page width)

   and

   EXTSCT=OBUF:nnn (buffer size)

   where nnn is octal and must be the same in both directives. The default value is 120 (octal).

   Note that the value of nnn affects the size of the task.

Figure A-1 lists the contents of the command file SYEBLD.CMD.

```
;
;
;
;        S Y E B L D . C M D
;
;USE: RSX11M SYE ERROR LOG DISPLAY PROGRAM
;
;
;THIS COMMAND FILE WILL BUILD THE ...SYE TASK FROM THE
;SYEBLD.ODL FILE FOR MAPPED SYSTEM
;
[1,54]SYE/MM/CP/-FP,MP:[1,34]SYE/-SP=[1,20]SYEBLD/MP
;
;
STACK=96
ASG=TI:1:5
TASK=...SYE
PAR=GEN:0:40000
;
;
;
;THIS DEFINES THE PAGE WIDTH FOR OUTPUT TO THE TERMINALS
;FROM THE TERMINAL MESSAGE PROCESSOR - THIS HAS NO EFFECT ON THE
;OUTPUT OF THE ERROR DISPLAYS
;ALL VALUES ARE IN OCTAL
;
GBLDEF=PT$WTH:110
;
;THIS IS THE SIZE OF THE TERMINAL BUFFER
;THIS VALUE MUST BE EQUAL TO THE VALUE USED FOR PT$WTH
;
EXTSCT=TBUF:110
;
;THIS DEFINES THE PAGE WIDTH FOR THE OUTPUT FROM THE ERROR
;PROCESSORS
;
;ALL VALUES ARE OCTAL
GBLDEF=PL$WTH:120
;
;THIS IS THE EXPANSION OF THE OUTPUT BUFFER
;THIS VALUE MUST BE THE SAME AS PL$WTH
;
EXTSCT=OBUF:120
;
;THIS IS THE DEFINITION OF THE PAGE LENGTH FOR THE OUTPUT FROM
;THE ERROR PROCESSORS
;
;ALL VALUES TYPED ARE OCTAL
;
GBLDEF=PL$LGH:74
;
/
```

Figure A-1 SYEBLD.CMD

APPENDIX B

ERROR LOGGING DATA FORMATS

This Appendix consists of tables that illustrate memory-resident data formats for the following types of errors:

- Device driver load and unload errors (Table B-1),
- Unexpected trap or interrupt errors (Table B-2),
- Device hardware and interrupt timeout errors (Table B-3), and
- Memory parity errors (Table B-17).

Tables B-4 through B-16 explain the device-dependent parameters shown in Table B-3.

Users should refer to these tables when implementing new devices or features into the error logging system, or when analyzing a memory-resident error (an error that appears in a memory-dump printout, for example).

Offsets - The values shown for the offset definitions do not include the link word and, hence, must be increased by 2 to obtain the appropriate address values.

Reserved Words - The term "reserved" means that the related word contains a value for a specific purpose; its contents must not be altered.

Blanks - The term "blank" means that the related word was allocated for future use; its contents have no meaning.

Time Indications - Time indications are recorded as bytes or words. In general, byte entries are made by Executive routines; word entries are made by the get time system directive.

Table B-1
Data Format
Device Driver Load or Unload

Word		Offset Value	Definition
0	Link word		
1	Size of core block in bytes	+0	E.SIZE
2	(Reserved) error code (EC.INI)	+2	E.CODE
3	ERRLOG fills in time here	+4	E.TIME
4	ERRLOG fills in time here	+6	
5	ERRLOG fills in time here	+10	
6	ERRLOG fills in time here	+12	
7	ERRLOG fills in time here	+14	
8	ERRLOG fills in time here	+16	
9	RSX-11M version number	+20	
10	Error Logger identification	+22	
11	Error Logger identification	+24	
12	Load(4)/Unload (10) code	+26	E.WHY
13	Configuration code	+30	
14	Driver name	+32	E.NAME
15	(Blank)	+34	
16	(Blank)	+36	
17	(Blank)	+40	

Table B-2
Data Format
Unexpected Trap or Interrupt Error

Word		Offset Value	Definition
0	Link word		
1	Size of core block in bytes	+0	E.SIZE
2	(Reserved) Error code	+2	E.CODE
3	Minute Second	+4	E.TIME
4	Day Hour	+6	
5	Year Month	+10	
6	Error sequence number	+12	E.SEQ
7	Active bit map	+14	E.ABM
8	Missed count Vector/4 of trap	+16	E.VCTR(16),E.LOST(17)
9	Processor Status before trap	+20	
10	Program Counter before trap	+22	E.OPC

Table B-3
Data Format
Device Hardware and Interrupt Timeout Errors

Word			Offset Value	Definition
0	Link word			
1	Size of core block in bytes		+0	E.SIZE
2	(Reserved)	Entry code	+2	E.CODE
3	Minute	Second	+4	E.TIME
4	Day	Hour	+6	
5	Year	Month	+10	
6	Error sequence number		+12	E.SEQ
7	Active bit map		+14	E.ABM
8	Maximum retries	Retries left	+16	E.RTRY
9	I/O in queue	Task priority	+20	E.IOC(21)
10	Task name		+22	E.TASK
11	Task name		+24	
12	Base address of task		+26	E.PAR
13	Programmer group	Programmer code	+30	E.UIC
14	UCB address		+32	E.UCB
15	Function code		+34	E.FNC
16	I/O packet parameters		+36	
17	(device-dependent)*		+40	
18	(device-dependent)		+42	
19	(device-dependent)		+44	
20	(device-dependent)		+46	
21	(device-dependent)		+50	
22	(device-dependent)		+52	
23	Count of registers		+54	E.RCNT
24	Device registers		+56	E.REGS
.	"	"	.	
.	"	"	.	
.	"	"	.	
n	Device registers		m	

* See I/O Packet Parameter Tables B-4 through B-16.

Table B-4
I/O Packet Parameters
RK03, RK05, RK05F

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	"RKDA" for I/O GO	+46
21	LBN or Transfer start	+50
22	(Reserved)	+52

Table B-5
I/O Packet Parameters
RP02, RP03

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	"RPCA" for I/O GO	+46
21	"RPDA" for I/O GO	+50
22	(Reserved)	+52

Table B-6
I/O Packet Parameters
RF11

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	"RFDAE" for I/O GO	+46
21	"RFDAR" for I/O GO	+50
22	(Reserved)	+52

Table B-7
I/O Packet Parameters
RS03, RS04

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	(Reserved)	+46
21	"RSDA" for I/O GO	+50
22	(Reserved)	+52

Table B-8
I/O Packet Parameters
RP04, RP05, RP06

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	"RPDC" for I/O GO	+46
21	"RPDA" for I/O GO	+50
22	(Reserved)	+52

Table B-9
I/O Packet Parameters
RK06

Word		Offset Value
16	Transfer memory address bits (High)	+36
17	Transfer memory address bits (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	"RKDA" for I/O GO	+46
21	"RKDC" for I/O GO	+50
22	(Reserved)	+52

Table B-10
I/O Packet Parameters
TA11

Word		Offset Value
16	Space count if spacing function or "KTAPR" contents to transfer data through if data function	+36
17	Virtual buffer address of transfer	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	(Reserved)	+46
21	(Reserved)	+50
22	(Reserved)	+52

Table B-11
I/O Packet Parameters
TS03, TE10, TU10

Word		Offset Value
16	Space count if space function or transfer memory address (High) if data function	+36
17	Transfer memory address (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	(Reserved)	+46
21	(Reserved)	+50
22	(Reserved)	+52

Table B-12
I/O Packet Parameters
TE16, TU16, TU45

Word		Offset Value
16	Space count if space function or transfer memory address (High) if data function	+36
17	Transfer memory address (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	(Reserved)	+46
21	(Reserved)	+50
22	(Reserved)	+52

Table B-13
I/O Packet Parameters
TC11

Word		Offset Value
16	Transfer memory address (High)	+36
17	Transfer memory address (Low)	+40
18	Transfer size in bytes	+42
19	(Reserved)	+44
20	(Reserved)	+46
21	LBN for transfer start	+50
22	(Reserved)	+52

Table B-14
I/O Packet Parameters
RL01

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Total transfer size in bytes	+42
10	Seek difference count	+44
11	"RLDA" for I/O GO	+46
12	This transfer size in bytes	+50
13	(Reserved)	+52

Table B-15
I/O Packet Parameters
RM03

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	"RMDC" for I/O GO	+46
12	"RMDA" for I/O GO	+50
13	(Reserved)	+52

Table B-16
I/O Packet Parameters
RX01

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	Current transfer size in bytes	+46
12	"RXDA" for I/O GO	+50
13	(Reserved)	+52

Table B-17
Data Format
Memory Parity Errors

Word			Offset Value
0	Link word		+0
1	Size of block in bytes		+2
2	Processor type*	Entry code=002	+4
3	Minute	Second	+6
4	Day	Hour	+10
5	Year	Month	+12
6	Error sequence number		+14
7	Trap PC		+16
8	Trap PS		+20
9	First word of task name		+22
10	Second word of task name		+24
11	First word of partition name		+26
12	Second word of partition name		+30
13	Partition base address		+32
14	Partition size		+34
15	Memory parity CSR's		+36
16	"		+40
17	"		+42
18	"		+44
19	"		+46
20	"		+50
21	"		+52
22	"		+54
23	"		+56
24	"		+60
25	"		+62
26	"		+64
27	"		+66
28	"		+70
29	"		+72
30	"		+74
31	Cache parity CSR's		+76
32	"		+100
33	"		+102
34	"		+104
35	"		+106
36	"		+110

* The processor type field contains 1 for an 11/40 or 11/45, 2 for an 11/70, and 3 for an 11/60.

APPENDIX C

ERR.TMP AND ERROR.TMP FILE RECORD FORMATS

This Appendix illustrates record formats that appear in a temporary file (either ERR.TMP or ERROR.TMP) created by the error logging task ERRLOG. Printouts generated by the DMP program of either temporary file contain the record formats described in the following tables. Specifically, the tables describe:

- Initial configuration records (Table C-1),
- Device initialization records (Table C-2),
- Device hardware and interrupt timeout errors (Table C-3),
- Unexpected trap or interrupt errors (Table C-17),
- Device driver load and unload errors (Table C-18),
- Memory parity errors (Table C-19), and
- Power fail records (Table C-20).

Tables C-4 through C-16 explain the device-dependent parameters shown in Table C-3.

Users should refer to these tables when implementing new devices or features into the error logging system.

In printouts generated by the DMP program, the record size in bytes precedes each variable length record description.

Reserved Words - The term "reserved" means that the corresponding word contains a value for a specific purpose; its contents must not be altered.

Blank Words - The term "blank" means that the corresponding word was allocated for future use; its contents have no meaning.

Time Indications - Time indications are recorded as bytes or words. In general, byte entries are made by Executive routines; word entries are made by the get time system directive.

Table C-1
ERR.TMP and ERROR.TMP File Record Format
Initial Configuration Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	(Reserved) Error code (EC.INI)	+2	E.CODE
2	Year	+4	E.TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	RSX-11M version number	+20	
9	Error logger Identification	+22	
10	Error logger Identification	+24	
11	Initial start (0) 'PSE' request (2)	+26	E.WHY
12	Configuration code	+30	
13	(Blank)	+32	E.NAME
14	(Blank)	+34	
15	(Blank)	+36	
16	(Blank)	+40	

Configuration Codes:

0 - Unmapped

2 - 18 bit addressing

4 - 22 bit addressing

Table C-2
ERR.TMP and ERROR.TMP File Record Format
Device Initialization Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	Init. SUB-CODE* Error code (EC.INI)	+2	E.CODE
2	UCB address	+4	
3	Device name in ASCII	+6	
4	Physical unit number System unit number	+10	
5	Vector/4 System controller number	+12	
6	Device CSR Address	+14	
7	IOABM mask	+16	
8	Device control words (CW1)	+20	
9	Device control words (CW2)	+22	
10	Device control words (CW3)	+24	
11	Device control words (CW4)	+26	
12	(Blank)	+30	
13	(Blank)	+32	
14	(Blank)	+34	

* 1 = Device is loaded for error logging (E.DDW).
2 = Device is unloaded for error logging (E.RDDW).

Table C-3
ERR.TMP and ERROR.TMP File Record Format
Device Hardware and Interrupt Timeout Errors

Word			Offset Value	Definition
0	Size of record in bytes		+0	E.SIZE
1	(Reserved)	Entry Code	+2	E.CODE
2	Minute	Second	+4	E.TIME
3	Day	Hour	+6	
4	Year	Month	+10	
5	Error sequence number		+12	E.SEQ
6	Active bit map		+14	E.ABM
7	Maximum retries	Retries left	+16	E.RTRY
8	I/O in Queue	Task priority	+20	E.IOC(21)
9	Task name		+22	E.TASK
10	Task name		+24	
11	Base address of task		+26	E.PAR
12	Programmer group	Programmer code	+30	E.UIC
13	UCB address		+32	E.UCB
14	Function code		+34	E.FNC
15	I/O Packet Parameters		+36	
16	(device-dependent)*		+40	
17	(device-dependent)		+42	
18	(device-dependent)		+44	
19	(device-dependent)		+46	
20	(device-dependent)		+50	
21	(device-dependent)		+52	
22	Count of registers		+54	E.RCNT
23	Device registers		+56	E.REGS
.	"	"	.	
.	"	"	.	
.	"	"	.	
n	Device registers		m	

* See I/O Packet Parameters in Tables C-4 through C-16.

Table C-4
I/O Packet Parameters
RK03, RK05, RK05F

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RKDA" for I/O GO	+46
20	LBN or transfer start	+50
21	(Reserved)	+52

Table C-5
I/O Packet Parameters
RP02, RP03

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RPCA" for I/O GO	+46
20	"RPDA" for I/O GO	+50
21	(Reserved)	+52

Table C-6
I/O Packet Parameters
RF11

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RFDAE" for I/O GO	+46
20	"RFDAR" for I/O GO	+50
21	(Reserved)	+52

Table C-7
I/O Packet Parameters
RS03, RS04

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	"RSDA" for I/O GO	+50
21	(Reserved)	+52

Table C-8
I/O Packet Parameters
RP04, RP05, RP06

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RPDC" for I/O GO	+46
20	"RPDA" for I/O GO	+50
21	(Reserved)	+52

Table C-9
I/O Packet Parameters
RK06

Word		Offset Value
15	Transfer memory address bits (High)	+36
16	Transfer memory address bits (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	"RKDA" for I/O GO	+46
20	"RKDC" for I/O GO	+50
21	(Reserved)	+52

Table C-10
I/O Packet Parameters
TA11

Word		Offset Value
15	Space count if spacing function or "KTAPR" contents to transfer data through if data function	+36
16	Virtual buffer address of transfer	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table C-11
I/O Packet Parameters
TS03, TE10, TU10

Word		Offset Value
15	Space count if space function or transfer memory address (High) if data function	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table C-12
I/O Packet Parameters
TE16, TU16, TU45

Word		Offset Value
15	Space count if space function or transfer memory Address (High) if data function	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	(Reserved)	+50
21	(Reserved)	+52

Table C-13
I/O Packet Parameters
TC11

Word		Offset Value
15	Transfer memory address (High)	+36
16	Transfer memory address (Low)	+40
17	Transfer size in bytes	+42
18	(Reserved)	+44
19	(Reserved)	+46
20	LBN for transfer start	+50
21	(Reserved)	+52

Table C-14
I/O Packet Parameters
RL01

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Total transfer size in bytes	+42
10	Seek difference count	+44
11	"RLDA" for I/O GO	+46
12	This transfer size in bytes	+50
13	(Reserved)	+52

Table C-15
I/O Packet Parameters
RM03

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	"RMDC" for I/O GO	+46
12	"RMDA" for I/O GO	+50
13	(Reserved)	+52

Table C-16
I/O Packet Parameters
RX01

Word		Offset Value
7	Transfer memory address bits (High)	+36
8	Transfer memory address bits (Low)	+40
9	Transfer size in bytes	+42
10	(Reserved)	+44
11	Current transfer size in bytes	+46
12	"RXDA" for I/O GO	+50
13	(Reserved)	+52

Table C-17
ERR.TMP and ERROR.TMP File Record Format
Unexpected Trap or Interrupt Error

Word			Offset Value	Definition
0	Size of record in bytes		+0	E.SIZE
1	(Reserved)	Error Code (EC.NSI)	+2	E.CODE
2	Minute	Second	+4	E.TIME
3	Day	Hour	+6	
4	Year	Month	+10	
5	Error Sequence Number		+12	E.SEQ
6	Active bit map		+14	E.ABM
7	Missed count	Vector/4 of Trap	+16	E.VCTR(16) E.LOST(17)
8	Processor Status before trap		+20	
9	Program Counter before trap		+22	E.OPC

Table C-18
ERR.TMP and ERROR.TMP File Record Format
Device Driver Load or Unload

Word			Offset Value	Definition
0	Size of record in bytes		+0	E.SIZE
1	(200)	Error code (EC.INI)	+2	E.CODE
2	Year		+4	E.TIME
3	Month		+6	
4	Day		+10	
5	Hour		+12	
6	Minute		+14	
7	Second		+16	
8	RSX-11M version number		+20	
9	Error logger Identification		+22	
10	Error logger Identification		+24	
11	Load(4)/Unload (10) code		+26	E.WHY
12	Configuration code		+30	
13	Driver name		+32	E.NAME
14	(Blank)		+34	
15	(Blank)		+36	
16	(Blank)		+40	

Table C-19
ERR.TMP and ERROR.TMP File Record Format
Memory Parity Errors

Word			Offset Value
0	Size		+0
1	Processor Type*	Entry Code	+2
2	Minute	Second	+4
3	Day	Hour	+6
4	Year	Month	+10
5	Error Sequence Number		+12
6	Trap PC		+14
7	Trap PS		+16
8	First word task name		+20
9	Second word task name		+22
10	First word partition name		+24
11	Second word partition name		+26
12	Partition base address		+30
13	Partition size		+32
14	Memory parity CSR's		+34
15	"		+36
16	"		+40
17	"		+42
18	"		+44
19	"		+46
20	"		+50
21	"		+52
22	"		+54
23	"		+56
24	"		+60
25	"		+62
26	"		+64
27	"		+66
28	"		+70
29	"		+72
30	Cache parity CSR's		+74
31	"		+100
32	"		+102
33	"		+104
34	"		+106
35	"		+110

* Refer to Table B-17 for processor type values.

Table C-20
ERR.TMP and ERROR.TMP File Record Format
Power Fail Record

Word		Offset Value	Definition
0	Size of record in bytes	+0	E.SIZE
1	Reserved Error Code (EC.PWR)	+2	E.CODE
2	Year	+4	E.TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	

APPENDIX D

ERROR.SYS FILE RECORD FORMATS

This Appendix defines record formats contained in the file ERROR.SYS. If the report generating task ERRLOG is not available, the user can consult these record formats to analyze the contents of ERROR.SYS. These formats should also be studied when adding features to the existing error logging system.

A user can run the system utility DMP to obtain a listing of ERROR.SYS. (See the RSX-11 Utilities Procedures Manual.) In such a printout, the record size in bytes precedes each variable length record.

Each record consists of one or more fields. The first field in every record is a header field that contains basic error information and optionally points to additional fields. The optional fields can follow in any order. Tables D-1 through D-6 define the six header field formats available for different types of errors; these errors are:

- Hardware or interrupt timeout errors (Table D-1),
- Memory parity errors (Table D-2),
- Error logging startup errors (Table D-3),
- Error logging termination errors (Table D-4),
- Power fail records (Table D-5)
- Errors caused by PSE's removal of the ERR.TMP file (Table D-6), and
- Unexpected trap or interrupt errors (Table D-7).

The first 8 words of any header field always contain the same type of information.

Tables D-8, D-22, D-23, and D-24 define four fields that may or may not be part of the record, depending on the type of error described. (Tables D-9 through D-18 describe device-dependent parameters referenced in Table D-8.) The four additional fields are:

- A program field, format 0, (Table D-8),
- A program field, format 2 (Table D-22),
- A bus activity field (Table D-23), and
- A device register field (Table D-24).

Reserved Words - The term "reserved" means that the corresponding word contains a value for a specific purpose; its contents must not be altered.

Blank Words - The term "blank" means that the related word was allocated for future use; its contents have no meaning.

Table D-1
ERROR.SYS File Record Format
Header Field Hardware (001) or Interrupt Timeout (140) Errors
Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (001/140) format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	UDE (device type and class)	+22	A$DCDT(22),A$DTYP(23)
10	UDE (controller and unit number)	+24	A$CNTN(24),A$UNTN(25)
11	Device mnemonic (ASCII)	+26	A$DVNM
12	Unit logical number	+30	A$SYUN
13	Offset to register field	+32	A$DRF
14	Offset to program field	+34	A$PGF
15	(Reserved)	+36	
16	(Reserved)	+40	
17	Offset to bus activity field	+42	A$BAF
18	(Reserved)	+44	

Table D-2
ERROR.SYS File Record Format
Header Field Memory Parity Error (002) Format 1

Word		Offset Value	Definition
0	Size of header Field in Bytes	+0	A$SIZE
1	Entry type (002) Format (1)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	UDE (device class and type)	+22	A$DCDT(22),A$DTYP(23)
10	UDE (controller and unit number)	+24	A$CNTN(24),A$UNTN(25)
11	Offset to program field	+26	A$MPF1
12	Offset to register field	+30	A$MRF1

Table D-3
ERROR.SYS File Record Format
Header Field System Start Up (040) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (040) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	(Reserved)	+22	
10	(Reserved)	+24	
11	(Reserved)	+26	

Table D-4
ERROR.SYS File Record Format
Header Field Error Logging Terminated (041) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (041) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	

Table D-5
ERROR.SYS File Record Format
Power Fail (042) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (42) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+18	A$SEQN

Table D-6
ERROR.SYS File Record Format
Header Field Error File (ERR.TMP) Re-initialized (043) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (043) Format (0)	+2	A$FMT(2),A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN

Table D-7
ERROR.SYS File Record Format
Header Field Unexpected Trap or Interrupt (141) Format 0

Word		Offset Value	Definition
0	Size of header field in bytes	+0	A$SIZE
1	Entry type (141) Format (0)	+2	A$FMT(2) ,A$ENTY(3)
2	Year	+4	A$TIME
3	Month	+6	
4	Day	+10	
5	Hour	+12	
6	Minute	+14	
7	Second	+16	
8	Error sequence number	+20	A$SEQN
9	Vector of undefined interrupt	+22	A$VCTR
10	Unexpected interrupts not logged because of this log effort	+24	A$UICT
11	PSW of interrupted process	+26	A$OPS
12	PC of interrupted process	+30	A$OPC
13	Offset to bus activity field	+32	A$UIPT

Table D-8
ERROR.SYS File Record Format
Program Field Format 0 (Hardware or Interrupt Timeout Errors)

The Program Field contains information about the program that caused the error or that was active when the error occurred.

Word		Offset Value	Definition
0	Size of program field in bytes	+0	
1	Address mode* Format (0)	+2	P$FMT(2) ,P$ADMD(3)
2	Task name in RAD50	+4	P$TN
3	Task name in RAD50	+6	
4	Programmer group Programmer code	+10	P$UIC
5	Base program load address**	+12	P$TA0
6	Function code	+14	P$FC
7	I/O packet information***	+16	P$PARM
8	(Device-dependent)	+20	
9	(Device-dependent)	+22	
10	(Device-dependent)	+24	
11	(Device-dependent)	+26	
12	(Device-dependent)	+30	
13	(Device-dependent)	+32	
14	Maximum retries Retries left****	+34	P$RT0
15	I/O still in Queue	+36	P$RQ

* 0 = Unmapped; 2 = 18-bit Addressing; 4 = 22-bit Addressing.

** Real if unmapped; 1/64 Real if 18-bit or 22-bit Addressing.

*** See I/O Packet Parameters, Tables D-9 through D-21.

**** If word is negative, error was not recovered.

Table D-9
I/O Packet Parameters
RK03, RK05, RK05F

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RKDA" for I/O GO	+26
12	LBN or transfer start	+30
13	(Reserved)	+32

Table D-10
I/O Packet Parameters
RP02, RP03

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RPCA" for I/O GO	+26
12	"RPDA" for I/O GO	+30
13	(Reserved)	+32

Table D-11
I/O Packet Parameters
RF11

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RFDAE" for I/O GO	+26
12	"RFDAR" for I/O GO	+30
13	(Reserved)	+32

Table D-12
I/O Packet Parameters
RS03, RS04

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	"RSDA" for I/O GO	+30
13	(Reserved)	+32

Table D-13
I/O Packet Parameters
RP04, RP05, RP06

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RPDC" for I/O GO	+26
12	"RPDA" for I/O GO	+30
13	(Reserved)	+32

Table D-14
I/O Packet Parameters
RK06

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RKDA" for I/O GO	+26
12	"RKDC" for I/O GO	+30
13	(Reserved)	+32

Table D-15
I/O Packet Parameters
TA11

Word		Offset Value
7	Space count if spacing function or "KTAPR" contents to transfer data through if data function	+16
8	Virtual buffer address of transfer	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table D-16
I/O Packet Parameters
TS03, TE10, TU10

Word		Offset Value
7	Space count if space function or transfer memory address (High) if data function	+16
8	Transfer memory address (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table D-17
I/O Packet Parameters
TE16, TU16, TU45

Word		Offset Value
7	Space count if space function or transfer memory address (High) if data function	+16
8	Transfer memory address (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	(Reserved)	+30
13	(Reserved)	+32

Table D-18
I/O Packet Parameters
TC11

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	(Reserved)	+26
12	LBN for transfer start	+30
13	(Reserved)	+32

Table D-19
I/O Packet Parameters
RL01

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Total transfer size in bytes	+22
10	Seek difference count	+24
11	"RLDA" for I/O GO	+26
12	This transfer size in bytes	+30
13	(Reserved)	+32

Table D-20
I/O Packet Parameters
RM03

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	"RMDC" for I/O GO	+26
12	"RMDA" for I/O GO	+30
13	(Reserved)	+32

Table D-21
I/O Packet Parameters
RX01

Word		Offset Value
7	Transfer memory address bits (High)	+16
8	Transfer memory address bits (Low)	+20
9	Transfer size in bytes	+22
10	(Reserved)	+24
11	Current transfer size in bytes	+26
12	"RXDA" for I/O GO	+30
13	(Reserved)	+32

Table D-22
ERROR.SYS File Record Format
Program Field Format 2 (Memory Parity Errors)

The Program Field contains information about the program that caused the error or that was active when the error occurred.

Word		Offset Value	Definition
0	Size of program field in bytes	+0	
1	Address mode* Format (2)	+2	P$FMT(2), P$ADMD(3)
2	Task name in RAD50	+4	P$TN
3	Task name in RAD50	+6	
4	Base address of load **	+10	P$TA2
5	Task maximum size **	+12	P$TS
6	Partition name in RAD50	+14	P$PN
7	Partition name in RAD50	+16	
8	Base address of partition	+20	P$PS
9	Processor Status word	+22	P$PSW
10	Program Counter	+24	P$PC

* 0 = Unmapped; 2 = 18-bit addressing; 4 = 22-bit addressing.

** Real if unmapped; 1/64th real if 18-bit or 22-bit addressing.

Table D-23
ERROR.SYS File Record Format
Bus Activity Field Format 0

The bus activity field contains information on bus activity at the time of the error.

Word			Offset Value	Definition
0	Size of bus activity field in bytes		+0	
1	(Reserved)	Format (0)	+2	B$fmt
2	Active vector/4	Active vector/4	+4	B$VEC
.	" "	" "	.	
.	" "	" "	.	
.	" "	" "	.	
n	Active vector/4	Active vector/4	+m	

Table D-24
ERROR.SYS File Record Format
Device Register Field Format 0

The device register field contains the contents of the control and status registers of the device on which the error occurred.

Word		Offset Value	Definition
0	Size of register field in bytes	+0	
1	(Reserved) Format (0)	+2	R$FMT
2	Device registers	+4	R$REG
.	" "	.	
.	" "	.	
.	" "	.	
n	Device registers	+m	

APPENDIX E

DEVICE DRIVER MODIFICATION

This Appendix describes the modifications made to the mass storage device drivers to enable error logging, as well as the common Executive routines that are supplied for the error logging subsystem. All error logging code is incorporated in the Executive or device driver at system generation. In addition to building the necessary device data structures, the system generation procedures enable all error logging code by defining the following conditional assembly symbols:

Symbol	Errors
E$$DVC	Hardware errors (including interrupt timeouts)
E$$NSI	Undefined interrupts
E$$PER	Memory parity errors

E.1 DEVICE DRIVER MODIFICATIONS

A device driver that is modified to enable error logging performs the following:

- Immediately before starting a device data type function, it calls the Executive routine "$BMSET" to set the appropriate bit, thereby indicating that the device has a data function active on the UNIBUS.
- It defines the entry point for interrupt handling via the system macro INTSE$.
- It calls the Executive routine $DVCER if it discovers a loggable error while executing the interrupt handling routine. The routine $DVCER then logs the error.
- It calls the Executive routine $DTOER if it discovers a device timeout error. The routine $DTOER then logs the error.
- For each error, it performs the required number of retries, and records in R2 the maximum number of possible retries (high byte) and the number of possible retries not taken (low byte). It then calls the Executive routine $IODON.

Before attempting to modify a device driver, a programmer should study a driver that has already been adapted for error logging, such as the RK05 (DKDRV) device driver.

E.2 EXECUTIVE ROUTINES

E.2.1 $BMSET

This routine is found in the ERROR module.

Calling sequence:

```
     CALL $BMSET
```

Description:

```
;+
; **-$BMSET-SET A DRIVER'S BIT IN THE I/O ACTIVE BITMAP
;
; THIS COROUTINE RAISES PROCESSOR PRIORITY TO SEVEN AND
; SETS THE MASK IN THE SCB IN $IOABM,
; LETS THE CALLER START THE FUNCTION, AND THEN,
; ALLOWS INTERRUPTS.
;
; INPUTS:
;
;        R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;        $IOABM IS MODIFIED
;-
```

E.2.2 $DVCER

THIS ROUTINE IS FOUND IN THE ERROR module.

Calling sequence:

```
     CALL $DVCER
```

Description:

```
;+
; **-$DVCER-DEVICE ERROR BIT SET
;
; THIS IS THE EMB FORMATTING ROUTINE WHEN DEVICE ERROR BIT
; ERRORS ARE RECOGNIZED BY THE DRIVER. ON THE
; FIRST OCCURRENCE OF AN ERROR, $DVCER ATTEMPTS TO LOG IT.
; IF ERRORS OCCUR ON RETRIES, THEY ARE NOT LOGGED.
;
; THE ERROR CODE "EC.DVC" IS PUSHED ON THE STACK,
; THE ERROR IN PROGRESS BIT IS SET IN THE SCB, THE LENGTH OF THE
; REQUIRED EMB IS CALCULATED AND "$ALEMB" IS CALLED. IF "$ALEMB" FAILS
; TO ALLOCATE A PACKET FOR ANY REASON, "$DVCER" EXITS
; AND THE POINTER IN THE SCB TO THE EMB IS CLEARED.
; ELSE, THE SAVED $IOABM IS COPIED FROM THE SCB TO THE EMB AND
; A POINTER TO THE EMB IS SAVED IN THE SCB. THE ERROR INFORMATION
; INCLUDING DEVICE REGISTERS IS PUT INTO THE EMB AND THE
; RETURN IS MADE.
;
; INPUTS:
;
;       R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;       IF SUCCESSFUL, THE EMB IS FILLED, AND THE SCB
;       CONTAINS A POINTER TO IT. AN ERROR IN PROGRESS BIT
;       IS SET IN THE SCB. ELSE, THE OCCURRENCE OF
;       THE ERROR IS COUNTED ONLY.
;
; NOTE: ALL REGISTERS ARE PRESERVED
;-
```

E.2.3 $DTOER

THIS ROUTINE IS IN THE ERROR module.

Calling sequence:

```
     CALL $DTOER
```

Description:

```
;+
; **-$DTOER-DEVICE TIMEOUTS
;
; THIS IS THE EMB FORMATTING ROUTINE WHEN
; TIMEOUT ERRORS ARE RECOGNIZED BY THE DRIVER. ON THE
; FIRST OCCURRENCE OF AN ERROR, "$DTOER" ATTEMPTS TO LOG IT.
; IF ERRORS OCCUR ON RETRIES, THEY ARE NOT LOGGED.
;
; THE ERROR CODE "EC.DTO" IS PUSHED ON THE STACK,
; THE ERROR IN PROGRESS BIT IS SET IN THE SCB, THE LENGTH OF THE
; REQUIRED EMB IS CALCUALTED AND "$ALEMB" IS CALLED. IF "$ALEMB" FAILS
; TO ALLOCATE A PACKET FOR ANY REASON, "$DTOER" EXITS
; AND THE POINTER IN THE SCB TO THE EMB IS CLEARED.
; ELSE, THE SAVED "$IOABM" IS COPIED FROM THE SCB TO THE EMB AND
; A POINTER TO THE EMB IS SAVED IN THE SCB. THE ERROR INFORMATION
; INCLUDING DEVICE REGISTERS IS PUT INTO THE EMB AND THE
; RETURN IS MADE. THE CONTENTS OF THE CSR THAT IS SAVED IS UNCHANGED
; FROM THE TIME OF TIMEOUT. AFTER THE CSR IS SAVED, DEVICE INTERRUPTS
; ARE DISABLED AND CPU PRIORITY IS LOWERED TO PR0.
;
; INPUTS:
;
;       R2=ADDRESS OF THE CSR
;       R4=ADDRESS OF THE SCB
;
; OUTPUTS:
;
;       C=0 IF FUNCTION WAS NOT A USER-MODE DIAGNOSTIC FUNCTION
;               THE EMB IS FILLED AND THE SCB CONTAINS A POINTER
;               TO IT AND ERROR IN PROGRESS FLAG IS SET IN THE SCB.
;       C=1 IF FUNCTION WAS A USER-MODE DIAGNOSTIC FUNCTION.
;               IN THIS CASE ONLY INTERRUPT ENABLE IS CLEARED AND
;               THE PRIORITY IS LOWERED TO 0.
;
;
; NOTE: ALL REGISTERS ARE PRESERVED
;-
```

E.2.4 $IODON

Refer to the RSX-11M Guide to Writing an I/O Driver.

E.2.5 INTSE$

$INTSE is the interrupt save routine for device drivers that support error logging. To generate the required interrupt service code in the driver, the programmer issues a call to the system macro INTSE$. INTSE$ also generates the appropriate global symbols for interrupt entry points.

The format of the INTSE$ macro is:

```
INTSE$    xx,pri,#ctlrs [,pssave,ucbsave]
```

where:

xx is the 2-character device mnemonic.

pri is the priority of the device (the priority that would be used in a call to $INTSE).

#ctlrs is the number of controllers the driver services.

pssave is an optional argument specifying a variable in which to save the PS word. If omitted, a variable named TEMP is used.

ucbsave is an optional argument specifying a vector in which to store the interrupting device's UCB address. If omitted, a vector named CNTBL is used.

Outputs: R4 is the controller index.

R5 is the UCB address.

Example:

```
INTSE$ PP, PR4,P$$P11
```

E.2.6 Additional Executive Routines

These routines exist within the Executive and are called by the above mentioned routines. This information is supplied to assist in the handling and understanding of the internal workings of the RSX-11M error logging sub-system.

E.2.6.1 $ALEMB or $ALEB1 - This routine is found in the ERROR module.

Calling sequence:

```
CALL $ALEMB
CALL $ALEB1
```

Description:

```
;+
; **-$ALEMB-ALLOCATE AN ERROR MESSAGE BLOCK
; **-$ALEB1-ALLOCATE AN ERROR MESSAGE BLOCK (ALTERNATE ENTRY)
;
; THIS ROUTINE IS CALLED BY ERROR SERVICING
; ROUTINES. IT COUNTS THE OCCURRENCE OF THE ERROR AND TRIES
; TO ALLOCATE A CORE BLOCK FROM THE POOL.
; IF IT IS SUCCESFUL, IT FILLS IN THE ERROR CODE  THE TIME AND THE
; ERROR SEQUENCE NUMBER. ELSE, IT RETURNS WITH C-SET.
;
; INPUTS:
;
;       2(SP)=ERROR CODE
;       0(SP)=RETURN
;         R1 =SIZE OF THE EMB TO ALLOCATE
;
; OUTPUTS:
;
;       IF C-CLEAR:
;       R0=ADDRESS OF THE FIRST UNFILLED BYTE
;       R1=ADDRESS OF THE EMB
;
;       IF C-SET, UNSUCCESSFUL
;
; NOTE: R2 AND R3 ARE DESTROYED BY $ALEMB THRU $ALOCB
;-
```

E.2.6.2 **$QEMB** - This routine is found in the ERROR module.

Calling sequence:

```
     CALL $QEMB
```

Description:

```
;+
; **-$QEMB-QUEUE AN EMB
;
; THIS IS THE COMMON POINT FOR ALL EMBS. THE EMB IS QUEUED
; FIFO IN THE ERROR QUEUE.  IF THERE ARE
; ENOUGH BYTES OF EMBS IN THE POOL, THE LOGGER TASK IS AWAKENED.
; ELSE, IF THE QUEUE WAS EMPTY, A SCHEDULE REQUEST IS MADE
; SO A QUEUED EMB WILL BE WRITTEN WITHIN A TIME LIMIT.
; ELSE, A RETURN IS MADE.
;
; INPUTS:
;
;       R1=ADDRESS OF THE EMB
;
; OUTPUTS:
;
;       THE EMB IS QUEUED. CONDITIONALLY, THE LOGGER IS WAKED
;       OR A SCHEDULE REQUEST IS MADE FOR THE LOGGER
;
; NOTE: REGISTERS R0-R3 ARE DESTROYED
;-
```

INDEX

RSX-11M
Error Logging
Reference Manual
AA-2542C-TC

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. Problems with software should be reported on a Software Performance Report (SPR) form. If you require a written reply and are eligible to receive one under SPR service, submit your comments on an SPR form.

Did you find errors in this manual? If so, specify by page.

_____

_____

_____

_____

_____

_____

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

_____

_____

_____

_____

_____

_____

Is there sufficient documentation on associated system programs required for use of the software described in this manual? If not, what material is missing and where should it be placed?

_____

_____

_____

_____

_____

_____

Please indicate the type of user/reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Non-programmer interested in computer concepts and capabilities

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code or Country _____

Please cut along this line.

---------- Fold Here ----------

---------- Do Not Tear - Fold Here and Staple ----------

FIRST CLASS
PERMIT NO. 33
MAYNARD, MASS.

BUSINESS REPLY MAIL
NO POSTAGE STAMP NECESSARY IF MAILED IN THE UNITED STATES

Postage will be paid by:

digital

Software Documentation
146 Main Street ML5-5/E39
Maynard, Massachusetts 01754

digital

digital equipment corporation

Printed in U.S.A.