pdp11

RSX-11M Error Logging Reference Manual

Order No. DEC-11-OMELA-A-D

digital

RSX-11M Error Logging Reference Manual

Order No. DEC-11-OMELA-A-D

RSX-11M Version 2

digital equipment corporation · maynard, massachusetts

First Printing, April 1976

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

PREFACE

Ø.1 MANUAL OBJECTIVES AND READER ASSUMPTIONS

The RSX-11M Error Logging Reference Manual contains all the information needed to operate the error logging subsystem. Its main purpose is to enable the user to produce error logging reports, which allow the user to monitor the hardware reliability of an RSX-11M system and 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

Ø.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.

Chapter 5 describes the procedures for generating the error logging subsystem from system generation through assembly, task building and task installation.

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

Ø.3 ASSOCIATED DOCUMENTS

The RSX-11M/RSX-11S Documentation Directory, Order No. DEC-11-OMUGA-B-D, 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 or DECtape error as it occurs, regardless of whether or not it is recoverable. Then at user-determined intervals, an analysis task can be run to generate individual error and/or summary reports on some or all of these errors. Without the error logging facility, the Executive automatically retries recoverable errors; after a successful recovery, however, the user is unaware that the error ever occurred.

Error logging may be implemented on any RSX-11M system that is 24K or larger.

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 reports generated by the analysis task, 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 can 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
4. Analyzes 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 Error Log Analyzer, generates 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 monitor three types of errors:

1. Interrupts through unused vectors
2. Device errors
3. Interrupt timeouts

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 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.

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 Analysis and Report Generation

When it is convenient or necessary to generate an error log report, the user first runs the pre-analyzer task PSE to make a preliminary analysis 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 a wide variety of individual error 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. The user may also request a report that contains only information on individual errors, one that contains only summary information, or one that contains both kinds of statistics. 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 status of the system at the time of each error.
- Controlling the amount of system dynamic memory used for error logging.

A set of common error routines performs these functions for all three types of detected errors (undefined interrupts, device errors and interrupt timeouts).

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 the relevant system generation question was answered affirmatively (see section 5.2), 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.

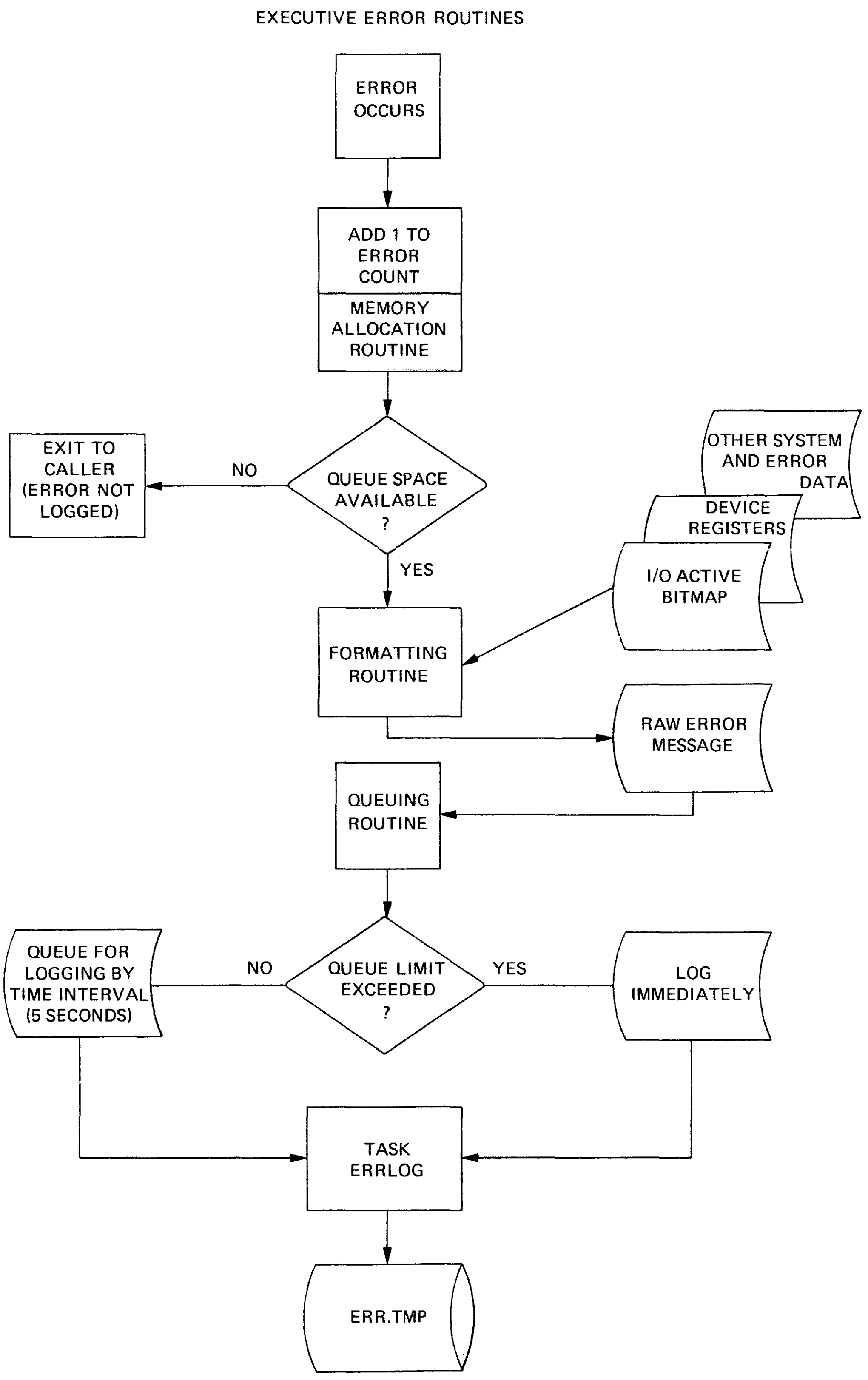

Figure 2-1 Executive Error Routines

2.1.1 Memory Allocation Routine

The memory allocation routine enforces a limit on the amount of system dynamic memory that can be used to queue error information. The task ERRLOG has usually recorded the raw information before this limit is reached. But if an unusual situation prevents ERRLOG from accessing the queued error information, which then fills up available space, the Executive starts to count the errors without logging them. When the space is again available, the Executive resumes queuing the raw error data.

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
- Any additional information needed to describe the error

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 in 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 it contents.

2.2 TASK INTERACTION

The primary function of the error logging tasks is to analyze 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, and Figure 2-2 demonstrates the file activity in a flow chart.

The error logging system on RSX-11M uses various defaults for its files. The default device and directory are SYØ: and [1,6]. The tasks use several standard filenames (including ERR.TMP, ERROR.TMP and ERROR.SYS). Users should not attempt to override these defaults (although it is possible to do so in some cases). One exception, however, is the device to which ERRLOG writes the raw data file ERR.TMP; it may be desirable to specify a dedicated, non-system device for this file. (This would be necessary, for instance, if the ERR.TMP files had to be analyzed at a different installation.)

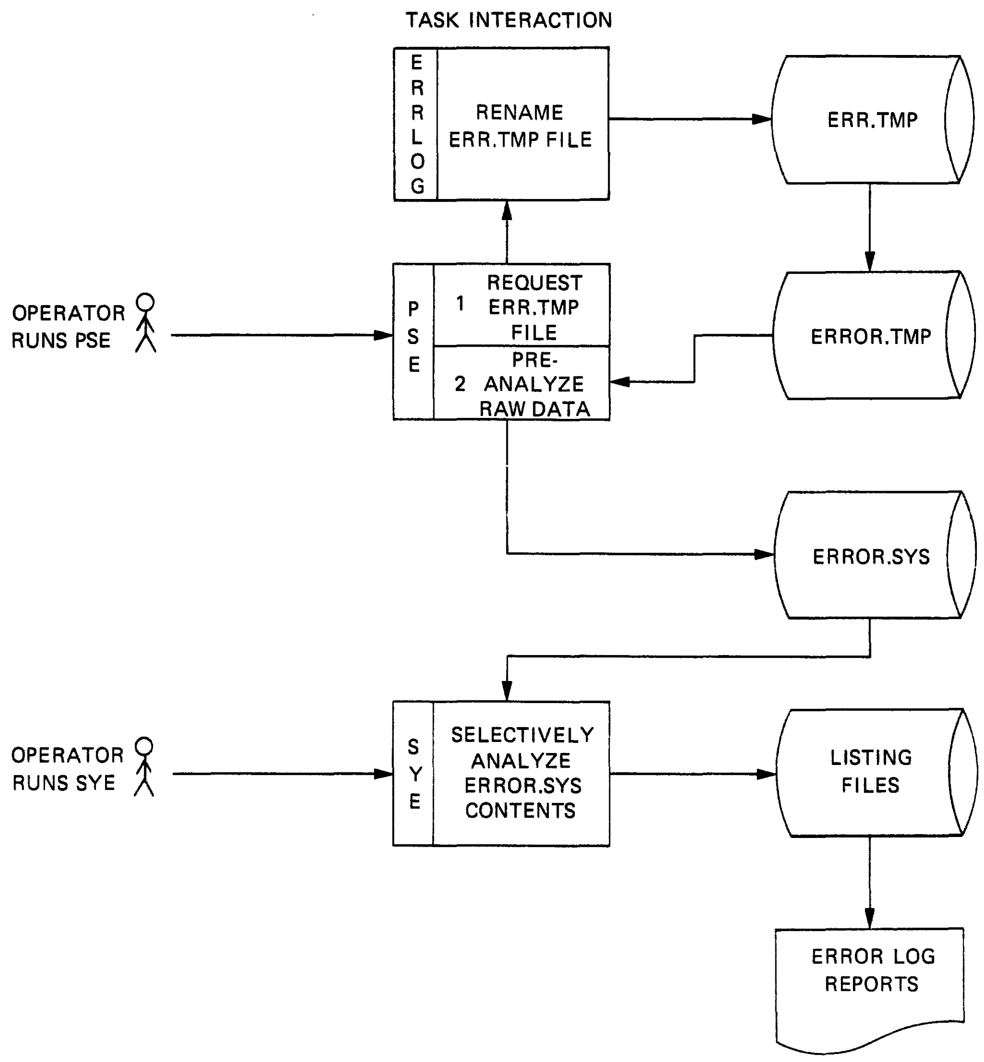

Figure 2-2 Task Interaction

2.2.1 ERRLOG Files (ERR.TMP and ERROR.TMP)

Each time ERRLOG is initialized (usually at System Startup), it creates a file called ERR.TMP and writes in system configuration information needed eventually by PSE. ERRLOG then periodically writes to the file the information it obtains from the Executive error message queue (see section 2.1.3). By default, the data is transferred to [1,6]ERR.TMP by means of LUN 4, assigned to SYØ: at task build. [1,6] and ERR.TMP are the only valid values for the directory and filename, but the device field may be changed (in which case, the LUN assignment would also have to be changed) at task build. See section 5.4.2. (The line in the file ERLBLD.CMD that assigns SYØ: to LUN 4 is "ASG=SY:4". A user could edit the assignment to read "ASG=DK2:4", for example. Otherwise, the REA command could be used to reassign the LUN. The REA command is described in the RSX-11M Operator's Procedures Manual.)

When the user runs PSE, ERRLOG renames the raw data file to ERROR.TMP, makes it available to PSE and then creates a new ERR.TMP file, which includes the required system configuration information. Only one ERR.TMP file should exist at any given time. There might be more than one if the system has crashed or been shut down improperly. In this case, use PIP to change the name to ERROR.TMP, but be careful to preserve the correct version number order.

2.2.2 PSE File (ERROR.SYS)

PSE (the Error File Preanalyzer) requests the file ERROR.TMP from ERRLOG, analyzes the raw error data, creates a file called ERROR.SYS and then deletes ERROR.TMP. If one or more ERROR.SYS files already exist, PSE appends the newly pre-analyzed data to the highest version. The data in ERROR.SYS is in a format that SYE can use to generate its reports.

When supplying the PSE command line, the user can override the output file's default device, directory and filename (SYØ:[1,6]ERROR.SYS), but such changes are not recommended. (Note that the input file specified in the SYE command line must be the same as the PSE output file; see sections 3.2 and 3.3.)

2.2.3 SYE Report Files

The output file from PSE becomes the input file to SYE. The name of the output file that SYE produces after analyzing the contents of ERROR.SYS is automatically generated and depends on the command line options issued to the task (see section 3.3).

See Chapter 4 for a detailed description of the listing files that SYE generates, the end product of the RSX-11M error logging subsystem.

CHAPTER 3

OPERATING PROCEDURES

This chapter describes the operating procedures for the three error logging and analysis tasks (ERRLOG, PSE and SYE) and for the task that terminates error logging (ERF). The error logging subsystem must be properly generated into the system before the user can invoke these tasks (see Chapter 5). 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 kinds of message are listed for each task in section 3.5.

3.1 RUNNING ERRLOG (THE ERROR LOGGER)

The task ERRLOG must be permanently installed (see Chapter 5). To activate error logging, initiate ERRLOG from a privileged terminal by typing:

```
>RUN ERRLOG
```

The system startup command file ([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-ANALYZER)

There are three ways to run the pre-analyzer task PSE. (PSE must be run from a privileged terminal.)

1. To install and run PSE in an unmapped system, type:

```
>INS [1,50]PSE

>PSE
```

2. To install and run PSE in a mapped system, type:

```
>INS [1,54]PSE

>PSE
```

3. To install, run and remove PSE in either a mapped or an unmapped system if the conditions described in section 5.5 have been met, type:

```
>RUN $PSE
```

When installing or running PSE by method 3 above in a system-controlled partition, append the /INC=xxx switch to the command line to increment the amount of space available to the task. 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-analyzer 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].

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

```
>INS [1,50]PSE

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

The defaults for omitted fields of the output file specifier are:

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

The default for indev: is 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 [1,54]PSE          ;INSTALL PSE
>PSE                    ;RUN PSE
PSE> SY:=SY:            ;GENERATE PRE-ANALYZED FILE
PSE> ^Z                 ;EXIT FROM PSE
>REM PSE                ;REMOVE PSE
```

3.3 RUNNING SYE (THE ERROR LOG ANALYZER)

There are three ways to run the error log analyzer task SYE. See Chapter 4 for a detailed description of the reports generated by SYE.

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

   ```
   >INS [1,50]SYE

   >SYE
   ```

2. To install and run SYE in a mapped system, type the following commands at a privileged terminal:

   ```
   >INS [1,54]SYE

   >SYE
   ```

3. To install and run SYE in either a mapped or an unmapped system if the conditions described in section 5.5 have been met, type the following command at any terminal:

   ```
   >RUN $SYE
   ```

The analyzer task responds with the prompt SYE> and waits for the user to type a command line. The format of the command line is

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

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

```
>INS [1,50]SYE

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

Note that the output filename and type are not included in the output file specifier. The filename corresponds to the values specified for xxx and yyy for the /BR: switch described below. The file type is 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 output file specifier are:

Field	Default
outdev:	SYØ:
ufd	user uic

The default values for the input file specifier are:

Field	Default
indev:	SY0:
ufd	[1,6]
file	ERROR
.typ	.SYS
switches	/BR:ALLSUM/NODE/SU

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
>
```

The following switches may be appended to the input file specifier to define the generated error log reports:

Switch	Description
/BR:xxxyyy	The /BR: (breakdown) switch determines what information is to be included in the report. The output filename is derived from the values assigned to xxxyyy. The output file type is LST.

xxx can have one of the following values:

ALL	Include error statistics for all disk and DECtape units.

NOTE

ALL is the default value for xxx.

DEV	Include only device-detected error statistics.
DSK	Include only error statistics for all disk units.
DTO	Include only error statistics related to interrupt timeouts.
MAG	Include only error statistics for all DECtape units.
SYS	Include statistics on system-related errors only.
UDI	Include statistics for undefined interrupt errors only.

yyy can have one of the following values:

ALL	Include both individual error and summary information in the report.
DSK	Include only disk errors in the report. (Only valid when xxx is ALL, DEV or DTO.)
MAG	Include only DECtape errors in the report. (Only valid when xxx is ALL, DEV or DTO.)
SUM	Include only the summary information in the report.

NOTES

1. SUM is the default value for yyy.
2. When yyy is either DSK or MAG, task SYE produces both individual error and summary information in its report.

/BG:time:date (BeGin with.) Include in the report only those errors that occurred on or after the specified time and date. The format of the time and date is

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

where all the numbers are decimal and all six fields must be specified.

Example:

/BG:2Ø:3Ø:Ø:3:2Ø:76 includes those errors that occurred after 8:3Ø PM on March 2Ø, 1976.

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

/DE Produce a detailed report that includes the following additional information:

-Description of the device error register bits
-The task and function that caused the error
-Statistics about the device that caused the error
-Information about concurrent UNIBUS activity

The default is /-DE or /NODE.

/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 RKØ3 or RKØ5 units be provided

/DV:DK1 requests that only the error statistics for RKØ3 or RKØ5 unit 1 be provided.

/ED:time:date (End Date.) Include in the report only those errors that occurred at or before the specified time and date. The format of the time and date is

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

where all the numbers are decimal and all six fields must be specified.

Example:

/ED:18:Ø:Ø:3:22:76 includes those errors that occurred before 6:ØØ PM on March 22, 1976.

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

/SU Include summary information in the report.

The default is /SU.

/-SU or /NOSU Do not include summary information in the report.

The default is /SU.

SYE command line examples:

1.
```
>SYE LP:=SYØ:[1,6]ERROR.SYS/BR:DTOMAG/DE
```

The command line requests a detailed error report to be output on the line printer. The /BR: switch instructs SYE to describe interrupt timeout errors (DTO) that occurred on DECtape units (MAG). The report will include both individual error descriptions and a summary description.

2.
```
>RUN $SYE
SYE> SYØ:=SYØ:/BR:ALLALL
SYE> ^Z
```

The /BR: switch instructs SYE to produce an error report on all the errors described in ERROR.SYS (the default filename in the SYE command line). The report will be stored in a listing file on the system disk, and it will contain both individual error descriptions and a summary description.

3.
```
>RUN $SYE
SYE> LPØ:=SYØ:/BG:2Ø:45:ØØ:Ø5:22:76/BR:UDISUM
SYE> LPØ:=SYØ:/BR:DTODSK/ED:Ø6:3Ø:ØØ:Ø5:24:76
SYE> ^Z
```

The first command line instructs SYE to produce on the line printer a summary description only of all undefined interrupt errors that occurred on or after 8:45 PM on May 22, 1976. The second command line requests a report on interrupt timeout errors related to disks that occurred on or before 6:3Ø AM on May 24, 1976. The second report, also output on the line printer, will include both individual error descriptions and a summary description.

3.4 RUNNING ERF (THE SHUTDOWN TASK)

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

1. For an unmapped system:

   ```
   >INS [1,50]ERF

   >ERF
   ```

2. For a mapped system:

   ```
   >INS [1,54]ERF

   >ERF
   ```

3. For either a mapped or an unmapped system if the conditions described in section 5.5 have been met:

   ```
   >RUN $ERF
   ```

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 opened a new file after passing its previous file to PSE.

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.

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 code.

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 ERL -- "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 using PIP, and subsequently analyzed (see the RSX-11M Utilities Procedures Manual).

ERL -- ERR.TMP OPEN FAILURE, CODE ddd

The file system failed 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 analysis 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. See Section 3.1 above.

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.

Try to analyze the file again.

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. See 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 BREAKDOWN SWITCH
/BREAKDOWN:text

Illegal values ("text") have been supplied with the breakdown (/BR:) switch. See section 3.3.

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.

> 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 OUTPUT FILE
FATAL ERROR - x.

> 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.

> 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.

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

3.5.3.3 Additional Messages

- The first message below is an error message; the second indicates that SYE has completed its analysis.

SYE -- SUMMARY TABLE OVERFLOW
REPORT CONTINUES WITHOUT SUMMARIES

> SYE is unable to produce a summary report because there are too many devices with associated errors. SYE produces a report without the normal summary. The /BR: switch should be changed to reduce the number of erring devices displayed. Alternatively, SYE could be modified to accommodate more devices. (See Appendix A, which describes how to modify SYE by means of the task builder command file.) Normally the analyzer is built to handle 40 (decimal) devices.

SYE -- x. PAGES
filename

> SYE displays this message when it has completed its analysis. "x." is the decimal number of pages in the report and "filename" is the name of the output file containing 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 (see section 3.2). PSE performs a preliminary analysis on raw data contained in files created by the error logging task ERRLOG. The PSE output file, which contains the pre-analyzed error data, becomes the input file to SYE.

SYE, the Error Log Analyzer, 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, unit or volume. See section 3.3 for a description of SYE operating procedures and all the available command line switch options.

The Analyzer produces an error report in the form of a printed listing or a listing file. By default, SYE creates a file called ALLSUM.LST on the system disk; this file contains a summary report of all errors on disk and DECtape units.

4.2 INDIVIDUAL ERROR REPORTS

Error logging detects and records three kinds of errors: interrupts through unused vectors, device errors and interrupt timeouts (see Section 1.2.1). This section describes in detail one or more sample reports on each kind of error.

4.2.1 Reports For Interrupts Through Unused Vectors (Traps)

Figures 4-1 and 4-2 are reports of two interrupts through unused vectors (such interrupts are also called traps). The following description of the two reports refers to each line of text by number, where the first line described is line 1. The first two lines, offset by two star lines, give header information about the error:

```
   ***************************************************************
1  TRAP THROUGH LOCATION 0
2  LOGGED AT 24:FEB:75 14:36:04            ERROR NUMBER 4.
   ***************************************************************

3  VECTOR THROUGH WHICH TRAP OCCURRED      0
4  NUMBER OF UNDF. INTR. MISSED WHILE LOGGING THIS ERROR   0.
5  PROCESSOR STATUS WORD PRIOR TO THIS ERROR              000242
6  PROGRAM COUNTER PRIOR TO THIS ERROR                    045670

7  VECTORS WITH ACTIVE IO
8          220
9          214
```

Figure 4-1 Error Report of a Trap Through Location Ø

```
   ***************************************************************
1  UNDEFINED INTERRUPT
2  LOGGED AT 24-FEB-75 14:52:05            ERROR NUMBER 5.
   ***************************************************************

3  VECTOR THROUGH WHICH TRAP OCCURRED      240
4  NUMBER OF UNDF. INTR. MISSED WHILE LOGGING THIS ERROR   2.
5  PROCESSOR STATUS WORD PRIOR TO THIS ERROR              000340
6  PROGRAM COUNTER PRIOR TO THIS ERROR                    043210

7  VECTORS WITH ACTIVE IO
8          220
```

Figure 4-2 Error Report of an Undefined Interrupt

- Line 1 describes the type of error (TRAP THROUGH LOCATION 0 or UNDEFINED INTERRUPT).
- Line 2 gives the date and time the error occurred and the error´s sequence number.

The first error to occur after error logging has been initialized is assigned a sequence number of 1. The sequence number then increments by 1 every time a loggable error occurs.

The next block of text (lines 3 through 6) contains information about the interrupt:

- Line 3 provides the address of the vector through which the interrupt occurred.
- Line 4 notes the number of undefined interrupts that occurred while the system was logging this error. (This counter permits the detection of bursts of undefined interrupts caused by UNIBUS noise corruption.)
- Line 5 is the processor status word immediately prior to the undefined interrupt.
- Line 6 is the value of the program counter immediately prior to the undefined interrupt.

The last block of data, line 7 onwards, lists all the vectors with active data transfers on the UNIBUS at the time of the error. The list includes the address of the device that caused the error. The block headed "VECTORS WITH ACTIVE IO" does not appear in the report if there was no I/O activity.

4.2.2 Device Error Reports

Figure 4-3 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 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 describes the type of error (DISK HARDWARE ERROR).
- Line 2 gives the date and time the error occurred and the error´s sequence number.

The first error to occur after error logging has been initialized is assigned a sequence number of 1. The sequence number then increments by 1 every time a loggable error occurs.

```
    RSX 11M SYSTEM ERROR REPORT COMPILED AT 20-MAY-75 11:00:13


    ***************************************************************
 1  DISK HARDWARE ERROR
 2  LOGGED AT 6-MAY-75 15:37:08            ERROR NUMBER 7.
    ***************************************************************

 3  DISK PARAMETERS
 4          UNIT NAME                 DK0
 5          DEVICE TYPE               RK05/RK03 UNIT-0 CONTROLLER-0

 6  DISK REGISTERS AT ERROR TIME
 7          RKDS            004760  WRITE PROTECT
 8          RKER            020000  WRITE VIOLATION
 9          RKCS            140702
10          RKWC            177406
11          RKBA            067636
12          RKDA            062000

13  ERROR DIAGNOSIS
14  RECOVERED
15  WRITE PROTECT
16  WRITE VIOLATION

17  RETRIES PERFORMED       8.

18  USER TASK PARAMETERS
19          TASK NAME                 F11ACP
20          TASK UIC                  1,1
21          PHYSICAL START ADDRESS  60000
22          USER FUNCTION REQUESTED          WRITE   (400)
23          FUNCTION INTERPRETED FROM REGISTERS     WRITE
24                  PHYSICAL BUFFER ADDRESS START   67622
25                  TRANSFER SIZE IN BYTES          1000
26                  LOGICAL BLOCK NUMBER AT IO GO   4540
27                  CYLINDER AT IO GO TIME          100.
28                  TRACK AT IO GO TIME             0.
29                  SECTOR AT IO GO TIME            0.
30                  CYLINDER AT ERROR TIME          99.
31                  TRACK AT ERROR TIME             1.
32                  SECTOR AT ERROR TIME            11.
33                  COUNT OF IO IN PROGRESS         0.

34  VECTORS WITH ACTIVE IO
35          220
```

Figure 4-3 Error Report of a Disk Hardware Error

The next block of data (DISK PARAMETERS) describes the erring device:

- Line 4 (UNIT NAME) shows the device mnemonic and unit number (DKØ).

- Line 5 (DEVICE TYPE) shows the DIGITAL name for the device (RKØ5/RKØ3), the physical unit number on the controller (UNIT-Ø) and the controller number (CONTROLLER-Ø).

The report then describes the state of the device registers at the time of the error (DISK REGISTERS AT ERROR TIME) in lines 6 through 12. The first column lists the register names; the second shows the contents of the registers. The device register names are those used in the PDP-11 Peripherals Handbook. The contents of the registers on retries are not recorded.

The text at the end of lines 7 and 8 (WRITE PROTECT and WRITE VIOLATION) describes the error bits in the device registers named on those lines. A description of the device error register bits only appears if the user has specified the /DE (detailed report) switch in the SYE command line.

- Line 14 indicates whether or not the Executive error recovery procedure corrected the error.

- Lines 15 and 16 repeat the description of the device error register bits (see lines 7 and 8). (This information only appears in a detailed report.)

- Line 17 shows how many times the Executive had to retry the error before it was successfully recovered. This line alternatively shows the number of times the Executive retried the error if recovery was unsuccessful.

The next block of data describes the user task that initiated the I/O request that failed. Only a detailed report will contain this information:

- Line 19 provides the 6-character task name (F11ACP).

- Line 2Ø is the User Identification Code (UIC) under which the task was running.

- Line 21 shows the task's physical start address, that is, the base address of the task in memory.

- Line 22 describes the function requested by the task (via a QIO directive) at the time of the error. The number in parentheses at the end of the line is the octal QIO function code.

- Line 23 describes the function that was implied by the contents of the register. For this error, the implied function is the same as the function requested by the task (WRITE).

NOTE

The function implied by the contents of the register (that is, the function that failed) can differ from the function requested by the task. For example, if a task requests a WRITE to a DECtape, the driver must perform a READ BLOCK NUMBER function, which could then cause an error.

- Lines 24 through 33. These 10 lines describe in detail the device activity at the time of the error. The information given depends on the type of device being described. This report includes the physical memory address of the data buffer and the transfer size, and the physical device's start address and estimated end address.

The last block of text (lines 34 and 35) lists the vectors with active data transfers on the UNIBUS at the time of the error. The list includes the address of the device that caused the error. The number on the second line (220) is the vector address through which a transfer complete or data error interrupt would be fielded. A long list of vector addresses could indicate that UNIBUS interaction caused the error.

The block headed "VECTORS WITH ACTIVE IO" does not appear if there was no I/O activity when the error occurred. Also, error reports on interrupt timeouts do not include this information because the long time element involved makes it irrelevant.

4.2.3 Interrupt Timeout Reports

Since interrupt timeout reports have the same format as device error reports, see section 4.2.2 for a description of the various entries.

4.3 SUMMARY REPORTS

Figure 4-4 is an example of a detailed summary. SYE includes detailed information in a report when the /DE switch has been specified in the command line. This part of the report provides summary information about all the errors analyzed in the report as a whole (but not necessarily about all the errors in the input file to SYE).

The command line issued to SYE determines which errors are to be analyzed; the summary displays this command line at the beginning of the report (see lines 2 and 3).

The next block of information, labelled "REPORT FILE ENVIRONMENT", describes the SYE input and output files and the time and date of the first and last entries in the input file (lines 4 through 8).

```
    RSX 11M SYSTEM ERROR REPORT COMPILED AT 20-MAY-75 11:00:39

 1  SYSTEM ERROR REPORT SUMMARY (SYSTEM)

 2  COMMAND LINE USED
 3  LP:=DK1:/BR:ALLALL/DE

 4  REPORT FILE ENVIRONMENT
 5          SOURCE FILE             DK1:[1,6]ERROR.SYS
 6          OUTPUT FILE             LP:(54,24]ALLALL.LST
 7          DATE OF FIRST ENTRY     6-MAY-75 15:21:31
 8          DATE OF LAST ENTRY      6-MAY-75 15:50:49
 9  ENTRIES PROCESSED                       12.
10  ENTRIES MISSING                          0.
11  UNKNOWN ENTRIES ENCOUNTERED              0.
12  UNKNOWN DEVICES ENCOUNTERED              0.
13  FIELD FORMAT ERRORS ENCOUNTERED          0.
14  DEVICE ERROR PROCESSED                  11.
15  UNDEFINED INTERRUPTS PROCESSED           0.
16  UNDF. INTR. MISSED DUE TO UNDF. INTR. PROCESSING        0.
17  TRAPS THROUGH LOCATION 0 PROCESSED       0.
18  DEVICE TIMEOUTS PROCESSED                0.
19  MEMORY MANAGEMENT ERRORS ENCOUNTERED
20  SYSTEM POWER FAILS ENCOUNTERED
21  SYSTEM PARITY ERRORS ENCOUNTERED
22          REPRODUCIBLE
23          NON REPRODUCIBLE
24  SYSTEM LOADS                             1.


25  SYSTEM ERROR REPORT SUMMARY     (DISK - MOVABLE HEAD)

26  RP03 UNIT-0 CONTROLLER-0
27  HARD    0.
28  SOFT    1.


29  SYSTEM ERROR REPORT SUMMARY     (TAPE)

30  TU56 UNIT-0 CONTROLLER-0
31  HARD    0.
32  SOFT    5.

33  TA11 UNIT-1 CONTROLLER-0
34  HARD    0.
35  SOFT    1.
```

Figure 4-4 Summary Error Report

The next 16 lines provide various error statistics:

- Line 9 is the number of error entries analyzed in the report.
- Line 1Ø is the number of errors missed because the occurrence of another error prevented a previous one from being logged.
- Line 11 is the number of unknown errors encountered by the analyzer. An unknown error is an entry that a version of SYE cannot analyze. This situation can occur if an old version of SYE has been run.
- Line 12 is the number of entries that referred to a device not supported by the analyzer. Such an entry may be encountered if the site has implemented error logging on a device that SYE does not recognize.
- Line 13 is the number of times the input file encountered a data structure error (field format error). Such encounters may indicate that the wrong version of the pre-analyzer PSE was used.
- Line 14 is the number of device errors analyzed in this report.
- Line 15 is the number of undefined interrupts analyzed in this report.
- Line 16 is the number of undefined interrupts that were not logged because another undefined interrupt was already being recorded.
- Line 17 is the number of traps through location Ø that were analyzed in this report.
- Line 18 is the number of interrupt timeouts analyzed in this report.
- Lines 19 through 23 are concerned with statistics that will be applicable to a future version of the error logging facility.
- Line 24 (SYSTEM LOADS) is the number of times that error logging was initiated in the time span covered by the report. Error logging is usually initiated only once when the system is loaded and started.

The remaining sections, lines 25 through 35, summarize error statistics according to the associated device type. The types defined for SYE summary reports are "disks - movable head", "disks - fixed head" and "tapes". A device type section appears only if the report analyzes errors associated with that type.

Each section describes:

- The DIGITAL name for the physical device (RPØ3 or TU56, for example)
- The physical unit numbers of the device and its controller (UNIT-Ø and CONTROLLER-Ø, for example)
- The number of associated hard (non-recovered) and soft (recovered) errors analyzed in this report.

CHAPTER 5

GENERATION PROCEDURES

5.1 INTRODUCTION

This chapter provides all the information needed to generate the error logging subsystem. Users should read the instructions carefully before proceeding to generate error logging for the first time.

The procedures consist of four basic operations:

1. Selecting error logging options at system generation.

2. Assembling the privileged tasks ERRLOG and ERF.

3. Task building all four error logging tasks.

4. Installing ERRLOG.

The assembly and task build procedures include the use of command files. If these files need to be modified in any way, the user should use an editor to do so before starting the procedures (see the RSX-11M Utilities Procedures Manual).

The files needed to generate the error logging facility can be found either

- On the source tape in the distribution kit for RP02, RP03 or RP04 configurations, or

- On the MCR and FCP source disk in the distribution kit for an RK05 configuration.

5.2 SYSTEM GENERATION OPTIONS

A complete system generation must be performed to incorporate error logging in an RSX-11M system. (See the RSX-11M System Generation Manual for a complete description of system generation procedures.)

The query program SGN (phase 1 of system generation) includes two questions that pertain to the error logging facility. The user should answer yes ("Y") to both these questions in order to include the necessary Executive features and data structures in the target system. If the user replies no ("N") to both these questions, the error logging facility cannot be generated without performing another complete system generation. The relevant questions are described below.

DO YOU WISH TO LOG DEVICE ERRORS AND INTERRUPT TIMEOUTS[Y/N]?

If the answer is yes:

- The disk and DECtape drivers will be assembled to enable error logging
- The formatting routine for device error and interrupt timeout messages will be included in the Executive
- The I/O data base will be extended for each controller.

DO YOU WISH TO LOG UNDEFINED INTERRUPTS[Y/N]?

If the answer is yes:

- The Executive will include vector address identification routines and a routine for formatting undefined interrupt error messages.
- All unused vectors will be filled with pointers to the Executive error logging routines.

If the answer is no, unused vectors will merely point to an RTI (Return from Interrupt) instruction.

The user must then complete both phases of system generation and start up the target system before proceeding to the next step, the assembly of the privileged tasks ERRLOG and ERF.

5.3 ASSEMBLING ERRLOG AND ERF

Only the two privileged tasks ERRLOG and ERF need to be assembled. The source files to be assembled are located on the system generation source volume (see section 5.1) in User File Directory [17,1Ø].

The command files used for assembly, also located on the source volume, are found in one of the following directories:

- [17,2Ø] for unmapped systems
- [17,24] for mapped systems

During assembly, the Executive configuration file RSXMC.MAC from phase 1 of system generation must be available under UIC [11,1Ø]. The MCR command UFD should be used if necessary to create an [11,1Ø] directory on the disk being used for the assembly. RSXMC.MAC contains conditional assembly parameters.

There are 2 steps in the assembly operation:

1. Making the source and command files available.
2. Assembling the tasks ERRLOG and ERF.

The description of each step is divided into subsections when the detailed instructions vary according to the type of configuration.

Only one subsection is applicable to a single configuration.

5.3.1 Preparing for Assembly

This step places the appropriate source and command files in the correct directories.

5.3.1.1 Dual RK Disk Configurations

Both the source and the command files required for assembly are already on the MCR/FCP source disk. However the user must transfer to that disk the RSXMC.MAC Executive configuration file for the system on with which error logging will run. This file exists on the mapped or unmapped target disk from the latest system generation. RSXMC.MAC is located in directory [200,200].

NOTE

> During system generation, the file RSXMC.MAC was copied to UIC [200,200] of the target system object disk.

Therefore, to prepare for assembly, mount and boot the MCR/FCP source disk on DK0: and mount the appropriate target disk on DK1:. Note that it may be necessary to install required MCR tasks.

Then issue MCR commands as follows:

```
     >MOU DK1:
     >SET /UIC=[11,10]
     >INS [1,50]UFD
     >UFD DK:[11,10]
     >RUN $PIP
     PIP> =DK1:[200,200]RSXMC.MAC
     PIP> ^Z
     >REM ...UFD
```

Proceed to section 5.3.2 when this step has been successfully completed.

5.3.1.2 RP02/03/04 Disk With Magnetic Tape Configurations

On this type of configuration, the Executive source and object files are on the system disk, but the source files for ERRLOG and ERF are on the FLX format source magnetic tape. To prepare for assembly, transfer the ERRLOG and ERF source files (held under UIC [17,10]) to the disk containing the Executive files. Note that it may be necessary to install the File Exchange Utility task FLX.

For example:

```
     >SET /UIC=[12,10]
     >RUN $FLX
     FLX> =MT:[12,10]LKLST.MAC
     FLX> ^Z
     >SET /UIC=[17,10]
     >RUN $FLX
     FLX> =MT:[17,10]*.MAC
     FLX> ^Z
```

NOTE

If the installation is using a TU16 magnetic tape drive, replace "MT:" with "MM:". For example:

```
FLX> =MM:[17,10]*.MAC
```

Proceed to section 5.3.2 when this step has been successfully completed.

5.3.1.3 RK Disk With RX Flexible Disk Configurations - On this type of configuration, the user must load the RX disk driver, mount the RX disk containing the current RSXMC.MAC configuration file (from phase 1 of system generation), then copy the file RSXMC.MAC to UFD [11,10] on the MCR/FCP source disk mounted on DK0:. If the configuration file RSXMC.MAC is not currently available on an RX flexible disk, the file can be found either:

- on the object disk in directory [200,200], or
- on the Executive source disk in directory [11,10]

Note that it may be necessary to install required MCR tasks.

Example:

```
>LOA DX:
>MOU DX:
>SET /UIC=[11,10]
>INS [1,50]UFD
>UFD DK:[11,10]
>REM ...UFD
>RUN $PIP
PIP> =DX:[1,2x]RSXMC.MAC
PIP> ^Z
```

where x in the last input file directory is 0 for an unmapped system and 4 for a mapped system.

Proceed to section 5.3.2 when this step has been successfully completed.

5.3.1.4 RK Disk With DECtape/Cassette/Magnetic Tape Configurations - The same set of pre-assembly instructions applies to all three of the configurations described in this heading.

The system disk should be the MCR/FCP source volume from the distribution kit. The appropriate device driver must be loaded and the volume containing the configuration file RSXMC.MAC for the current Executive must be available. Then use FLX to copy RSXMC.MAC to UFD [11,10] on the system disk. Note that it may be necessary to install required MCR tasks, such as UFD.

If the configuration file RSXMC.MAC is not currently available on a non-disk volume, the file can be found either:

- on the object disk in directory [200,200], or
- on the Executive source disk in directory [11,10]

In the following example, replace nn with the appropriate device name (DT for DECtape, CT for cassette, MT or MM for magnetic tape).

```
>LOA nn:
>INS [1,50]UFD
>UFD DK:[11,10]
>REM ...UFD
>SET /UIC=[11,10]
>RUN $FLX
FLX> =nn:[1,2x]RSXMC.MAC
FLX> ^Z
```

where x in [1,2x] is 0 for an unmapped system and 4 for a mapped system.

Proceed to section 5.3.2 when this step has been successfully completed.

5.3.2 Assembly

The following assembly instructions apply to all configurations.

The first step is to set the UIC either to [17,20] for an unmapped system or to [17,24] for a mapped system. The [17,20] or [17,24] directory must not contain any object modules; if necessary, use PIP to delete any such files.

The next step is to assemble the ERRLOG source files by means of an indirect command file (@ERLASM) that issues commands to the MACRO assembler. The resultant object modules are then concatenated into a single object file called ERL.OBS. Old object files are then deleted, and ERL.OBS is renamed ERL.OBJ.

Another indirect command file (@ERFASM) completes the operation by instructing the MACRO assembler to assemble the ERF source file.

For example:

```
>SET /UIC=[17,2x]
>RUN $PIP
PIP> *.OBJ;*/DE
PIP> ^Z
>RUN $BIGMAC
MAC> @ERLASM
MAC> ^Z
>RUN $PIP
PIP> ERL.OBS=*.OBJ
PIP> *.OBJ;*/DE
PIP> .OBJ/RE=ERL.OBS
PIP> ^Z
>RUN $BIGMAC
MAC> @ERFASM
MAC> ^Z
```

where x in [17,2x] is 0 for an unmapped system and 4 for a mapped system.

5.4 TASK BUILDING

In order to build the error logging tasks, the object and other necessary files must be transferred to the correct target disk. Therefore the task building operation consists of two steps:

1. Transferring the error logging files to the appropriate disk.

2. Building the tasks.

5.4.1 Transferring the Object Files

The file transfer instructions depend on the type of configuration. In all cases, however, the commands described transfer to the correct disk the libraries, command files and overlay description files needed to build PSE and SYE, and the object module files, command files and overlay description files needed to build ERRLOG and ERF.

5.4.1.1 Dual RK Disk Configurations - Firstly, mount the mapped or unmapped object disk on DKØ: and the MCR/FCP source disk on DK1:.

If the system is mapped, set the UIC to [1,2Ø] for non-privileged tasks and files, and to [1,24] for privileged tasks and files. (See the second example below.) If the system is unmapped, set the UIC to [1,2Ø] for both privileged and non-privileged tasks and files. Then use PIP to transfer the files to DKØ:.

For an unmapped system:

```
>MOU DK1:
>SET /UIC=[1,2Ø]
>RUN $PIP
PIP> =DK1:PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB
PIP> =DK1:PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*
PIP> =DK1:[17,2Ø]ERL.OBJ,ERF.OBJ
PIP> ^Z
```

For a mapped system:

```
>MOU DK1:
>SET /UIC=[1,2Ø]
>RUN $PIP
PIP> =DK1:PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB
PIP> ^Z
>SET /UIC=[1,24]
>RUN $PIP
PIP> =DK1:PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*
PIP> =DK1:[17,24]ERL.OBJ,ERF.OBJ
PIP> ^Z
```

Proceed to section 5.4.2.

5.4.1.2 RP02/03/04 Disk With Magnetic Tape Configurations - On this type of configuration, the required non-privileged object libraries and all command files reside on the distribution kit FLX format source magnetic tape. The privileged object files are already on the disk. The following commands transfer the remaining files to the system disk and set up the proper UICs.

NOTE

> If the installation's magnetic tape drive is a TU16, use MM instead of MT in the following examples.

For unmapped systems:

```
     >SET /UIC=[1,20]
     >RUN $PIP
     PIP> =[17,20]ERL.OBJ,ERF.OBJ
     PIP> ^Z
     >RUN $FLX
     FLX> =MT:[1,20]PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*
     FLX> =MT:[1,20]PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB
     FLX> ^Z
```

For mapped systems:

```
     >SET /UIC=[1,24]
     >RUN $PIP
     PIP> =[17,24]ERL.OBJ,ERF.OBJ
     PIP> ^Z
     >RUN $FLX
     FLX> =MT:[1,24]PSEBLD.CMD,SYEBLD.CMD,ERFBLD.*,ERLBLD.*
     FLX> ^Z
     >SET /UIC=[1,20]
     >RUN $FLX
     FLX> =MT:[1,20]PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB
     FLX> ^Z
```

Proceed to section 5.4.2 when the transfer has been completed successfully.

5.4.1.3 RK Disk With RX Flexible Disk Configurations - On this type of configuration, the user must transfer the required files to the object disk in two phases:

1. Copy the files from the MCR/FCP source disk to a scratch flexible disk.

2. Then copy the files from the flexible disk to the mapped or unmapped object disk.

Between these two phases, the user must bring down the system and remove the MCR/FCP disk; then reboot the system with the object disk mounted on DK0: to complete the transfer.

The following examples demonstrate how to effect the 2-phase transfer on either a mapped or an unmapped system.

For an unmapped system:

```
>DMO DX:
>INS [1,50]INI
>INI DX:
>REM ...INI
>MOU DX:
>INS [1,50]UFD
>UFD DX:[1,20]
>SET /UIC=[1,20]
>RUN $PIP
PIP> DX:=PSE.OLB,SYE.OLB,PSEBLD.ODL,SYEBLD.ODL
PIP> DX:=PSEBLD.CMD,SYEBLD.CMD,ERFBLD.*,ERLBLD.*
PIP> DX:=[17,20]ERL.OBJ,ERF.OBJ
PIP> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the unmapped object disk and reboot.

```
>SET /UIC=[1,20]
>MOU DX:
>RUN $PIP
PIP> =DX:*.*
PIP> ^Z
```

Proceed to section 5.4.2 when all the files have been successfully transferred.

For a mapped system:

```
>DMO DX:
>INS [1,50]INI
>INI DX:
>REM ...INI
>MOU DX:
>INS [1,50]UFD
>UFD DX:[1,20]
>UFD DX:[1,24]
>SET /UIC=[1,20]
>RUN $PIP
PIP> DX:=PSE.OLB,SYE.OLB,PSEBLD.ODL,SYEBLD.ODL
PIP> ^Z
>SET /UIC=[1,24]
>RUN $PIP
PIP> DX:=PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*
PIP> DX:=[17,24]ERL.OBJ,ERF.OBJ
PIP> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the mapped object disk and reboot.

```
>SET /UIC=[1,20]
>MOU DX:
>RUN $PIP
PIP> =DX:*.*
PIP> ^Z
>SET /UIC=[1,24]
>RUN $PIP
PIP> =DX:*.*
PIP> ^Z
```

Proceed to section 5.4.2 when all the files have been successfully transferred.

5.4.1.4 <u>RK Disk with DECtape or Magnetic Tape Configurations</u> - On this type of configuration, the user must transfer the required files to the object disk in two phases:

1. Copy the files from the MCR/FCP disk to a scratch tape (either DECtape or magnetic tape).

2. Then copy the files from the tape to the mapped or unmapped object disk.

Between these two phases, the user must bring down the system and remove the MCR/FCP disk; then reboot the system with the object disk mounted on DKØ: to complete the transfer.

The following examples demonstrate how to effect the transfer on either a mapped or an unmapped system. Where the commands use nn to represent a device name, insert the appropriate name for the installation (DT for DECtape, MT or MM for magnetic tape).

For an unmapped system:

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> nn:/ZE
FLX> nn:/DO=PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB/RS
FLX> nn:/DO=PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*/RS
FLX> nn:/DO=[17,2Ø]ERL.OBJ,ERF.OBJ/RS
FLX> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the unmapped object disk and reboot.

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> =nn:*.*
FLX> ^Z
```

Proceed to section 5.4.2 when all the files have been successfully transferred.

For a mapped system:

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> nn:/ZE
FLX> nn:/DO=PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB/RS
FLX> ^Z
>SET /UIC=[1,24]
>RUN $FLX
FLX> nn:/DO=PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*/RS
FLX> nn:/DO=[17,24]ERL.OBJ,ERF.OBJ/RS
FLX> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the mapped object disk and reboot.

The exact commands to be issued for the next step differ according to the type of configuration.

Magnetic tape configurations require the following commands:

```
>SET /UIC=[1,20]
>RUN $FLX
FLX> =nn:*.*
FLX> ^Z
>SET /UIC=[1,24]
>RUN $FLX
FLX> =nn:*.*
FLX> ^Z
```

where nn is MM or MT.

DECtape configurations require the following commands:

```
>SET /UIC=[1,20]
>RUN $FLX
FLX> =DT:PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB
FLX> ^Z

>SET /UIC=[1,24]
>RUN $FLX
FLX> =DT:PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*
FLX> =DT:ERL.OBJ,ERF.OBJ
FLX> ^Z
```

Proceed to section 5.4.2 when all the files have been successfully transferred.

5.4.1.5 <u>RK Disk with Cassette Configurations</u> - On this type of configuration, the user must transfer the required files to the object disk in two phases:

1. Copy the files from the MCR/FCP source disk onto three scratch cassettes.

2. Then copy the files from the cassettes onto the mapped or unmapped object disk.

Between these two phases, the user must bring down the system and remove the MCR/FCP disk; then reboot the system with the object disk mounted on DK0: to complete the transfer.

The following examples demonstrate how to effect the transfer on either a mapped or an unmapped system.

For an unmapped system:

Mount the first cassette.

```
>SET /UIC=[1,20]
>RUN $FLX
FLX> CT0:/ZE
FLX> CT0:/DO=PSEBLD.ODL,SYEBLD.ODL,PSE.OLB/RS
FLX> CT0:/DO=PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*/RS
FLX> CT0:/DO=[17,20]ERL.OBJ,ERF.OBJ/RS
```

Mount the second cassette.

```
FLX> CTØ:/ZE
FLX> CTØ:/DO=SYE.OLB/RS
FLX -- END OF VOLUME ON CASSETTE
CTØ:[1,2Ø]
 MOUNT NEW CASSETTE? (Y, Z (OUTPUT ONLY), OR CR)
```

Mount the third cassette.

```
FLX> Z
```

("Z" instructs FILEX to initialize the new cassette and then continue the copy operation.)

```
FLX> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the unmapped target disk on DKØ: and reboot.

Mount the first cassette.

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> =CTØ:*.*
```

Mount the second cassette.

```
FLX> =CTØ:*.*
FLX -- END OF VOLUME ON CASSETTE
CTØ:[1,2Ø]*.*
 MOUNT NEW CASSETTE? (Y, Z (OUTPUT ONLY), OR CR)
```

Mount the third cassette.

```
FLX> Y
```

("Y" instructs FILEX to continue the copy operation.)

```
FLX> ^Z
```

Proceed to section 5.4.2. when all the files have been transferred.

For a mapped system:

Mount the first cassette.

```
>SET /UIC=[1,24]
>RUN $FLX
FLX> CTØ:/ZE
FLX> CTØ:/DO=PSEBLD.CMD,SYEBLD.CMD,ERLBLD.*,ERFBLD.*/RS
FLX> CTØ:/DO=[17,24]ERL.OBJ,ERF.OBJ/RS
FLX> ^Z
```

Mount the second cassette.

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> CTØ:/ZE
FLX> CTØ:/DO=PSEBLD.ODL,SYEBLD.ODL,PSE.OLB,SYE.OLB/RS
FLX -- END OF VOLUME ON CASSETTE
CTØ:[1,2Ø]
 MOUNT NEW CASSETTE? (Y, Z (OUTPUT ONLY), OR CR)
```

Mount the third cassette.

```
FLX> Z
```

("Z" instructs FILEX to initialize the new cassette and then continue the copy operation.)

```
FLX> ^Z
```

Now bring down the system, remove the MCR/FCP disk, mount the mapped target disk on DKØ: and reboot.

Mount the first cassette.

```
>SET /UIC=[1,24]
>RUN $FLX
FLX> =CTØ:*.*
FLX> ^Z
```

Mount the second cassette.

```
>SET /UIC=[1,2Ø]
>RUN $FLX
FLX> =CTØ:*.*
FLX -- END OF VOLUME ON CASSETTE
CTØ:[1,2Ø]*.*
 MOUNT NEW CASSETTE? (Y, Z (OUTPUT ONLY), OR CR)
```

Mount the third cassette.

```
FLX> Y
```

("Y" instructs FILEX to continue the copy operation.)

```
FLX> ^Z
```

Proceed to section 5.4.2 when all the files have been transferred.

5.4.2 Task Builder Instructions

The commands described below build the error logging tasks ERRLOG, PSE, SYE and ERF. The system disk must be the mapped or unmapped object disk from the system generation distribution kit. All the necessary error logging files are on the system disk if the directions in section 5.4.1 were successfully followed.

Any changes to the TKB command files must be made before running the task builder.

For an unmapped system:

```
>SET /UIC=[1,2Ø]
>RUN $LBR
LBR> ERL/CR:14.:64.:64.=ERL.OBJ
LBR> ^Z
>RUN $BIGTKB
TKB> @ERLBLD
TKB> @ERFBLD
TKB> @PSEBLD
TKB> @SYEBLD
TKB> ^Z
```

For a mapped system:

```
     >SET /UIC=[1,20]
     >RUN $BIGTKB
     TKB> @[1,24]PSEBLD
     TKB> @[1,24]SYEBLD
     >SET /UIC=[1,24]
     >RUN $LBR
     LBR> ERL/CR:14.:64.:64.=ERL.OBJ
     LBR> ^Z
     >RUN $BIGTKB
     TKB> @ERLBLD
     TKB> @ERFBLD
     TKB> ^Z
```

The following four task image files exist in directory [1,50] on an unmapped system or [1,54] on a mapped system when task building has completed successfully:

1. ERL.TSK
2. ERF.TSK
3. PSE.TSK
4. SYE.TSK

5.5 TASK INSTALLATION

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

To install ERRLOG in an unmapped system, type the following at a privileged terminal:

```
     >INS [1,50]ERL
```

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

```
     >INS [1,54]ERL
```

The other tasks may be executed by the install/run/remove option of the RUN command ("RUN $PSE", for example) if the restrictions explained below are met.

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.

When installing or running PSE in a system-controlled partition, the user should append the /INC=xxx switch to the command line to increment the amount of space available to the task. PSE needs additional space at the rate of 1Ø(decimal) words per device unit, where device unit is each device that could cause an error to be logged. When running in a user-controlled (task) partition, PSE itself determines the space available and uses the amount it needs.

APPENDIX A

MODIFYING SYE AT TASK BUILD

The command file SYEBLD.CMD contains instructions to the task builder to create the ...SYE task (see section 4.4.2). 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 of 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
   ```

   and

   ```
   EXTSCT=TBUF:nnn
   ```

   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
   ```

   and

   ```
   EXTSCT=OBUF:nnn
   ```

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

   A user may want to increase the value of nnn to take advantage of a wide line printer, for example.

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

3. The maximum number of devices that SYE can handle in summary reports. The option statements related to the maximum number are:

   ```
   GBLDEF=MX$DEV:nn
   ```

   and

   ```
   EXTSCT=V1:xxx
   ```

where nn is the octal number representing the maximum number of devices that SYE can handle, and xxx is an octal number equal to 8. (decimal) times the value of MX$DEV.

The default values are 50 (octal) and 500 (octal) respectively.

Each single increment to the maximum number of devices increases the SYE task size by four words.

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

```
;
;SYEBLD.CMD
;
;THIS COMMAND FILE WILL BUILD THE ...SYE TASK FROM THE
;SYEBLD.ODL FILE FOR UNMAPPED SYSTEMS
;
[1,50]SYE/-CP/-MM,[1,30]SYE/-SP=SYEBLD/MP
;
;
STACK=64
ASG=TI:1
TASK=...SYE
PAR=GEN:40000:40000
;
;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$WTH:110
;
;THIS DEFINES THE PAGE WIDTH FOR THE OUTPUT FROM THE ERROR
;PROCESSORS
;
;ALL VALUES ARE OCTAL
GBLDEF=PL$WTH:120
;
;THIS IS THE DEFINITION OF THE PAGE LENGTH FOR THE OUTPUT FROM
;THE ERROR PROCESSORS
;
;ALL VALUES TYPED ARE OCTAL
;
GBLDEF=PL$LGH:74
;
;THIS IS THE MAXIMUM NUMBER OF DEVICES WHICH THE ANALYZER CAN HANDLE
;SUMMARIES OF SYSTEM LOGGED ERRORS.
;
;************** N O T E ****************
;
;WHEN CHANGING THIS VALUE THE NEXT VALUE OF THE SUMMARY TABLE
;LENGTH MUST BE ALSO CHANGED.
;
;
;ALL VALUES ARE TYPED AS OCTAL
;
GBLDEF=MX$DEV:62
;
;THIS IS THE SIZE OF THE SUMMARY TABLE TO BE USED IN THE ANALYXER
;WHEN CHANGING THE NUMBER OF DEVICES WHICH THE ANALYZER CAN
;HANDLE SUMMARIES ON THIS VALUE ALSO MUST BE CHANGED.
;
;****************** N O T E ******************
;
;TO CHANGE THIS VALUE THE NEW VALUE MUST BE EQUAL TO
;EIGHT (8.) TIMES THE VALUE OF "MX$DEV" ABOVE.
;
;
;ALL VALUES TYPED ARE OCTAL
;
EXTSCT=V1:620
//
```

Figure A-1

INDEX

RSX-11M Error Logging
Reference Manual
DEC-11-OMELA-A-D

READER'S COMMENTS

NOTE: This form is for document comments only. Problems with software should be reported on a Software Problem Report (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 ________

If you require a written reply, please check here. ☐

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 Communications
P. O. Box F
Maynard, Massachusetts 01754

digital

digital equipment corporation

Printed in U.S.A.